



AIN SHAMS UNIVERSITY
FACULTY OF ENGINEERING
Electronics and Communication Dept.

Ultra Low Power Sub 1-GHz Transceiver

Supervised By: Prof. Dr. Hani Fikry Ragai

Sponsored by: Silicon Vision

THESIS

Submitted in partial fulfillment of the requirements

for the degree of Bachelor of Science in Electrical Engineering

June, 2013

Group Members Names:

Adel Abdullah Taha Ammar	Chapter 8,10
Ahmed Medhat Helmy Ezzat Mohamed	Chapter 11,15
Ahmed Sayed Ali Sayed	Chapter 4,7
Alaa El-Deen Fathy Rezk	Chapter 1,3
Amgad Abdelwahab Mohamed Abdelwahab Ghonim	Chapter 2,7
Amr Ibrahim Mohammed Amen Ahmed	Chapter 6,7
Amr Yehia Hussien Mahmoud Arafat	Chapter 9
Kariem Gamal Mohamed El-Kholy	Chapter 13
Khaled Mohamed Abd El-Hafez Abd El-Aal Hassan	Chapter 14,15
Ramy Mohammed Abdelrahman Ibrahim Rady	Chapter 12
Omar Abdel-Rahman Mohamed Abdel-Raouf Mohamed	Chapter 5,7

ACKNOWLEDGEMENTS:

We wish to extend our deepest gratitude to our supervisor **Prof. Dr. Hany Fikry Ragai**, We want to extend out most sincere gratitude and infinite thanks to our advisors in silicon vision Company **Eng Ahmed Saad, Eng Hossam Ali, Eng Mohamed Mohsen, Eng Sherif Diao** and **Eng Nour El Deen Hany** for their support during all this year never accepting less than our best efforts.

We wish to give thanks to all those who gave us the motivation to complete this work from the college staff **Dr. Mohamed El Nozahy, Eng. Joesph Riad, Eng Omar El Aasser, Eng Samer Sayed Bahr, Eng Amr Tarek, Eng Mazen Erfan** and **Eng Mohamed Ibrahim** for their guidance and help.

Finally, we would like to thanks our families for their prayers and encouragement that always helps us taking the right steps.

Preface:

The rapid growth of transceivers created increasing demand for low-cost, low-power and reduced size and weight equipment's. An increasingly higher level of integration is needed to meet these requirements. Thanks to the advancement in deep sub-micron CMOS technology, this is easily achievable for digital signals and low-frequency signal processing. However, in order to reach the final goal of system on a chip (SOC) solution, the final piece of puzzle is still missing-the RF front end. Designing efficient, fully integrated transceivers that could operate from very low supply voltage and for biomedical implantable electronic systems is a major challenge. The work presents a transceiver that works at 915 MHz, on standard technology of 130 nm that favors low power, short range applications.

1. Contents

1	Receiver System Overview	31
1.1	RX specifications.....	31
1.1.1	Maximum input level	31
1.1.2	SNR.....	31
1.1.3	Sensitivity.....	31
1.1.4	Dynamic range	31
1.1.5	Non-Linearity	31
1.1.6	Cascaded nonlinear stages	34
1.1.7	Noise	34
1.1.8	SFDR	35
1.2	RX architecture.....	35
1.2.1	Super heterodyne	35
1.2.2	dual down conversion.....	36
1.2.3	Image rejection RX.....	37
1.2.4	Direct-conversion(zero-IF)	39
1.2.5	Low IF RX.....	42
1.2.6	Band pass sampling RX.....	44
1.2.7	Comparisons	45
1.3	System Design	46
1.3.1	Receiver System specifications	46
	So we can take BW about 280Khz	49
	Matlab could be found in appendix	49
	Sensitivity:	49
	Receiver Noise figure:	49
	Receiver IIP3:	49
	1-dB Compression point:	49
1.3.2	Receiver Type and Frequency planning.....	49
	Blocker Profile:	50
	Filter	51
	ADC	51
1.3.3	Gain and phase mismatch of PLL local frequency:	51
	Settling Time:	52
	Gain strategy:	52
	Specifications Distribution:	53

What achieved:	55
1.4 References:.....	55
2 The LNA.....	57
2.1 Introduction.....	57
2.2 Noise.....	58
2.2.1 Thermal noise	58
2.2.2 Flicker noise	60
2.2.3 Classical Two-Port Noise Theory	60
2.3 Matching	61
2.3.1 Impedance Transformation	61
2.3.2 Impedance Transformation Using Lumped Elements	62
2.3.3The Choice of Topology	64
2.3.4 Insertion Loss.....	64
2.4 Linearity.....	64
2.4.1 Optimum biasing	64
2.4.2 Linear feedback	66
2.4.3 Feed forward cancellation	67
2.5 Topologies of LNA	67
2.5.1 Common Source with resistive load	68
2.5.2 Common Source with resistive feedback	68
2.5.3 Common gate LNA.....	69
2.5.4 Common source LNA with inductive degeneration	71
2.5.5 Comparisons between Topologies	73
2.6 Design Methodology	73
2.6.1 The way of thinking	73
2.6.2 Tradeoffs in LNA	74
2.6.3 Hand analysis.....	75
2.7 Differential vs. single ended LNA	77
2.8 Simulation and Results	77
2.8.1 Notes, problems that I face	77
2.8.2 Circuit.....	78
2.8.2 LNA.....	80
2.8.3 Attenuator	85
2.8.4 Quality factor of the input stage	88
2.8.5 Specs achieved.....	88

2.9	Layout.....	89
2.9.1	Introduction to Layout.....	89
2.9.2	Layout of LNA circuit.....	90
2.10	Conclusion	90
2.11	References.....	91
3	MIXER.....	92
3.1	Introduction:	92
3.2	Mixer Definitions:	92
3.3	Mixer Topologies	93
3.4	Specifications required and what achieved	96
3.5	Proposed Mixer	96
	Mixer specifications Analysis for proposed Mixer:	97
3.6	Design Methodology	100
3.7	Analysis Explanation:	105
3.8	Simulations and Results:	107
4	IF Filter	124
4.1	Introduction.....	124
4.2	Topology survey and selection.....	125
4.2.1	Main topologies and general features of filters	125
4.2.2	The Active-RC and MOS-C filters.....	128
4.2.3	The Gm-C filters	130
4.2.4	The Switched-Capacitor filters.....	133
4.2.5	The switched current filters	135
4.2.6	The selected topology (Switched Capacitor filter).....	137
4.3	General issues	138
4.3.1	What is the complex filters , and how complexity generated ?	138
4.3.2	The selected circuit for the S.C. filter	139
4.3.3	The filter's main switch design	140
4.3.4	The schematic and symbol of the single complex integrator stage	145
4.3.5	How to measure the response of the complex filters in CADENCE.....	146
4.4	Failed iterations , and tradeoffs solved throughout it	147
4.4.1	The first iteration	147
4.4.2	the second iteration.....	148
4.4.3	The third iteration.....	150
4.4.4	The fourth iteration	153

4.5	The filter design.....	155
4.5.1	New modifications from last iteration.....	155
4.5.2	The general structure of the filter	156
4.5.3	The AA filter design.....	156
4.5.4	The core S.C. filter.....	163
4.5.5	The reconstruction filter	168
4.5.6	The circuit of all of the filter	168
4.6	The results of the filter simulation.....	169
4.6.1	The response (transfer function) of the filter	169
4.6.2	The time domain examples on the filter	171
4.6.3	The corners simulation results.....	176
4.6.4	The NF & IIP3 simulation results.....	177
4.6.5	The results SUMMERY	178
5	Automatic Gain Control	178
5.1	Introduction	178
5.2	AGC Topologies	179
5.3	AGC Main Blocks.....	180
5.4	Programmable Gain Amplifier.....	180
5.4.1	Introduction.....	180
5.4.2	Design Parameters	180
5.4.3	PGA Topologies.....	181
5.4.4	Required Specifications	183
5.4.5	Design Approach	184
5.4.6	Simulation Results	186
5.4.7	Corners Results	191
5.4.8	Layout	194
5.5	Peak Detector	195
5.5.1	Introduction.....	195
5.5.2	Design Parameters	196
5.5.3	PD Topologies	196
5.5.4	Required Specifications	199
5.5.5	Design Approach	199
5.5.6	Simulation Results	201
5.5.7	Corners Results	205
5.5.8	Layout	206

5.6	Flash A/D	209
5.6.1	Introduction.....	209
5.6.2	Modeling of Flash A/D	209
5.6.3	Design Parameters	210
5.6.4	Simulation Results	210
5.7	Digital Control.....	210
5.8	Simulation Result of all AGC blocks	210
5.8.1	Control Methodology.....	210
5.8.2	Simulation on small received signal	211
5.8.3	Simulation on saturated received signal	211
5.9	Conclusion.....	212
5.10	Reference.....	213
6	ADC.....	214
6.1	Abstract:	214
6.2	New challenge in ADCs design	215
6.3	High speed ADCs topologies:	215
6.3.1	Flash ADC:	216
6.3.2	Flash ADC Specifications:.....	216
6.3.3	Pipeline ADC:.....	217
6.3.4	SAR ADC:	217
6.3.5	Comparison of different types of ADCs:	217
6.4	SAR ADC:.....	217
6.4.1	Real converters:	218
6.4.2	Specifications:.....	218
6.4.3	Static performance:	218
6.4.4	Offset error:.....	218
6.4.5	DNL (differential non linearity).....	218
6.4.6	INL (Integral non linearity)	219
6.4.7	Full scale error:	220
6.4.8	Monotonicity:.....	220
6.4.9	Dynamic performance:.....	220
6.5	SAR ADC blocks:	221
6.5.1	Introduction:.....	221
6.5.2	Sample and hold:.....	222
6.5.3	Comparator:	222

6.6	Performance parameters:.....	223
6.6.2	Kick back noise:.....	224
6.6.3	Metstability:	224
6.6.4	Slew rate and propagation delay:	225
6.6.5	Comparator Architecture:	225
6.6.6	Basic blocks:	225
6.6.7	Open-loop:	225
6.6.8	Latched comparator following pre-amp:	226
6.6.9	Static comparator:	226
6.6.10	Rail to rail comparator:	227
6.6.11	Comparison:.....	228
6.6.12	Proposed comparator:	229
6.6.13	Successive approximation registers:	229
6.7	DAC:	230
6.7.1	Binary-Weighted Capacitor Array:.....	230
6.7.2	C-2C Capacitor Array:.....	230
6.7.3	Comparison:.....	231
6.7.4	Proposed DAC:	232
6.7.5	Capacitor value selection:	233
6.7.6	KT/C noise:.....	233
6.7.7	Parasitic capacitors:	233
6.7.8	Mismatch of capacitor:	234
6.8	Design methodology:	234
6.8.1	Capacitor value selection:	234
	SAR logic control:.....	235
6.8.2	Sampling switch:.....	235
6.8.3	Ideal switch:	235
6.9	Practical sampling issues:.....	236
6.9.1	1.3.4.3.3 Charge injection:.....	237
6.10	Proposed sampling switch:.....	237
6.10.1	Simulation results:	239
6.10.2	Dessouky switch:	239
6.10.3	Sadollahy switches [2]:	240
6.11	Proposed sampling switch:.....	241
6.11.1	Comparison:.....	242

6.12	Comparator:.....	243
6.12.1	Achieved specifications:	244
6.12.2	Corners simulation:	244
6.13	Full system simulations:.....	245
6.13.1	Corners summary:	248
6.14	Mismatch analysis:.....	249
6.14.1	Conclusion:	249
6.15	Final conclusion:	250
6.16	Layout:	251
6.16.1	Switches layout:	251
	Reference switch:	252
1.3.5.3	252
	Capacitor Array:.....	253
6.17	Summary:	253
6.18	Future work:	253
6.19	References:	254
7	Receiver System verification.....	256
7.1	The LNA-Mixer interface :	256
7.2	The LNA-Mixer-Filter interface :.....	257
7.3	The AGC-ADC interface	258
7.4	The ALL RX chain interfacing results	260
8	Transmitter & PLL System Overview	262
8.1	Introduction.....	262
8.2	Specification Extraction.....	262
	Frequency Range	262
	Frequency Resolution	262
	Frequency Accuracy.....	262
	Locking Time	262
	Phase noise	262
	Spurs Level.....	262
8.2.1	Frequency Range.....	262
8.2.2	Frequency Resolution	262
8.2.3	Locking Time	262
8.3	Brief Introduction for the phase locked loop (PLL)	263
8.3.1	Brief Review for PLL system	263

8.3.2	Building the linearized PLL model.....	263
8.4	Frequency Synthesizer Architectures.....	264
8.5	Calculating Loop Parameters	265
8.5.1	Calculation for VCO gain and Tuning range	265
8.5.2	Calculating the loop bandwidth.....	265
8.5.3	Calculating Loop filter component elements	266
8.6	Phase Noise analysis.....	269
8.6.1	Reference oscillator phase noise	269
8.6.2	PFD & charge pump noise.....	271
8.6.3	VCO phase noise	273
8.6.4	Sigma delta phase noise	275
8.7	Reference spurs.....	278
8.8	CPPSIM system design	279
9	Crystal oscillator	281
9.1	Introduction.....	281
9.1.1	Crystal oscillator in PLL	281
9.1.2	Why crystal oscillator?	282
9.1.3	Crystal oscillator applications	282
9.1.4	What is crystal oscillator?	282
9.1.5	Piezoelectricity.....	282
9.1.6	Some historical notes.....	283
9.2	Crystal oscillator specifications	283
9.3	Crystal modeling.....	284
9.3.1	Series or Parallel Resonant Crystals.....	285
9.4	Theory of oscillation.....	285
9.4.1	Negative resistance.....	285
9.5	Single transistors devices	286
9.5.1	Lossless linear circuit	286
9.5.2	Derivations for critical and optimum transconductance.....	287
9.5.3	Active transistor topologies	288
9.6	Amplitude control topologies	289
9.6.1	Analog amplitude control crystal oscillator	289
9.6.2	Digital amplitude controlled crystal oscillator	289
9.7	Design methodology	290
9.7.1	Resonator.....	290

9.8	Active circuit	293
9.8.1	Design.....	293
9.9	Active circuit with current control loop.....	297
9.9.1	Simulation results.....	298
9.10	Buffer.....	300
9.11	Total circuit.....	301
9.12	Summary	302
9.13	Layout.....	302
9.13.1	OTA Layout.....	303
9.13.2	Buffer Layout.....	304
9.14	Conclusion	304
9.15	References.....	305
10	PFD and Charge pump	306
10.1	Phase frequency detector	306
10.1.1	Phase Detectors	306
10.1.2	Phase frequency detector.....	307
10.2	Design methodology	309
10.3	PFD simulation results.....	312
10.4	Charge pump.....	313
10.4.1	Introduction	313
10.4.2	Charge pump architectures	313
10.4.3	PFD & CHARGE PUMP non-idealities	317
10.4.4	Design methodology	319
10.4.5	Simulation results	321
11	Voltage controlled oscillator “VCO”	323
11.1	Introduction	323
11.2	VCO specification.....	323
11.2.1	Phase noise.....	323
11.2.2	VCO gain (K_{vco}).....	325
11.2.3	Tuning range	325
11.2.4	Output swing.....	325
11.2.5	Supply sensitivity.....	326
11.2.6	Frequency pulling	326
11.2.7	Drive capability.....	326
11.2.8	Power dissipation.....	326

11.3	Basic concepts	326
11.3.1	Feedback model	326
11.3.2	One port model	327
11.4	Topologies survey	327
11.4.1	Ring oscillator.....	327
11.4.2	LC oscillator.....	328
11.5	Design methodology	331
11.5.1	Tank circuit	332
11.5.2	Active devices design	338
11.5.3	Tail current.....	339
11.5.4	Buffer design.....	339
11.5.5	Calibrator design.....	342
11.6	Important simulations.....	344
11.6.1	Transient output	344
11.6.2	Tuning range	345
11.6.3	VCO gain (K_{vco}).....	346
11.6.4	Phase noise.....	348
11.6.5	16 covered sub-bands.....	350
11.7	Summary of the result	350
11.8	Design schematics and test bench	351
11.8.1	VCO core, Buffer, capacitor bank	351
11.8.2	VCO core schematic	352
11.8.3	Capacitor bank schematic	353
11.8.4	Buffer schematic	354
11.9	References	355
12	Frequency Divider	356
12.1	Introduction:	356
12.2	General Consideration:.....	356
12.2.1	Fractional N PLL:	357
12.2.2	Divider Phase noise	357
12.2.3A	Modular Architecture.....	357
12.3	Architecture Review.....	358
12.3.1	PULSE SWALLOW DIVIDER	359
12.3.2	A Multi modulus Divider (MMD).....	360
12.3.3	Generic MMD.....	363

12.3.4 Comparison & Conclusion	364
12.4 Circuit review	365
12.4.1 CMOS Rail to Rail (static)	365
12.4.2 Current Mode Logic (CML)	366
12.4.3 TSPC (True single phase clocking)	367
12.4.4 OTHER STYLES	367
12.4.5 Comparison & Conclusion	367
Required design Specs Given: choose a MMD to be implemented	369
12.5 Design Methodology	369
12.5.1 Design method for output Divider quadrature generator.....	369
12.5.1.1 CML LATCH	369
12.5.2 Design method for feedback divider	376
12.5.2.1 Design method For CML 2/3 Divider	376
12.5.2.1.1 CML LATCH & CML NAND	377
12.5.2.1.3 CML Level shifter	377
12.5.2.2 Design method For CML to CMOS converter	379
12.5.2.3 Design methodology for CMOS divider	382
12.5.2 Required Vs. Achieved.....	386
12.5.3 Corner simulation	387
12.6 Conclusion.....	387
13 Sigma Delta	389
13.1 Introduction.....	389
13.2 Basics Concepts of the Sigma Delta	389
13.3 Quantization noise and over sampling effects.....	389
13.4 Noise-Shaping Effect	392
13.5 Sigma Delta Modulators in Fractional-N frequency synthesizers.....	393
13.6 Spur Reduction Techniques in Fractional-N Frequency Synthesis.....	393
13.7 Architectures of Sigma Delta Modulators.....	394
13.7.1 First Order Sigma Delta Modulator.....	395
13.7.2 MASH 1-1-1 Modulator.....	396
13.7.3 Single Stage Modulator.....	398
13.7.4 Performance Comparison	399
13.8 Phase Noise Due to SD Converters	399
13.9 Calculation of the Required Resolution	400
13.10 Dithering Techniques.....	400

13.11	Deterministic techniques.....	400
13.11.1	Stochastic techniques.....	400
13.12	Types of Quantizers	401
13.12.1	Truncation Quantizer SDM.....	401
13.12.2	Variable Modulus SDM.....	401
13.13	Simulations & Results	401
13.14	Conclusion	404
13.15	Acknowledgements	404
1.1	References.....	404
14	Power amplifier	406
14.1	Introduction	406
14.2	General considerations	406
14.2.1	Efficiency.....	406
14.2.2	Linearity.....	407
14.2.3	Single-Ended and Differential PAs.....	407
14.2.4	Cascode output stage VS common source stage.....	408
14.3	Topology survey.....	408
14.3.1	Linear classes	408
14.3.2	Non-linear classes	413
14.4	Comparison between topologies	417
14.5	Design methodology	418
14.5.1	Bias Circuit	419
14.5.2	Pre-amplifier	419
14.5.3	Driver	422
14.5.4	Switch	425
14.5.5	Output stage	425
14.6	Simulated results and test benches	431
14.7	Conclusion.....	439
14.8	References	440
15	Transmitter & PLL System Verification	441
15.1	VCO with Divider	441
15.1.1	Circuit schematic.....	441
15.1.2	Simulations.....	442
15.1.3	PSS result.....	444
15.1.4	Divider with PA.....	444

15.1.5	Introduction	444
15.1.6	Divider–PA test bench.....	444
15.1.7	Coupling DC of Divider	445
15.1.8	Pre-amplifier	446
15.1.9	Simulated results	446
15.1.10	Corners simulation	448
15.2	VCO with Divider and PA.....	449
15.2.1	Circuit schematic.....	449
15.2.2	Transient result	449
15.2.3	PSS result.....	451
I.	Appendix A: Ocean Script Codes.....	452
II.	Appendix B: Mat Lab Codes	503
15.2.4	Matlab Code (GFSK BW calculation)	523
15.2.5	525
III.	Appendix C: Verilog-A Codes	539
IV.	Appendix D: VHDL Codes	555

List of Figures

Figure 1.1.1	: cross modulation	32
Figure 1.1.2	: iip3.....	33
Figure 1.1.3	: IIP2	33
Figure 1.1.4	: definition iip2.....	33
Figure 1.1.5	: cascoded devices	34
Figure 1.1.6	: thermal noise	34
Figure 1.1.7	: super heterodyne.....	35
Figure 1.1.8	: image problem	35
Figure 1.1.9	: image filter	36
Figure 1.1.10	: dual conversion RX.....	36
Figure 1.1.11	: 90 phase shift.....	37
Figure 1.1.12	: algorithm.....	37
Figure 1.1.13	: hartley RX.....	38
Figure 1.1.14	: waver architecture.....	39
Figure 1.1.15	: direct conversio RX	39
Figure 1.1.16	: 1/f noise	40
Figure 1.1.17	: lo leakage	40
Figure 1.1.18	: capacitive coupling	41
Figure 1.1.19	: negative feedback.....	41
Figure 1.1.20	: offset cancellation.....	41
Figure 1.1.21	: even order distortion	42
Figure 1.1.22	: low IF RX.....	43

Figure 1.1.23 : band pass RX	44
Figure 3.1 Single Balanced passive Mixer	93
Figure 3.2 Double balanced passive Mixer	93
Figure 3.3 Active Gilbert Mixer (double balanced).....	94
Figure 3.4 Folded Mixer	94
Figure 3.5 Folded Mixer Topologies.....	94
Figure 3.6 Bulk injection	95
<i>Figure 3.7 Proposed active Active Gilbert Mixer</i>	<i>97</i>
<i>Figure 3.8 Summary of Specifications</i>	<i>100</i>
<i>Figure 3.9 Mixer Parts.....</i>	<i>101</i>
Figure 3.10 Constant gm circuit	101
<i>Figure 3.11 Transconductor Stage.....</i>	<i>103</i>
<i>Figure 3.12 Switch Part.....</i>	<i>104</i>
Figure 3.13 Load.....	104
<i>Figure 3.14 Test_Bench</i>	<i>107</i>
<i>Figure 3.15 Transconductor Bias</i>	<i>108</i>
<i>Figure 3.16 Transconductor Biasing DC-Bias</i>	<i>108</i>
<i>Figure 3.17 switches DC biasing.....</i>	<i>109</i>
<i>Figure 3.18 Load Bias</i>	<i>109</i>
<i>Figure 3.19 Typical Gain</i>	<i>110</i>
<i>Figure 3.20 Typical Noise Figure</i>	<i>110</i>
<i>Figure 3.21 Typical Compression point.....</i>	<i>111</i>
<i>Figure 3.22 IIP3.....</i>	<i>111</i>
<i>Figure 3.23 Harmonics values</i>	<i>112</i>
<i>Figure 3.24 RF-IF isolation.....</i>	<i>112</i>
<i>Figure 3.25 RF-LO Isolation</i>	<i>113</i>
<i>Figure 3.26 LO-IF isolation.....</i>	<i>113</i>
<i>Figure 3.27 LO-RF isolation</i>	<i>114</i>
<i>Figure 3.28 IF-RF isolation.....</i>	<i>114</i>
<i>Figure 3.29 IF-LO isolation.....</i>	<i>115</i>
<i>Figure 3.30 Corners gain.....</i>	<i>116</i>
<i>Figure 3.31 Noise Figure across corners</i>	<i>117</i>
Figure 4.4.1 : operating regions for various filters topologies	125
Figure 4.4.2 : the Butterworth responses.....	125
Figure 4.4.3 : the responses of Chybeshev 1 filters	126
Figure 4.4.4 : the Chebyshev2 responses	126
Figure 4.4.5 : the Elliptic responses	126
Figure 4.4.6 : this figure summarizes all filters parameters	127
Figure 4.4.7 : Various OP-AMPs that can be used for Active-RC and MOS-C filters.....	128
Figure 4.4.8 : the various types of VLSI CAPs.....	128
Figure 4.4.9 : the Various VLSI resistors can be used	129
Figure 4.4.10 : the first order Active-RC filters	129
Figure 4.4.11 : various second order Active-RC and MOS-C filters	129
Figure 4.4.12 : the transcontuctors definition.....	130
Figure 4.4.13 : the different integrators circuits using Gm-C cells.....	130
Figure 4.4.14 : EMULATING resistors , inductors , adders and amplifiers using Gm-C cells only	131
Figure 4.4.15 : the first order Gm-C filters.....	131

Figure 4.4.16 : second order Gm-C filter that can be employed as low , high and band pass filters depending on the values of V_{i1} and V_{i2}	132
Figure 4.4.17 : second order Gm-C filters using direct mapping from the active-RC filters	132
Figure 4.4.18 : how the S.C. can emulates a resistor.....	133
Figure 4.4.19 : the switches implementation in general (conceptual)	134
Figure 4.4.20 : the second order S.C. filters examples	134
Figure 4.4.21 : the first order S.C. filters examples	134
Figure 4.4.22 : the main idea of the switched current filters	135
Figure 4.4.23 : the main cells made by the SI circuits	136
Figure 4.4.24 : the first order SI filter example.....	136
Figure 4.4.25 : the second order SI filter example	136
Figure 4.4.26 : the most general idea of the complex filters.....	138
Figure 4.4.27 : transferring the low pass filter into COMPLEX filter	138
Figure 4.4.28 : making complex filter using only integrators and adders	139
Figure 4.4.29 : the single S.C. integrator cell used in the filter.....	140
Figure 4.4.30 : the main circuit of the single stage complex integrator used in our filter in reality.....	140
Figure 4.4.31 : the transmission gate's resistance Vs. input voltage.....	141
Figure 4.4.32 : R_{on} versus the transistor width	142
Figure 4.4.33 : the variation of the R_{on} with the transistor length	142
Figure 4.4.34 : the output of the charge injection test bench.....	143
Figure 4.4.35 : the error due to charge injection	143
Figure 4.4.36 : the main switch schematic used in the filter.....	144
Figure 4.4.37 : the main symbol used in ALL filter circuits	144
Figure 4.4.38 : the schematic of the single stage complex integrators.....	145
Figure 4.4.39 : the symbol of the single stage complex integrators	145
Figure 4.4.40 : how to measure the response of the complex filters	146
Figure 4.4.41: the circuit of the first iteration	147
Figure 4.4.42 : the response of the first iteration filter	148
Figure 4.4.43 : the AA and the reconstruction filters of the second iterations.....	149
Figure 4.4.44 : the circuit of the second iteration	149
Figure 4.4.45 : the response of the second iteration	149
Figure 4.4.46 : the input stage of the third iteration.....	150
Figure 4.4.47 : the response of the A.A. of the third iteration	151
Figure 4.4.48 : the output stage of the third iteration	151
Figure 4.4.49 : the circuit of the third iteration	151
Figure 4.4.50 : the response of the third iteration	152
Figure 4.4.51 : the phase response of the third iteration	152
Figure 4.4.52 : the input stage of the fourth iteration	154
Figure 4.4.53 : the response of the various stages of the A.A. filter in the fourth stage	154
Figure 4.4.54 : the response of the fourth iteration.....	154
Figure 4.4.55 : the circuit of the fourth iteration	154
Figure 4.4.56 : the anatomy of the S.C. filters	156
Figure 4.4.57 : the input stage of the AA filter	157
Figure 4.4.58 : the constant Gm cell used in all of AA filter	157
Figure 4.4.59 : the test bench of the input stage of the AA filter.....	158
Figure 4.4.60 : the gain of the input stage of the AA filter	158
Figure 4.4.61 : gain variation across corners in about 1 dB which is satisfactory enough.....	158

Figure 4.4.62 : the noise figure of the input stage of the AA filter	159
Figure 4.4.63 : the proceeding stage of the AA filter	160
Figure 4.4.64 : the gain in the typical and corners for the proceeding stage of the AA filter.....	161
Figure 4.4.65 : the schematic of the AA filter	162
Figure 4.4.66 : the response of the AA filter after each stage of its stages	162
Figure 4.4.67 : the core OTA circuit	163
Figure 4.4.68 : the biasing cell of the core OTA.....	164
Figure 4.4.69 : the CMFB circuit used in the core OTA.....	164
Figure 4.4.70 : the gain of the core OTA (typical and corners).....	165
Figure 4.4.71 : the output of the square wave	166
Figure 4.4.72 : the PSRR of the core OTA	166
Figure 4.4.73 : the CMRR of the core OTA.....	166
Figure 4.4.74 : measurement of the SR for the core OTA	167
Figure 4.4.75 : the reconstruction filter.....	168
Figure 4.4.76 : the filter circuit (all).....	169
Figure 4.4.77 : the response of the filter and the rejections.....	169
Figure 4.4.78 :the BW of the filter	170
Figure 4.4.79 : the IRR and the image side response of the filte.....	170
Figure 4.4.80 : the phase response of the filter.....	171
Figure 4.4.81 : the signal at IF frequency (1.1 MHz) at the positive frequency side in the middle of the S.C. core filter.....	172
Figure 4.4.82 : : the signal at IF frequency (1.1 MHz) at the positive frequency side after the reconstruction filter	172
Figure 4.4.83: : the signal at IF frequency (1.1 MHz) at the positive frequency side in the middle of the S.C. core filter "zoomed".....	172
Figure 4.4.84 :the signal at 5.1 MHz frequency at the positive frequency side in the middle of the S.C. core filter (full rejection).....	173
Figure 4.4.85 : : the signal at IF frequency (1.1 MHz) at the negative frequency side (image) after the reconstruction filter	173
Figure 4.4.86 : the signal at IF frequency (1.1 MHz) at the negative frequency side (image) in the middle of the S.C. core filter “ IRR is shown to be -55 dB form this simulation ”	173
Figure 4.4.87 : the signal at 100 KHz frequency at the positive frequency side after the reconstruction filter.....	174
Figure 4.4.88 : the signal at 100 KHz frequency at the positive frequency side in the middle of the S.C. core filter.....	174
Figure 4.4.89 : the signal at 5.1 MHz frequency at the positive frequency side after the reconstruction filter (full rejection).....	174
Figure 4.4.90 : the signal at frequency of 14.1 MHz “alias frequency” in the middle of the S.C. core filter ... and we can see clearly how the AA filter eliminates the signal completely	175
Figure 4.4.91 : the signal at frequency of 14.1 MHz “alias frequency” after the reconstruction filter ... and we can see clearly how the AA filter eliminates the signal completely	175
Figure 4.4.92 : a zoomed version of the corners test	176
Figure 4.4.93 : the corners test results on the entire filter	176
Figure 4.4.94 : the noise measurement test bench.....	177
Figure 4.4.95 : the V2/Hz noise power spectral density for the pass band region	177
Figure 4.4.96 : the curve of the V/sqrt(Hz) , noise voltage magnitude “RMS” in the pass band region....	178
Figure 5.5.1; Simplified Radio Receiver Blocks	179

Figure 5.5.2; Feedback AGC Feed Forward AGC	179
Figure 5.5.3; PGA with resistive feed back	181
Figure 5.5.4; OTA with gm-Boosting	182
Figure 5.5.5; Multiplier VGA	182
Figure 5.5.6; Proposed Folded Cascode OTA in ^[4]	183
Figure 5.5.7; Variable Gain Amplifier by change Transconductance of source coupled.....	183
Figure 5.5.8; three stages used in PGA	184
Figure 5.5.9; gm/ID method.....	185
Figure 5.5.10; ID vs. Vgs	185
Figure 5.5.11; gm/ID vs. Vod.....	186
Figure 5.5.12; Proposed Folded Cascode OTA.....	186
Figure 5.5.13; Open Loop Gain and Phase Margin of Folded Cascode OTA TT	187
Figure 5.5.14; Figure 14 AC response of CMFB TT.....	187
Figure 5.5.15; Open Loop Gain and Phase Margin of Folded Cascode OTA SS	187
Figure 5.5.16; AC response of CMFB SS.....	188
Figure 5.5.17; Open Loop Gain and Phase Margin of Folded Cascode OTA FF	188
Figure 5.5.18; Figure 18 AC response of CMFB FF.....	188
Figure 5.5.19; Proposed PGA	189
Figure 5.5.20; 1-dB comparison point TT.....	189
Figure 5.5.21; Figure 21 NF of PGA TT	190
Figure 5.5.22; Proposed PGA AC response	190
Figure 5.5.23; PGA output gain vs. gain code word.....	191
Figure 5.5.24; Open Loop gain of Folded Cascode OTA	192
Figure 5.5.25; Loop gain of Folded cascode CMFB	192
Figure 5.5.26; PM of CMFB across corners	192
Figure 5.5.27; Max PGA gain across corners	193
Figure 5.5.28; NF variation across corners	193
Figure 5.5.29; 1-dB comparison point SS.....	193
Figure 5.5.30; 1-dB comparison point FF.....	194
Figure 5.5.31; Floor Planning of OTA	194
Figure 5.5.32; Layout of Folded Cascode OTA	194
Figure 5.5.33; Layout of PGA	195
Figure 5.5.34; Peak Detector transient response	196
Figure 5.5.35; Conventional RC Peak Detector	197
Figure 5.5.36; Op-amp plus diode based peak detector topology	197
Figure 5.5.37; Schematic diagram of the peak detector block.....	198
Figure 5.5.38; OTA plus current mirror closed-loop topology.....	199
Figure 5.5.39; current mirror closed loop peak detector	199
Figure 5.5.40; Folded Cascode OTA used in PD1	200
Figure 5.5.41; Proposed PD2 based on closed loop current mirror	200
Figure 5.5.42; Folded Cascode OTA used in PD2	201
Figure 5.5.43; Settling Time PD1 431KHz	201
Figure 5.5.44; Attack Time PD1 431 KHz.....	202
Figure 5.5.45; Release Time PD1 431KHz	202
Figure 5.5.46; Ripples in 100mV Amplitude	202
Figure 5.5.47; Loop gain of CMFB.....	203
Figure 5.5.48; Settling Time PD2 40MHz	204

Figure 5.5.49; Attack Time PD2 2MHz	204
Figure 5.5.50; Release Time PD2 2MHz	204
Figure 5.5.51; Open Loop Variation across corners.....	205
Figure 5.5.52; Loop gain of CMFB	205
Figure 5.5.53; PM variation across corners of PD.....	206
Figure 5.5.54; Floor planning of PD1	206
Figure 5.5.55; Floor Planning of PD2	207
Figure 5.5.56; Layout of PD1.....	208
Figure 5.5.57; Layout of PD2.....	208
Figure 5.5.58; Flash A/D used in AGC	209
Figure 5.5.59; Flash A/D simulation result.....	210
Figure 5.5.60; Total AGC system	211
Figure 5.5.61; Min Signal Received	211
Figure 5.5.62; Max Signal Received	212
Figure 6.6.1; Ideal transfer function of a 3-bit ADC.....	214
Figure 6.6.2 : ADC classification based on speed and resolution	216
Figure 6.6.3 : Schematic of a flash ADC	216
Figure 6.6.4 : Comparison of high-speed ADCs	217
Figure 6.6.5 : Quantization noise	217
Figure 6.6.6 : ADC and DAC characteristics	218
Figure 6.6.7 : offset and full scale error	219
Figure 6.6.8 : DNL.....	219
Figure 6.6.9 : INL	219
Figure 6.6.10 : INL transfer function	219
Figure 6.6.11 : Full scale error.....	220
Figure 6.6.12 : ERB	221
Figure 6.6.13 : SAR algorithm	222
Figure 6.6.14 : SAR ADC	222
Figure 6.6.15 : comparator	222
Figure 6.6.16 : comparator characteristics	223
Figure 6.6.17 : comparator component.....	223
Figure 6.6.18 : comparator offset	224
Figure 6.6.19 : static and dynamic offset.....	224
Figure 6.6.20 : open-loop comparator.....	225
Figure 6.6.21 : open loop gain	226
Figure 6.6.22 : static comparator.....	226
Figure 6.6.23 : dynamic comparator.....	227
Figure 6.6.24 : fully latched comparator	227
Figure 6.6.25 : rail to rail comparator	228
Figure 6.6.26 : proposed comparator	229
Figure 6.6.27 : SAR logic.....	229
Figure 6.6.28 : binary weighted DAC	230
Figure 6.6.29 : two-stage Dac	230
Figure 6.6.30 : C-2C DAC	231
Figure 6.6.31 : proposed dac	232
Figure 6.6.32 : algorithm.....	233
Figure 6.6.33 : KT/C noise	233

Figure 6.6.34 : table of mime-cap.....	234
Figure 6.6.35 : DNL, INL Matlab simulation	235
Figure 6.6.36 : SAR algorithm	235
Figure 6.6.37 : ideal switch	236
Figure 6.6.38 : ideal switch diagrams	236
Figure 6.6.39 : thermal noise	236
Figure 6.6.40 : charge injection	237
Figure 6.6.41 : dessouky switch	239
Figure 6.6.42 : sadollahy switch.....	240
Figure 6.6.43 : proposed switch.....	241
Figure 6.6.44 : comparator	243
Figure 6.6.45 : simulation	243
Figure 6.6.46 : settling time and delay	243
Figure 6.6.47 : Monte Carlo simulation and offset.....	244
Figure 6.6.48 : corner simulation.....	245
Figure 6.6.49 : SAR ADC	245
Figure 6.6.50 : algorithm.....	246
Figure 6.6.51 : PSD simulation	247
Figure 6.6.52 : monte-carlo simulation.....	249
Figure 6.6.53 : switch layout.....	251
Figure 6.6.54 : reference buffer layout.....	252
Figure 6.6.55 : comparator layout	252
Figure 6.6.56 : Capacitor array.....	253
Figure 7.7.1 : the input and the output of LNA.....	256
Figure 7.7.2 : the Mixer output.....	256
Figure 7.7.3 : the NF of both LNA and mixer	257
Figure 7.7.4 : the filter output when the RF-input signal is in the channel side	257
Figure 7.7.5 : I&Q signals in the input and the output of the filter when the input RF signal is at the Image side	258
Figure 7.7.6 : the schematic of the AGC-ADC interface.....	258
Figure 7.7.7 : the AGC output while interfacing	259
Figure 7.7.8 : the signals throughout the ADC.....	259
Figure 7.7.9: ADC signal tracking	260
Figure 7.7.10 : the schematic of the all Rx.....	260
Figure 7.7.11 : the outputs of the each block of the Rx chain.....	261
Figure 8.1 Integer N-frequency synthesizer	263
Figure 8.2 VCO linear tuning	263
Figure 8.8.3 : fractional N synthesizer	265
Figure 8.8.4: second order loop filter	265
Figure 8.5: Loop stability.....	268
Figure8.8.6: phase noise transfer function of all blocks.....	269
Figure 8.8.7 : reference oscillator phase noise.....	270
Figure 8.8.8: lesson equation.....	270
Figure 8.8.9 : Reference oscillator in the output phase noise.....	270
Figure 8.8.10 : Charge pump noise transfer function	271
Figure 8.8.11 : Charge pump noise form	272
Figure 8.8.12 : Charge pump phase noise	272

Figure 8.8.13 : Charge pump noise contribution in the output.....	273
Figure 8.8.14 : VCO noise transfer function.....	273
Figure 8.8.15 : VCO phase noise	274
Figure 8.8.16 : VCO phase noise contribution in the output.....	275
Figure 8.8.17 : SDM noise transfer function.....	275
Figure 8.18: SDM noise psd	277
Figure 8.8.19 : Reasons for reference spurs	279
Figure 8.8.20 : CPPSIM parameters	279
Figure 8.8.21 : CPPSIM noise results.....	280
Figure 9.1:crystal oscilltor.....	281
Figure 9. 92.: Crystal Oscillator	281
Figure 9.3: Phase Noise.....	283
Figure 9.9.4: Phase Noise.....	283
Figure 9. 95.: model impedance-imaginary part.....	293
Figure 10.10.1 : XOR waveform	306
Figure 10.10.2 : SR latch phase detector	306
Figure 10.10.3 : Tri state PFD.....	307
Figure 10.10.4 : Tri state PFD finite state machine.....	307
Figure 10.10.5 : Tri state PFD signals	308
Figure 10.10.6 : D flip flop implementation.....	308
Figure 10.10.7 : tri- state PFD	309
Figure 10.10.8 : D flip flop (connected with labels from inside)	310
Figure 10.10.9 : CMOS NOR gate	311
Figure 10.10.10 : CMOS NAND gate.....	312
Figure 10.10.11 : PFD transfer function.....	312
Figure 10.10.12 : Switch in drain single ended charge pump.....	314
Figure 10.10.13 : Switch in gate charge pump	314
Figure 10.10.14 : Switch in source charge pump.....	315
Figure 10.10.15 : NMOS differential charge pump.....	315
Figure 10.10.16 : Current steering charge pump.....	316
Figure 10.10.17 : Current steering charge pump with unity gain buffer.....	316
Figure 10.10.18 : PFD &CHARGE PUMP dead zone	317
Figure 10.10.19 : Current mismatch effect.....	318
Figure 10.10.20 : Timing mismatch.....	318
Figure 10.10.21 : Charge sharing	319
Figure 10.10.22 : SWITCH IN SOURCE CHARGE PUMP CIRCUIT	320
Figure 10.10.23 : DC mismatch	321
Figure 10.10.24 : PFD charge pump transfer function.....	321
Figure 10.10.25 : CHARGE PUMP output RMS noise summary	322
Figure 10.10.26 : Output noise of charge pump.....	322
Figure 11.11.1 VCO	323
Figure 11.11.2 spectrum without and with phase noise	324
Figure 11.11.3 phase noise effect in time domain	324
Figure 11.11.4 Phase noise specification.....	324
Figure 11.11.5 phase noise effect at the RX	325
Figure 11.11.6 output frequency vs. vctrl.....	325
Figure 11.11.7 feedback model	326

Figure 11.11.8 one port model	327
Figure 11.11.9 3 stage ring oscillator	327
Figure 11.11.10 ideal LC tank circuit	328
Figure 11.11.11 CS amplifier with a tank circuit	329
Figure 11.11.12 CMOS cross-coupled LC oscillator	329
Figure 11.11.13 PMOS cross-coupled LC oscillator	329
Figure 11.11.14 NMOS cross-coupled LC oscillator	329
Figure 11.11.15 Q of symmetrical inductor	332
Figure 11.11.16 Q of the standard (STD) inductor $Q=9 @ 1.8G$ & $Q=9.8 @ 2G$	333
Figure 11.11.17 Q of the Centre tap (CT) inductor $Q=16.67 @ 1.8G$ & $Q=17.48 @ 2G$	333
Figure 11.11.18 inductor series resistance (ohm) $R_s = 1.952$ ohm	334
Figure 11.11.19 equivalent circuit for Q calculation	334
Figure 11.11.20 KVCO variation (solid) across the compliance range KVCO varies from 5 MHz/v to 40 MHz/v	335
Figure 11.11.21 Capacitor coupled Varactor	336
Figure 11.11.22 show the effect of R on the Phase noise for $R = 500$ ohm, P.N @1 MHz = -119.4 dbc/Hz for $R = 1K$, P.N @1 MHz = -122.1 dbc/Hz for $R = 2K$, P.N @1 MHz = -121.1 dbc/Hz for $R = 3K$, P.N @1 MHz = -123.2 dbc/Hz	337
Figure 11.11.23 KVCO variation KVCO varies from 75 MHz/v to 95 MHz/v average = 90 MHz/v	337
Figure 11.11.24 VCO core schematic	339
Figure 11.11.25 source follower topology	340
Figure 11.11.26 source follower output vs. transistor width	341
Figure 11.11.27 source follower output bandwidth	341
Figure 11.11.28 input (dotted) & output (solid) of the buffer	342
Figure 11.11.29 counter based frequency calibrator	342
Figure 11.11.30 output frequency calibration	343
Figure 11.11.31 Buffer input (dotted) & buffer output (solid)	344
Figure 11.11.32 covered frequency range in FF, TT, and SS	345
Figure 11.11.33 tuning curve sub-band for the required range in TT	346
Figure 11.11.34 tuning curve (dotted) & VCO gain for sub-band 1 (TT) average $K_{vco} = 90$ MHz/v	346
Figure 11.11.35 Tuning curve (dotted) vs. VCO gain (solid) for sub-band 2 average $K_{VCO} = 95$ MHz/v ...	347
Figure 11.36 KVO variation with tempretur	347
Figure 11.11.37 P.N.(TT) -123.2 dbc/hz @ 27^0 -120.5 dbc/hz @ 85 -125.7 dbc/hz @ -40	348
Figure 11.38 PN (SS) -120.2dbc/hz(85^0),-122.7dbc/hz(27^0),-125.2dbc/hz (-40^0)	348
Figure 11.11.39 PN(FF) -121.5 dbc/hz (85) -124.5 dbc/hz (27) -126.3 dbc/hz (-40)	349
Figure 11.40: covered sub-bands	350
Figure 11.11.41 VCO core & Buffer & cap bank	351
Figure 11.11.42 VCO Core schematic	352
Figure 11.11.43 Capacitor bank Schematic	353
Figure 11.11.44 Buffer Circuit	354
Figure 12.12.1: Synchronous Divider	356
Figure 12.12.2: A noise spectrum at the output of the divider	357
Figure 12.12.3: Divide by two	358
Figure 12.12.4: Divide by three circuits	359
Figure 12.5: Fully Programmable Divider based on Dual Modulus Prescaler (PULSE SWALLOW)	359
Figure 12. 12.6: A divide by 2/3 with a Modin control and Modout signals added	360
Figure 12. 12.7: Waveforms of divide by 2/3 circuit, assuming that Modin is high	360

Figure 12.12.8: two cascaded divide by 2/3 stages	361
Figure 122.12.9: A divide by 4/7 circuit made from two 2/3 stages shown at latch Level	361
Figure 12.10: Waveforms in divide-by-7mode (R1 = high and R2 = high)	362
Figure 12. 12.11: Waveforms in divide-by-5 mode (R1 = high; R2 = low).	362
Figure 12.12: MMD Architecture	362
Figure 12.13: A generic MMD architecture	363
Figure 12.14 : MMD chosen as Feedback Divider with Sigma-Delta interface	365
Figure 12.12.15: quadrature output generator Figure 12.16: : Fixed Divide by 2 with Mixer and PA.	365
Figure 12.17: Static latches	366
Figure 12.18: a) CML D latch – b) regeneration mode Equivalent circuit - c) waveform	366
Figure 12.12.19: CML NAND/AND Figure 12.12.20: CML AND combined with Latch	367
Figure 12.21: TSPC Flip FLOP.....	367
Figure 12.22: Power consumed VS. Frequency	368
Figure 12.23: Quadrature outputs Generator	369
Figure 12.24: Quadrature Output generator from inside.....	370
Figure 12.12.25: CML Latch used to generate IQ signals.....	372
Figure 12.26: CML Buffer	373
Figure 12.12.27 : Quadrature Bias and Poly current	373
Figure 12.28: Transient Simulation of input, IQ outputs, Frequencies , Delay \equiv IQ Mismatch $=T/4 = 0$ Mismatch	374
Figure 12.29 : Monte Carlo Simulation after Adding bias Cells	374
Figure 12.12.30: I and Q signals across corners , and max swing across corners is 393 mV from 350 mV	375
Figure 12.12.31: overall Feedback Divider.....	376
Figure 12.32: Feedback divider construction	376
Figure 12.33: A CML MMD Construction	377
Figure 12.34 : CML latch	378
Figure 12.35: CML NAND	378
Figure 12.12.36: CML LEVEL SHIFTER	379
Figure 12.377: CML TO CMOS Converter.....	379
Figure 12.38: Inverter as ampilifer.....	381
Figure 12.391: Overall CMOS MMD DIVIDER	382
Figure 12.12.40: INSIDE ANY DIVIDER CELL	382
Figure 12.41: CMOS SR LATCH	383
Figure 12.42: Combined AND gate with SR latch with sizing of the reference latch with minimum length	383
Figure 12.43: CMOS NAND	384
Figure 12.12.44: CMOS NOT	384
Figure 12.12.45 : VCO input , output at max Division Ratio & their Frequencies	385
Figure 12.46: Variable Division Ratio by Varying Digital Input at VCO = 1.83GHZ	385
Figure 12.47: Division Ratio of 72 worst-case Corner VCO=2.3GHZ output=31.9444MHZ.....	387
Figure 13.13.1: Transfer Characteristics of a multi bit quantizer	389
Figure 13.13.2: The quantization error as a function of the input.	390
Figure 13.13.3: Quantization noise reduction by increasing OSR: (up) low sampling frequency, and (down) higher sampling frequency.	391
Figure 13.13.4: Feedback model of a noise-shaping system.....	392

Figure 13.13.5: Output of an oversampling system with and without feedback.....	392
Figure 13.13.6: Fractional-N frequency synthesizer with a SD modulator.....	393
Figure 13.13.7: Discrete time model of the MASH 1-1-1 $\Sigma\Delta$ modulator.....	394
Figure 13.13.8 : Discrete time model of the 3rd order single loop modulator using CIFB.....	395
Figure 13.13.9: Discrete time model of the 3rd order single loop modulator using CIFF.....	395
Figure 13.13.10 : : conversion of a first order SD in an ADC to a first order SD modulator in a fractional –N accumulator: (a) modulator for an ADC, (b) modulator in a fractional-N accumulator, (c) Equivalent Circuit	395
Figure 13.13.11: : Triple loop MASH 1-1-1 SD modulator	396
Figure 13.13.12: an alternative topology utilizing 3 accumulators and delay elements for simple hardware implementation	396
Figure 13.13.13: a third order MASH 1-2 topology	398
Figure 13.13.14: Conceptual drawing of the single-stage SD modulator.....	398
Figure 13.13.15: A third-order, single-stage SD modulator.....	399
Figure 13.13.16 : A fractional accumulator with dithering scheme.	400
Figure 13.13.17 : A first-order DSM with a single-bit variable modulus quantizer	401
Figure 13.13.18: Output levels for a sinusoidal input of amplitude 1.2 V.....	402
Figure 13.13.19: Output Levels for a d.c. input of 0.125 V	402
Figure 13.13.20 : Power Spectral Density when using LSB dithering	402
Figure 13.13.21: Output Levels for d.c. input of 0.5 V.....	402
Figure 13.13.22 : Output Levels for a sinusoidal input of 0.5 V.....	402
Figure 13.13.23: Output power spectral density of MASH 1-1-1 without dithering.....	403
Figure 13.13.24: Verified VHDL testbench output.....	403
Figure 13.13.25: Synthesized modulator using Xilinx.....	403
Figure 14.14.1 : Upconverter with balun connection ^[1]	407
Figure 14.14.2 : Feedback in a single ended PA due to bond wire ^[1]	407
Figure 14.14.3 : less problematic situation in a differential PA ^[1]	407
Figure 14.14.4 : Cascoded PA ^[1]	408
Figure 14.14.5 : Voltage & current waveform	408
Figure 14.14.6 : General power amplifier model.....	408
Figure 14.14.7 : Current & Voltage waveforms	410
Figure 14.14.8 : Class B power amplifier	410
Figure 14.14.9 : Class C power amplifier	411
Figure 14.14.10 : Current & Voltage waveforms	411
Figure 14.14.11 : Efficiency VS condition angle ^[1]	412
Figure 14.14.12 : Output power VS condition angle [1]	412
Figure 14.14.13 : Output stage with switching transistor ^[1]	413
Figure 14.14.14 : Voltage & current waveforms ^[2]	414
Figure 14.14.15 : Class D power amplifiers ^[2]	414
Figure 14.14.16 : Class E load ^[1]	415
Figure 14.14.17 : Voltage & Current waveforms ^[2]	415
Figure 14.14.18 : Class E power amplifier.....	415
Figure 14.14.19 : Class F power amplifier.....	416
Figure 14.14.20 : Voltage and current waveforms	416
Figure 14.14.21 : Inverse class F current & voltage waveform ^[2]	417
Figure 14.14.22 : Switching Amplifiers	418
Figure 14.14.23 : PA sub-blocks	419

Figure 14.14.24 : Bias Circuit	419
Figure 14.14.25 : Test-bench	420
Figure 14.14.26 : Optimization curves	420
Figure 14.14.27 : $I_d/(W/L) / (g_m/I_d)$ Curve	421
Figure 14.14.28 : Pre-amplifier	421
Figure 14.14.29 : Preamplifier test bench	422
Figure 14.14.30 : Induction tuned load PA driver	422
Figure 14.14.31 : Class F driver	423
Figure 14.14.32 : Output voltage waveform.....	423
Figure 14.14.33 : Driver	424
Figure 14.14.34: Switch.....	425
Figure 14.14.35 : Output stage	426
Figure 14.14.36: Class F termination network.....	427
Figure 14.14.37 : Inverse class F termination	428
Figure 14.14.38 : Output stage core	430
Figure 14.14.39 : Output node bond wire model	430
Figure 14.14.40 : Supplies and ground bond wire model.....	430
Figure 14.14.41 : PA test-bench.....	431
Figure 14.14.42 : PA schematic.....	431
Figure 14.14.43: Pout Vs Pin	432
Figure 14.14.44 : Power spectrum measurements	432
Figure 14.14.45 : Drain efficiency	433
Figure 14.14.46 : Power aided efficiency.....	433
Figure 14.14.47 : Power gain	434
Figure 14.14.48 : 1-dB compression point.....	434
Figure 14.14.49 : Kf stability factor	435
Figure 14.14.50 : B1f stability measurements	435
Figure 14.14.51: Voltage and current waveforms	436
Figure 14.14.52 : Output power & PAE VS supplies variation	436
Figure 14.14.53 : Output power VS frequency	437
Figure 15.15.1 : VCO with Feedback Divider.....	441
Figure 15.2 : VCO with quadrature generator (IQ) divide by 2.....	441
Figure 15.15.3 Transient simulation of VCO and IQ divide by 2 and their Freq	442
Figure 1515.15.4 : Transient simulation of VCO and IQ divide by 2 and their Freq at worst case corner .	442
Figure 15.15.5 : Transient simulation of VCO and Feedback divider with variable division ratio.....	443
Figure 15.6 : Transient simulation of VCO and Feedback divider with variable division ratio At worst case corner.....	443
Figure 1515.15.70 : Pre-amplifier	446
Figure 1515.15.82 : Transient analysis 2.....	447
Figure 1515.15.93 : Power spectrum.....	447
Figure 1515.15.10 : Pout Vs Pin	448
Figure 1515.15.115 : Circuit schematic for VCO with Divider & PA	449
Figure 15.15.124 : VCO transient output.....	449
Figure 15.15.137 : PA Output	450
Figure 1515.15.148 : Transient output of VCO, Divider & PA	450

List of Tables

Table 2.1 : shows the specs that must be achieved across all corners	57
Table 2.2 : shows the achieved specs in the typical case	57
Table 4.3 : comparison between main filters models	127
Table 4 : the parameters of the core OTA	167
Table 5.5; Comparison between Feedback and feed forward AGC Loop.....	180
Table 5.6; Required Specifications for PGA	183
Table 5.7; Advantages and disadvantages of proposed PGA topology	184
Table 5.8; Required Specification for Peak Detector	199
Table 5.9; Comparison between PD1 and PD2	204
Table 6.10 : monte-carlo simulation.....	235
Table 2: Phase Noise after Current Control Loop	Error! Bookmark not defined.
Table 3: Comparison between with and without Current loop.....	Error! Bookmark not defined.
Table 4: Summary of results	302
Table 11.15 LC oscillator vs. Ring oscillator	330
Table 11.16 tank circuit values at 1.8Ghz	338
Table 11.17 summary of the result.....	350
Table 14.18: Max efficiency and Output power capability.....	416
Table 14.19: Comparison between different topologies.....	417
Table 14.20: Specification VS achieved	437
Table 14.21: Current consumption of each stage.....	437
Table 14.22 : DC programmability power levels (8 levels recommend).....	438
Table 14.23 : Corners simulation	438

1 Receiver System Overview

1.1 RX specifications

1.1.1 Maximum input level

Max input level is the max. power of the received signal to achieve the required SNR. We must put margin to this value to avoid transmitter leakage.

1.1.2 SNR

Signal to noise ratio is the power signal to total rx noise. Depends on the modulation scheme and data rate. Required by digital system.

1.1.3 Sensitivity

Minimum signal power that can be detectable by the receiver. Noise and non-linearity affect sensitivity.

We define acceptable quality as a sufficient SNR.

1.1.4 Dynamic range

Maximum input level that the receiver can tolerate divided by minimum input level that it can detect (sensitivity).

Also dynamic range can be define as SFDR. DR is limited by compression at the upper end and noise at the lower end.

1.1.5 Non-Linearity

A system is linear if its output can be expressed as a linear combination (superposition) of responses to in individual parts For a linear system $Y1(t)=F(X1(t))$, $Y2(t)=F(X2(t))$, $aY1(t)+bY2(t)=F(aX1(t)+bX2(t))$.

1.1.5.1 Time invariance

A system is time invariant if a time shift in its input results in the same time shift in its output

$$Y(t)=F(X(t)) \quad > \quad Y(t-T)=F(X(t-T)).$$

A system is called memory less or static if the output doesn't depend on the past values of its input. a system is called dynamic if the output depends on the past values of its inputs or outputs.

1.1.5.2 1-dB compression point

In the presence of a blocker Gain is reduced due to nonlinearity , This impacts the weak desired Signal. Typically specified by the 1 dB Compression point.

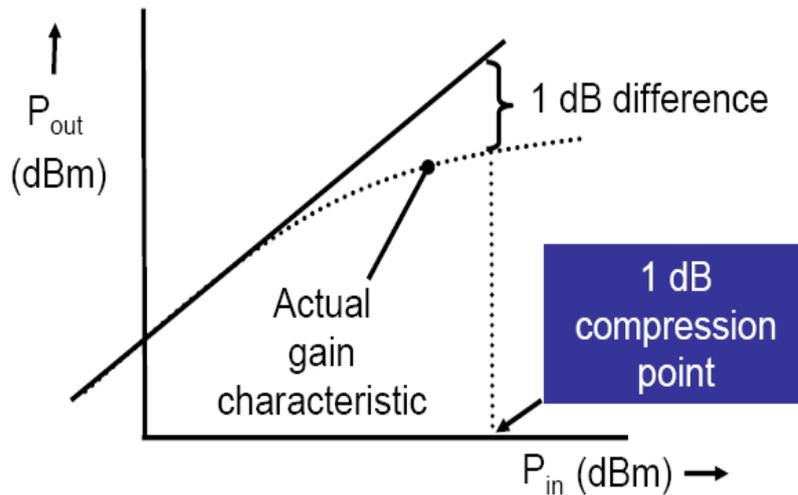


Figure 1.1 : 1-dB compression point

1.1.5.3 Harmonic distortion

If a sinusoid is applied to a non linear system the out put generally exhibits frequency components that are integer multiples of the input frequency but in many RF circuits harmonic distortion is unimportant or an irrelevant indicator of the effect of nonlinearity. For example an amplifier operating at 2.4 GHz has second harmonic at 4.8 GHz which is greatly suppressed if the circuit has narrow bandwidth.

1.1.5.4 Cross modulation

Another phenomena that occurs when a weak signal and a strong interferer pass through a nonlinear system is the transfer of modulation from the interferer to the signal. This is a problem for amplitude modulation but not for frequency modulation.

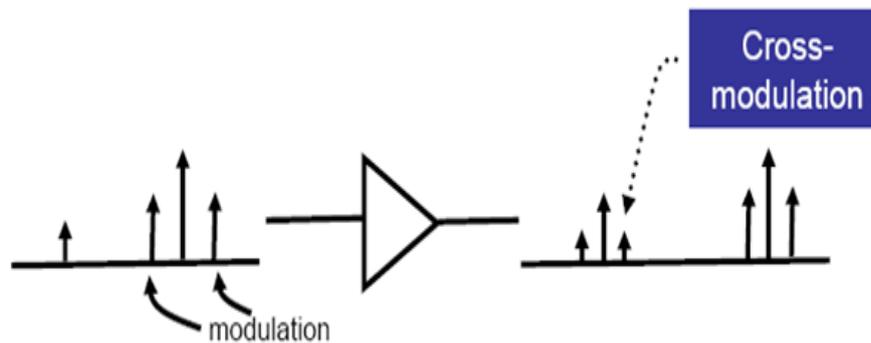


Figure 1.1.1 : cross modulation

1.1.5.5 Intermodulation

IMD occurs as two or more signals pass through a two-port network with a nonlinear transfer function. The spectrum at the output of the device is comprised of the original signals and additional spurious signals. The additional spurious signals can cause interference within the original system or in other systems. When the spurious signals are of sufficient amplitude, they can overpower the signal of interest, resulting in interference and, in extreme cases, loss of transmitted information, such as voice, data, or video. The undesirable effects of IMD can be mitigated, by improving the linearity of system components

(amplifiers, other semiconductors, and even passive elements), which cause IMD. The need to measure inter modulation is important a common method of IM characterizations is the two tone test where by two pure sinusoids of equal amplitudes are applied to the input; the amplitude of the output IM normalized to the fundamentals of the output.

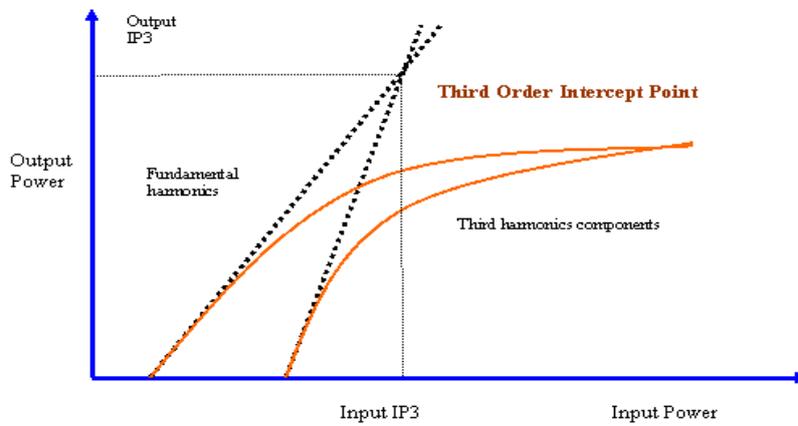


Figure 1.1.2 : iip3

1.1.5.6 IIP2

the second order distortion component are far from the desired band and has no effect on the receiver linearity but in direct conversion receivers the second order intermodulation can feed through the mixer and disturb the signal at the baseband . For differential design the IIP2 is very high for the whole receivers and the second order distortion has very little effect, and can be removed by a ac coupling capacitors.

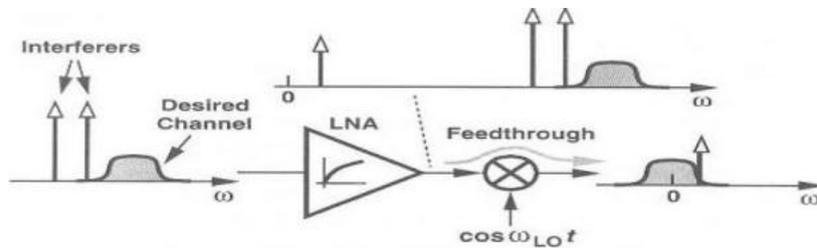
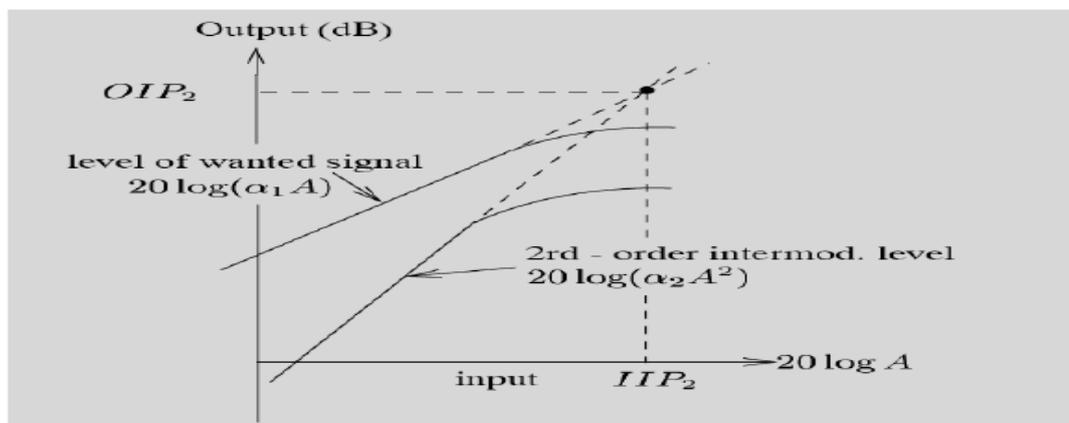


Figure 1.1.3 : IIP2



The IIP2 is the input power where the wanted signal and the second order intermodulation signal are equal (extrapolated point).

Figure1.1.4 : definition iip2

1.1.6 Cascaded nonlinear stages

This is if each stage in cascaded has a gain greater than unity the nonlinearity of the latter stages becomes increasingly more critical because the third intercept point of each stage is equivalently scaled down by the total gain of preceding stage. generally: $1/AIP_3 = 1/AIP_{3,1} + a_1^2 / (1/AIP_{3,2}) + (a_1 * a_2)^2 * (AIP_{3,3})$

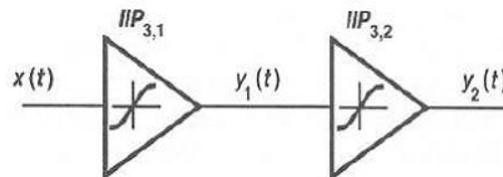


Figure 1.1.5 : cascaded devices

1.1.7 Noise

1.1.7.1 Thermal noise

Due to random motion of carriers due to thermal agitation. It is white noise. Noise power = $KT = -174 \text{ dbm/hz}$.

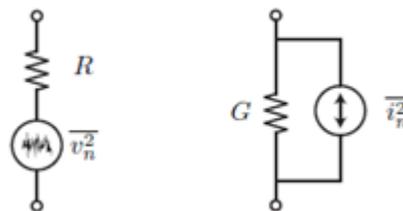


Figure 1.1.6 : thermal noise

- For MOSFETS:

- $\overline{i_{ind}^2} = 4KT\gamma g_{d0}\Delta f$.
- g_{d0} is the drain source conductance at zero V_{ds} .
- $\gamma = 1$ at zero V_{ds} and about $\frac{2}{3}$ at saturation for long channel.
- Gate noise: The fluctuating channel potential couples capacitively with the gate causing noisy gate current.

- $\overline{i_{ng}^2} = 4KT\delta g_g \Delta f$
 where: $g_g = \frac{\omega^2 C_{gs}^2}{5g_{d0}}$ and $\delta \cong \frac{4}{3}$ for long channel.

1.1.7.2 Flicker noise

Also called 1/f noise. It decreases with frequency. For minimum flicker noise decrease the voltage across the resistor.

1.1.7.3 Noise figure

It describes the degradation of SNR. It is the total output noise to output noise due to input.

1.1.8 SFDR

Spurious free dynamic range. Difference in dB between the max. input to the max. spur in band.

1.2 RX architecture

1.2.1 Super heterodyne

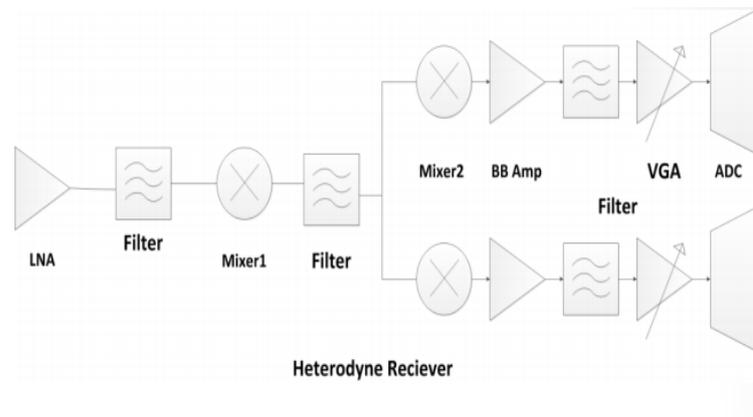


Figure 1.1.7 : super heterodyne

1.2.1.1 Image problem

There are a lot of signals transmitted and if desired signal and any signal far by $2 \cdot W_{if}$ so its called IMAGE. $\omega_{im} = \omega_{in} + 2 \cdot W_{if} = 2 \cdot \omega_{lo} - \omega_{in}$.

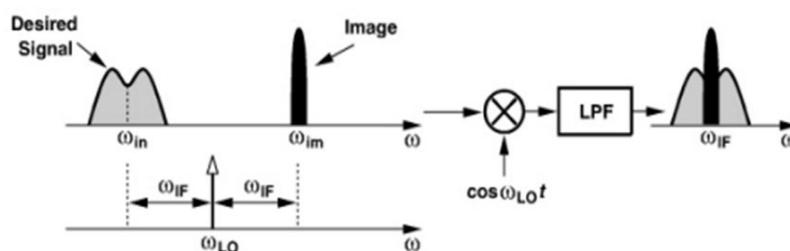


Figure 1.1.8 : image problem

We must use image reject filter before mixer and after LNA WHY?

We need first to increase signal gain because the filter make small loss in the desired signal and large attenuation in the image band. To maximize image rejection, it is desirable to choose a large value of W_{if} , large difference between desired signal and image one, but there is a trade-off between imagerejection and channel selection (filter)

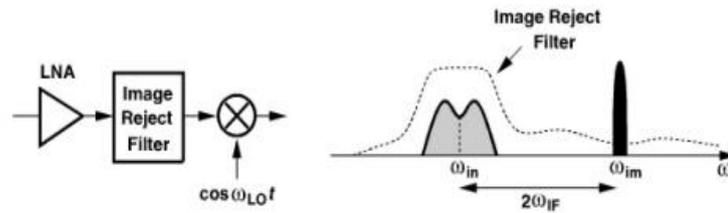


Figure 1.1.9 : image filter

More Filters => More Images => More Filters.

- High-Q filters are hard to build and expensive.
- Use more filters each with lower Q.
- But that requires more mixing in the receive chain which leads to image problems and needs more filters...
- we can chose the ω_{LO} smaller than the desired band (low side injection) or larger than it (high side injection). This way we can choose the position of the image frequency to be in one side or the other. If we know that there are less interferers in one of the two sides of our band, we'll fix the ω_{LO} in order to have the image frequency in the side with less interferers .

1.2.2 dual down conversion

This is ideal dual downconversion , due to non-ideality of mixer (specially) and others there is non-linearity and noise .

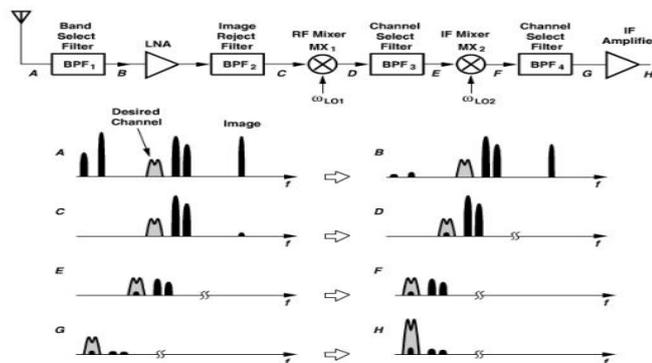


Figure 1.1.10 : dual conversion RX

• **Advantages :**

- Excellent selectivity: selection of small signals in the presence of strong interfering signals (interferers).
- No dc offset problem (no zero if).

• **Disadvantages :**

- Image problem.
- Off-chip filters consume more power , area.

- Power dissipation.
- trade-off between image rejection and channel selection.

1.2.3 Image rejection RX

- Heterodyne must deal with the image and mixing spurs .
- Direct conversion must deal with even-order distortion and flicker noise .
- Image – reject architecture suppress the image without filtering .
- So avoiding the trade – off between image – rejection and channel selection .
- Avoiding challenges between sensitivity and selectivity .

1.2.3.1 90 phase shift

- The shift by 90 is done using ‘ Hilbert transform ‘ .
- The positive frequency contents are multiplied by -j , and negative contents by +j .
- 90 phase shift can be done using RC-CR network

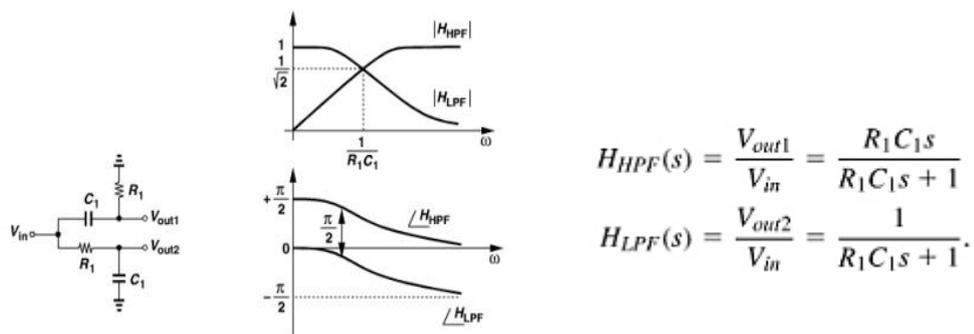


Figure 1.1.11 : 90 phase shift

Angle(hpf) = $\pi/2 - \tan^{-1}(R \cdot C \cdot \omega)$. Angle(lpf) = $-\tan^{-1}(R \cdot C \cdot \omega)$. Angle(hpf) - angle(lpf) = $\pi/2$ at all frequencies close to $\omega = 1/(R \cdot C)$. So, we can consider V_{out2} as a Hilbert transform of V_{out1} at freq. close to $1/(R \cdot C)$. We can use also Hilbert

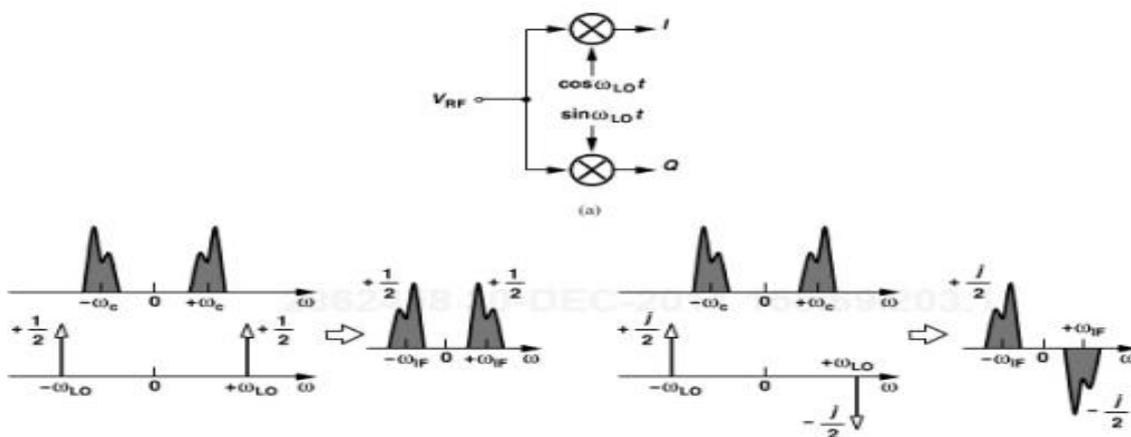


Figure 1.1.12 : algorithm

1.2.3.2 Hartley architecture

- It is based on downconverted each of signal and image , then suppress high component frequency then shift one of them by 90 and adding together so suppress image as picture mention.
- By adding $X_a + X_c$, it is appear that image will be suppressed .
- There are some drawbacks on this architecture will be mentioned now .

The phase shift actually can't be done by shifting 90 but , shift +45 in one path and -45 in another path because shifting by 90 difficult while circuit component vary with process and temp.

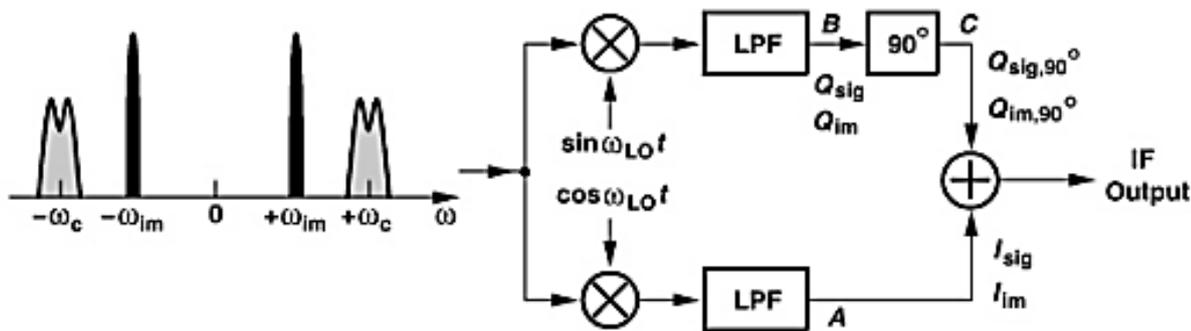


Figure 1.1.13 : hartley RX

1.2.3.2.1 Drawbacks

- Drawbacks of Hartley is mainly due to mismatch.
- The two local oscillator must be identical (quadrature sure).
- Mismatch occur due to amplitude error ϵ and phase error $\Delta\theta$.
- If we put sin lo as a ideal and there is mismatch in cos as $(1 + \epsilon) \cos(\omega_{LO}t + \Delta\theta)$.
- We define IRR as ‘‘ image – rejection ratio ‘‘ which result from dividing ratio of the i/p to that of o/p.
- For $\Delta\theta \ll 1$ rad and $\epsilon \ll 1$, so IRR will be $IRR = 4/(\epsilon^2 + \Delta\theta^2)$.
- With various mismatch arising in the LO , the IRR falls below 35 dB.
- Another critical drawback come from values of R and C in RC-CR network due to temp. and other parameters.
- Another drawback come from change the frequency $(1/R \cdot C)$ due to R and C are variable with its parameters.
- Another drawback from the noise of the adder , which realized by diff. pair , which convert voltages to current and sum them then convert the result to voltage.

1.2.3.3 Waver architecture

- Our analysis of the Hartley architecture has revealed several issues due to RC-CR network

- The weaver receiver avoids these issues by using another 2 mixer and lpf instead of using phase shifter.
- Continue the equation we derive that

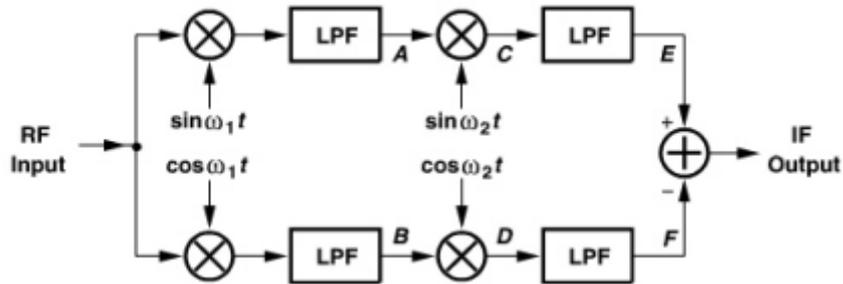


Figure 1.1.14 : waver architecture

- So the image removed
- The weaver receiver avoids issues related to RC-CR network : resistance , caps variation , degradation of IRR , attenuation and noise.
- For image – rejection well above 40 dB , it must be calibration to cancel the gain and phase mismatches.

1.2.4 Direct-conversion(zero-IF)

A direct-conversion receiver , also known as homodyne ,or zero-IF receiver.Wanted channel directly converted to baseband(0 Hz).

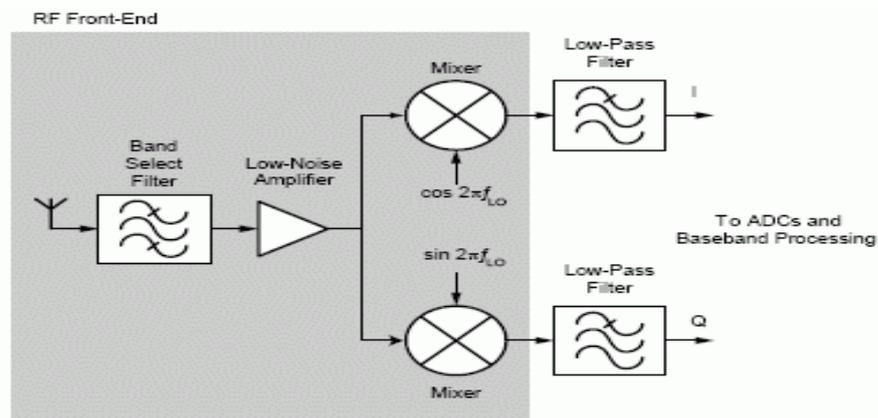


Figure 1.1.15 : direct conversio RX

- For phase and frequency modulated signals, direct-con-ersion requires I/Q signals.
- In current cellular systems where signals are either frequency- or phase-modulated, direct down-conversion must provide quadrature outputs so as to avoid loss of information, since the two sidebands of the RF spectrum contain different phase information.
- In direct-conversion we convert the frequency to baseband 0 hz

1.2.4.1 1/f noise(flicker)

- $1/f$ noise = $k/(W*L*C_{ox}*F)$.
- At low frequency this noise is high .
- To reduce it , increase the value of L.
- Due to non-ideality of silicon formation , there are traps that catch slow electrons(low frequency) but at high frequency ,electrons move by high speed and can't be trapped in silicon.
- Conclusion of the above , $1/f$ noise affect only in zero frequency or low frequency

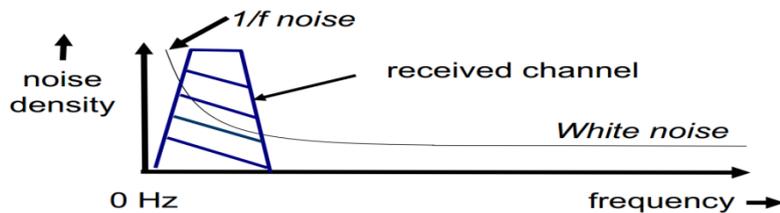


Figure 1.1.16 : 1/f noise

1.2.4.2 LO leakage

- Due to substrate , in low frequency there is a capacitor between LO and antenna which make a leakage.
- Due to LNA matched $S_{22}=1$ so , this signal reflected again to mixer.
- Multiplying two sin wave together result in a DC OFFSET .
- $\sin(\omega_{lo}) * \sin(\omega_{lo}) = 0.5 * (1 + 2 * \cos(2 * \omega_{lo}))$.
- If this dc is for example 1mv and multiply by gain of 1000 so it is 1v !!.
- Any corruption of the DC level can move the bias point of the blocks of the receiver
- DC offset also make ADC high performance.

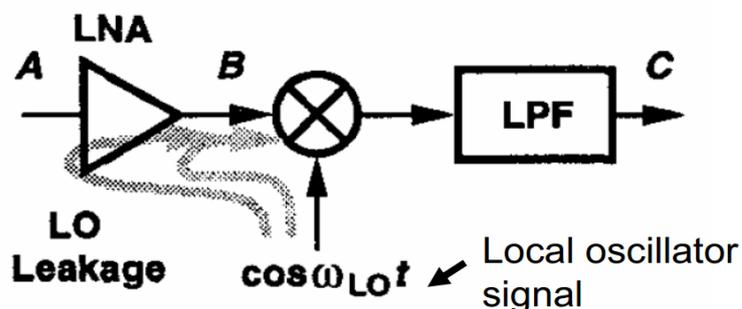


Figure 1.1.17 : lo leakage

In order to minimize this problem of LO leakage and re-radiation, it is important to use differential LO and RF inputs to the receiver IC to cancel out common mode signals. In addition, LO leakage is further reduced by fully integrating the RF VCO tank on chip.

1.2.4.3 DC offset

Dc offset due to leakage or another reasons that may saturate the receiver. Can be solved by many architecture but each has its drawbacks

1.2.4.3.1 Capacitive Coupling

- Requires a large capacitor

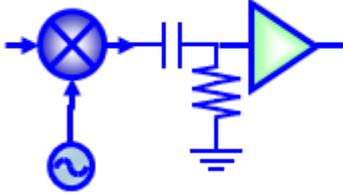


Figure 1.1.18 : capacitive coupling

1.2.4.3.2 Negative Feedback

- Nonlinear

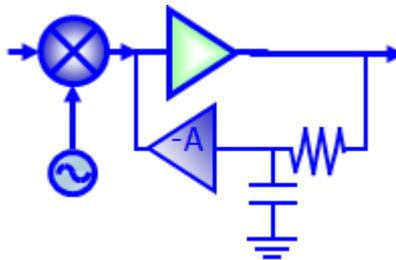


Figure 1.1.19 : negative feedback

1.2.4.3.3 TDMA Offset Cancellation

- Requires a large capacitor

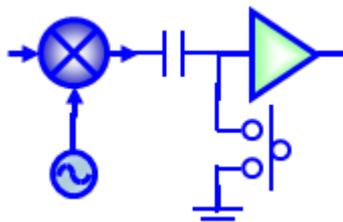


Figure 1.1.20 : offset cancellation

1.2.4.4 Even-Order Distortion

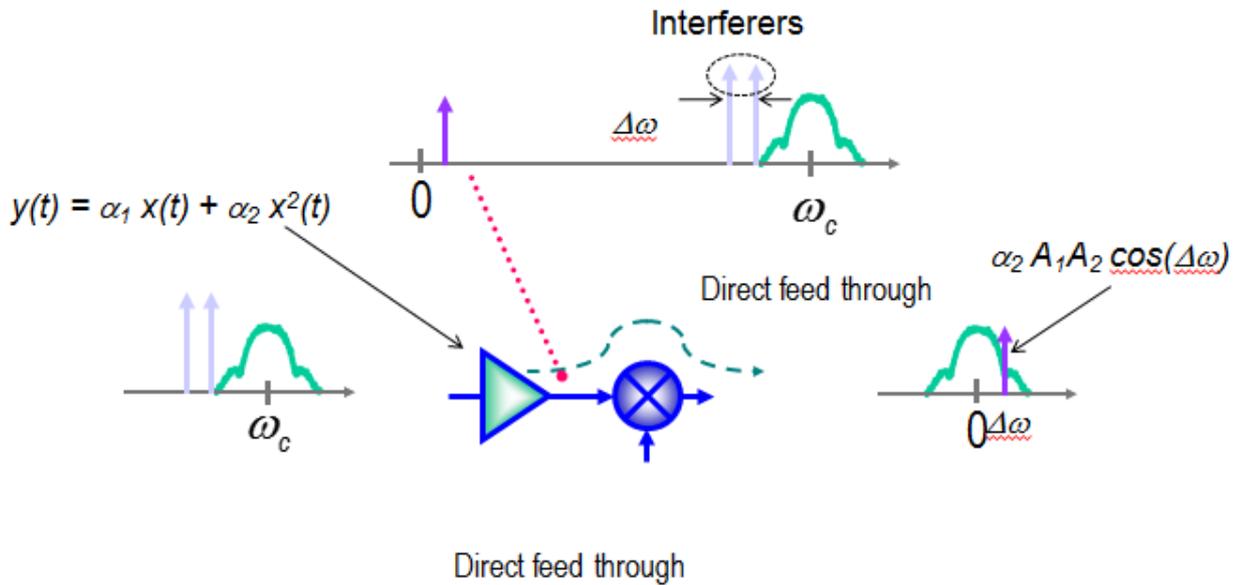


Figure 1.1.21 : even order distortion

1.2.4.5 Advantages :

- No image problem since in baseband.
- Need for LPF instead of HPF and low quality so easy to be on-chip.
- Easier to integrated.

1.2.4.6 Dis-advantages

- Dc offset problem due to baseband.
- LO leakage .
- 1/f noise in 0 Hz.
- I/Q mismatches.
- Flicker noise
- Distortion due to LO

1.2.5 Low IF RX

- Low if receiver combines the advantages of both super-heterodyne and direct-conversion.
- Suitable for integration since providing on-chip image rejection , and channel selection can be done using low Q filter.
- Unlike direct conversion , low if not sensitive to dc offset or LO leakage (in low frequency not 0).
- Image is suppressed in a Low-IF receiver by a poly-phase filter.
- There is folded noise due to non-ideal image filter that added to noise figure.

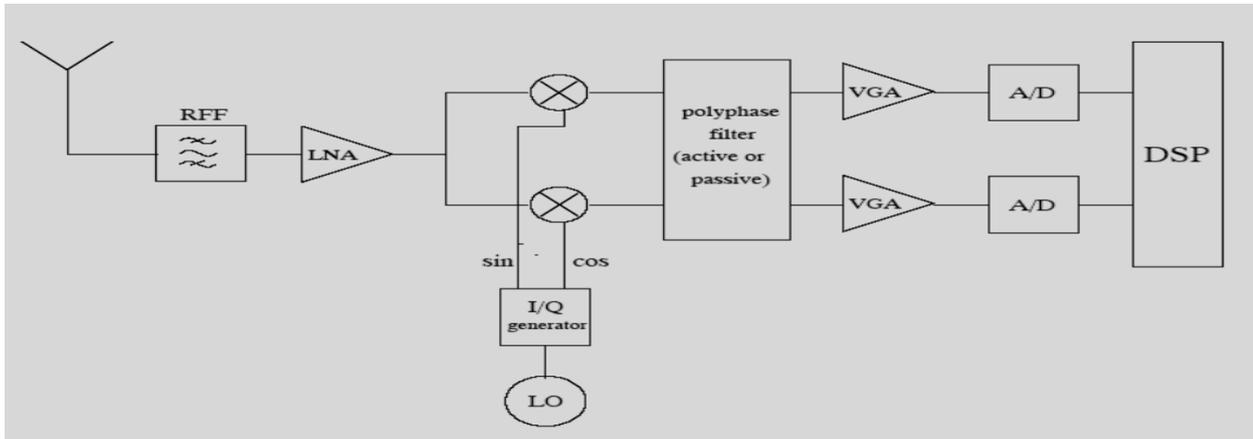


Figure 1.1.22 : low IF RX

1.2.5.1 Advantages

- No dc problem.
- Easy IC integration.
- On-chip filter.
- Low quality filter.
- Image is suppressed by a poly-phase filter.
- A high-pass filter may be used after the mixer to remove unwanted DC components, assuming the used modulation allows removal of part of the energy in the spectrum around DC.
- Low 1/f noise

1.2.5.2 Dis-advantages

- I/Q mismatches.
- Require high performance ADC (high Fs than in zero-if).
- Power dissipation and area due to poly phase filter.
- IIP2 is strong(second distortion).

1.2.6 Band pass sampling RX

- Received signal is filtered by an RF BPF then LNA.
- Signal converted to digital domain.

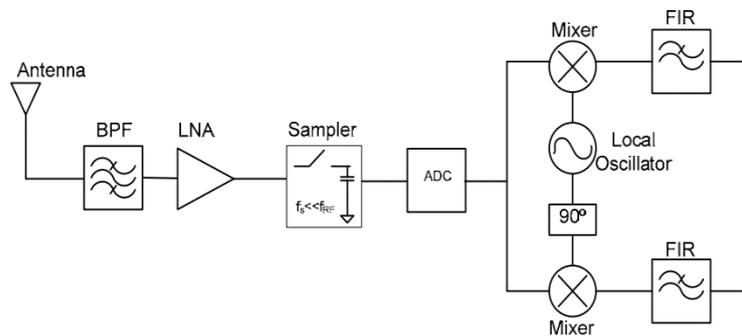


Figure 1.1.23 : band pass RX

1.2.6.1 Advantages

- Flexibility.
- Lower cost , area.
- Minimize dc problem and RF problem.
- Digital mixer and filters.
- Easier I/Q matching.

1.2.6.2 Dis-advantages

- Clock jitter.
- noise figure.
- Power consumption.
- Very high performance ADC (used in RF 2.4 GHz so $F_s=5$ GHz for example which is very complicated to ADC).

1.2.7 Comparisons

Architecture	Advantages	disadvantages
Super-heterodyne	<ul style="list-style-type: none"> • Selectivity • Sensitivity • No-dc problem 	<ul style="list-style-type: none"> • High Q filters • Image problem • Off-chip filters
Direct-conversion	<ul style="list-style-type: none"> • Integration • LPF instead of BPF • No image problem 	<ul style="list-style-type: none"> • Dc problem • I/Q mismatch • LO problems • Flicker noise and distortion
Low-IF receiver	<ul style="list-style-type: none"> • No dc problem • Integration • Low Q filter • On-chip filter • Low 1/f noise 	<ul style="list-style-type: none"> • I/Q mismatch • High performance ADC • Image rejection using poly phase filter
Hartley	<ul style="list-style-type: none"> • Good IRR • Image rejection 	<ul style="list-style-type: none"> • I/Q mismatch • RC-CR network • Shift by 90 and adder
Weaver	<ul style="list-style-type: none"> • Similar to Hartley • Avoid RC-CR network 	<ul style="list-style-type: none"> • Increase no of mixers • Mismatch • Increased no of component
Band pass sampling	<ul style="list-style-type: none"> • Flexibility • Minimize dc problem • Easier matching (digital) 	<ul style="list-style-type: none"> • Clock jitter • Noise figure • Power dissipation

Arch. Items	super heterodyne	Direct conversion	Low IF receiver
Sensitivity	++	--	+
Image rejection	++	No need for image rejection (at baseband)	+
Dc offset Immunity	No dc offset	--	+
Flicker noise immunity (1/f)	No flicker noise (high if than others)	--	+
I/Q mismatch	-	--	--
Integration Capability	--	++	++
Simplicity	--	++	++

1.3 System Design

1.3.1 Receiver System specifications

1.3.1.1 Receiver Overview:

We will design our receiver for one Modulation type which is GFSK of 915 Band which has certain specifications. We will analyze this these specifications in the following parts.

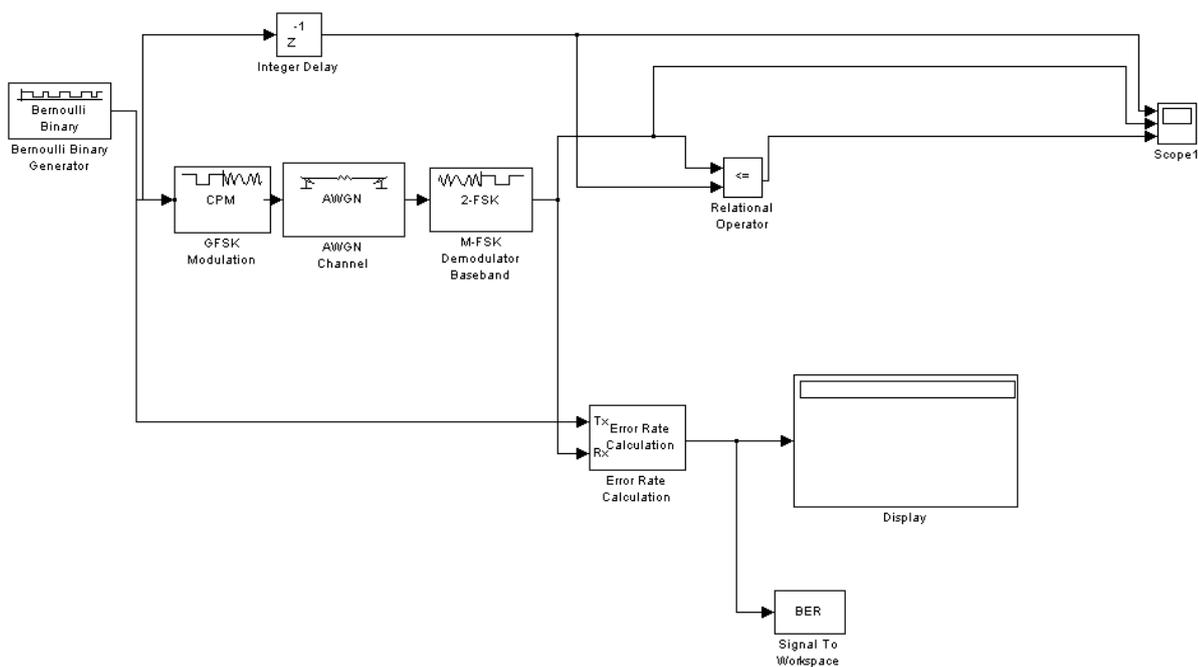
1.3.1.2 GFSK Modeling

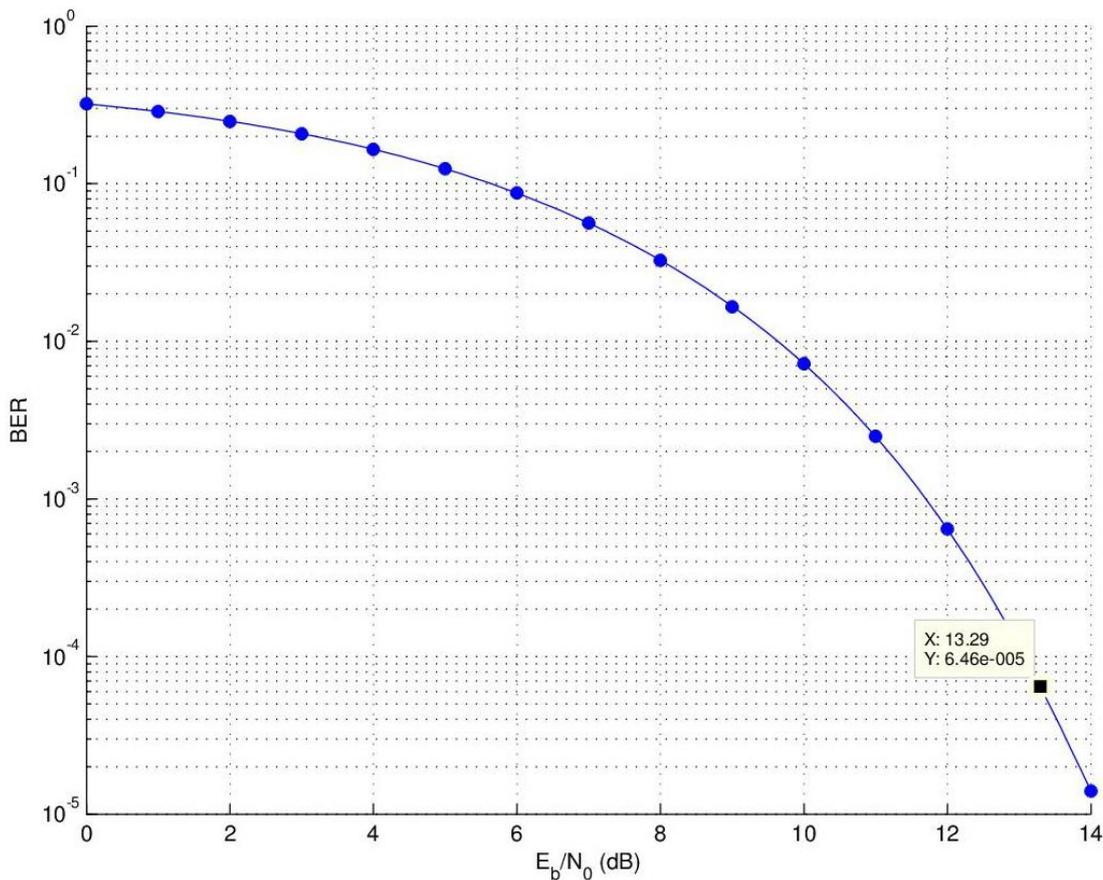
To be able to get required SNR of receiver, we need to get (BER vs E_b/N_o) curve to get required the SNR value from specific BER.

From datasheet, we have Δf (frequency deviation) =127 KHz , Bit rate=250Kbps , BT=0.5 , PER (packet error Rate)=1% , Packet length =20bytes.

Then we can get h (modulation index) = $\frac{2 * \Delta f}{R_B} = 1.016$.

We make GFSK model in Simulink to draw BER vs E_b/N_o curve for BT=0.5 and h=1.016.





It is assumed in datasheet that bits are independent, So we can assume probability that packet is True equal to probability that all packets are true , so we apply the following equation:

$$(1 - PER) = (1 - BER)^N , \text{ where } N \text{ is bits number in packet which is equal } = 8 * 20 = 160, \text{ so}$$

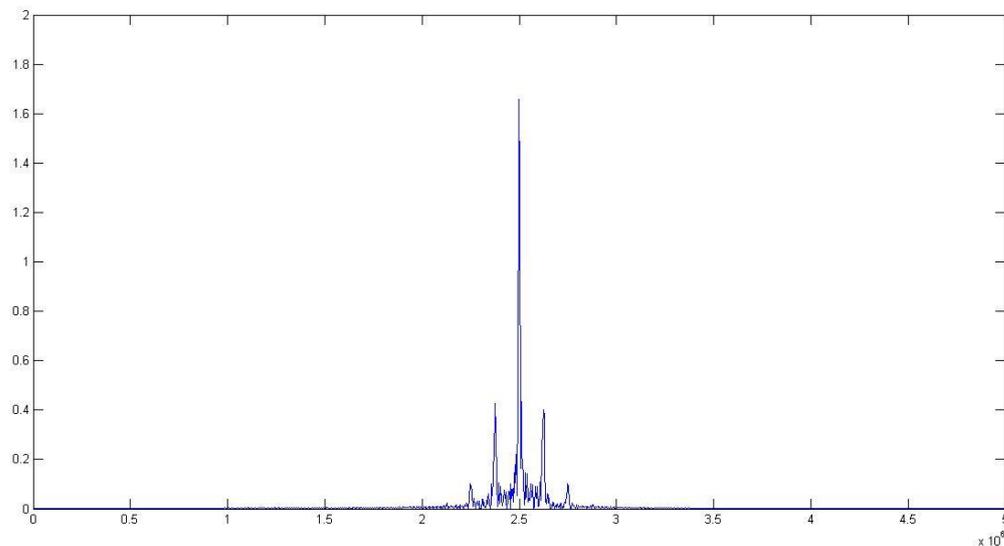
$$(1 - 0.01) = (1 - BER)^{160} \rightarrow BER = 6.28 * 10^{-5}$$

From BER vs E_b/N_o curve @ $BER = 6.28 * 10^{-5}$,we get $E_b/N_o = 13.3$.

Since $SNR = \frac{E_B * R_B}{N_o * BW}$ where BW is the noise effective BW which is approximately the same of channel filter. Then $SNR = 12.51 \text{ dB}$. Take Margin = 3dB (To accommodate for conversion between amplitude noise and frequency noise as we concern here with Frequency noise at demodulator. Also to accommodate for SNR degradation which is small around 0.2 dB []). So $SNR = 15.5 \text{ dB}$

1.3.1.3 BW Calculation:

A model is established to measure BW of GFSK. It was found for $BT=0.5$ and $h=1.016 \rightarrow$ BW will be 260KHz for power of 95% and will be about 270 KHz for power of 98% and more for more power.



So we can take BW about 280Khz

Matlab could be found in appendix

Sensitivity:

As specified from datasheet, Receiver Sensitivity = -95dbm.

Receiver Noise figure:

$$NF = S_i - (-174 + 10 \log(BW)) - SNR = 8.73 \text{ dB.}$$

Receiver IIP3:

As specified by vendor, the receiver IIP3 = -24 dBm.

1-dB Compression point:

As we know that 1-dB compression is below IIP3 is about 10 dB , So

$$P_{1\text{dB}} = IIP3 - 10 = -34 \text{ dB.}$$

1.3.2 Receiver Type and Frequency planning

Due to advantages specified before of Low-if receiver, Low-if receiver will be our proposed architecture due to low effect of DC offset and flicker noise and low effect of second order intermeditation.

Second we want to design receiver @ certain IF frequency,

For low ω_{IF}

- We have Flicker noise and DC offset problem.

While for high ω_{IF} :

- We have power consumption and selectivity problem (higher required Quality factor)

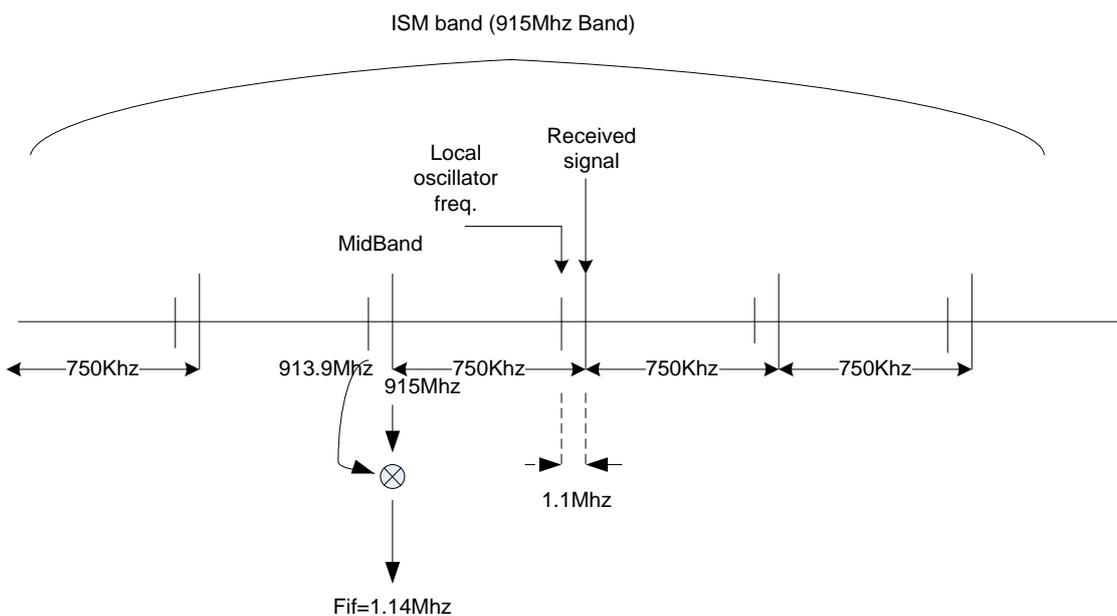
So a good compromise between two values is $f_{IF} = 1.1\text{MHz}$.

As we see frequency planning

We work on ISM band (915MHz) which range from 902MHz to 928 MHz Also channel spacing is 750Khz so we need signal to be transmitted @ frequencies $915\text{MHz} + (-)0.75*n, n=0,1,2,3, \dots$

Also we need local oscillator frequency to be $913.9\text{MHz} + (-) 0.75*n, n=1, 2, 3, 4$.

This is clarified in the following figure.



Blocker Profile:

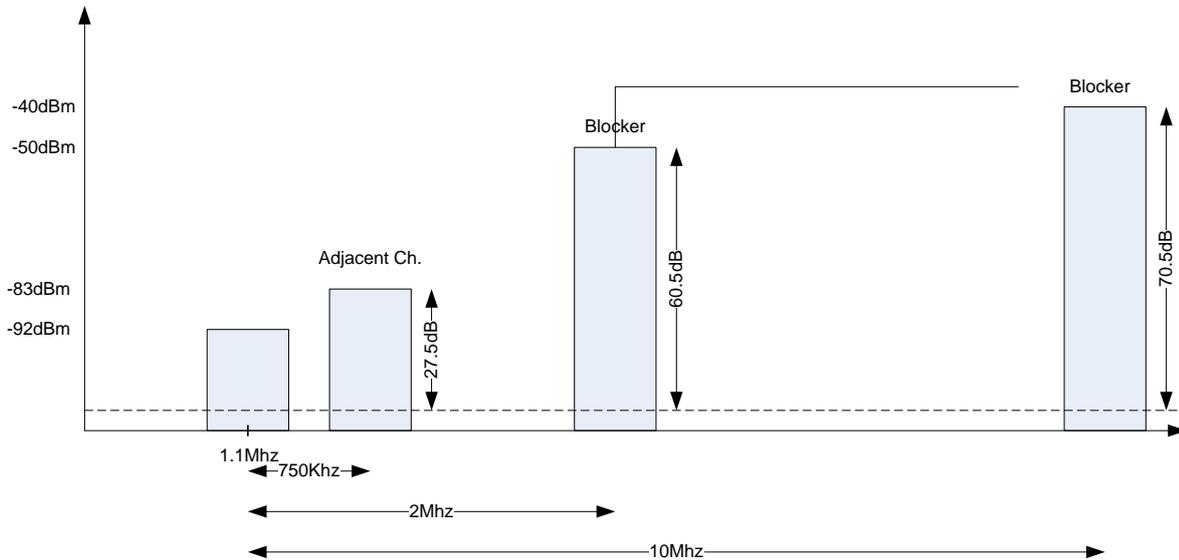
Blockers are specified when desired channel 3dB above sensitivity limit.

Blockers value and rejection required are shown in figure

As summary:

Blocker @ 2MHz → Rejection required >60.5dB.

Blocker @ 10MHz → Rejection required >70.5dB.



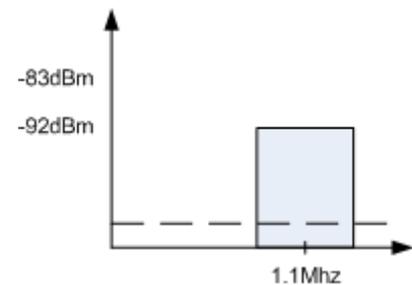
Filter

Complex Filter will be used in order to remove image

Butter worth filter is used:

- Passband is maximally flat (no ripples).
- Rolls off towards zero in stopband.

$$f_{if}=1.1\text{MHz} , \text{ BW}=280\text{KHz} , \text{ Then } f_{c2}=1.24\text{MHz}$$



ADC

@ ADC we design system that blockers and adjacent channel are rejected, so what will retain is the signal. We leave Margin about 5.5dB below V_{ref} of ADC and take 15 dB below the noise floor of signal in order to have insignificant effect of ADC quantization noise on SNR of signal So

$$DR_{ADC}=5.5 + SNR + 15 = 36 \text{ dB}$$

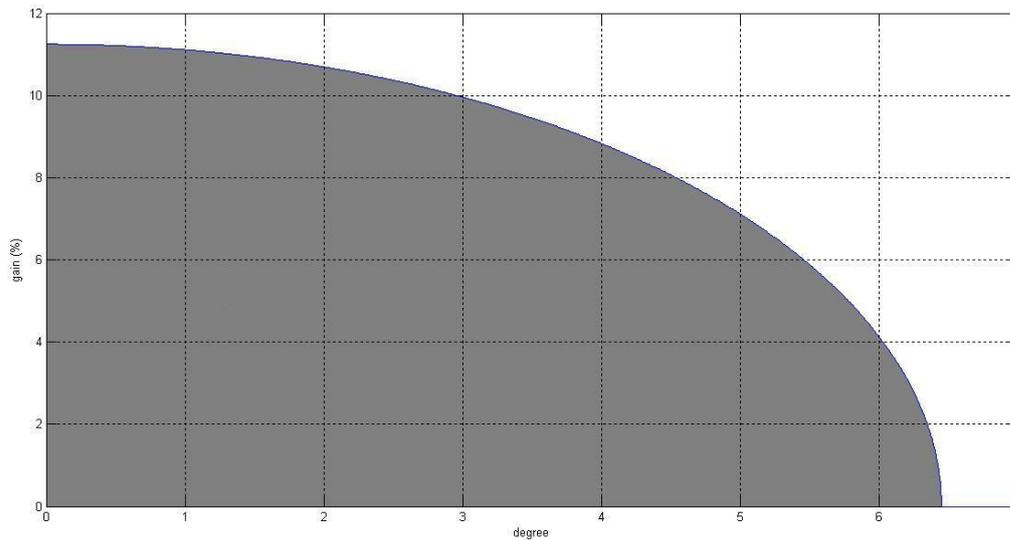
So Sampling Frequency of ADC is $>2*(1.14+0.28/2) = 2.56 \rightarrow$ Take it 2.6 MHz

1.3.3 Gain and phase mismatch of PLL local frequency:

Since we depend on complex filters to reject Images . And this done when real RF signal is multiplied by e^{jLO} , Practically performed using a quadrature mixer, which basically consists of two mixers whose LO inputs are in quadrature phase, In the complex signal representation , the desired signal at the mixer output is located at the positive IF frequency while the image signal is located at the negative IF frequency. In the real implementation, the desired (image) signal in the I branch leads (lags) the Q branch by 90° . Phase and gain imbalances at the mixer output, due to LO and mixer mismatches, will cause the image signal at $-f_{if}$ to spill over the image band at f_{if} . As a result the image rejection ratio (IRR) will be limited by these mismatches. It can be shown that the rejection limit (in dB) is given by [2]:

$$IRR(\text{dB})=10\log \left(\sin^2\left(\frac{\theta}{2}\right) + \left(\frac{A}{2}\right)^2 \cos^2\left(\frac{\theta}{2}\right) \right) \approx 10\log \left(\sin^2\left(\frac{\theta}{2}\right) + \left(\frac{A}{2}\right)^2 \right).$$

This equation can be approximated to simple circle, simulation results show the boundary values of phase and gain mismatch constraint on receiver (especially MIXER+LNA), where values inside this arc is accepted values.



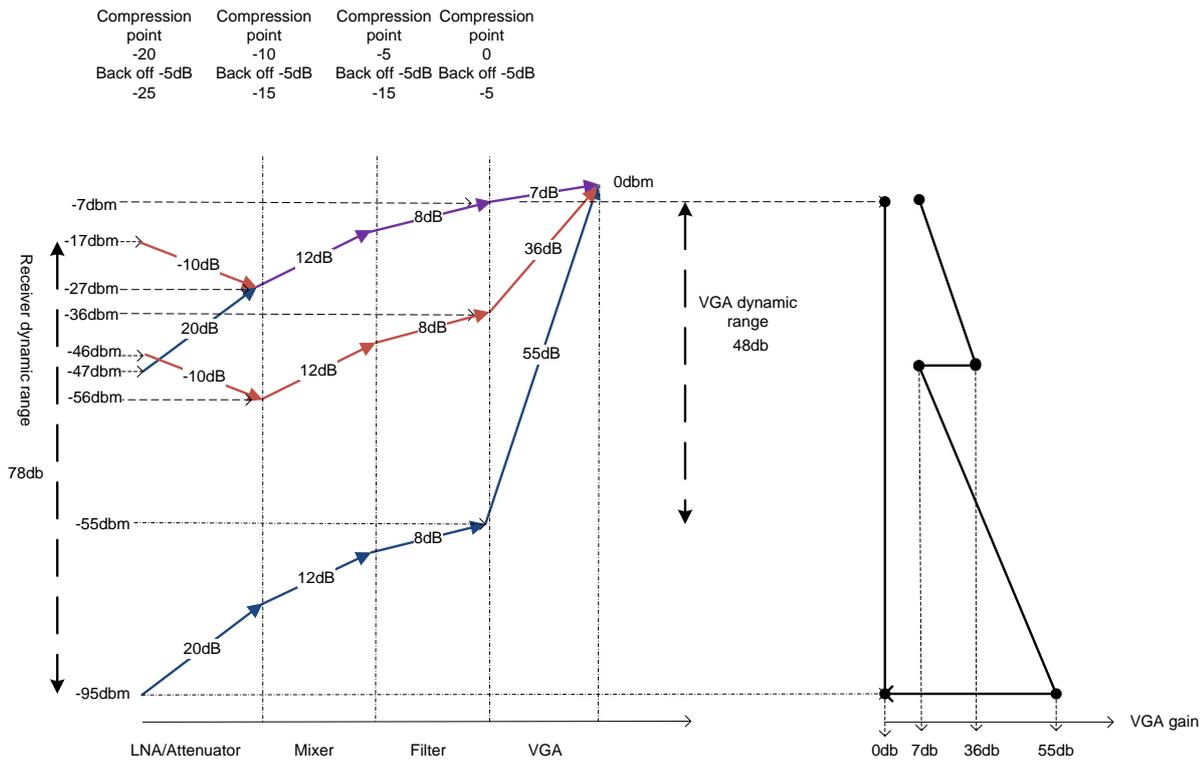
Settling Time:

In Ethernet frame there is initial parts is called permeable bytes .This part has advantage ,It is used for receiver to have sufficient time to be able to receive data bits (payload) correctly and receiver must have sufficient time less than it to settle .Recommended Preamble bytes are 4 bytes , Meaning that settling time of receiver must be less than $16\mu\text{s}$

Gain strategy:

ADC input

- Minimum signal = -95 dBm and saturation = -17 dBm.
- We target that signal reach ADC input @certain input value regardless the value of signal @ front end.
- This achieved using VGA after filter and Attenuator parallel to LNA
- It is designed that when input RF signal have value $>-47\text{dbm}$, The attenuator will bypass the LNA ,where in this condition no need for gain @ LNA as signal become large and have high SNR @ ADC input regardless using Attenuator
- Gain distribution depend that signal never reach any block at a value greater than its (compression point + (Back off Margin taken to be $\approx 5\text{ dB}$))
- We plan that signal reaches ADC 600mV(peak) differential (300m peak @one side) $\rightarrow 5.5\text{dbm}$
- Leave margin for VGA step and settle time so take margin 5.5dB $\rightarrow 0\text{dBm}$
- V_{ref} of ADC = 0.6V.
- Total max gain = $0 - (-95) = 95\text{dB}$



Parameter	LNA/Attenuator	Mixer	Filter	VGA	ADC	Total	Spec.	Unit
-----------	----------------	-------	--------	-----	-----	-------	-------	------

Specifications Distribution:

Distribution of specifications across blocks depend on the following issues

1. Noise figure of first block in receiver (LNA) have to be the smallest Noise figure as it contributes more to total Noise figure.
2. In case of linearity , As we know that blockers after Filter not affect the receiver linearity(I mean IIP3) especially we designed a filter to reject the blockers so the rejected blockers can't make significant contribution to total intermodulation .So will not take the effect of blocks after filter in IIP3 calculations. And since last block is the more effective one to linearity as blockers become large so we have to design filter to have highest linearity.
3. Gain distribution trades off between Noise figure and linearity as more gain in receiver front end Improves NF but degrades Linearity. And if more gain given to last blocks better linearity but worse NF.

After that survey was done for blocks to know the usual values and maximum values of Noise figure and linearity for different blocks. Then iterations done to reach to the desired values of receiver .The final distribution is shown in following table.

Gain	20/-10	12	8	50	0	95	95	dB
NF	3.9/25	12	32	25	30	5.6	8.7	dB
IIP3	-10/-2	2.5	10	10	20	-23.52	-24	dBm
Compression or more	-20/-12	-10	0	0	10			dbm

What achieved:

	LNA		Mixer		Filter		VGA		ADC		Units
	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	
Gain	17	23	9.4	12.3	8	10.2	-	-	0	0	dB
NF	2	3.9	8.8	11.3	-	32.5	-	25	-	20	dB
IIP3	-8	-6.8	0.7	52	-	17	-	7	-	20	dB

1.4 References:

1. RF Microelectronics, Second Edition, By Behzad Razavi.
2. BLUETOOTH/WLAN RECEIVER DESIGN METHODOLOGY AND IC IMPLEMENTATIONS , By AHMED ELADAWY EMIRA
3. RF SYSTEM DESIGN OF TRANSCEIVERS FOR WIRELESS COMMUNICATIONS, By Qizheng Gu
4. Automatic Calibration of Modulated Fractional-N frequency Synthesizers ,by Daniel R.McMahill

2 The LNA

2.1 Introduction

LNA stands for Low noise amplifier. LNA is the first on-chip component in the chain of the receiver. LNA is considered to be the most important block in the receiver as it can't be changed or replaced. The function of LNA is to give a suitable gain to the input signal with adding a small noise to it. As LNA is the first block in the chain as shown in the figure (2.1) so it is the most effective block in the calculation of the noise figure of the chain. The gain of LNA mustn't be very large or small but it must be suitable to give enough gain to the following stages to decrease their effect in noise figure contribution but without affecting the linearity of the input signal and the IIP3 of the system.

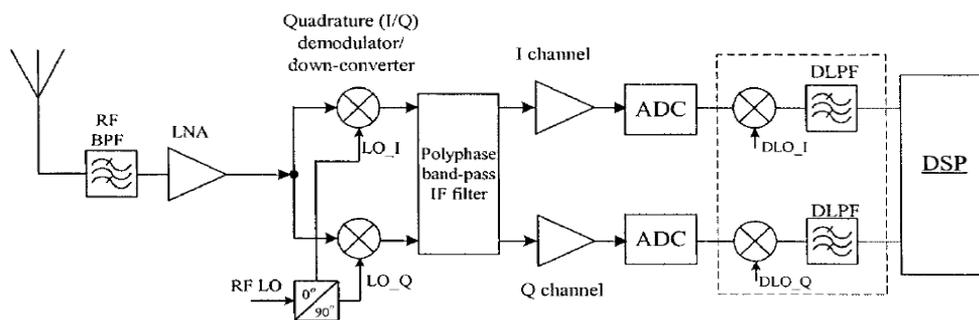


Figure 2.1: shows the block diagram of the receiver and the symbol of the of LNA

In this chapter I'll talk about the design of LNA, the factors affecting the design and tradeoffs between them, topologies and why I used the inductive degenerated topology. The specs that must be achieved across all corners are listed in table (2.1) as shown below.

Gain	NF	S11	IIP3	power
20dB	3.9 dB	<-10 dB	-9 dBm	2mA

Table 2.1 : shows the specs that must be achieved across all corners

From the shown specs we observe that the gain required is high, NF (noise Figure) required is not very low but I must achieve low one to pass all corners. We observe also that the given power to LNA is low (as I use differential LNA so this power is considered low power). Also, the achieved specs in the typical case are listed in table (2.2). The achieved specs across all corners will be at the end of this chapter.

Gain	NF	S11	IIP3	power
21dB	2.55dB	<-15 dB	-7 dBm	2mA

Table 2.2 : shows the achieved specs in the typical case

This chapter will be divided in to eleven sections. Section one is an introduction ,In section two I will talk about noise and its different sources as it is a very important parameter in LNA block, in section three I'll take about matching between LNA and the antenna (or the off chip component) as it is very important to make matching as in this case we talk about power not volt so matching is very important ,in section four I'll take about linearity techniques, in section five I'll talk about different topologies of LNA, in section six I'll show the design methodology of differential inductive degenerated topology which is the chosen topology, in section seven I'll show comparisons between differential and single ended input output LNA(using inductive degenerated topology) , in section eight I'll show the simulation results , in section nine I'll show the layout of the designed inductive degenerated LNA, in section ten I'll make some conclusion and comments, in section eleven I'll give the name of the references and thesis used.

2.2 Noise

The sensitivity of the communication systems is limited by the noise. The noise can be defined as any random interference unrelated to the desired signal and the noise can be divided in to three major classes which are the noise due to the interference, noise due to CMOS components and the quantization noise due added by the ADC. However, the major contributions in a CMOS technology is that added by CMOS components, the most effective types of the noise added by the CMOS components are thermal noise and flicker noise.

2.2.1 Thermal noise

Sometimes called Johnson-Nyquist Noise is the noise generated by the fluctuations of the electric current inside an electrical conductor, due to the random thermal motion of the charge carrier (the electrons). Because the noise process is random, one can't identify a specific value of voltage at a particular time(in fact, the amplitude has a Gaussian distribution), and the only way is to characterize the noise with statistical measures, such as the mean square or root mean square values as shown in the following different types of thermal noise.

2.2.1.1 Resistance thermal noise

Resistance thermal noise is generated by the thermal agitation of the charge carriers (electrons) inside a resistor, which happens regardless of any applied voltage. The only way to reduce this noise is to lower the circuit's temperature or minimize the resistance. This noise is independent of the resistor's material .The equation of the thermal noise is shown

$$\overline{Vn^2} = 4kTR\Delta f$$

Or

$$\overline{in^2} = \frac{4KT \times f}{R}$$

Where k is Boltzmann's constant in joules per Kelvin, T is the resistor's absolute temperature in Kelvin, and R is the resistor value in ohms (Ω)

The thermal noise of the resistance can be modeled in to two ways as a voltage source or a current source or both of them but we take must take care that they are correlated. Figure (2.2) shows the modeling of the resistor's noise

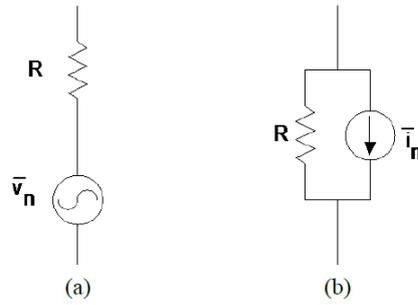


Figure 2.2: Resistor thermal noise models: (a) equivalent voltage source, (b) equivalent current source.

I talked about this noise as the antenna has a 50 ohm resistance that will contribute in noise figure calculations. Also for wide band topologies that use resistive matching at the input it will be very effective but in topology like inductive degenerated topology it is not very effective except under certain cases that will be clarified clearly in the following chapters.

2.2.1.2 Thermal noise in MOSFETS

Since FETs (both junction and MOS) are essentially voltage-controlled resistors, they exhibit thermal noise [THOMAS H.LEE 2004]. There are different types of noise which are drain current noise, gate noise and gate induced noise.

2.2.1.2.1 Drain current noise

In the triode region of operation the mosfet is considered as resistance. Indeed, detailed theoretical considerations lead to the following expression for the drain current noise of FETS:

$$\overline{ind}^2 = 4kT\gamma g_{do} \Delta f$$

Where g_{do} is the drain-source conductance at zero V_{Ds} . The parameter γ has a value of unity at zero V_{Ds} and, in long devices, decreases toward a value of 2/3 in saturation [THOMAS H.LEE 2004].

We must note that $g_{do} = g_m$ where g_m is the transconductance a quick derivation of this

$$I_{Ds} = \beta \times ((V_{GS} - V_{th}) \times V_{Ds} - \frac{V_{Ds}^2}{2}) \Rightarrow g_m = \frac{\partial I_{DS}}{\partial V_{GS}} = \beta \times (V_{GS} - V_{th})$$

$$\Rightarrow g_{do} = \left\{ \frac{\partial I_{DS}}{\partial V_{Ds}} \Big|_{V_{Ds} = 0} \right\} = \beta \times (V_{GS} - V_{th})$$

So, they are equal as shown from the above derivation.

2.2.1.2.2 Gate noise

This noise is due to the noise of the resistance of the gate that can be reduced by using techniques like fingers this will reduce the resistance of the gate very well.

We can say that

$$R_{new} = \frac{R_{old}}{n^2}$$

Where n is the number of fingers used.

2.2.1.2.3 Gate induced noise

The gate induced noise is an effective source of noise at high frequency. It shares the common origin of the drain current noise so they are correlated. The gate induced noise is due that the fluctuating channel potential couples capacitively into the gate terminal, leading to noisy gate current it can be modeled as shown in figure(2.3)

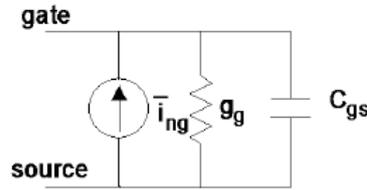


Figure 2.3: shows the model of the induced gate noise (after van der Zeil)

Van der Zeil has shown that the gate noise can be expressed as

$$i_{ng}^2 = 4KT\delta g_g \Delta f \quad , \text{ Where the parameter } g_g = \frac{\omega^2 C_{gs}^2}{5g_{do}}$$

Where δ is the gate noise coefficient with a value of about twice γ for long channel MOSFET, for short channel MOSFETS is taken to be 2-4, ω is the angular frequency and C_{gs} is the gate source capacitance.

2.2.2 Flicker noise

It is an effective type of noise at high frequency. It is inverse proportional with the frequency of operation. The effect of it appears in resistors only when direct current passing through it. Also, Flicker noise appears in MOSFET due to traps at the surface and the equation of this noise is

$$\overline{i_n^2} = \frac{K}{f} \times \frac{g_m^2}{WLC_{ox}^2} \times \Delta f \approx \frac{K}{f} \times \omega_T^2 \times A \times \Delta f$$

Where A is the area of the gate (=WL) and K is a device-specific constant.

There are other sources of noise but these sources are the most effective sources of noise in CMOS technology. In LNA the most effective source of noises are the drain current noise, induced gate noise and the gate noise. The gate noise can be reduced too much by using fingers as stated above, as we will see the effect of the gate induced noise and the drain gate noise are the most effective sources of noise in the my design of the inductive degenerated LNA in the following sections. The gate induced noise I think it appears due to the large capacitance i used between the gate and the source as will see.

2.2.3 Classical Two-Port Noise Theory

We can model any noisy circuit (not ideal circuit) with two noise sources and noiseless circuit. The two sources of noise are voltage and current sources, they are correlated to each other. Figure (2.4) shows the model

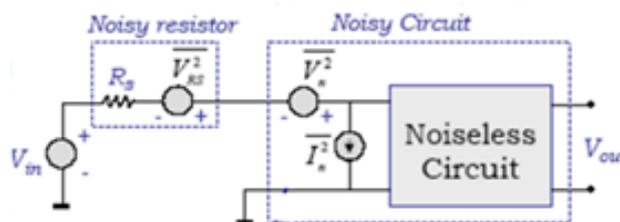


Figure 2.4: Model of noisy circuit

After derivations we can get some complex results that show how to get the minimum noise to LNA from the off chip band pass filter (BPF). The condition to get the least noise from BPF is not the same condition to get the maximum signal that made by matching, but we must know that what we want is to the largest possible signal to noise ratio not least noise and most signal reflected back to the BPF. So, I'll make impedance matching between the BPF and LNA. I'll take about how to make impedance matching in the next section.

2.3 Matching

Impedance matching to minimize reflections is achieved by making the load impedance equal to the source impedance. Ideally, the source and load impedances should be purely resistive. In figure 1 the signal is received by the antenna then to an off chip band pass filter to select the desired band only and to avoid the out band interference signals then to the LNA. The band pass filter is an off chip component to have high quality factor to get the signal of the desired band and filter out the out band interference signals. As the BPF is an off chip block there must be a good matching between the BPF and LNA to receive the maximum power from the signal. I'll talk in details in this section about different types of matching techniques.

2.3.1 Impedance Transformation

An Impedance transformation is the process of converting certain load impedance to a different one seen by the source.

Required to: Increase the maximum power transfer and Increase the SNR and reduce the NF in receivers

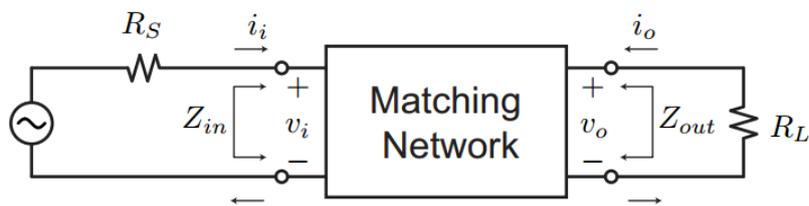
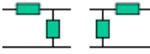
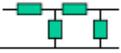


Figure 2.5: Shows a block diagram Transformation Network to make matching

Types of impedance matching

- Single lumped element (either L or C) 
- Dual lumped elements (L impedance matching network) 
- Triple lumped elements (Pi or T impedance matching network) 
- More lumped elements (ladder type) 

2.3.2 Impedance Transformation Using Lumped Elements

“RF design is all about impedance matching.” Inductors and capacitors are handy elements at impedance matching. An impedance matcher changes a given load resistance R_L to a source resistance R_S . Without loss of generality, assume $R_S > R_L$ and a power match factor of $m = R_S/R_L$ is desired. In fact any matching network that boosts the resistance by some factor can be flipped over to do the opposite matching.

2.3.2.1 Capacitive and Inductive Dividers

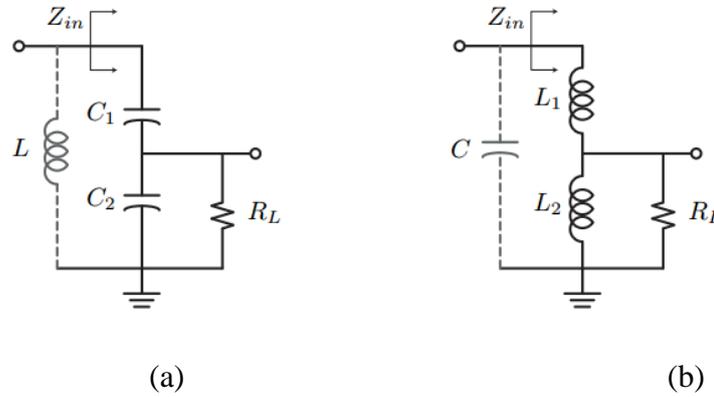


Figure 2.6: capacitive and inductive dividers

Perhaps the simplest matching networks are simple voltage dividers. Consider the capacitive voltage divider. At RF frequencies, if $R_L \gg X_2$, then we can see that the circuit will work as advertised. Assuming that negligible current flows into R_L , the current flowing into the capacitors is given by

For (a)

$$i = \frac{V_i}{j(X_1 + X_2)}$$

$$V_o = V_{C_2} = jX_2 \times i = V_i \times \frac{X_2}{X_1 + X_2} = V_i \times \frac{1}{1 + \frac{C_2}{C_1}}$$

$$V_o \times i_o = V_i \times i_i$$

$$\text{So, } R_{in} = \left(1 + \frac{C_2}{C_1}\right)^2 \times R_L$$

Where $C_{equivalent} = \frac{C_1 + C_2}{C_1 \times C_2}$, so we put parallel L to resonant at the desired band.

For (b)

$$\text{Similarly, } V_o = V_{L_2} = jX_2 \times i = V_i \times \frac{X_2}{X_1 + X_2} = V_i \times \frac{1}{1 + \frac{L_1}{L_2}}$$

$$V_o \times i_o = V_i \times i_i$$

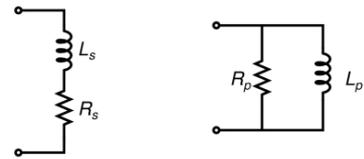
$$R_{in} = \left(1 + \frac{L_1}{L_2}\right)^2 \times R_l$$

Where $L_{equivalent} = L_1 + L_2$, so we put parallel C to resonant at the desired band.

2.3.2.2 The impedance transformation from parallel to series or vice versa

$$R_s = \frac{R_p}{(1+Q_p^2)}$$

$$L_s = \frac{L_p}{\left(1 + \frac{1}{Q_p^2}\right)}$$



$$R_s = \frac{R_p}{(1+Q_p^2)}$$

$$C_s = C_p \left(1 + \frac{1}{Q_p^2}\right)$$

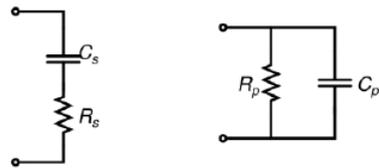


Figure 2.7: series to parallel and parallel to series conversion

This transformation is valid at a specific frequency, $Q_s = Q_p$ such that both circuit are equivalent, For $Q > 3$ $L_p = L_s$ and The quality factor is not the circuit quality factor.

2.3.2.3 L-Match

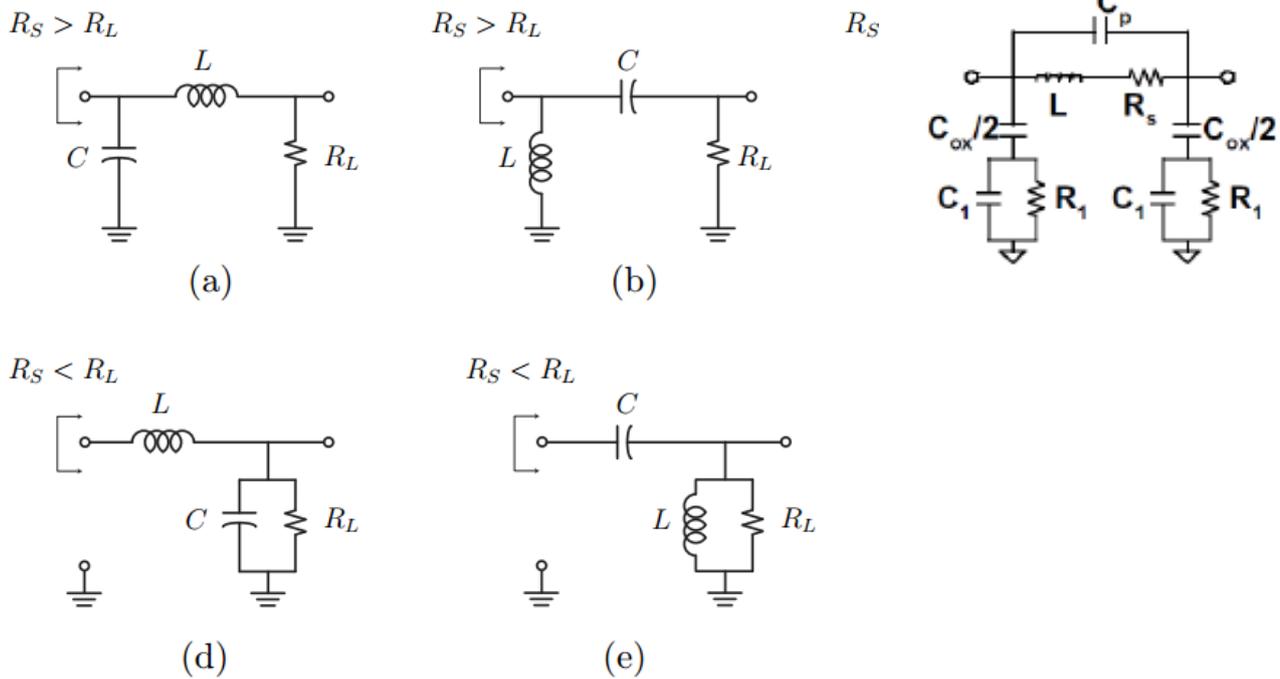


Figure 2.8: L-Match networks

Consider the L-Matching networks, named due to the topology of the network. We shall see that one direction of the L-match boosts the load impedance (in series with load) whereas the other lowers the load impedance (in shunt with the load).

2.3.3 The Choice of Topology

The choice of topology depends on the application. For instance some provide AC coupling (DC isolation) which may be required in many applications. In other applications a common DC voltage may be needed, making the networks with DC coupling the obvious choice. Also, depending on the values of inductor and capacitor used to make matching is it acceptable for my application or not.

2.3.4 Insertion Loss

There would be losses due non idealities in the passive components as we will see in later sections that it will affect my design of LNA, so non idealities must be considered while designing RF blocks. We must take care that inductor may be converted to capacitor so take care!

2.4 Linearity

In addition to noise figure, gain and input matching, linearity is an important consideration because an LNA must do more than simply amplify signals without adding much noise. It must also remain linear even when strong signals are being received. In particular, the LNA must maintain linear operation when receiving weak signal in the presence of a strong interfering one [THOMAS H.LEE 2004].

This section describes the various techniques used to achieve high linearity in an LNA and their advantages and drawbacks. The linearity techniques can be broadly classified into four different categories namely, optimum biasing, linear feedback, optimum out-off band terminations and feed forward.

2.4.1 Optimum biasing

The non-linearity of a MOS transistor arises from its voltage to current (V-I) conversion.

The drain current in a MOSFET can be modeled in terms of its gate-source voltage as shown

$$i_d = g_{m1} \times V_{gs} + g_{m2} \times V_{gs}^2 + g_{m3} \times V_{gs}^3$$

Where g_{m1} is its trans conductance, g_{m2} represents its second order non-linearity obtained by the second order derivative of FET transfer characteristics (I_d - V_{gs}) and g_{m3} is its third order non-linearity obtained by the third order derivative of FET transfer characteristics. The IIP3 is given in the above-mentioned-terms as follow

$$IIP3 = \sqrt{\frac{4}{3} \times \frac{|g_{m1}|}{|g_{m3}|}}$$

The I_d - V_{gs} transfer characteristics of a common source transistor along with g_{m1} , g_{m2} , g_{m3} are shown in Figure(2.10) for the case of a transistor in 0.35 μ m CMOS process. It can be seen that in the region of moderate inversion, in-between weak inversion and strong inversion, the third order derivative (g_{m3}) becomes zero over a narrow region. As shown in figure (2.10), it can be seen that IIP3 approaches infinity as g_{m3} becomes zero. Thus any transistor biased at this point can achieve high linearity. But the problem with this mechanism is that the region over which this linearity boost can be obtained is very narrow and due to process variations this bias point is bound to change leading to a very sensitive and limited improvement. Also, the transistor has to be biased in moderate inversion at the “sweet spot” hence placing a restriction on the trans conductance of the input stage. This restricts the maximum gain that can be obtained and thus affects the noise figure (NF) which is highly undesirable. Various bias circuit techniques have been proposed here the input transistor can be optimally biased such that $g_{m3} = 0$ It has been proven that the actual point of bias at which high levels of IIP3 can be achieved is slightly offset from the bias point at which $g_{m3} = 0$. But such a bias circuit is again prone to process variations resulting in poor linearity and would require fancy process to minimize the mismatch between transistors.

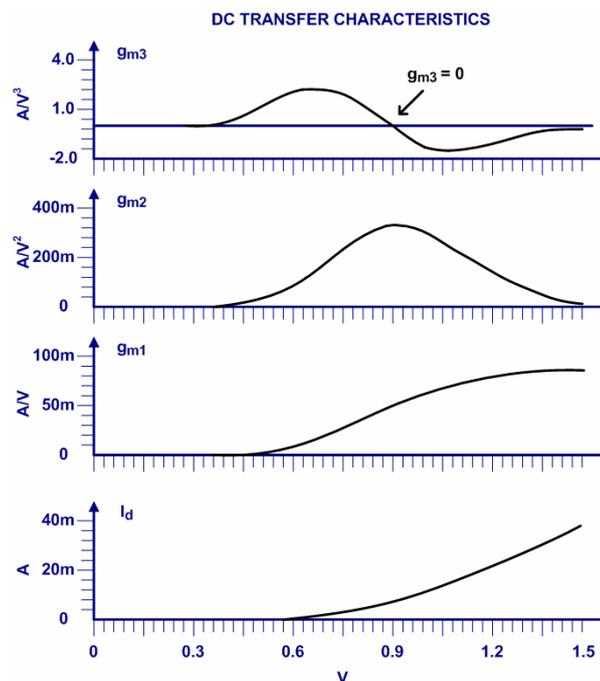


Figure 2.10: MOSFET transfer characteristics

2.4.2 Linear feedback

The most popular technique in base-band circuits to obtain high linearity is through the use of negative feedback. Figure (2.11) shows the configuration of a negative feedback nonlinear amplifier with gain 'A' and a linear feedback factor ' β '.

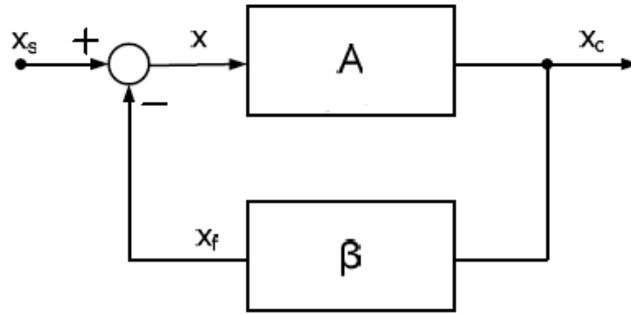


Figure 2.11: Non-linear amplifier with linear feedback

In the feedback method, a fraction of the output signal X_o is fed back to the input X_s through a linear feedback network β and is subtracted from the input to generate X_i an error signal which is fed to the amplifier A. The amplifier transfer function is given as follows.

$$X_o(t) = a_1 \times X_i(t) + a_2 \times X_i(t)^2 + a_3 \times X_i(t)^3$$

$$X_i(t) = X_s(t) - X_f(t) = X_s(t) - fX_o(t)$$

The closed loop transfer function is given by

$$X_o(t) = b_1 \times X_s(t) + b_2 \times X_s(t)^2 + b_3 \times X_s(t)^3$$

The coefficients b_1, b_2, b_3 are calculated, and the third order intermodulation distortion can be given as follows.

$$IIP3 = \sqrt{\frac{4}{3} \times \left| \frac{a_1}{a_3} \right| \times (1 + a_1 f)^3}$$

Without approximations the expression will be

$$A_{IIP3} = \sqrt{\frac{4}{3} \times \left| \frac{a_1}{a_3} \right| \times \frac{(1 + T_o)^3}{1 - \frac{2a_2^2}{a_1 \times a_3}} \times \frac{T_o}{T_o + 1}}$$

Where $T_o = a_1 f$, it can be seen that the IIP3 is also affected by the second order nonlinearity a_2 . In CMOS circuits which operate in strong inversion region, the coefficients a_1 and a_3 are of opposite signs leading to further reduction in IIP3. Note the concept of negative feedback is not that compatible with RF circuits since the gain of the amplifier is in the order of 10-25dB as opposed to base band circuits where a gain of 60dB can be easily achieved (the high gain is needed to make avoid the effect of a_2).

2.4.3 Feed forward cancellation

In this technique, scaled versions of the input signal are fed to two different amplifiers whose outputs are added to obtain the final output. The input signals are scaled such that the third order distortion is eliminated at the final output. This feed forward cancellation technique is used to achieve high IIP3. But the gain of the amplifier is reduced. Due to the reduced gain, the noise figure (NF) worsens. Further, more noise is added due to more active components in the circuit. This technique is highly sensitive to mismatch between the main and auxiliary gain stages and errors in the signal scaling factor. This configuration also consumes more power due to two amplifier stages being used. This technique is the most suitable one for RF circuits.

There are many approaches to get high linearity. But my specs of IIP3 are low so I'll not use linearity techniques.

2.5 Topologies of LNA

Generally to design LNA there are common things that must be taken in to consideration like good matching S_{11} *must* be less than -10 dB in the desired band to be sure that the desired signal not reflected back and 90% of the power received by the LNA, high but suitable gain to less the effect of the noise of the following stages but not affect the linearity of the system (generally the gain of LNA in the range between 15 dB to 25 dB depending on the system specs), low noise figure to said that the LNA is good it must be from 2 to 3dB for narrow band LNA and it may reach 5 dB to the wide band LNA.

There are different topologies of LNA that we can choose between them depending on the specs required from LNA to achieve the specs of the system. I'll talk about these topologies in details in this section.

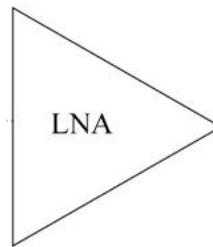


Figure 2.12: the symbol of LNA

First of all we must determine first the method to determine the Noise Figure:

1-Deactivate current or voltage sources in the circuit.

2- Noise contribution of each noise source $\implies \overline{V_{on}^2}$.

3- $\overline{V_{ont}^2} = \sum \overline{V_{on}^2}$.

4 -Get gain of the circuit.

5- $\overline{V_{int}^2} = \frac{\overline{V_{on}^2}}{gain^2}$

6- Noise Factor = $\frac{\overline{V_{int}^2}}{4KTR}$

7-Noise Figure= $10 \times \log(\text{Noise Factor})$

2.5.1 Common Source with resistive load

To make matching R_1 must be equal to R_L

$$\overline{V_{on(RL)}}^2 = 4KTR_L$$

$$\overline{V_{on(M1)}}^2 = 4KT\gamma g_m R_L^2$$

$$\overline{V_{out}}^2 = 4KTR_L + 4KT\gamma g_m R_L^2 + 4KT(R_S + R_1) \times gain^2$$

$$\text{Gain of the circuit} = \frac{1}{2} \times g_m \times R_L$$

$$\overline{V_{int}}^2 = 16 \times \frac{kTR_L}{(g_m R_L)^2} + 16 \times \frac{KT\gamma}{g_m} + 4KT(R_S + R_1)$$

After neglecting small terms

$$\text{Noise Factor} = 2 + \frac{4\gamma}{g_m R_S}$$

So, Noise Figure is more than 3 dB in typical case so it cannot achieve all the system specs in all corners.

Also, we must take care about the stability so we must add cascade MOSFET to separate the input from the output to make LNA stable and to decrease the Miller capacitance of M1 and thus the effect of C_{gd} is neglected.

The noise added by this cascade device can be neglected and it will be a small term in Noise Figure calculations.

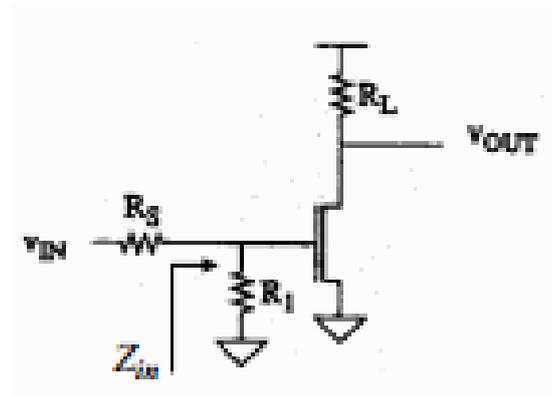


Figure 2.13: Common Source with resistive load

2.5.2 Common Source with resistive feedback

Using miller to get 50 ohm

$$R_{in} = \frac{R_f}{\text{Gain}} = \frac{R_f}{g_m \times R_f} = \frac{1}{g_m}$$

$$\text{To make matching } R_S = R_{in} = \frac{1}{g_m}$$

$$\text{So, gain} = \frac{1}{2} \times g_m \times R_f$$

$$\overline{V_{on(RF)}}^2 = 4KTR_F$$

$$\overline{V_{on(RS)}}^2 = 4KTR_S \times gain^2$$

$$\text{To get } \overline{V_{on(M1)}}^2$$

$$g_m \times V_{gs} + \overline{v_{nM1}} = \frac{-V_{out}}{(R_S + R_F)}$$

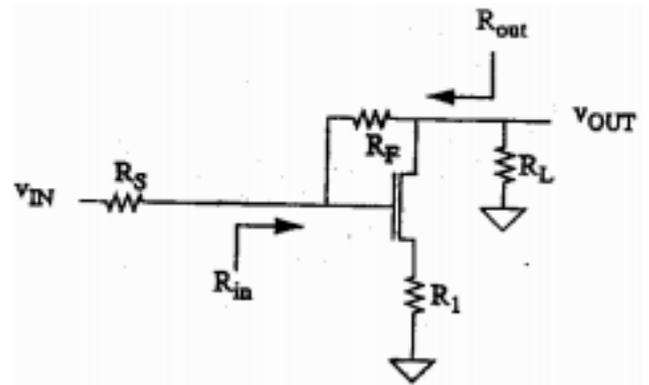


Figure 2.14: Common source with resistive feedback

$$V_{gs} = \frac{V_{out} \times R_S}{(R_S + R_F)}$$

Then $\overline{i_{nM1}} = -V_{out} \times \left(\frac{1}{(R_S + R_F)} + \frac{g_m \times R_S}{(R_S + R_F)} \right) = -V_{out} \times \frac{1 + g_m \times R_S}{(R_S + R_F)} = -V_{out} \times \frac{2}{R_F}$

$$\overline{V_{onM1}^2} = \overline{i_{nM1}^2} \times \left(\frac{R_F}{2} \right)^2 = 4KT\gamma g_m \left(\frac{R_F}{2} \right)^2$$

$$\overline{V_{int}^2} = 4kTR_S + \frac{16KT}{g_m^2 \times R_F} + \frac{4KT\gamma}{g_m}$$

$$\text{Noise Factor} = 1 + \frac{4}{g_m^2 \times R_F \times R_S} + \frac{\gamma}{g_m \times R_S}$$

After neglecting small terms

$$\text{Noise Factor} = 1 + \gamma \text{ where } \gamma \approx \frac{4}{3}$$

Noise figure is approximately 3.7 dB so this topology can't achieve specs across all corners.

2.5.3 Common gate LNA

To make matching make $R_{in} = R_S$

$$R_{in} = \frac{1}{g_m}$$

$$\text{Gain} = \frac{1}{2} \times g_m \times R_L$$

$$\overline{V_{onM1}^2} = \frac{4kT\gamma}{g_m} \times \left(\frac{g_m \times R_L}{1 + g_m \times R_S} \right)^2 = \frac{4kT\gamma}{g_m} \times \left(\frac{g_m \times R_L}{2} \right)^2$$

$$\overline{V_{on(R_S)}^2} = 4kTR_S \times \text{gain}^2$$

$$\overline{V_{on(R_L)}^2} = 4kTR_L$$

After neglecting R_L as its noise effect is small

$$\overline{V_{int}^2} = 4kTR_S + \frac{4kT\gamma}{g_m}$$

$$\text{Noise Factor} = 1 + \gamma$$

So, Noise figure is approximately 3.7 that can't achieve the specs at all corners.

So, we must think about a power full topology that achieves the required specs. We must think about matching using inductors and capacitors as they ideally not add noise to the system and also to select the required band. I will proceed as follow find Z_{in} that can be matched to the 50 ohm

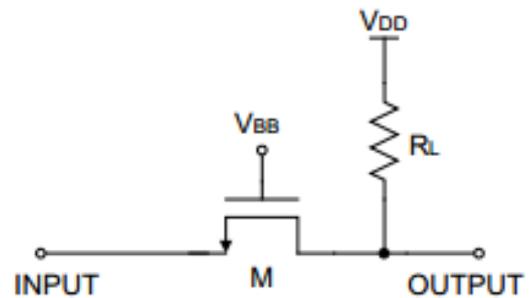


Figure 2.15: Common gate LNA

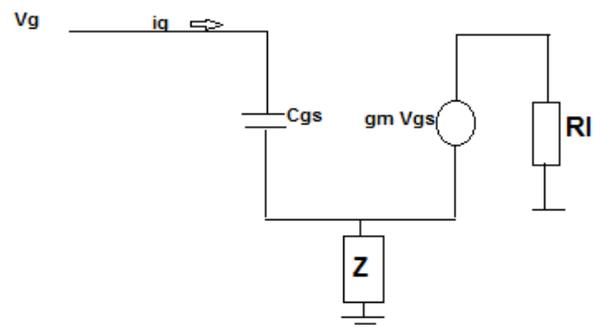


Figure 2.16: MOSFET with impedance degeneration

of the BPF and also select the desired band.

We start with figure (2.16) to reach this topology

$$\frac{V_g - V_{gs}}{Z_s} = g_m \times V_{gs} + i_g$$

$$V_g = i_g \times Z_s + V_{gs}(1 + g_m \times Z_s)$$

$$V_{gs} = \frac{i_g}{sC_{gs}}$$

$$V_g = i_g \times Z_s + \frac{i_g}{sC_{gs}} \times (1 + g_m \times Z_s)$$

$$Z_{in} = \frac{V_g}{i_g} = Z_s + \frac{1}{sC_{gs}} \times (1 + g_m \times Z_s) = Z_s + \frac{1}{sC_{gs}} \times (1 + g_m \times Z_s)$$

$$\text{Then } Z_{in} = Z_s + \frac{1}{sC_{gs}} + \frac{g_m \times Z_s}{sC_{gs}}$$

We will put Resistance, capacitor and inductor to Z_s respectively and see what happen

$$1- Z_s = R_s$$

$$Z_{in} = R_s + \frac{1}{s \times C_{gs}} + \frac{g_m \times R_s}{s \times C_{gs}}$$

$$C_{equivalent} = \frac{C_{gs}}{1 + g_m \times R_s} \approx \frac{1}{\omega_T \times R_s}, \text{ it is very small value so we don't want to make this.}$$

$$2- Z_s = \frac{1}{sC_s}$$

$$Z_{in} = \frac{1}{sC_s} + \frac{1}{sC_{gs}} + \frac{g_m}{s^2 C_{gs} C_s}$$

$$Z_{in}(j\omega) = \frac{1}{j\omega} \left[\frac{C_{gs} + C_s}{C_{gs} \times C_s} \right] - \frac{g_m}{\omega^2 \times C_{gs} \times C_s}$$

So, there is negative resistance that makes oscillation.

$$3- Z_s = sL_s$$

$$Z_{in} = sL_s + \frac{1}{sC_{gs}} + \frac{g_m L_s}{C_{gs}}$$

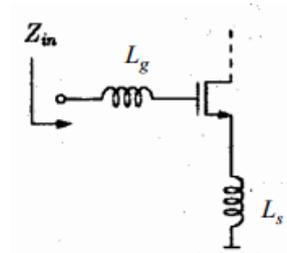


Figure 2.18: Input stage

As shown in figure (2.17)

So, we get the most suitable component that can be added for matching as the resistance is positive so no oscillations. So, we make matching without adding noise ideally.

This is introduction to the most important topology in LNA which is called common source with Inductive degenerated LNA (Super LNA), this will be discussed in details.

2.5.4 Common source LNA with inductive degeneration

The inductive degenerated LNA is as seen by figure (2.18), I'll study the input stage in details and make some observations that show up the great advantages of this topology.

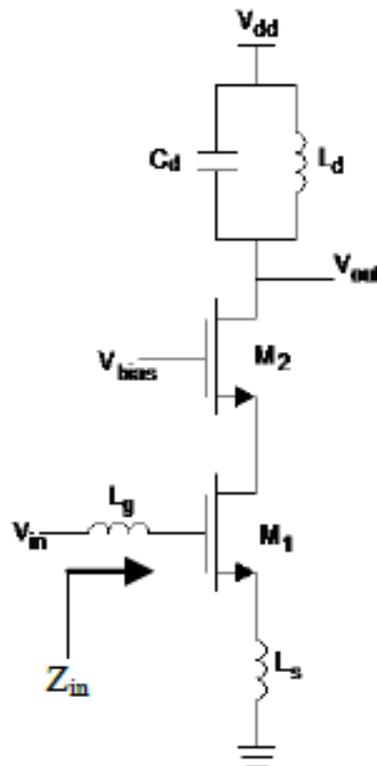
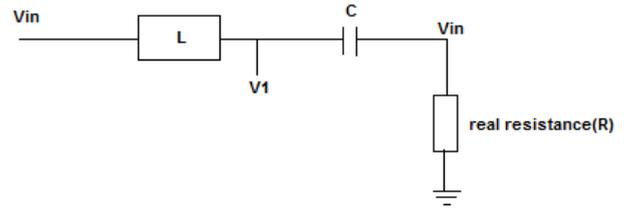


Figure 2.18: Inductive degenerated LNA

Input stage of LNA is shown in figure (18.2), we want to get the voltage on C_{gs}



The input stage can be represented as shown in figure (2.19)

$$V_{in} = I \times (Z) = I \times (SL + \frac{1}{SC} + R)$$

At resonance

$$V_{in} = I \times R$$

Figure 2.19: Input stage of LNA

$$V_1 = V_{in} - I \times SL = V_{in} - \frac{V_{in}}{R} \times SL = V_{in} \left(1 - \frac{SL}{R}\right) = V_{in}(1 - Q)$$

$$\Delta V_c = V_{in} - V_{in}(1 - Q) = V_{in}Q$$

So, it is very important result as it shows that the input signal multiplied by a gain which is the quality factor of the input stage, so we have a stage that provides a gain without adding noise in the ideal case. This is result is very useful result that makes this topology is the best topology to be used as no noise added to make matching(ideally) also it adds gain.

To get noise:

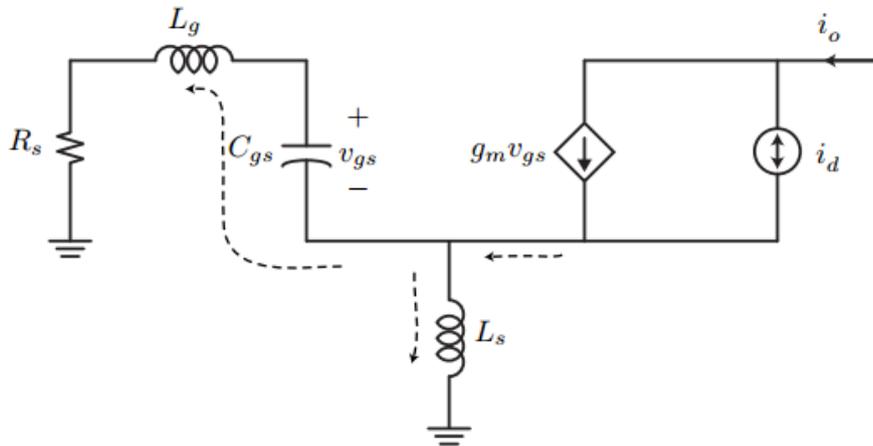


Figure 2.20: small AC signal model of inductive degenerated LNA

$$Q = \frac{1}{\omega_o C_{gs} \times 2R_s}$$

$$G_m = Q \times g_m = \frac{1}{\omega_o C_{gs} \times 2R_s} \times g_m$$

$$V_{gs} = (id + g_m V_{gs}) \times \frac{SL_s}{SL_s + \frac{1}{SC_{gs}} + SL_g + R_s} \times \frac{1}{SC_{gs}}$$

At resonance

$$V_{gs} = (id + g_m V_{gs}) \times \frac{SL_s}{R_s} \times \frac{1}{SC_{gs}}$$

$$V_{gs} \left(1 + g_m \times \frac{L_s}{R_s \times C_{gs}} \right) = id \times \frac{L_s}{R_s \times C_{gs}}$$

As we know $R_s = \frac{g_m \times L_s}{C_{gs}}$

Then $V_{gs}(2) = \frac{id}{g_m}$ $g_m V_{gs} = \frac{id}{2}$ it means that $\frac{1}{4}$ of the noise only will go the output

So, $\overline{v_{ont}^2} = \frac{\overline{id^2}}{4} + G_m^2 \times \overline{V_s^2}$

Noise Factor = $1 + \frac{\overline{id^2}}{4 \times G_m^2 \times \overline{V_s^2}} = 1 + \frac{4KT\gamma g_m}{4 \times 4KTR_s} \times \left(\frac{\omega_o \times 2R_s}{\omega_T} \right)^2 = 1 + \gamma g_m R_s \times \left(\frac{\omega_o}{\omega_T} \right)^2$

So, the noise figure that obtains from this Topology is very small and can achieve the required specs.

2.5.5 Comparisons between Topologies

	Common Source	Resistive feedback	Common Gate	Inductive degenerated
Noise Figure	High	Moderate	Moderate	least
Input Matching	Easy	Easy	easy	Complex
Sensitivity to parasitic	Less	Less	less	large
Gain	Moderate	Moderate	Moderate	high
Draw backs	Large NF	Stability	Large NF	Area

Table 2.3: Comparison between topologies

2.6 Design Methodology

2.6.1 The way of thinking

First we must ask some questions about LNA performance and the desired specs to be achieved across all corners. Figure (2.21) shows what determines and controls the LNA performance and what limits the desired specs.

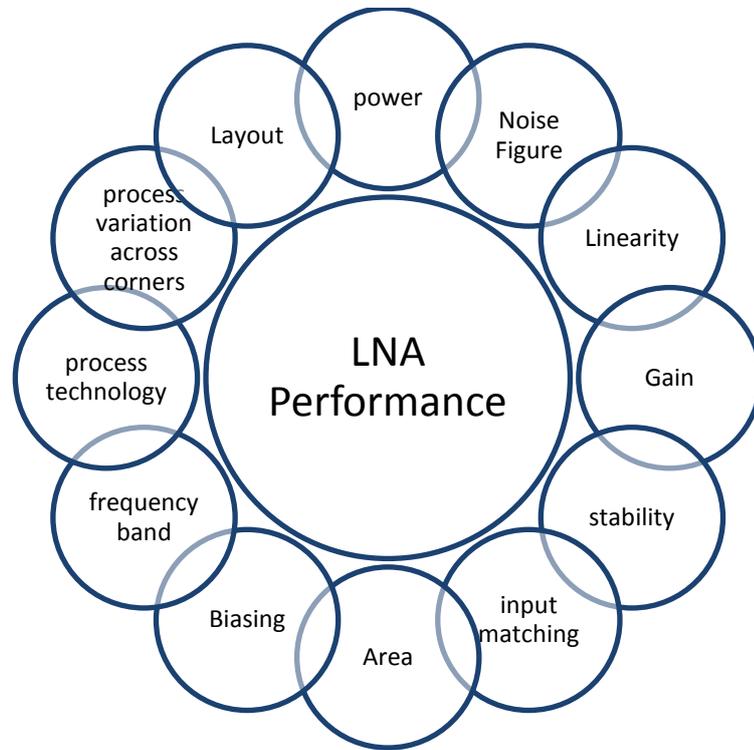


Figure 2.21: Shows what determines and controls LNA performance

2.6.2 Tradeoffs in LNA

In general, I list the tradeoffs that must be considered in LNA that may be happened or not depending on the circuit then I'll list the tradeoffs in my design.

In general these tradeoffs may happen while designing LNA:

- 1- Between noise matching and power matching.
- 2- Between power and Noise Figure.
- 3- Between linearity and Noise Figure.
- 4- Between gain and Noise Figure.

For my design the main trade off was between:

- 1- Between power and Noise Figure.
- 2- To choose the components to have the highest possible quality factors at the input stage to add the least noise to LNA.

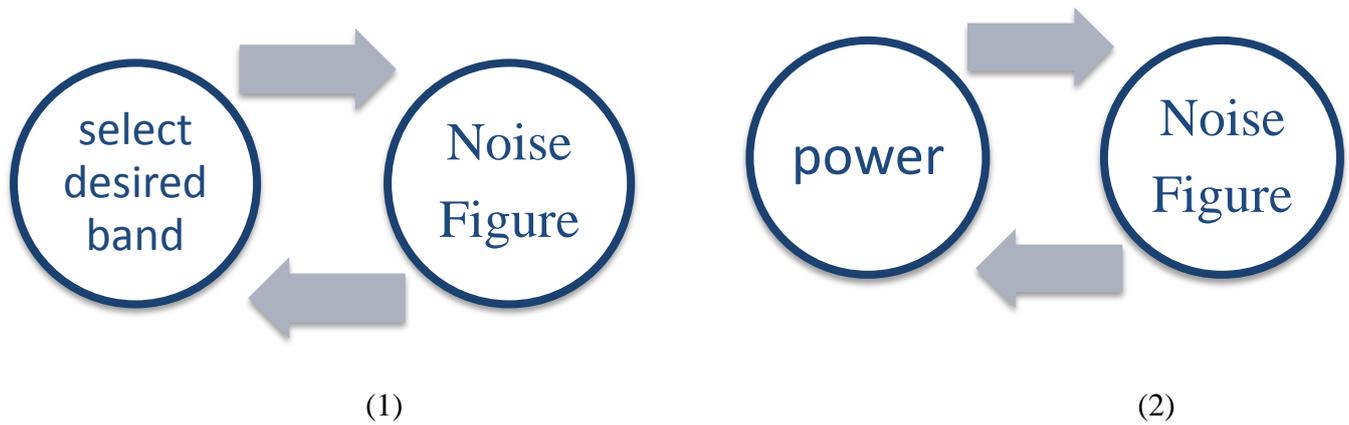


Figure 2.22: Tradeoffs

2.6.3 Hand analysis

First of all we write the design equations

$$1 - \omega_o = \frac{1}{\sqrt{(L_s + L_g) \times (C_{gs})}}$$

$$2 - R_s = \frac{g_m \times L_s}{C_{gs}}$$

$$3 - \text{Noise Factor} = 1 + \gamma g_m R_s \times \left(\frac{\omega_o}{\omega_T}\right)^2$$

$$4 - Q = \frac{1}{\omega_o C_{gs} \times 2R_s} = \frac{1}{2R_s} \times \sqrt{\frac{(L_s + L_g)}{C_{gs}}}$$

$$5 - \text{Gain} = \frac{1}{2} \times Q \times g_m \times R_L$$

$$6 - g_m = \sqrt{2 \times I_d \times \frac{W}{L} \times \mu_n \times C_{ox}}$$

$$7 - g_m = \frac{2 \times I_d}{V_{od}}$$

$$8 - C_{ox} = \frac{3.9 \times \epsilon_{ox}}{t_{ox}}$$

2.6.3.1 Design steps that I follow

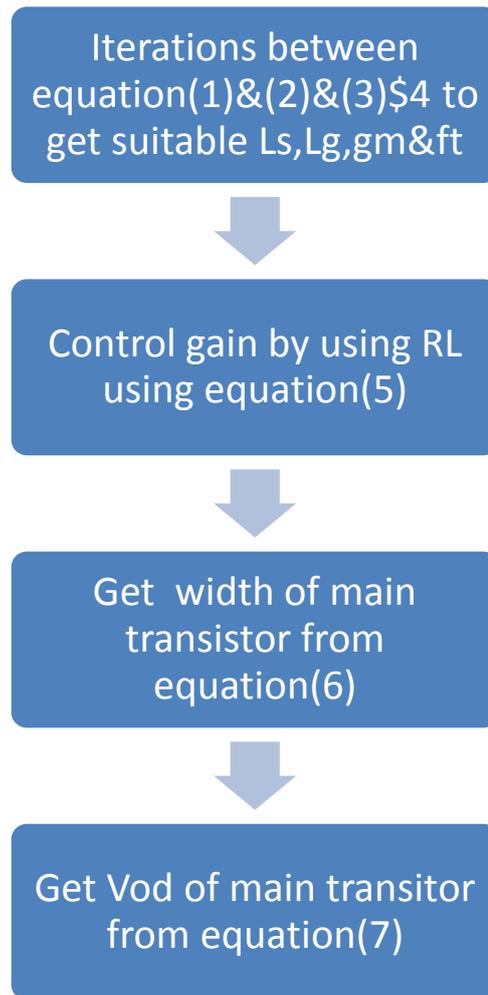


Figure 2.23: Design procedures

2.6.3.2 Explanation of design procedures that I flow

From equation 1, 2, 3&4, we must

- 1- achieve the required frequency
- 2- Take care about the power constraint which is 1mA for single input-output LNA (I'll use differential LNA for reasons I'll discuss them in the next section).
- 3- The f_t must be large enough to make the second term in the noise figure equation small & C_{gs} is of small value as it has great effect in NF.
- 4- V_{od} is moderate to have the IIP3 required Not high so need of large V_{od} .
- 5- Take care that Q at least more than one.

From equation 5

We get the gain which is controlled easily by R_L .

From equation 6

I get the width as I have g_m from the previous procedures and let $I=0.9$ mA.

From equation 7

I get V_{od} (take care that V_{od} moderate to have the IIP3 required not high so need of large V_{od})

After making these steps:

I get these values

L_g	21 nH
L_s	2 nH
C_{gs}	1.315 PF
gm	16.44 m
R_L	682.49 ohm
Gain	21 dB
Noise Figure	1dB

Table 2.4: the values of components from hand analysis

Note: this noise figure is for single ended but for Differential it will be approximately doubled.

2.6.3.3 Problems

L_g is of Large value as frequency is not very high as blue tooth for example that will affect the design, in practical L_g is not ideal but it has a resistance value so this large value of L_g will have not a small value of resistance that will increase Noise Figure by 1 dB approximately!!! . I'll discuss this issue in details.

2.7 Differential vs. single ended LNA

	Differential LNA	Single ended LNA
Advantages	1-Rejection of common mode noise and interferer.	1-Compact layout size. 2- less power for same NF and linearity
Disadvantages	1-Double area and current. 2-Need Balun at input. 3- Linearity limited by bias current.	1- drive single-balance mixer; or use output Balun to drive double balance mixer

Table 2.5: Comparison between Differential vs. single ended LNA

2.8 Simulation and Results

2.8.1 Notes, problems that I face

I'll use Differential LNA to avoid the common mode noise that may make LNA don't achieve the required specs if i used single input-output LNA. By mirroring t i can get the differential input-output LNA. I'll balloon at the input of LNA as my design become differential topology.

After hand analysis i test what I get from it, I found that in ideal case when using components from analog.lib the results were very good, but when using practical components from TSMC 0.13 i found that i must make some changes to the components value to achieve the specs due to non ideal components.

The big problem that i face that the frequency is not very high like Bluetooth as an example which make me use large components that will take large area in layout. Also, due to non ideal component and large value of these components it results in considerable resistance that affected the noise figure these resistances make the Noise Figure increases by 1dB approximately .The most effective component to make this increase in the NF is the resistance of L_g , Also the large value of C_{gs} I think it makes the induced gate noise of effective value as i observed from simulations.

2.8.2 Circuit

2.8.2.1 The whole circuit

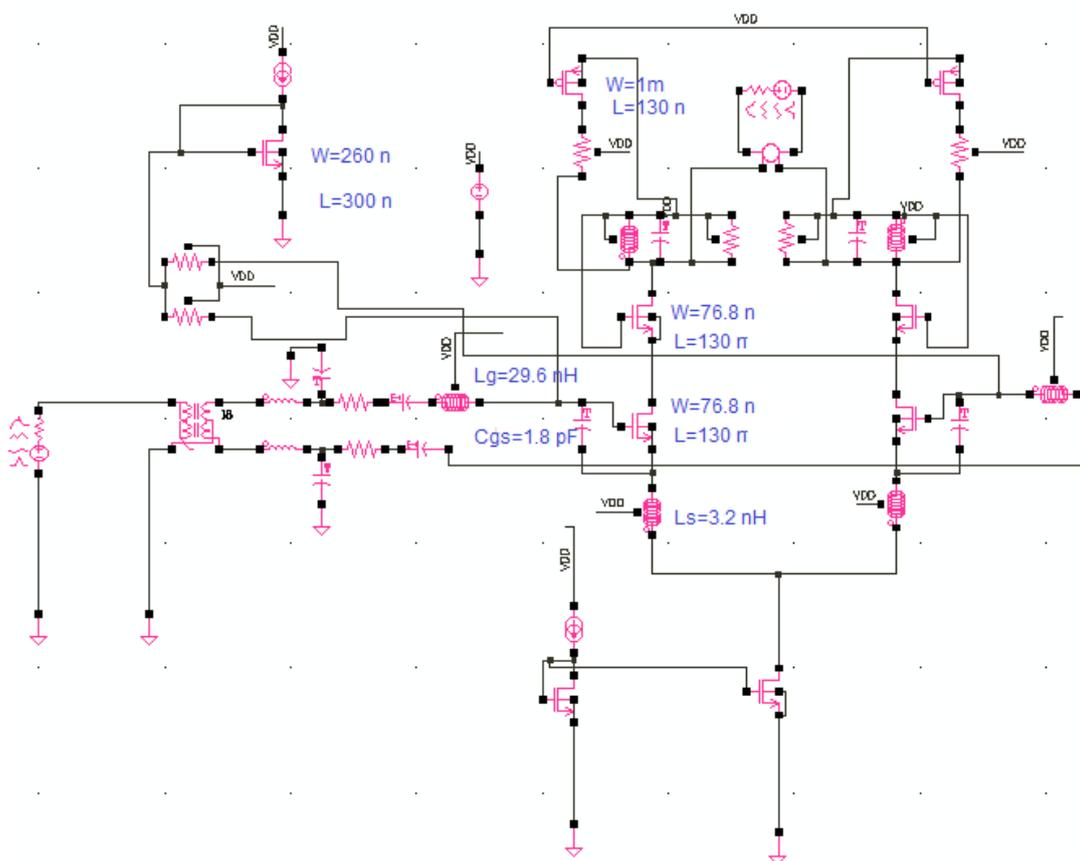


Figure 24.2: LNA Circuit with attenuator

2.8.2.2 Biasing circuit

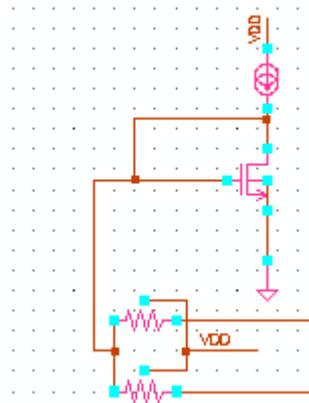


Figure 2.25: Biasing Circuit

2.8.2.3 Switch to choose attenuation mode

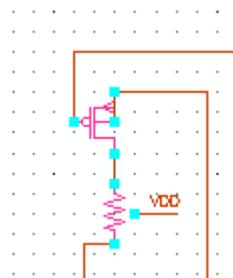


Figure 2.26: switch

2.8.2.3 Ballun and Bond wire

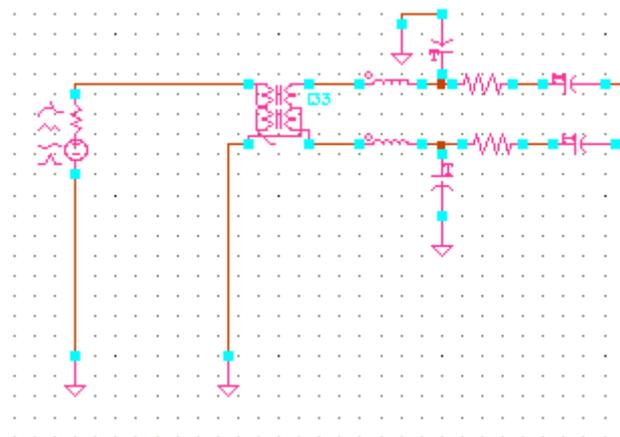


Figure 2.27: Ballun and Bond wire

2.8.2 LNA

2.8.2.1 Typical-Typical

2.8.2.1.1 Noise Figure

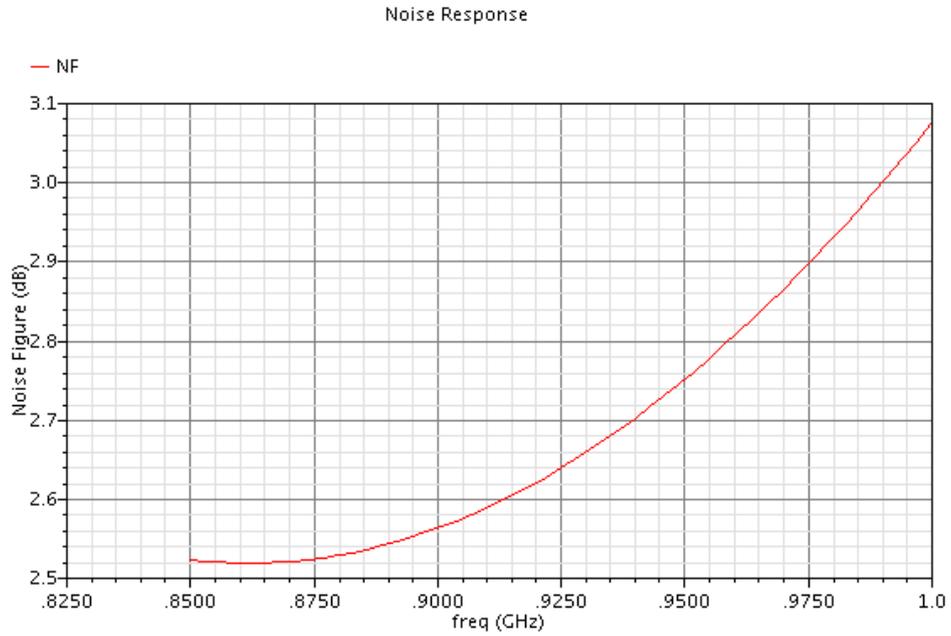


Figure 2.28: Noise Figure (T-T)

2.8.2.1.2 (S11)

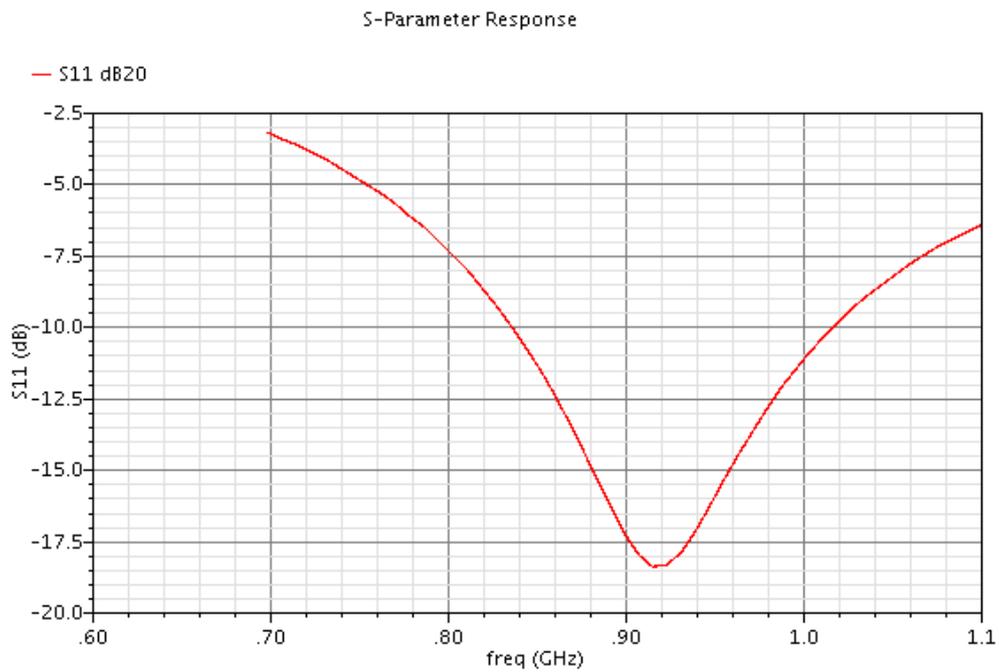


Figure 2.29: S11 (T-T)

2.8.2.1.3 (S21)

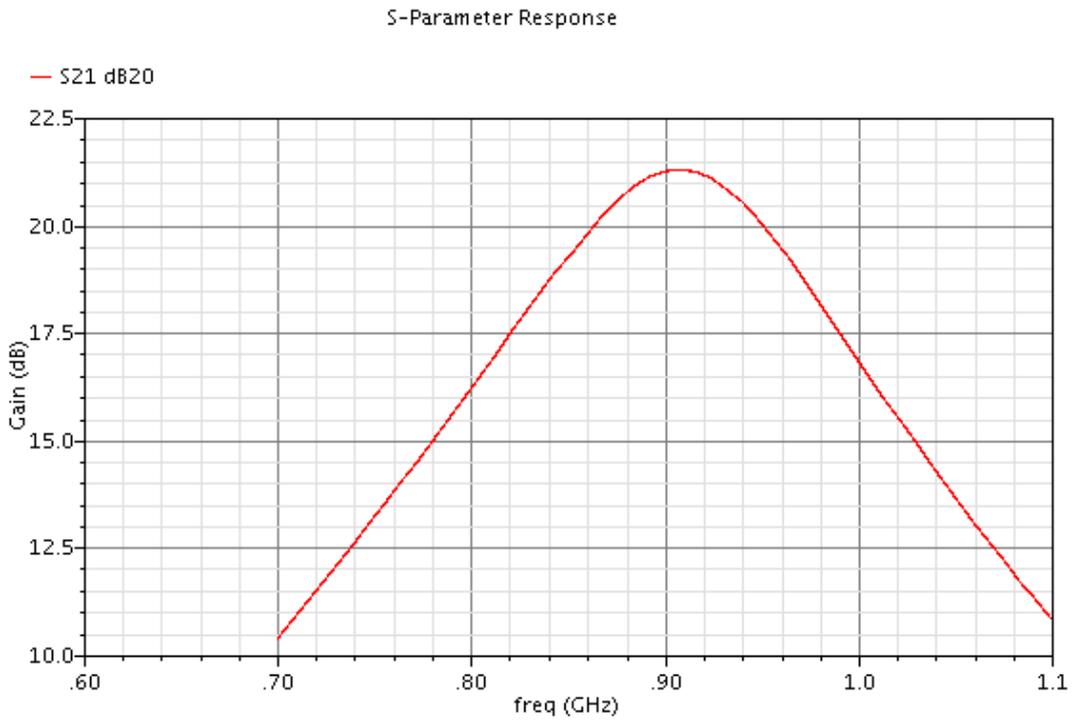


Figure 2.30: S21 (Typical-Typical)

2.8.2.1.4(S12)

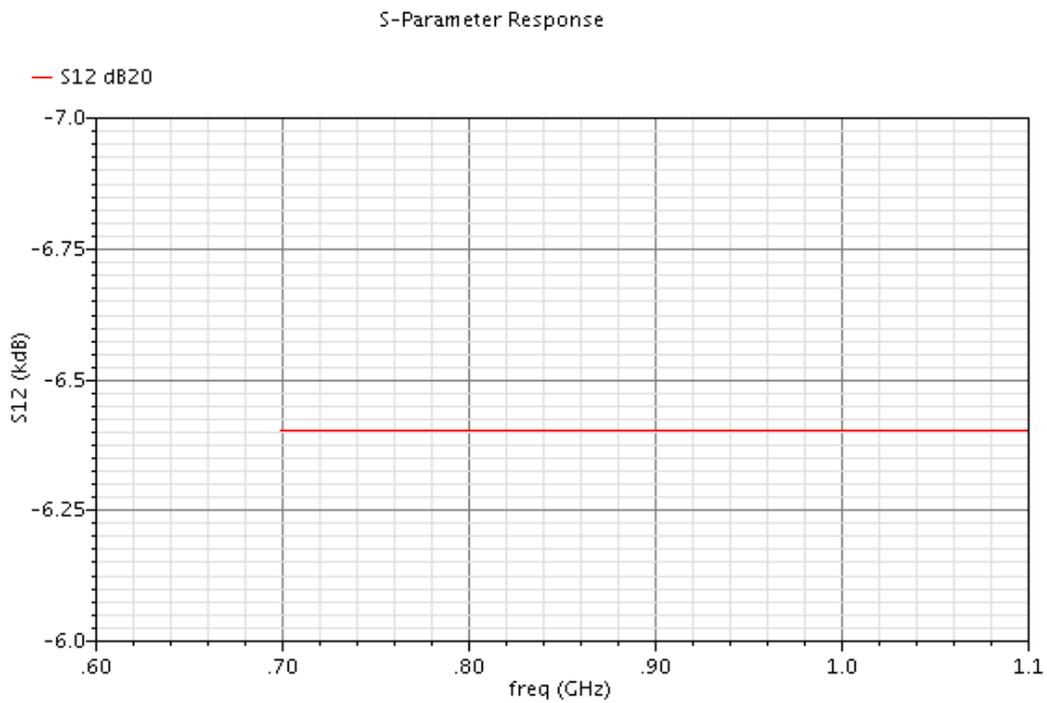


Figure 2.31: S12 (Typical-Typical)

2.8.2.1.6 Compression Point

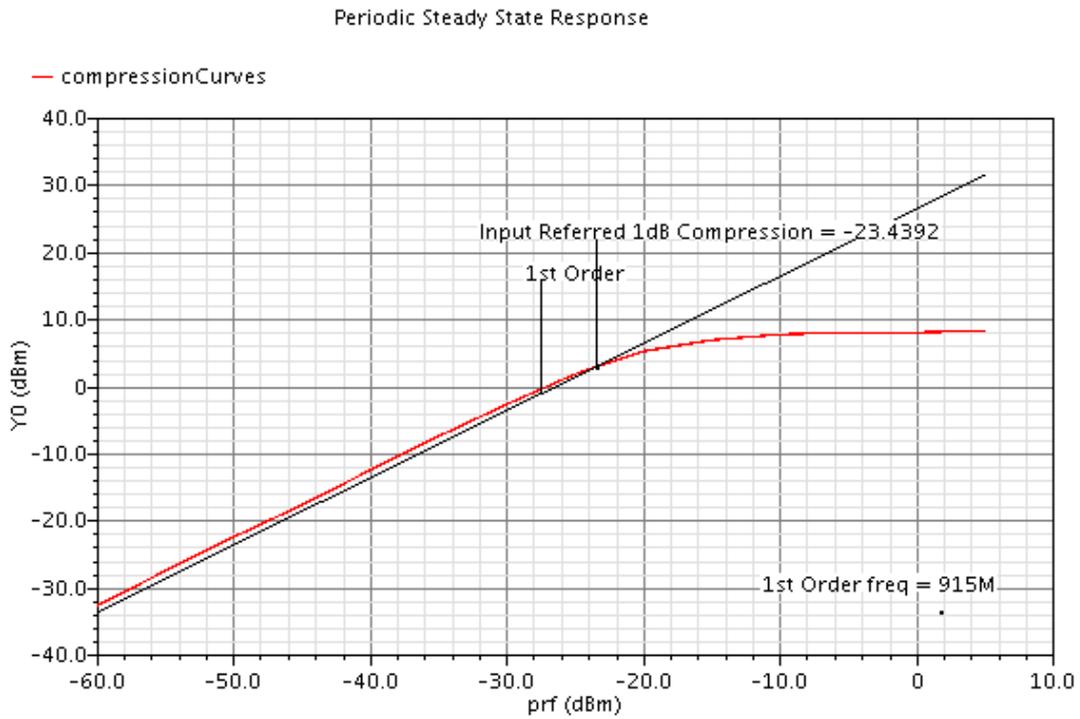


Figure 2.32: Compression Point (Typical-Typical)

2.8.2.1.7 IIP3

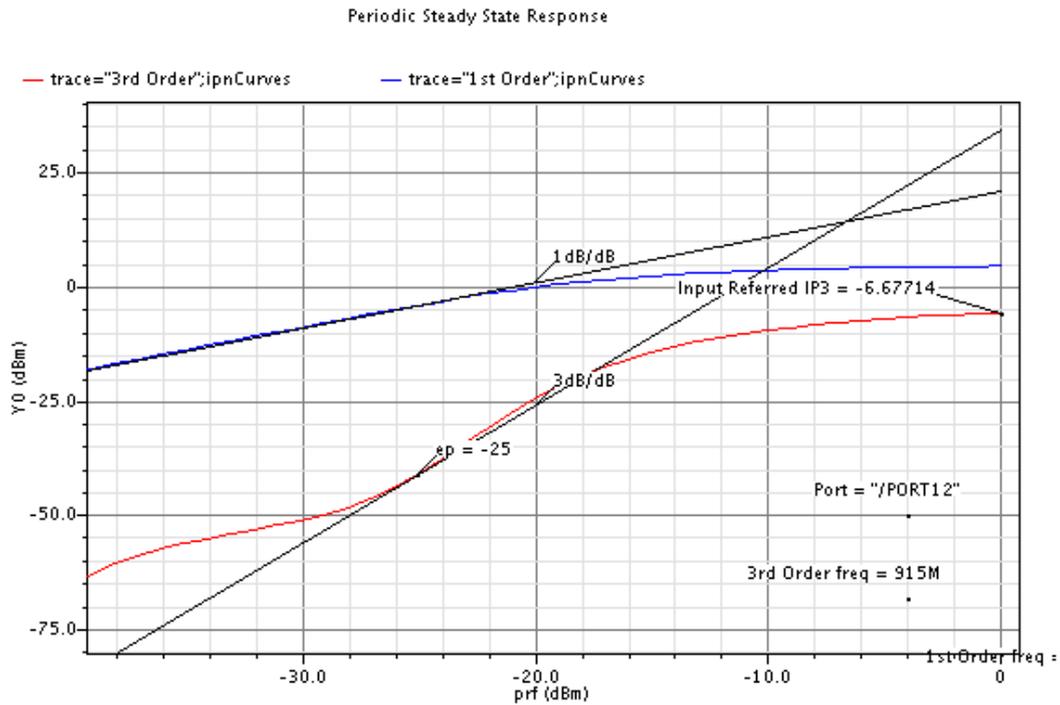


Figure 2.33: IIP3 (Typical-Typical)

2.8.2.1.8 Check Stability

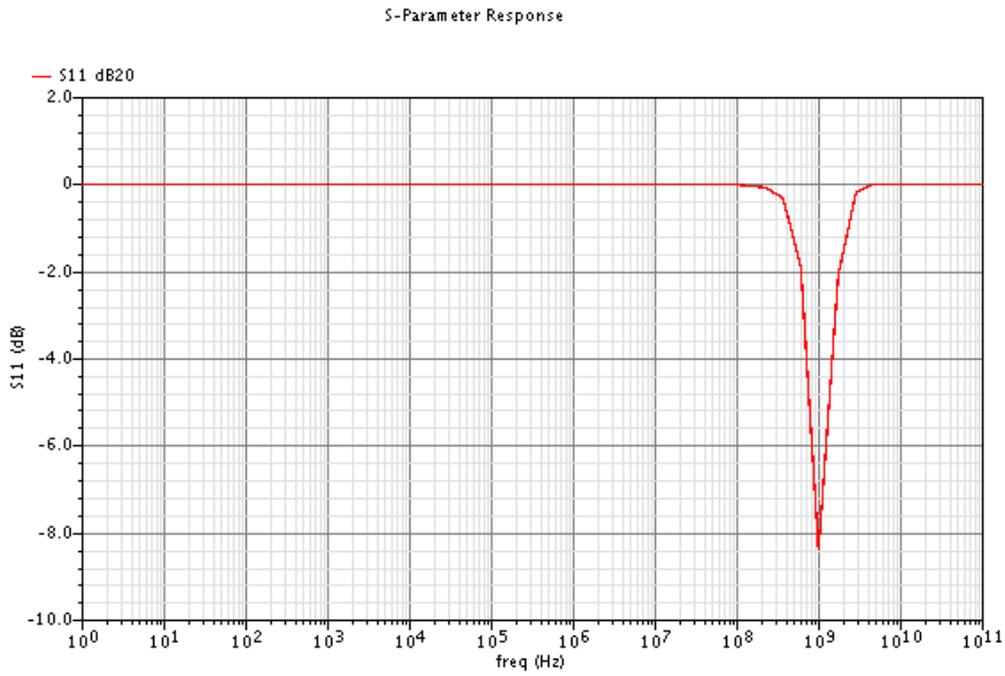


Figure 2.34: Checking Stability (Typical-Typical)

2.3.2.2 Corners

2.3.2.2.1 Noise Figure

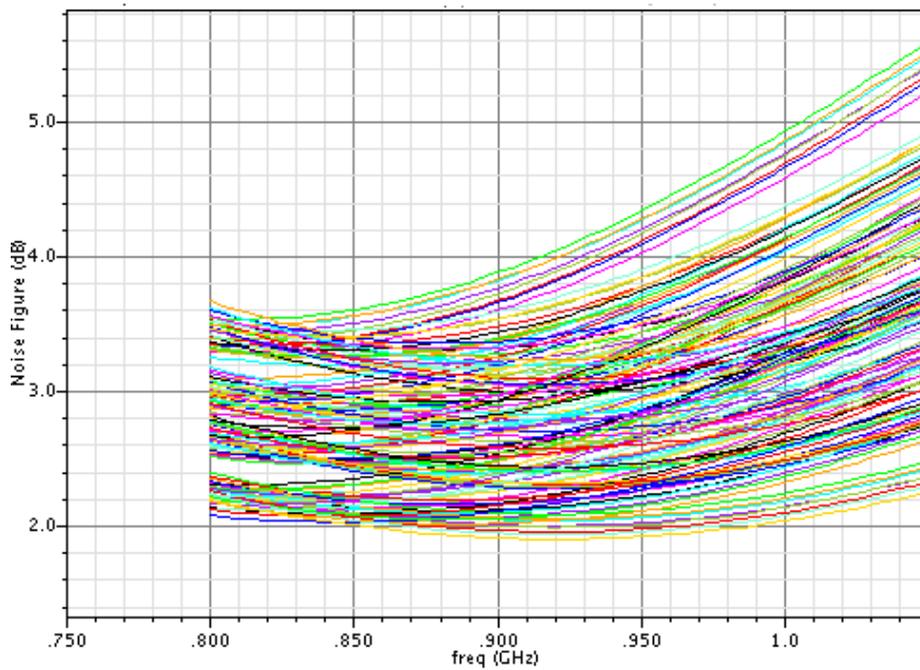


Figure 2.35: Noise Figure (across corners)

2.3.2.2.2 (S11)

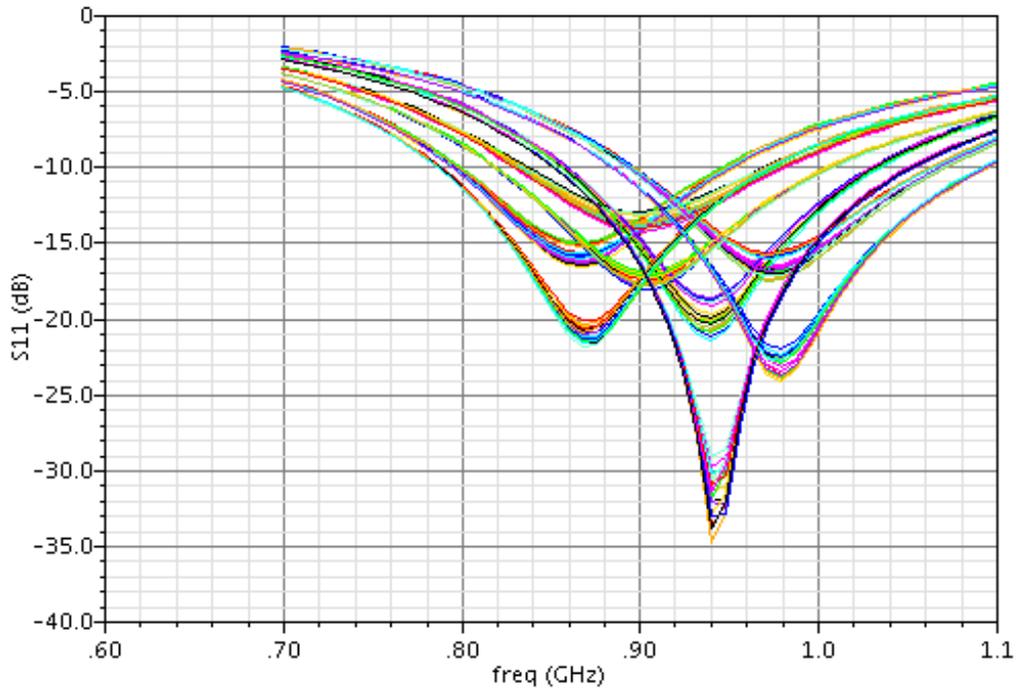


Figure 2.36: S11 (across corners)

2.3.2.2.3 (S21)

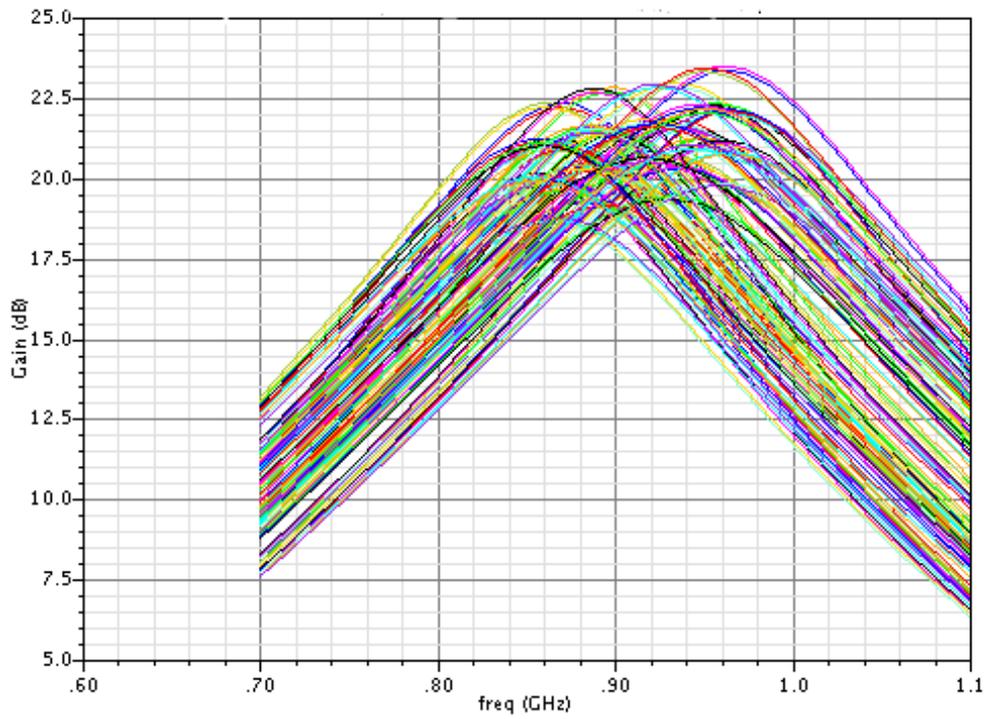


Figure 2.37: S21 (across corners)

2.8.3 Attenuator

2.8.3.1 Typical-Typical

2.8.3.1.1 Noise Figure

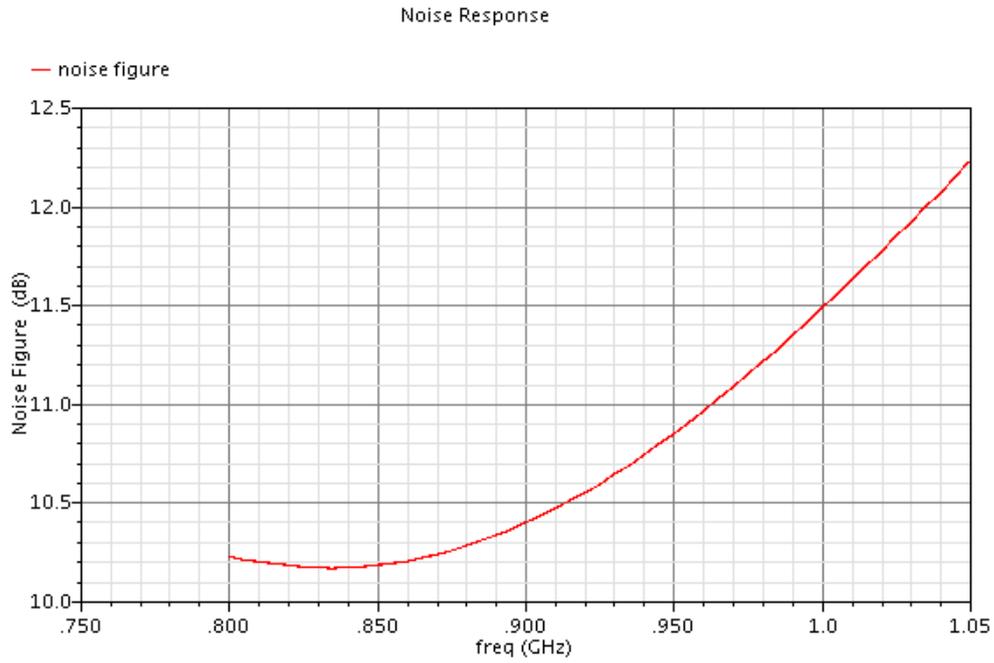


Figure 2.38: Noise Figure (Typical-Typical)

2.8.3.1.2 (S21)

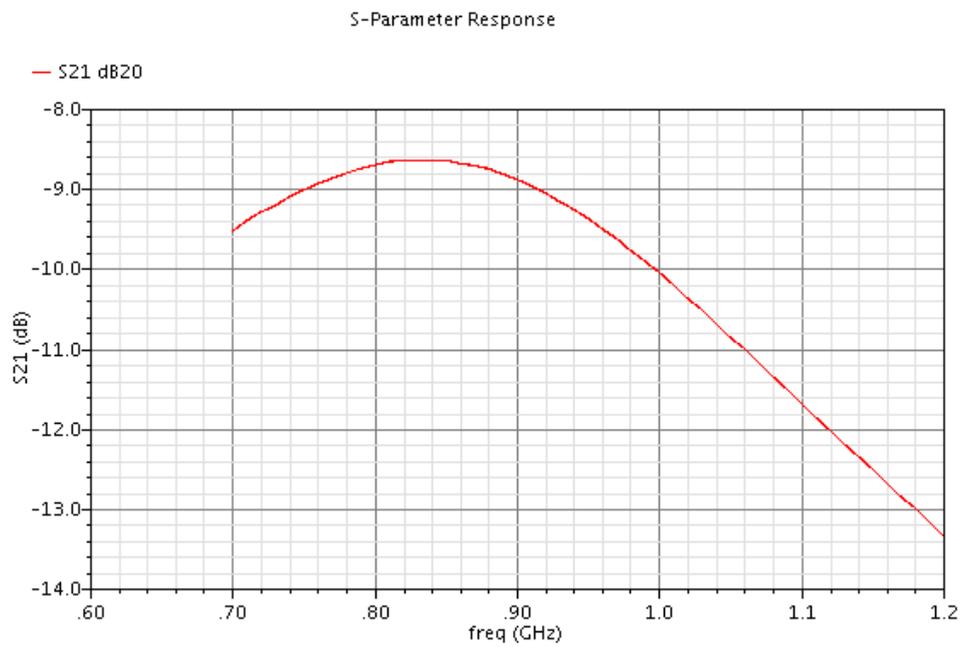


Figure 2.39: S21 (Typical-Typical)

2.8.3.1.3 Compression Point

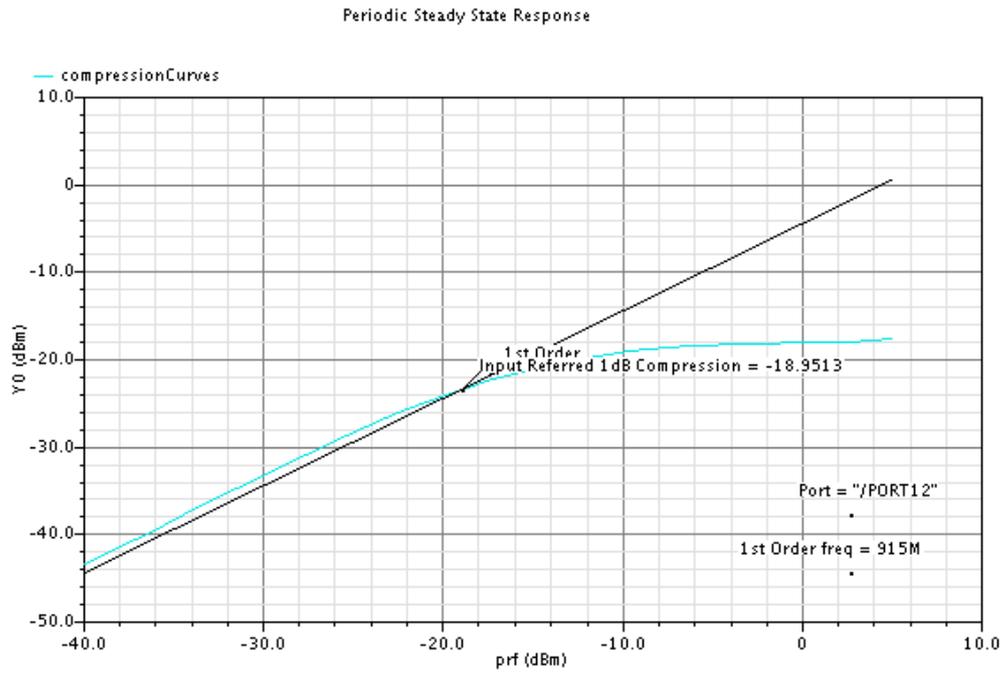


Figure 2.40: Compression Point (Typical-Typical)

2.8.3.1.4 IIP3

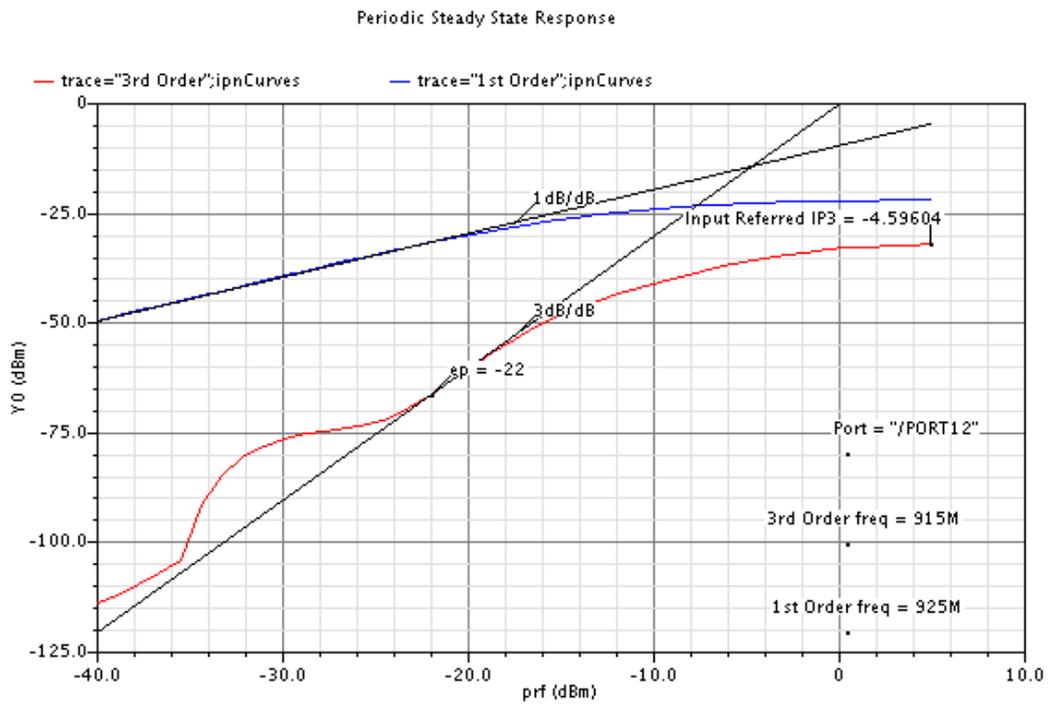


Figure 2.41: IIP3 (Typical-Typical)

2.8.3.2 Corners

2.8.3.2.1 Noise Figure

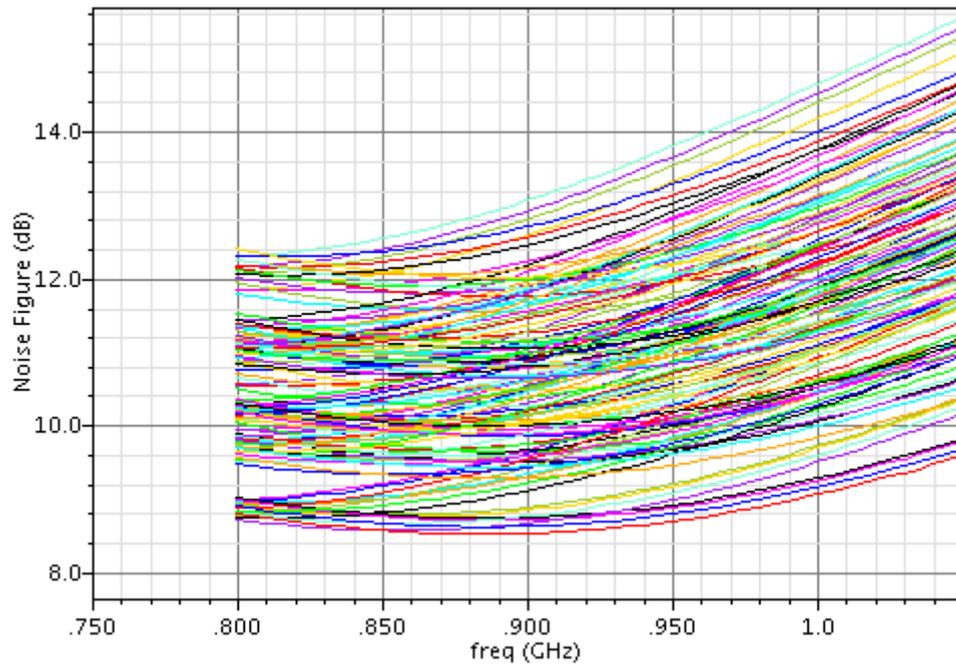


Figure 2.42: Noise Figure (across corners)

2.8.3.2.2 (S21)

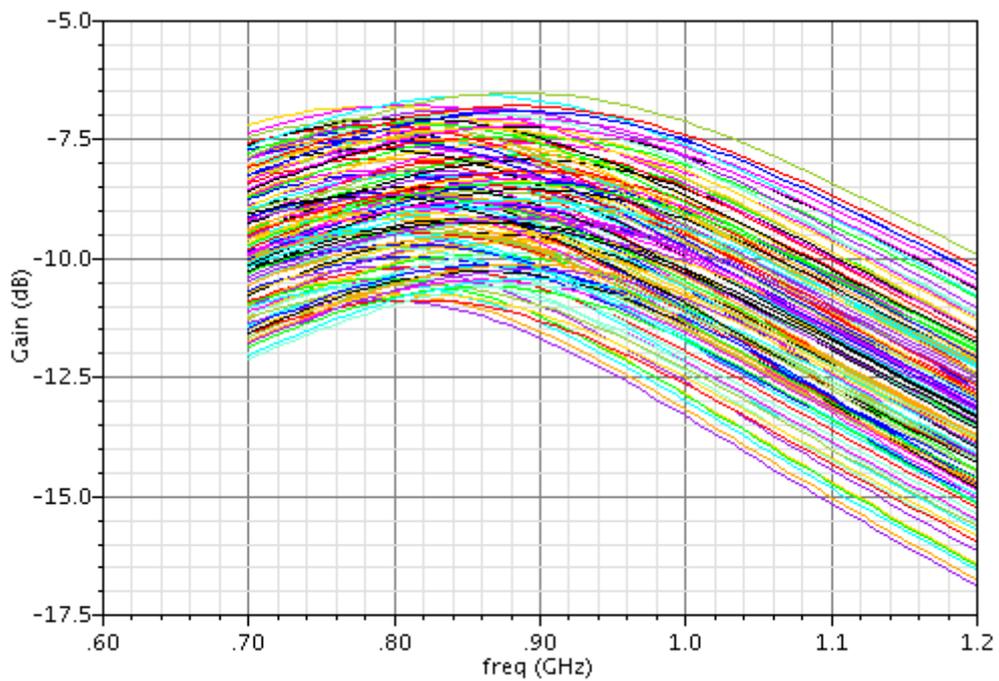


Figure 2.43: S21 (across corners)

2.8.4 Quality factor of the input stage

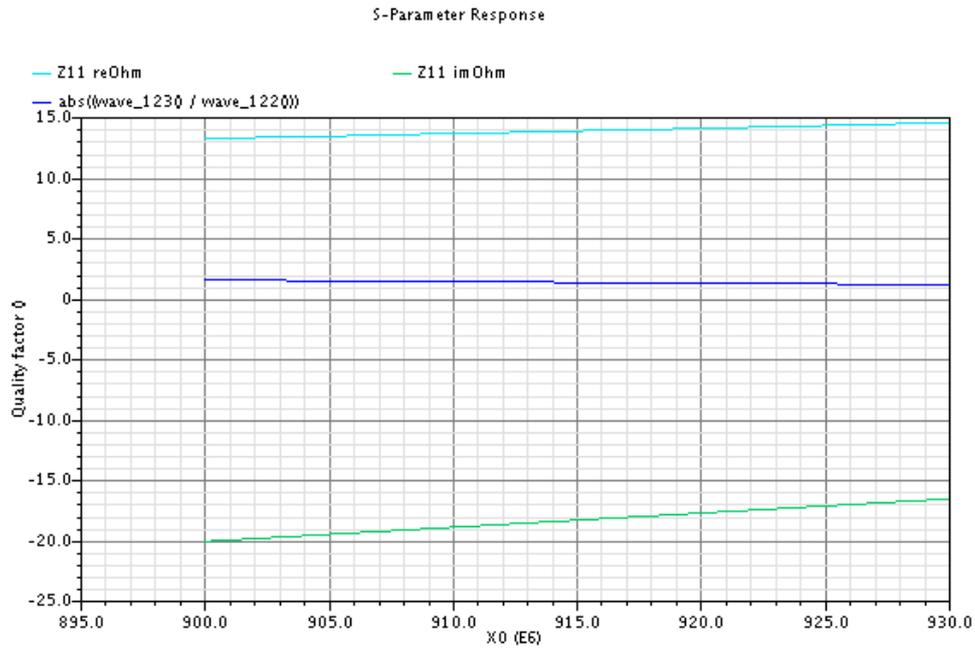


Figure 2.44: Quality factor of the input stage

The Quality factor is equal to 1.3 that will give gain to the signal before it passes through the main transistor.

2.8.5 Specs achieved

2.8.5.1 LNA

	Gain	NF	S11	IIP3	power	Compression Point
Typical- Typical	21 dB	2.55 dB	<-15 dB	-7 dBm	2 mA	-23.4 dBm
Fast- Fast	21 dB	2 dB	<-10 dB	-7.6 dBm	2.14 mA	-23.02 dBm
Slow- Slow	18 dB	3.8 dB	<-10 dB	-3.2 dBm	19.3 mA	-19.8 dBm

Table 2.6: Specs achieved

2.8.5.2 Attenuator

Gain	NF	S11	IIP3	power	Compression point
-9 dB	10.5 dB	<-15 dB	-4.59 dBm	2mA	-18.9 dBm

Table 2.7: Specs achieved

The simulation for all corners in the appendix.

2.9 Layout

2.9.1 Introduction to Layout

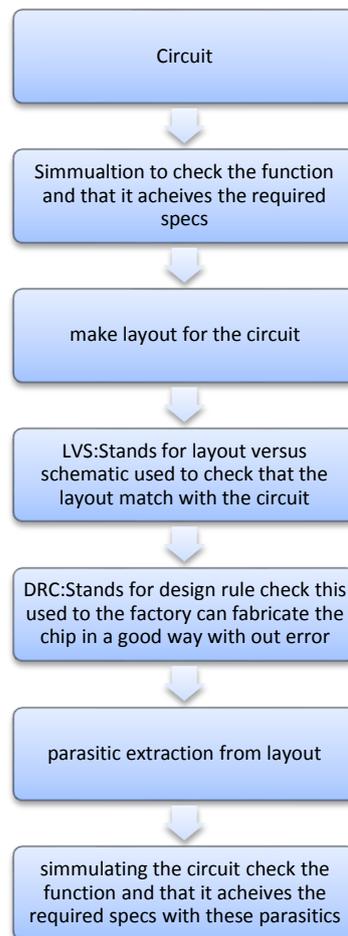


Figure 2.45: Layout process

2.9.2 Layout of LNA circuit

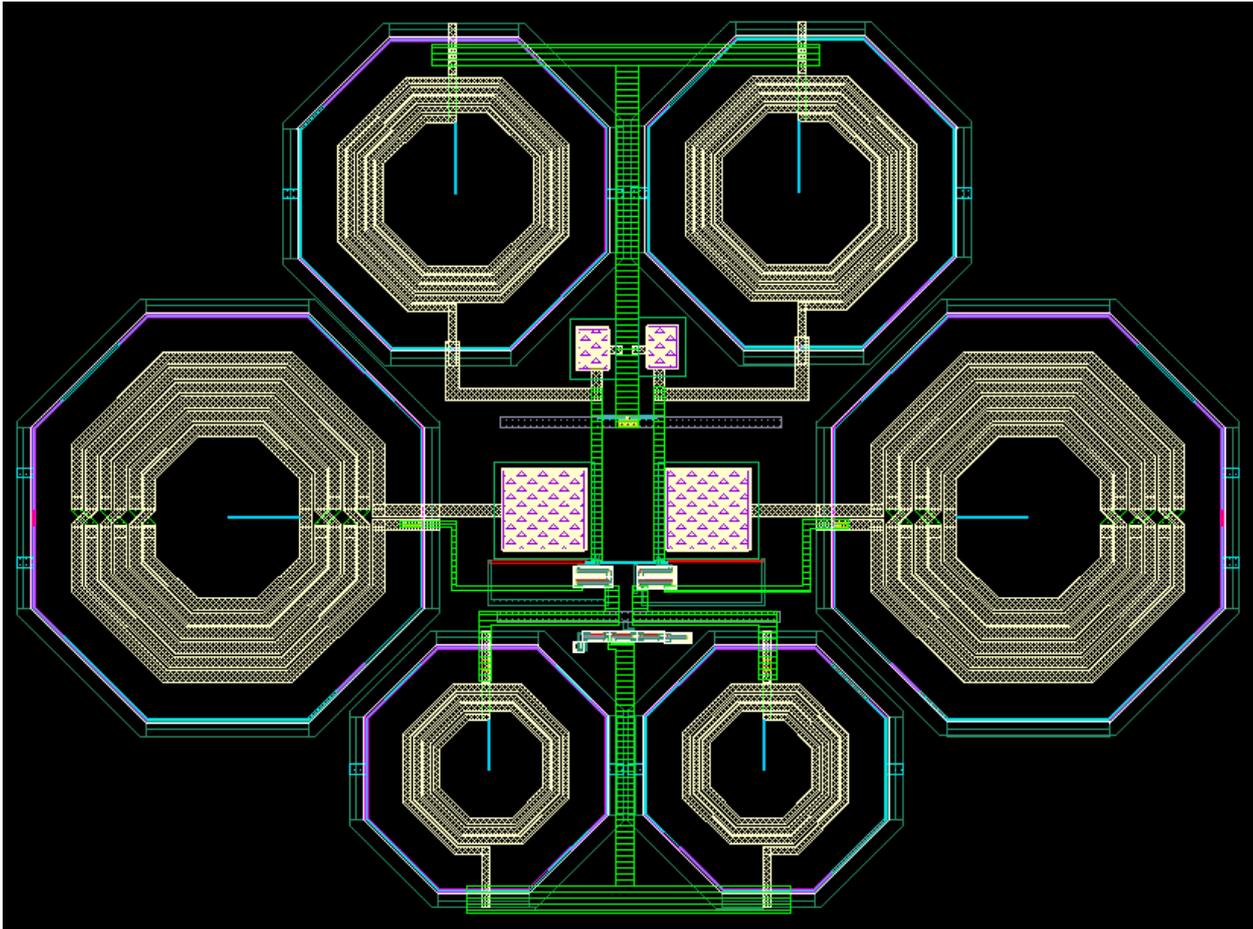


Figure 2.46: Layout of LNA circuit

This layout is untested as I have neither DRC nor LVS

2.10 Conclusion

LNA is considered the most important block in the receiver chain that its gain and Noise Figure affect too much on the system and it can't be replaced or removed from the chain of receiver till now.

We observed that the inductive degenerated LNA is the most powerful topology in LNA that used to get the least Noise Figure. Also, we observed that frequency band from 902 M to 928M leads us to use large components and these large components increase the Noise Figure by approximately 1 dB due to parasitic.

2.11References

- [1] Behzad Razavi, “RF microelectronics” second edition.
- [2] Thomas H. Lee, “The design of CMOS Radio-Frequency integrated circuits” second edition.
- [3] Highly Linear Low Noise Amplifier. (May 2006) by Sivakumar Ganesan, B.E. (Hons.).
- [4] http://amesp02.tamu.edu/~sanchez/Heng_LNA.pdf
- [5] Design of Low Power / Low Sensitivity to Process Variations Integrated RF Receivers in Deep Sub-micron CMOS Technology by DR. Sameh Ahmed Assem Mostafa Ibrahim.
- [6] weble.upc.es/rfcs/Material/Tema5_LNA.pdf
- [7] <http://www.eel.upc.edu/rfcs/>
- [8] MOS Differential LNA Design Tutorial
- [9] Maximintegrated.com
- [10] <http://petrus.upc.es/rfcs/Material/LNA.pdf>
- [11] <http://rfic.eecs.berkeley.edu/142/pdf/lect10.pdf>
- [12] Practical Considerations for Low Noise Amplifier Design By Tim Das

3 MIXER

3.1 Introduction:

This chapter introduces a Mixer designed for low voltage low-if receiver (915 ISM band). Mixer has a tradeoff between its linearity, Gain and Noise figure requiring effort to get an optimum value. The proposed design is folded gilbert mixer for low voltage applications. It is designed in tsmc 0.13 μ m. The mixer gain has suffered from large variation across corners leading to the use of constant gm-cell decreasing variation from 6.5dB to 2.9 dB. Sections are organized as following: Section 3.2 talks about Mixer definition, Section 3.3 talks Mixers topologies, Section 3.4 talks about Specifications Required, Section 3.5 talks about Proposed Topology and more deep analysis, Section 3.6 talks about design Methodology, Section 3.7 explain analysis used in Simulation and Section 3.8 shows typical results and corners Results

3.2 Mixer Definitions:

Mixer is a block used to convert down RF signal to low-IF signal or even zero-IF in order to be able to be converted to digital. The mixer has gain property during conversion. Mixer has different definitions about Noise Figure and Gain. These definitions are clarified in the following definitions.

Noise Figure:

There are two definitions for Noise figure which are SSB (single side band) and DSB (double side band) NF. This can be clarified in case of direct-conversion receiver, where RF signal is converted down to zero IF frequency. In this case there is no image, but there is an important issue concerning type of signal, if it is SSB or DSB. In case of DSB signal here RF signal appears to be found around carrier frequency so when converted to zero frequency, this signal is converted with its noise so when we look at positive frequency only, we see that both signal and noise are doubled. But in case of SSB where RF signal is only in one side of carrier, when conversion happens the signal of SSB is converted but the noise in both sides is also converted, so after conversion and looking at positive frequency only, we found that **noise** in signal band is double where signal is not. Meaning that SNR of DSB signal is double that of SSB. So we define two definitions for SSB signal named as SSB NF and other for DSB named as DSB NF and SSB NF is 3dB higher than DSB.

This definition is extended to non-zero IF receivers where we have image. As image signal contains noise and this noise is folded down in addition to signal noise, to signal band making it higher Noise figure than that of direct conversion receiver (DSB signal) of same signal.

Anyway the concerned type, depends on Receiver architecture. In case of low-if receiver using complex filter after Mixer, this removes image and its noise, making us interested in DSB NF of mixer in system design.

Gain:

Mixer convert signal from certain frequency to another one so we have to redefine the gain definition, because the output frequency is different from input frequency. This gain is called **Conversion gain** and it is defined as signal amplitude of down converted signal to that of RF signal

Linearity:

Mixer has nonlinear effect with LO signal and linear effect with RF signal

Mixer like any block has compression point and IIP3. But here intermodulation product appears in down converted signal band not in RF as we talk here about frequency conversion and after mixer we are interested in down converted signal band not the RF one.

Feed-through (port-to-port) isolation:

As in mixer we have two input different signals of different frequencies (in case of non-zero IF receiver) and output has different frequency. So any signal from any port can feed-through to over port causing problems

So there are important definitions which are RF-IF, IF-LO, LO-RF (each one in both direction) feed-through, which measure amount of signal feed-through from port to another port. RF-LO feed-through can have serious problem it may cause frequency pulling of oscillator, LO-RF have problem of self mixing leading to DC-offset, RF-IF feed-through cause leaked even order distortion.

Unbalanced, single balanced and double balanced Mixer:
 Unbalanced mixer mean that we see @ output of Mixer, RF and LO signals beside IF(converted signal).
 Single balanced mean that there is RF or LO feed-through @ output.
 Double balanced mean No RF or LO feed through @ output.

3.3 Mixer Topologies

Passive Mixer:

Idea:

We have here switches M1 and M2 are driven by differential LO frequencies, commutating input signal to outputs (switching ON and OFF in each branch). We notice here that these switches not carry any DC current As M1 and M2 act as switches for input RF signal, mean from time domain point of view as square wave of frequency LO multiplied by RF signal (which is from frequency domain point of view frequency translation). This square wave contains harmonics where each is multiplied with RF signal. Using Fourier Series square wave is decomposed to $[0.5 + \frac{2}{\pi} \cos(\omega_{LO}t) + \frac{2}{3\pi} \cos(3\omega_{LO}t) + \dots]$. Since we are interested in first harmonic so we can conclude gain will be $\frac{2}{\pi} = -4\text{dB}$ (passive Mixer)

We notice that this topology doesn't suffer from **RF-IF feed-through** but suffer from **LO-IF feed-through**. This can be solved using the double balanced version, as shown in figure

Here gain is the same as (signal become differential)
 This topology is convenient for receiver having differential LNA
 Noise:

This core does not suffer from flicker noise (as no current pass through switches) and this is great advantage for passive mixers.
 But it has thermal noise due to switches and R_{Load}

Active Mixer:

Its Simplest configuration (see figure). Here RF signal converted to current acquiring gm of the transconductor M_{RF}, then pass through (M_{SW}) suffering from switching then converted By R_L to output voltage

Like passive mixer, Active mixer has double balanced version
 This topology has disadvantage of many cascaded transistors (there

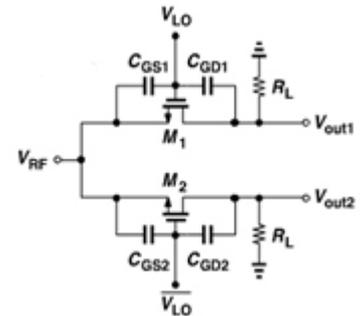


Figure 3.1 Single Balanced passive Mixer

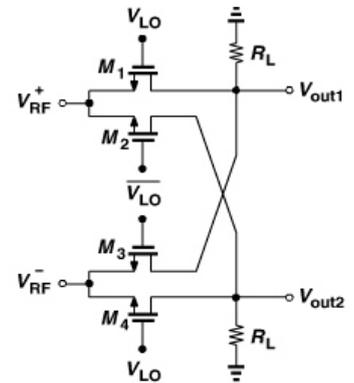
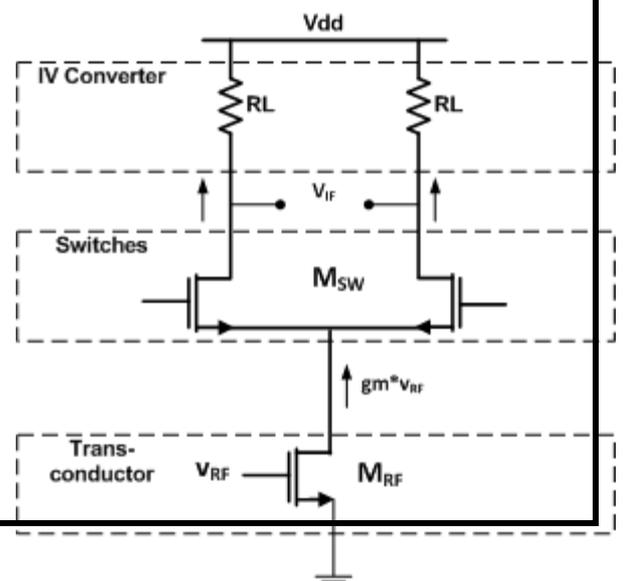


Figure 3.2 Double balanced passive Mixer



is also current source for fully balanced one) which make inconvenient for low voltage applications.

Folded mixer:

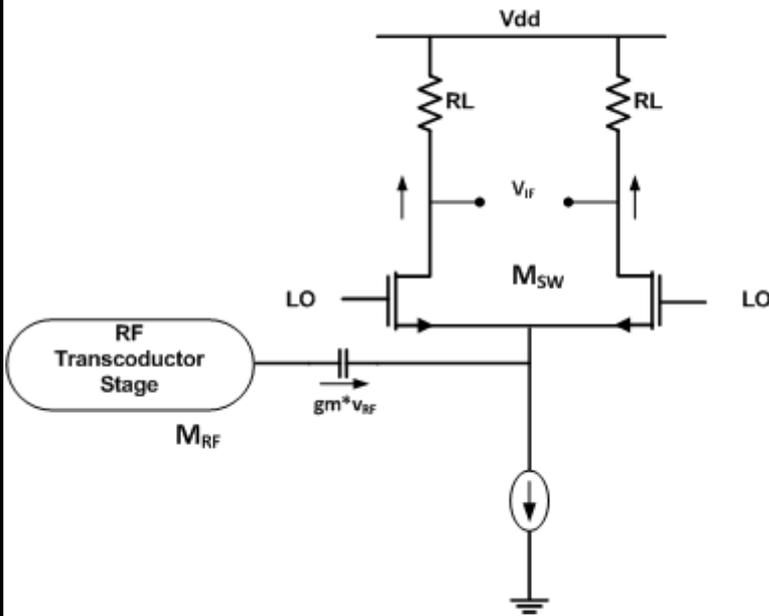


Figure 3.3 Active Gilbert Mixer (double balanced)

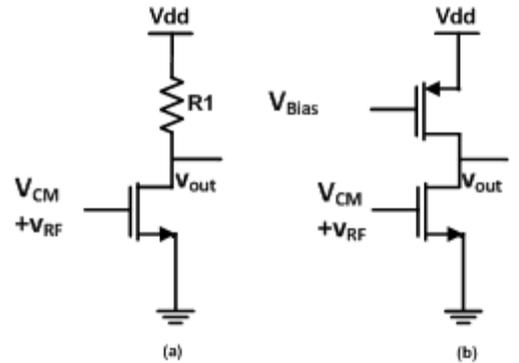


Figure 3.4 Folded Mixer

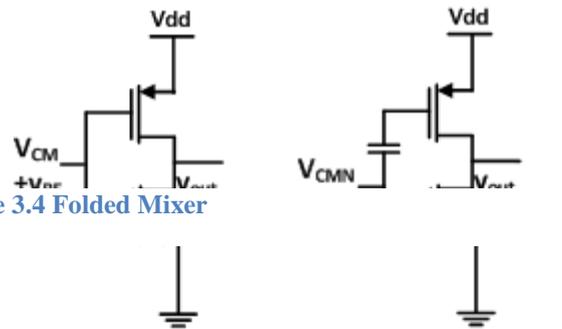


Figure 3.5 Folded Mixer Topologies

common source with pmos load (c) inverter (d) modified inverter

Here headroom problem is solved by folding RF transistors from switches to have independent bias, but folded mixer needs more Power which is a principle drawback, Folded In put stage can be implemented like [figure 1.5(a)] .but this topology suffer from ac signal leakage through R1 which itself has constraint on its size due to headroom .This can be solved using Pmos load (need CMFB to adjust CM @ output) [figure 1.5 (b)].

We can also get benefit of pmos load and use it to amplify signal (current reuse), increasing gain with same current making whole input stage act as inverter [figure 1.5(c)].But using inverter has constraint on min supply which make it not suitable for low voltage application this thing can be clarified by finding min. supply ,which is

$$V_{ddmin} = V_{gsn} + V_{gsp} = V_{ovm} + V_{ovp} + V_{thn} + V_{thp}$$

This last issue can be solved by separating bias of Nmos and Pmos [figure 1.5(d)].

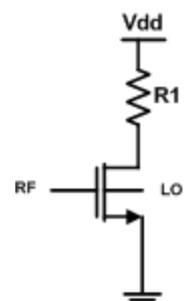
Bulk injection Mixer:

Despite passive and active Mixers, RF and LO frequency are entered in same Mosfet.

As here V_{th} is function in LO so multiplication happens inside Mosfet.

This topology has advantage over Gilbert mixer as number of stacked Mosfets decrease by one, making it suitable for low voltage applications.

From Noise figure point of view ,Bulk injection is noisier than active mixer as LO signal is inserted directly through body terminal contributing to noisier drain current o Mosfet transistor



Topologies Comparison:

	Passive	Active	Bulk injection
Gain	small	Large	Large
Noise Figure	smaller	Medium	Larger
IIP3	High	High	High

3.4 Specifications required and what achieved

	Min	Typical	Max	Spec. required	Unit
Gain	9.4	11.46	12.3	12	dB
NF (SSB)	11.8	12.55	14.3	15	dB
IIP3	0.7	3.07	5.2	2.5	dBm
Comp.	-11.3	-8.54	-5.8		dBm
Power	=	2.15	-	2-3	mA

3.5 Proposed Mixer

- We use Active Mixer for the gain specified by system specifications on Mixer.
- We use folded topology to be convenient for low voltage technology we use and for high linearity need on Mixer.
- We use last one explained of folded circuits (current reusing one) to reuse current consumed.
- Modified constant gm bias circuit is used to bias Transconductor circuit (to decrease gain variation) and modeled poly current biasing used for Mixer core (to have constant CM output of Mixer) .

These variations can be decreased using

- combination of resistance has less variation with temp
- Using const gm biasing
- Improved const gm as will be explained in Design Methodology.

Linearity:

Linearity of mixer is important issue must be taken in consideration.

Linearity of mixer is affected by many parts of circuit where signal flow and each one can be dominant for up certain extent.

Where linearity is affected by:

- Increasing V_{ov} of Transconductor mosfets increase linearity ,but that decrease gm (for same power)→ decrease gain and increase noise
- Compression can happen at switch sources seen impedance of switches and that must be minimized by increase gm by increase current -->more power consumption → also can decrease Cm output put which can in return make switches enter triode at large swing of signal which in return increase impedance seen at switches sources which in return cause compression at same point
- Signal compression happen when signal be clipped due to V_{dd} → good design of CM output and output swing
- Switches entering triode cause nonlinearity signal. On switches must not enter triode when other switch not enters off region yet.

Noise:

Transconductor mosfets

- **Thermal noise:**

Can be decreased using higher gm -->more power and can affect linearity.

- **Flicker noise:**

It is not important as it will be unconverted to LO and 3 LO..., but can appear at output by small effect due offset in switching happen dur to switch flicker noise (direct)

Current source (Transconductor stage)

- **Thermal noise:**

Thermal noise appears @ output →This noise appears with folding while conversion with LO harmonics.

But @ gradual transition of switching these noise not appear @ output as we take differential (As the mixer core act at this instant like a differential amplifier seeing this noise as CM signal)

Solution: Decrease Current source gm by decrease current (which have side affect on gain and linearity)

- **Flicker noise**

It is not important as it will be unconverted to LO and 3 LO...

Switches Mosfets :

- **Flicker noise**

1. **Direct switch noise:** which has modulation effect of switching? This leads to advance or retard of zero crossing point. This leads to noise sampling @ frequency $2\omega_{LO}$.

This noise current can be proved to be as average= $I*V_n(f)/A/\pi$ see darabi page 3

A is sin wave amplitude and I is current and $V_n(f)$ flicker noise .To decrease noise we have to increase area. Decrease current will have back side effect on signal gain .

2. **Indirect switch noise:** This is due to finite capacitance at node X . It can be shown that it has

current value at output $= 2 * f_{LO} * C_p * V_{n-switch}(f)$

- Thermal noise :
 - Like flicker noise, It is affected by sampling at ω_{LO} and its harmonics.
 - It can be proven that thermal noise current @ output to be $= 4KT \gamma \frac{I}{\pi * A}$, where A is sin wave amplitude.

Port isolation

As we use double balanced topology we don't not suffer here from RF-IF feed-through or LO-IF feed-through.

Summary:

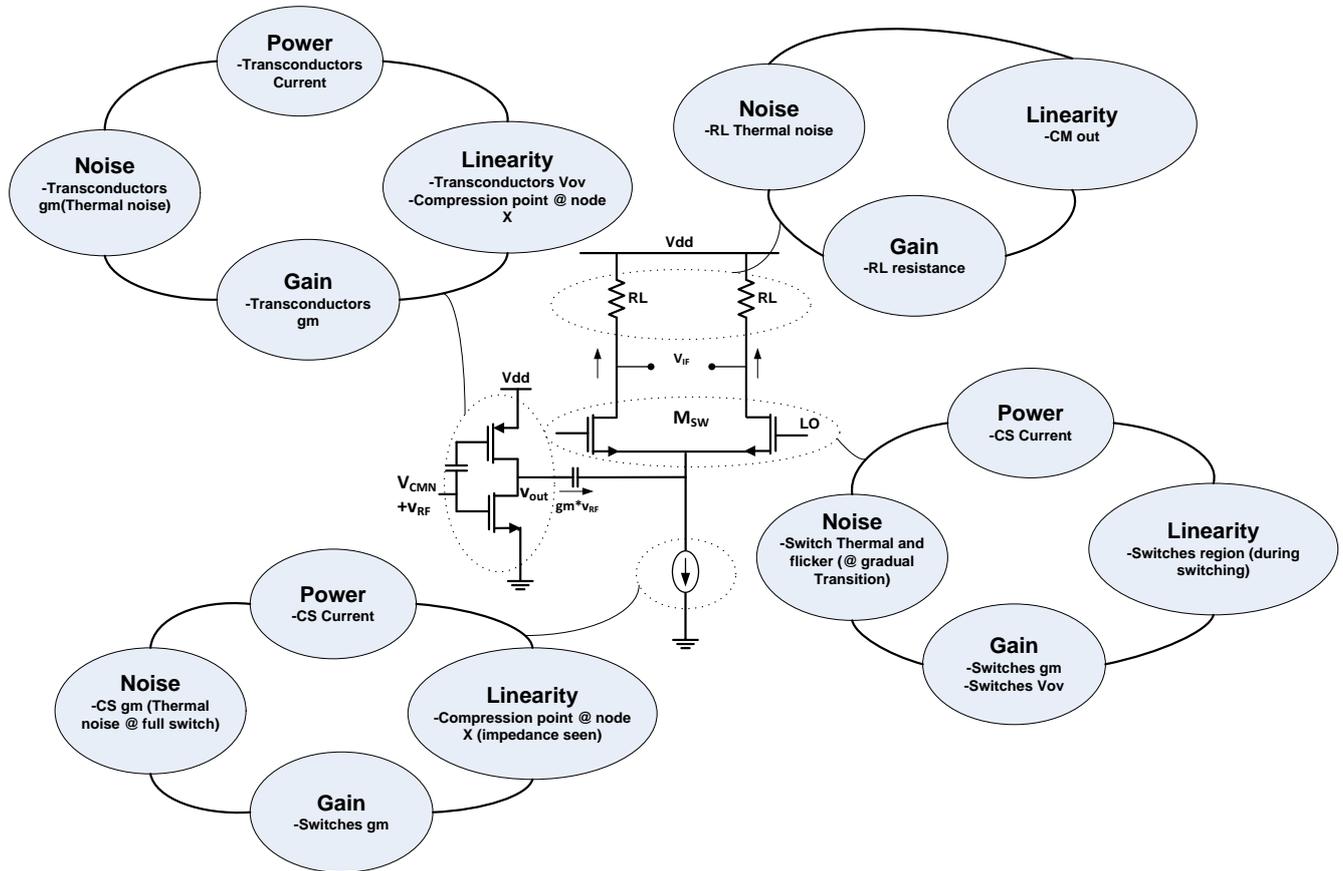


Figure 3.8 Summary of Specifications

3.6 Design Methodology

For simplicity the mixer can be divided into 3-parts: Transconductor, switch and load.

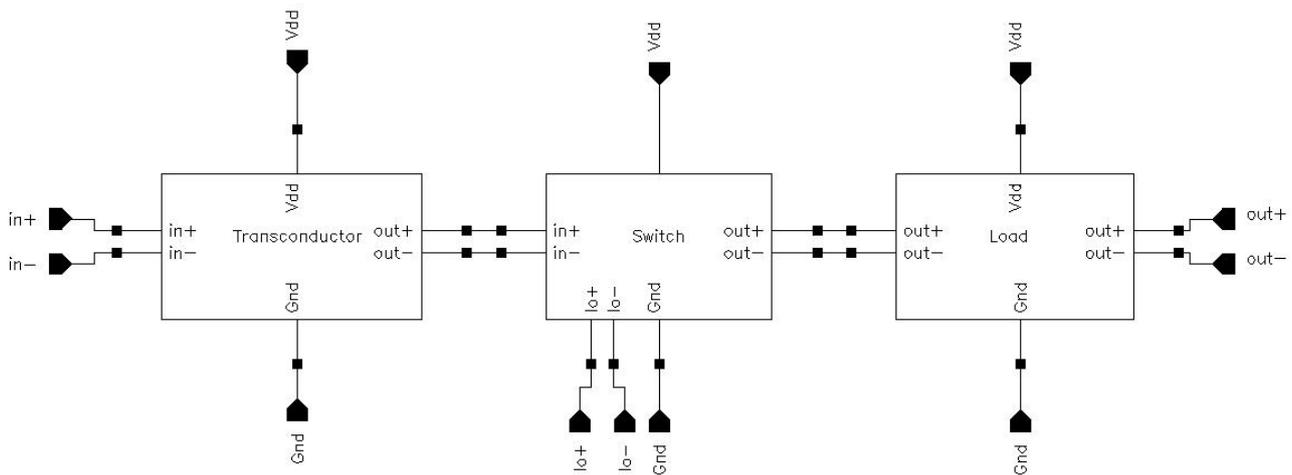


Figure 3.9 Mixer Parts

Transconductor Stage

Transconductor Mosfets:

Here this stage is very low gain amplifier as the load seen by stage is switches source and that not make restriction on swing in the stage (but to some extent as explained before)

For linearity issue ,to avoid compression at RF Mosfets .We assume stage as a simple differential amplifier, since the target compression point is about -8 dBm equating it to $20\log_{10}(V_{OV}^2 \cdot 10)$ as simple approximation we get $V_{OV} > 90\text{mV}$.And we take length to be min to decrease parasitic caps for BW issue.

Current Source:

We have Power constraint for all mixers from 2 to 3mA. So we will consume less than 1mA for this stage

To be in strong version region and to pass corners \rightarrow take V_{OV} be around 100mV. We specify great length for CS to decrease channel length modulation effect (350nm \rightarrow max. of rfnmos). We adjust width for needed current which give us needed total gm for gain and noise spec (850mA).

Since this stage see small output resistance in AC but in DC (CM) point of view seen high impedance node \rightarrow so this topology needs CMFB circuit to stabilize CM output. As we have here very small gain amplifier so we can use simple CMFB circuit implemented using resistances only (15 K Ω) (as shown figure) connected from output to pmos input mosfets gate (having advantage of no need of bias for pmos)

Biasing:

Transconductor Mosfets is biased using diode connected Mosfet from constant current source and connected by Resistance of 30Kohm to isolate AC signal from DC bias and its large value to decrease loading effect on LNA. CS is biased using current mirror from constant gm circuit.

After designing the circuit and running across corners ,It was found that Mixer suffer from large variation across corners about 6.5dB and that was solved using constant gm circuit as shown in figure [1.10].

Constant gm-circuit:

This circuit is used to bias a current source to a circuit, this current bias the desired circuit in a mode to have constant gm. As we know corners effect V_{th} of Mosfet and in addition to that the mobility. So we need to have a circuit give us a current depend on mobility not constant so that have we get constant gm.

Since $gm = \sqrt{2I_D \mu_n C_{OX} \frac{W}{L}}$, so we need to have $I_D = \frac{M}{\mu_n C_{OX} \frac{W}{L}}$, where M is constant, so

that we can eliminate effect of mobility.

The circuit as shown contains of 2 NMOS , 2 PMOS and Resistance.

The 2 current Mirror PMOS enforce current in the two branches are equal.

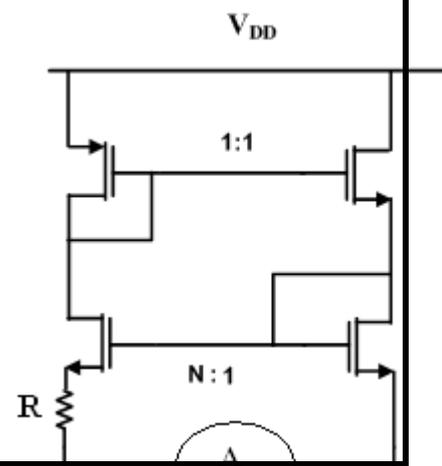


Figure 3.10 Constant gm circuit

But the two lower Mosfets have ratios N:1 . by solving voltage equation in loop A and solving current equation we can prove that current be:

$$I_D = \frac{1}{R^2} \left(1 - \frac{1}{\sqrt{N}}\right) \frac{2}{\mu_n C_{OX} \frac{W}{L}}$$

So we must have $N > 1$ and R to be constant resistance across corners which can be achieved by off chip resistor. Then we get gm value equal:

$$gm = \frac{1}{R} \sqrt{G}, \text{ where } G \text{ depends on } N .$$

Return to mixer . Since gain $= \frac{2}{\pi} gm * R_L$, where R_L as we say before have variation across corners. So if we have constant gm we will not have constant gm due to variation in R_L . but this effect can be decreased by making R in constant gm circuit of same type of R_L . So we have gain depends only on ratio between R_s of same type.

Also this eliminate the issue of parasitic capacitance of off chip resistance which may affect stability of circuit. Practically this will not make variation =0, due to channel modulation effect in mirroring currents in this circuit itself and to the Transconductor.

This circuit has disadvantage:

- Large variation in power in Circuits.

Design:

N was taken to be equal =3. The circuit Mosfets are chosen to have Lengths =10μ, to decrease effect of channel modulation and we will consume current in every branch in typical value = 15μ.

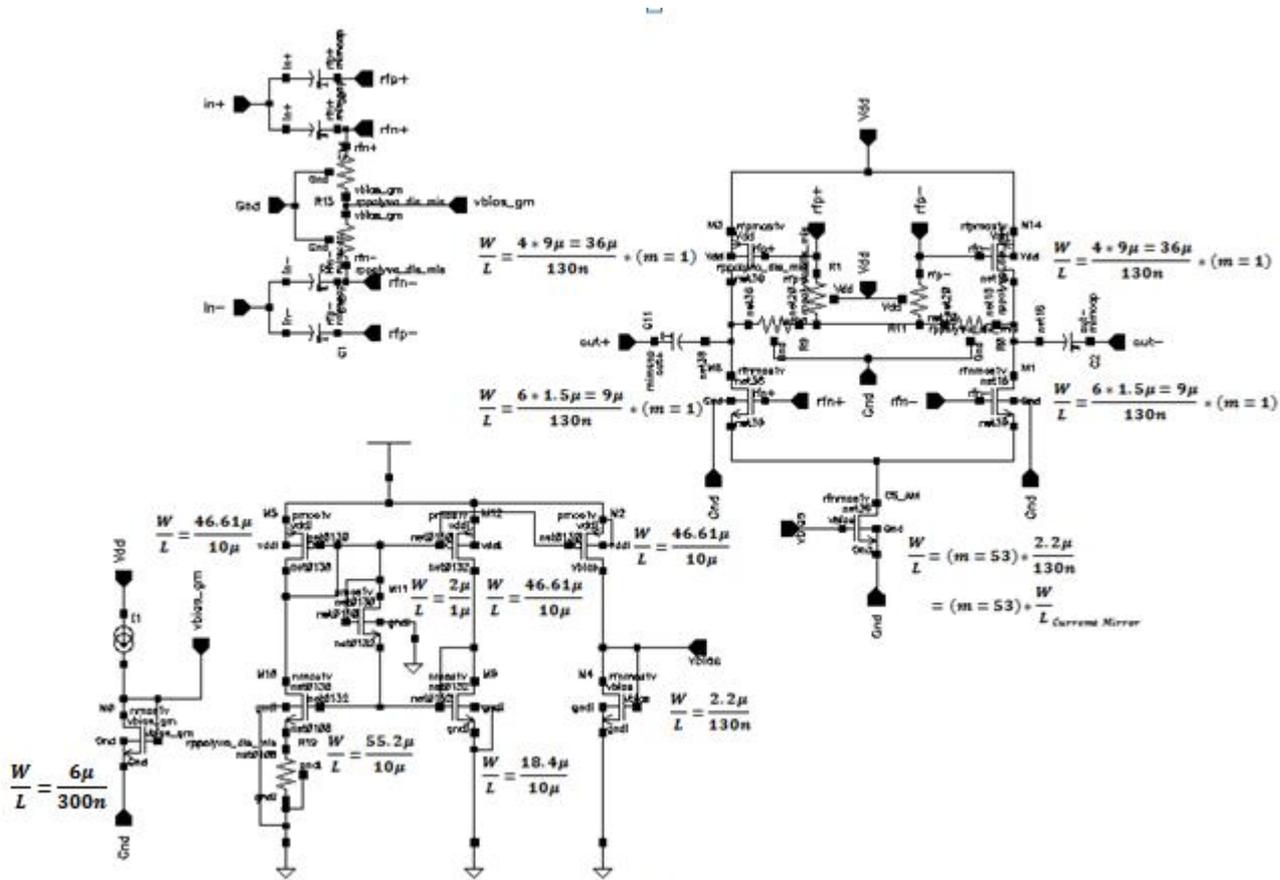


Figure 3.11 Transconductor Stage

Switches:

As we say before that compression depend on node X as signal may compress here and we need to have gain of signal to this point is $g_m \cdot R$, R is the resistance at this node which is dominant by resistance seen into switch sources (switches in saturation) this $R \sim (1/g_m \text{ switch})$. So to have no compression at this point we need to increase signal gain to this point by increase g_m by increase current but trades off with CS_Switch thermal noise. We start to design each pair of switches to carry total current = 200mA. Then simulation of linearity and noise must be done and as explained before we see tradeoff between them so optimum design must be achieved

Switches must be designed to have small V_{ov} to increase slope of switching (decrease gradual transition time) which affect the gain and noise as we say before. But we have to design it to be able to pass all corners. Length of switches are chosen to be maximum to decrease flicker noise (which can have effect of increase indirect flicker noise if switches capacitance is dominant). V_{ov} value is determined by switches dimensions (W/L) and current pass through and after simulation across corners. We reach to V_{ov} as good value = 90mV

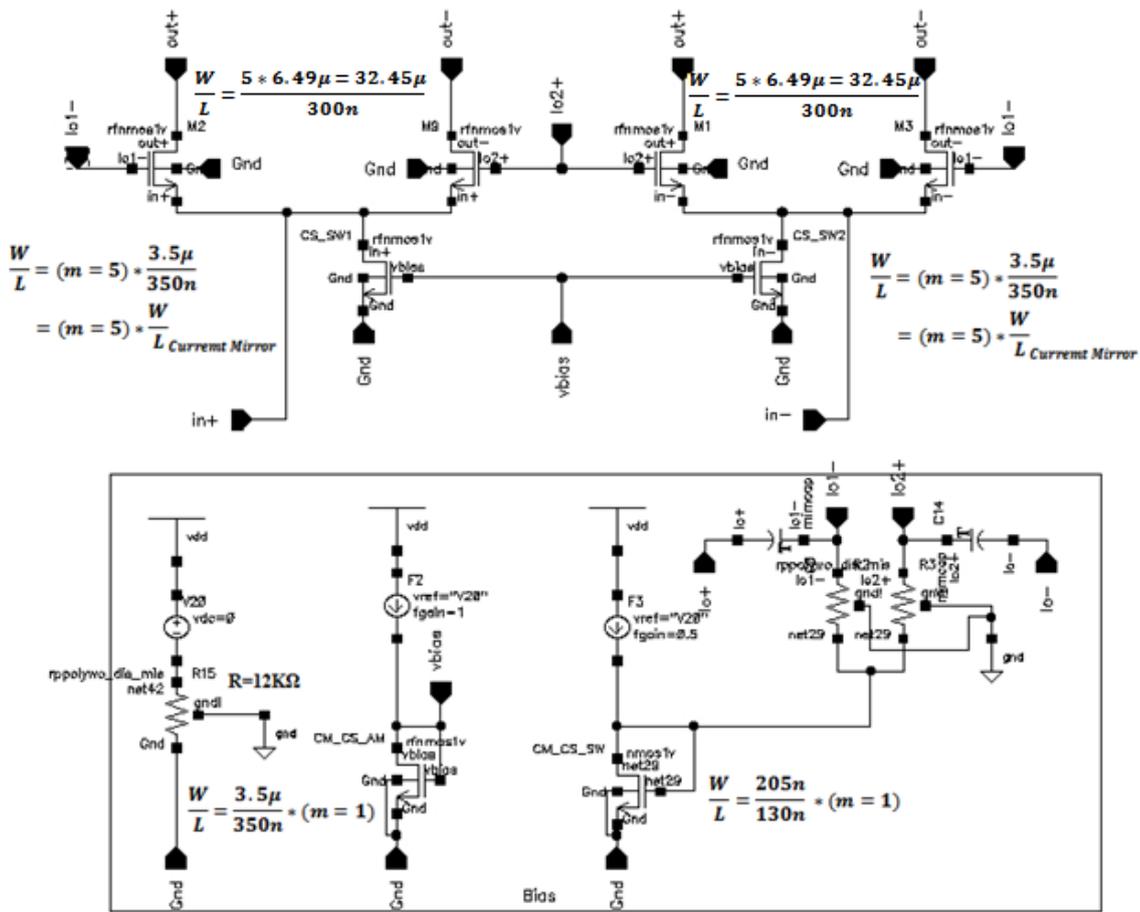


Figure 3.12 Switch Part

Load:

Load is chosen to be R (rppoly) which is free flicker noise and no need for more CMFB circuit. This value is determined by the needed gain which is trades off with linearity (as R value with current flow in determine CM output). As good value we need CM output to be around 750 mV to be able to have good swing. C_L is taken to be over specification about 5pF.

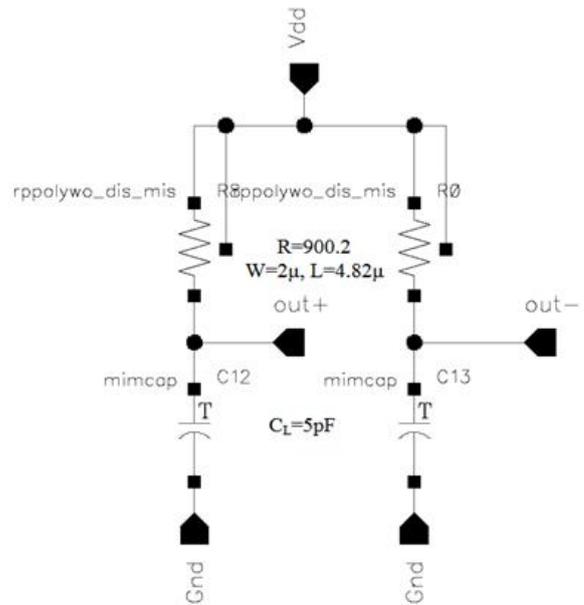


Figure 3.13 Load

3.7 Analysis Explanation:

DC-biasing:

Like any DC-bias analysis used to know DC-operating point of circuit.

Gain measurement:

As we want to measure the gain of mixer we can introduce small RF signal and measure output so we have gain. But in mixer there is another signal drive the Mixer which LO signal, this signal is large signal and we can't consider it as small signal because as we know that this signal changes the circuit characteristics periodically by switching it on and off. This thing leads us to an important thing that circuit must be solved at this signal then we apply small-signal. Like in a simple amplifier we first solve it about dc-operating point then we apply the small signal to circuit, we need here to solve circuit about certain periodic signal which can be achieved using PSS analysis. Then we apply small-signal using PAC analysis (PSS analysis must be run first before PAC analysis).

So to measure gain in our Mixer we run the Mixer with PSS analysis using LO frequency=314.86MHz as large signal, then apply PAC signal with swept-frequency signal from 915MHz to 920MHz, where small-signal takes a value of one volt. Then we measure output which is the gain itself.

Also gain can be measured using PXF analysis instead of PAC analysis.

Noise figure Measurement

The same thing like gain measurement we need to solve circuit @ LO signal for reasons we say before also this signal has an important effect on noises especially noise folding. After running PSS analysis we need to run noise analysis to measure the Noise figure.

We run the Mixer with PSS analysis using LO frequency=314.86MHz as large signal, then apply noise analysis with swept-frequency from 1KHz to 5MHz.

IIP3 Measurement:

IIP3 test depends on assigning two blockers onto Mixer and measuring output intermodulation then hence we can IIP3 of Mixer.

As we know that Mixer is driven by default by LO frequency, and we have solved steady state solution of circuit about this LO signal. Since two-tone test is made by driving circuit by two-tone RF signals (f_1 and f_2), where $f_1 - f_2 = \Delta$ ($f_1 < f_2$) and $f_{\text{desired}} - f_1 = \Delta$, so we have third intermodulation fall @ $2f_1 - f_2 = f_{\text{desired}}$. Where one is taken as small signal (f_2) and other (f_1) is large signal (which will introduce non-linearity to circuit). This approach is good as it saves time and speeds up simulation and improves accuracy also we have no restriction on frequency of small tone as no need to be commensurate with other frequencies.

This is done using qpss analysis to solve circuit about LO signal and tone (f_1). Then we enter the second small signal tone (f_2) in qpac analysis. What happens is that f_1 modulates the characteristics of the circuit, which causes any signal at f_2 to mix down with $2f_1$ to $2f_1 - f_2$ (intermodulation product).

As we know we can measure IIP3 without sweep input tone. We can only take one point and measure difference ΔP to get IIP3 but that constraint that we must take a point far away below compression point to be inside the asymptotic range for both the fundamentals and the intermodulation terms and to get the right IIP3

From system design

$f_1 = f_{\text{rf}} + 2 \text{ MHz} = 915 \text{ MHz} + 2 \text{ MHz} = 917 \text{ MHz}$ and $f_2 = f_{\text{rf}} + 4 \text{ MHz} = 915 \text{ MHz} + 4 \text{ MHz} = 919 \text{ MHz}$.

Port to Port Isolation

Isolation analysis can be simulated using pss + pac for RF-IF and RF-LO but to measure LO-RF isolation we need another analysis which is PXF analysis which gives the capability to measure Transfer functions from any point in circuit to a certain point specified in the PXF analysis (opposite to PAC analysis which measures Transfer function from certain point specified in PAC analysis to any point in circuit). So we are capable of measuring other port to port analysis like LO-IF and LO-RF and IF-RF ...

Analysis Summary:

Analysis	Parameter measured	Condition
DC	DC operating point	-
Transient	Time domain analysis	-
PSS	Transient Analysis	-
PAC	Gain ,Port Isolation	PSS needed
PXF	Gain check ,Port Isolation	PSS needed
PNOISE	Noise	PSS needed
QPSS	Compression point	-
QPAC	IIP3	QPSS needed

3.8 Simulations and Results:

Test-Bench

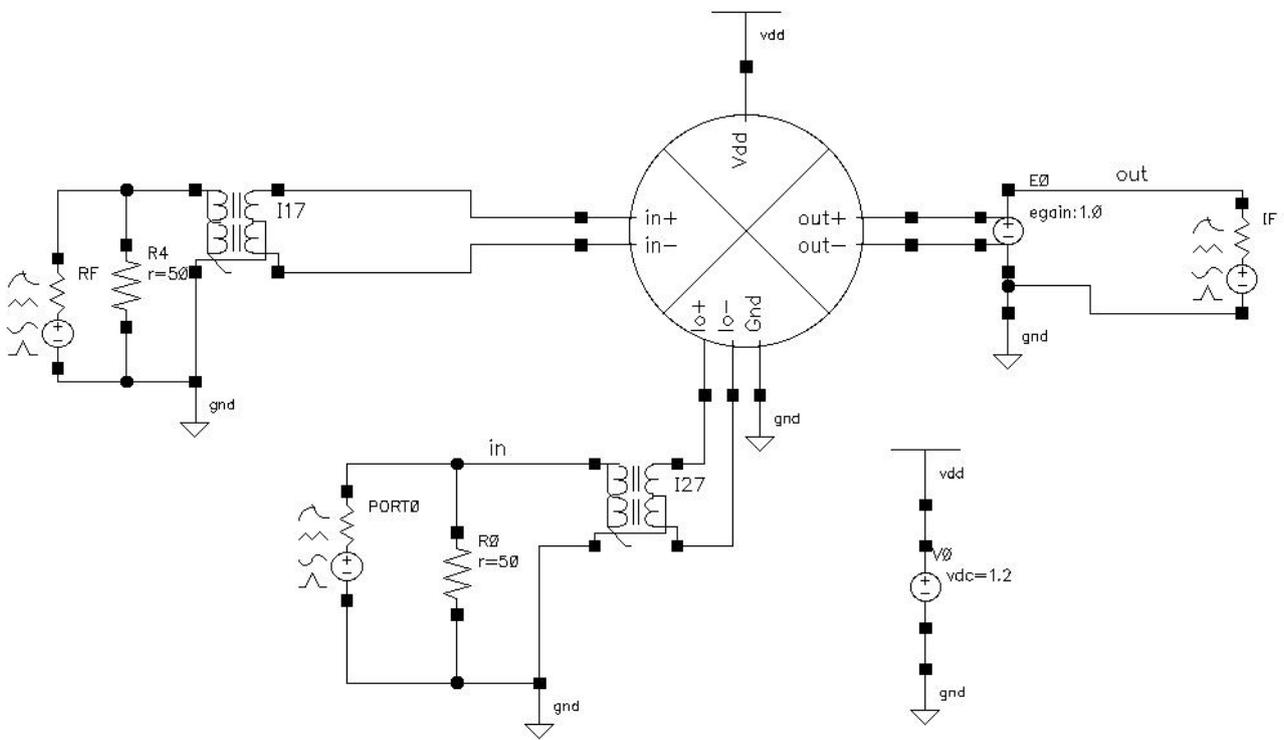


Figure 3.14 Test_Bench

Switches

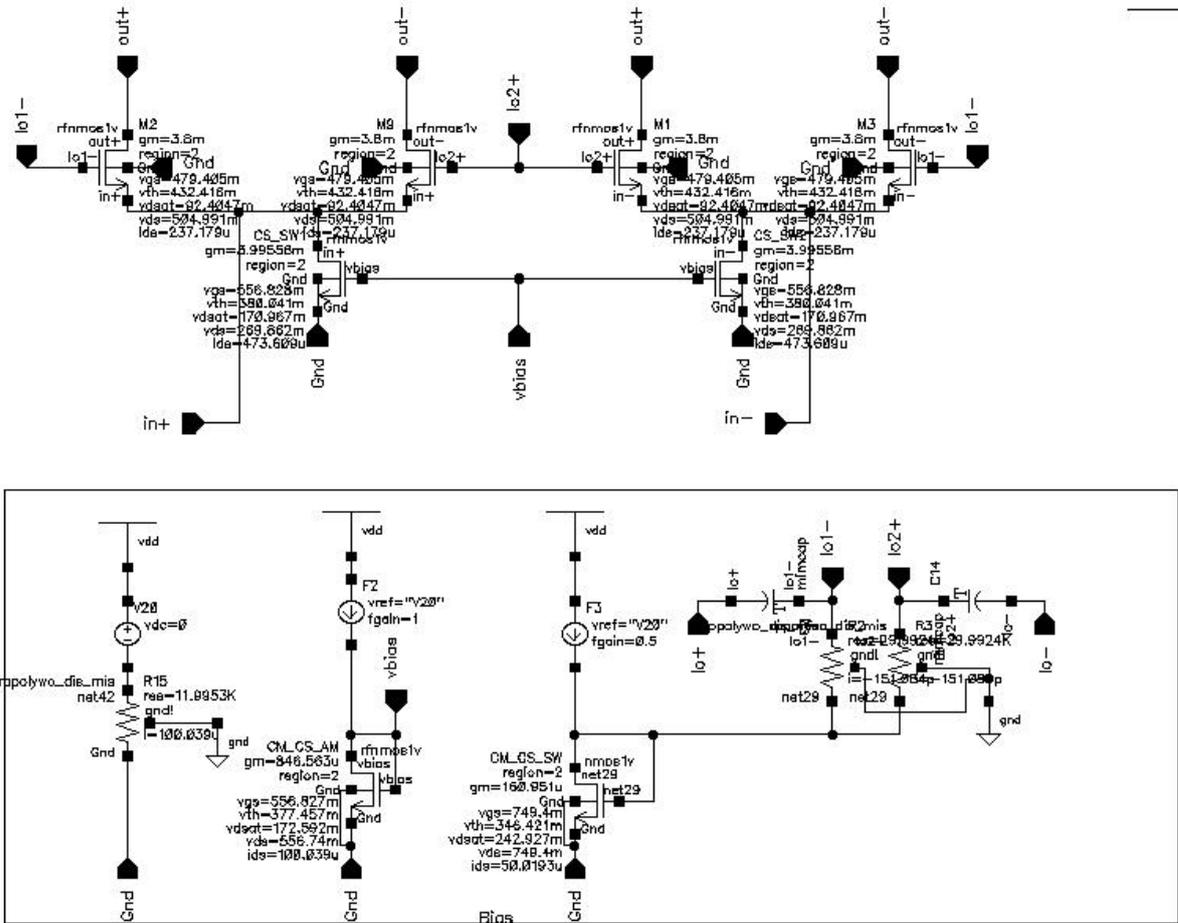


Figure 3.17 switches DC biasing

Load:

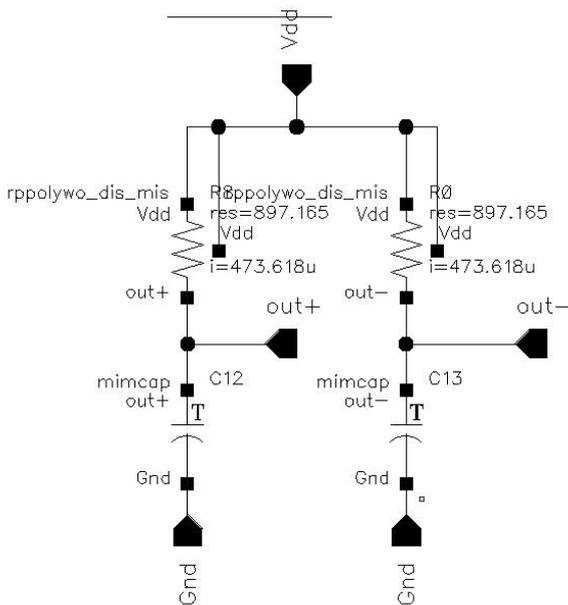


Figure 3.18 Load Bias

Typical:

Gain:

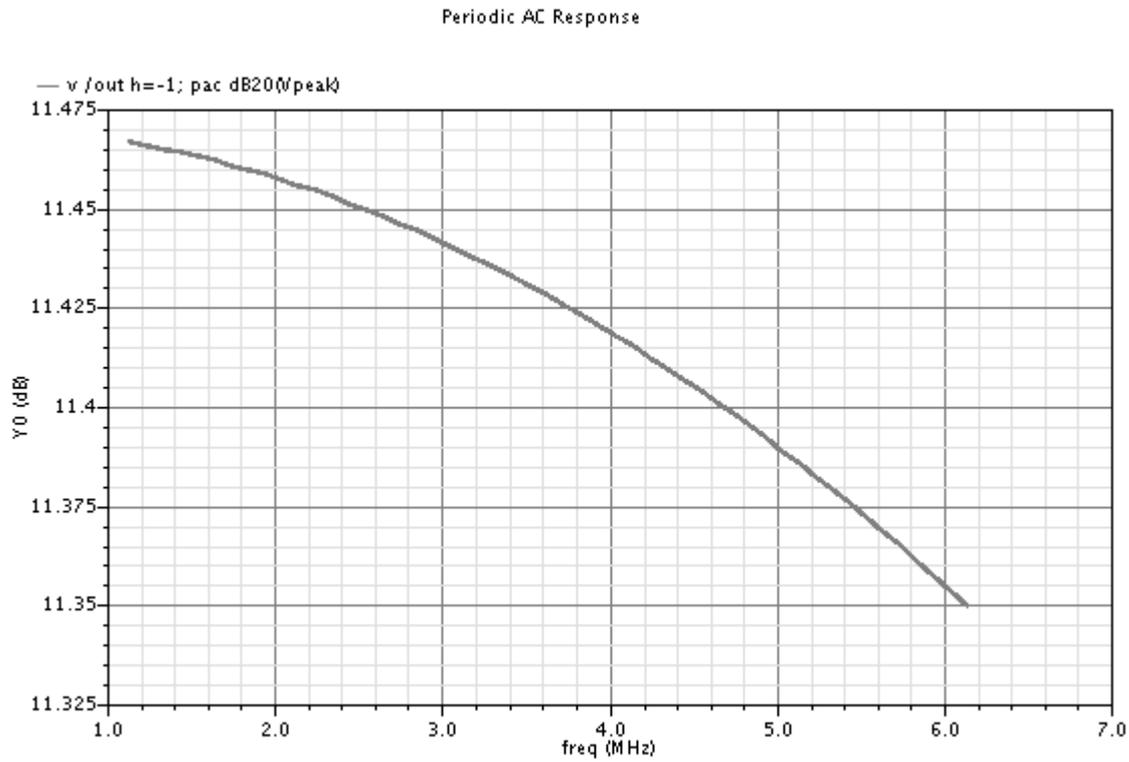


Figure 3.19 Typical Gain

Gain was found to be about **11.46dB**

Noise Figure

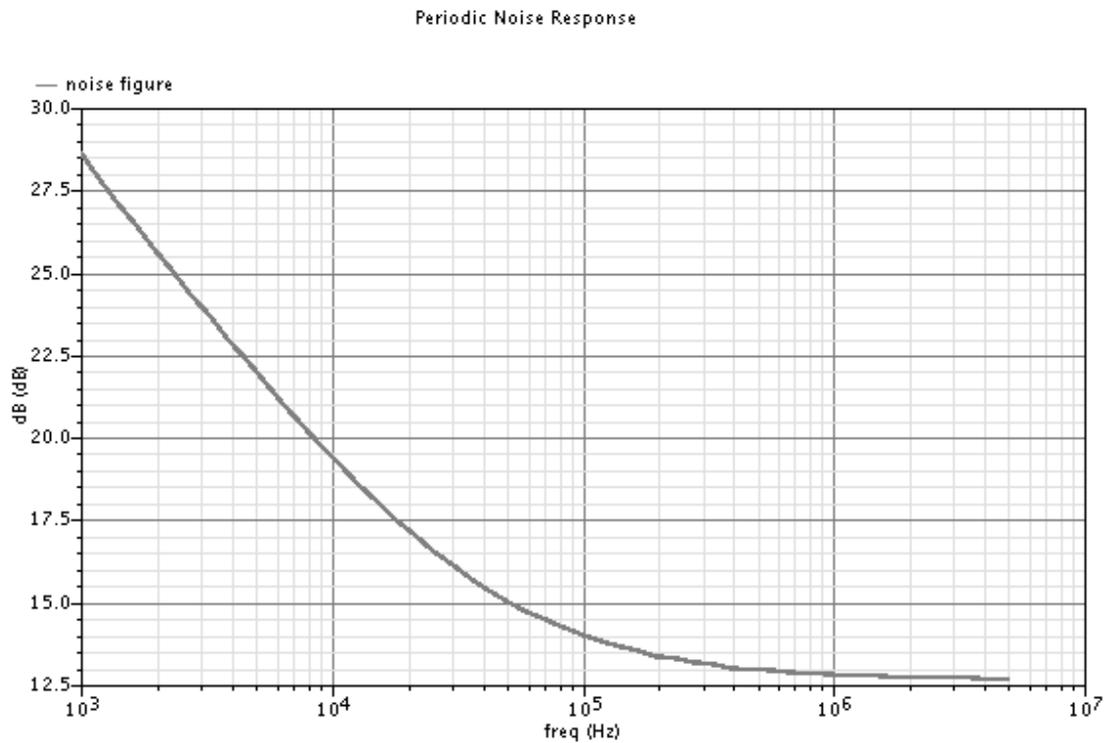


Figure 3.20 Typical Noise Figure

Noise Figure (SSB) @ desired frequency 1.14 MHz to be about 12.55dB

Compression point

Quasi-Periodic Steady State Response

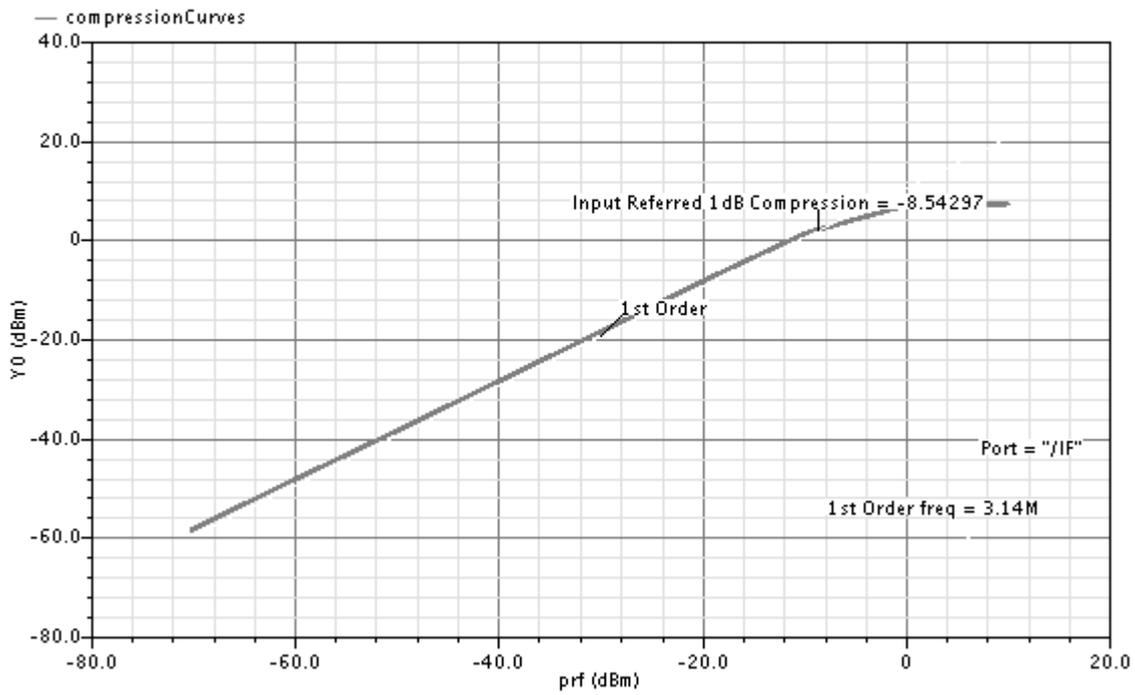


Figure 3.21 Typical Compression point

Compression point = -8.54dBm.

IIP3

Quasi-Periodic AC Response

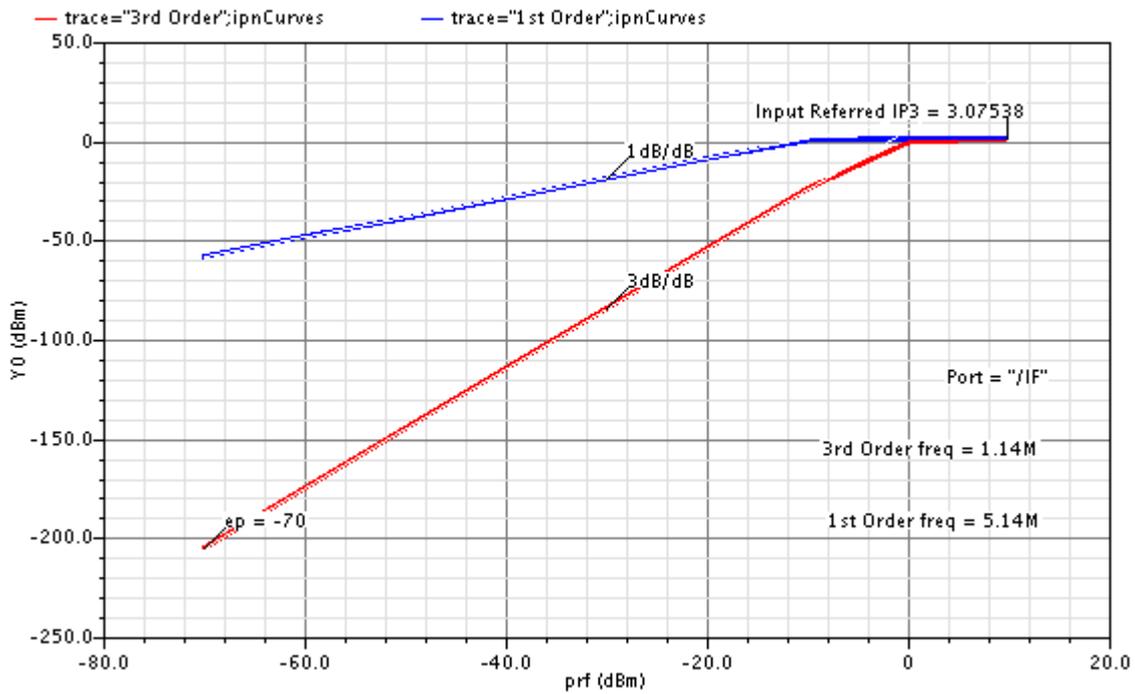


Figure 3.22 IIP3

IIP3 = 3.07

Harmonic distortion

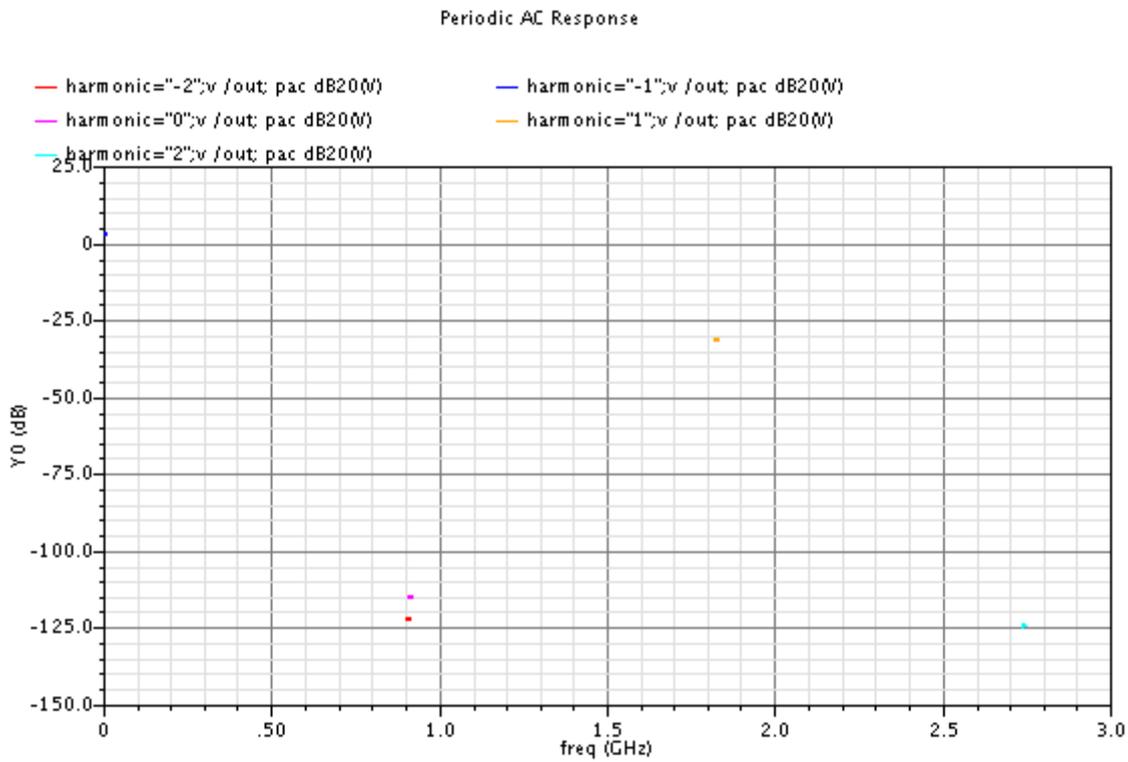


Figure 3.23 Harmonics values

Isolation
RF-IF isolation

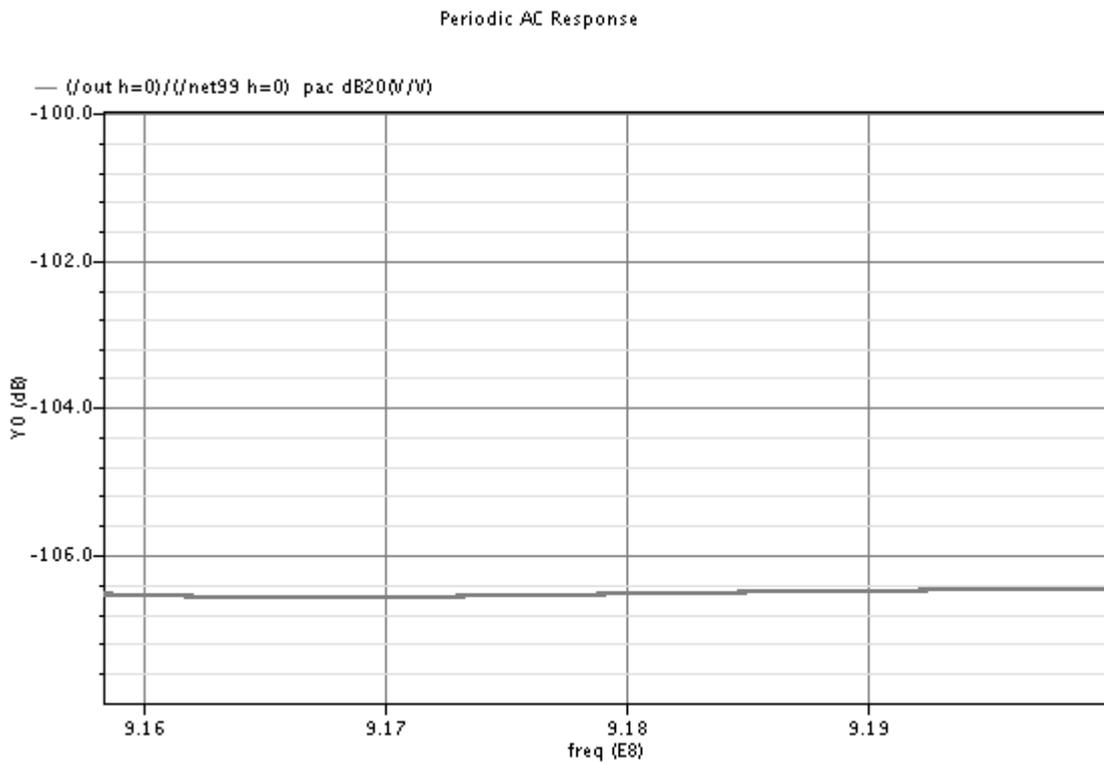


Figure 3.24 RF-IF isolation

RF-LO isolation

Periodic AC Response

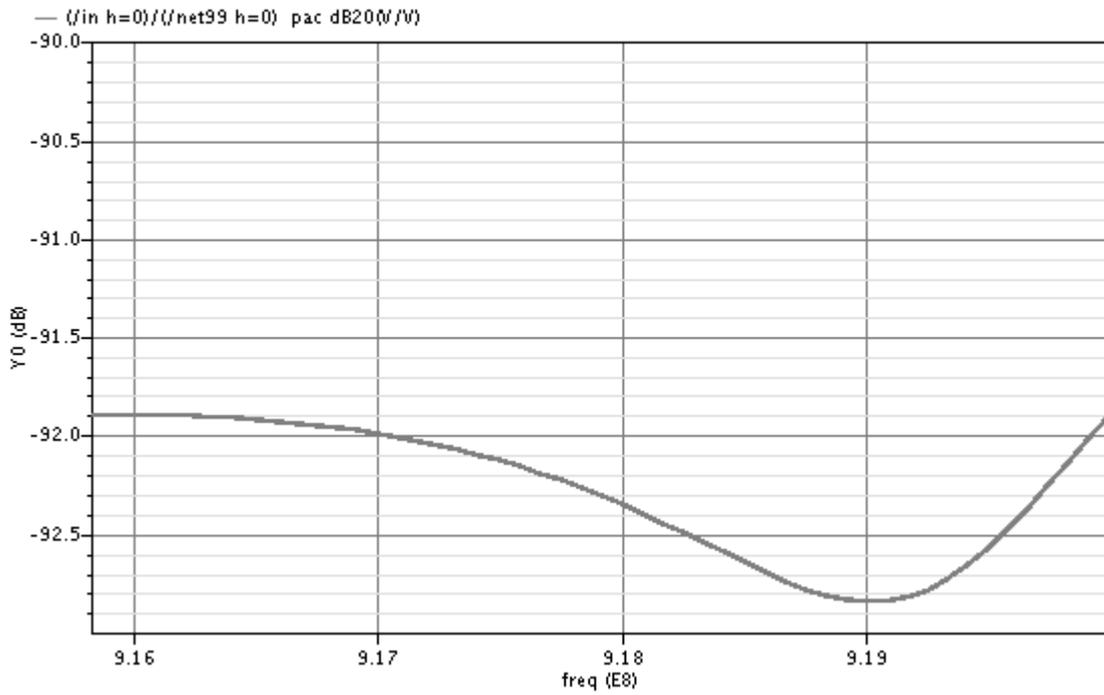


Figure 3.25 RF-LO Isolation

LO-IF isolation

Periodic XF Response

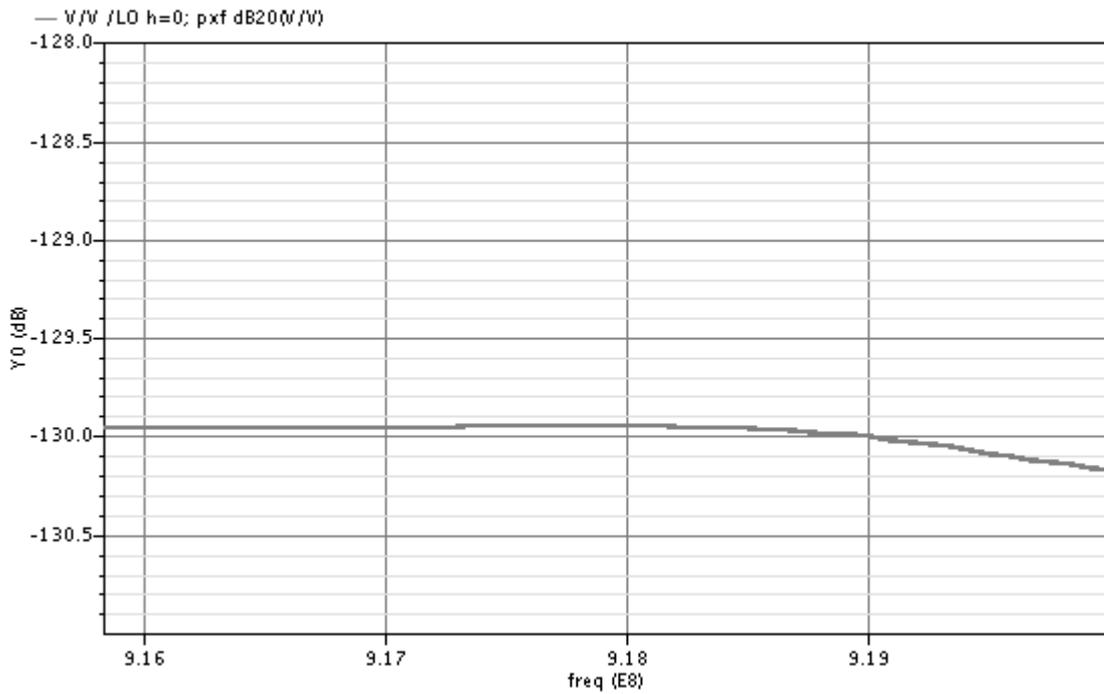


Figure 3.26 LO-IF isolation

LO-RF isolation

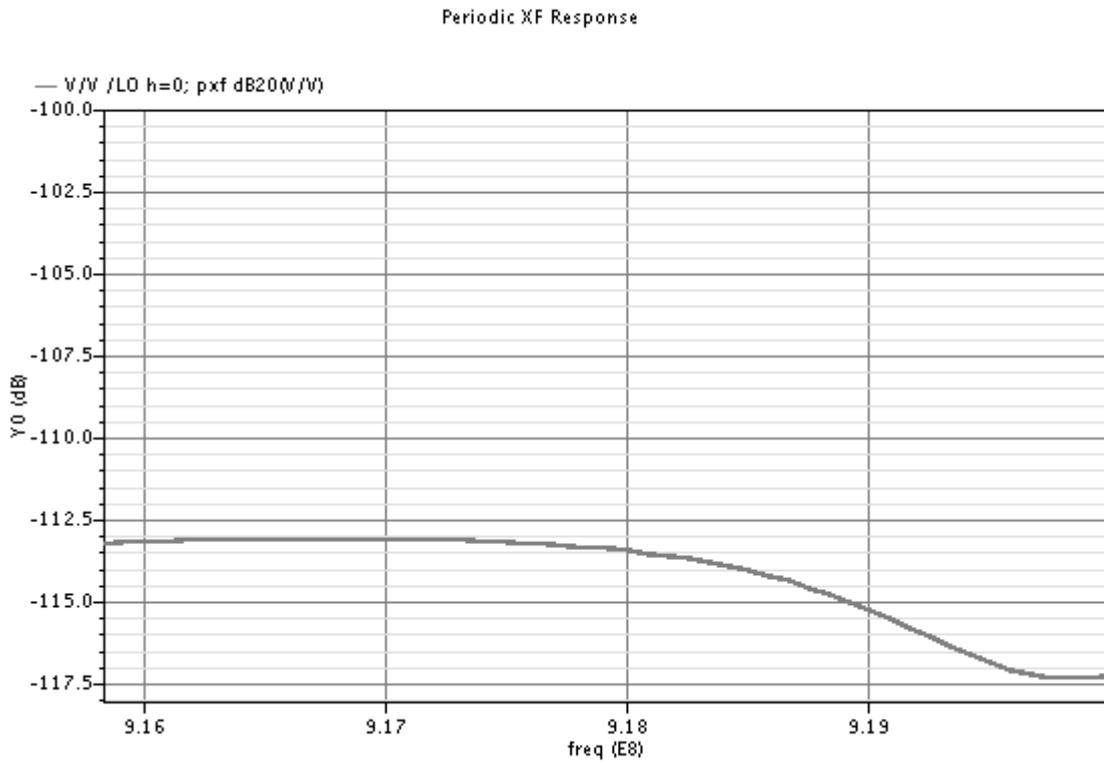


Figure 3.27 LO-RF isolation

IF-RF isolation

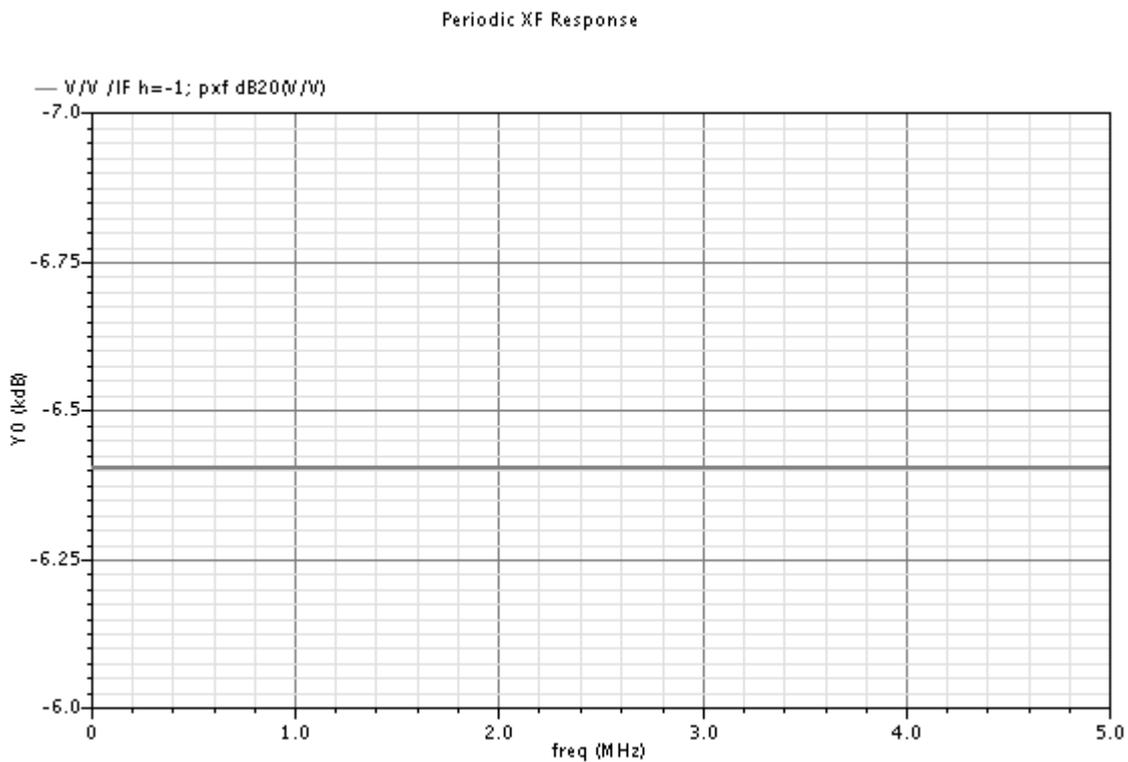


Figure 3.28 IF-RF isolation

IF-LO isolation

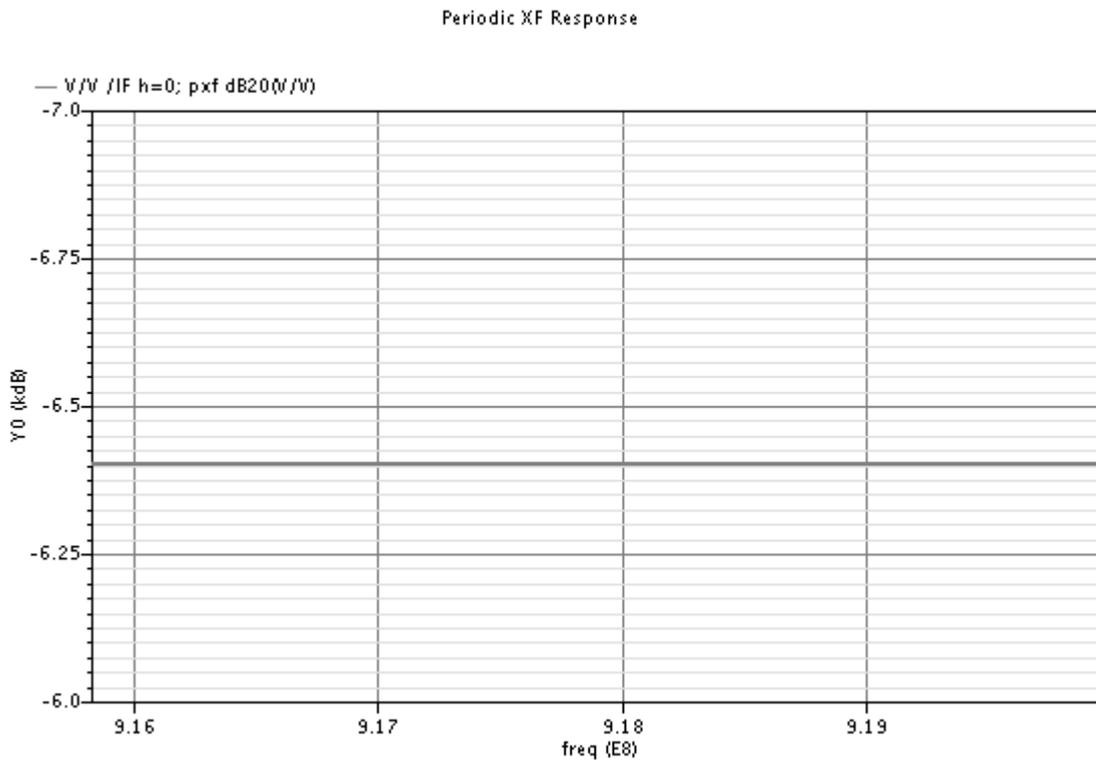


Figure 3.29 IF-LO isolation

Corners:

Gain, NF and IIP3 tests are simulated using ocean script language, code can be found in the appendix. Results are shown here:

Gain

After using modified constant gm-cell gain variation has decreased from 6.5dB to 2.9dB

- Maximum value=**12.3** @**corner** Mosfet "FF" Capacitance "SS" ,Resistance "SS" and Temperature 0° Supply 1.3
- Minimum value=**9.4** @**corner** Mosfet "SS" Capacitance "FF" ,Resistance "FF" and Temperature 85° Supply 1.1

Gain across corners is shown below

In appendix you will find a table contains all corners gain

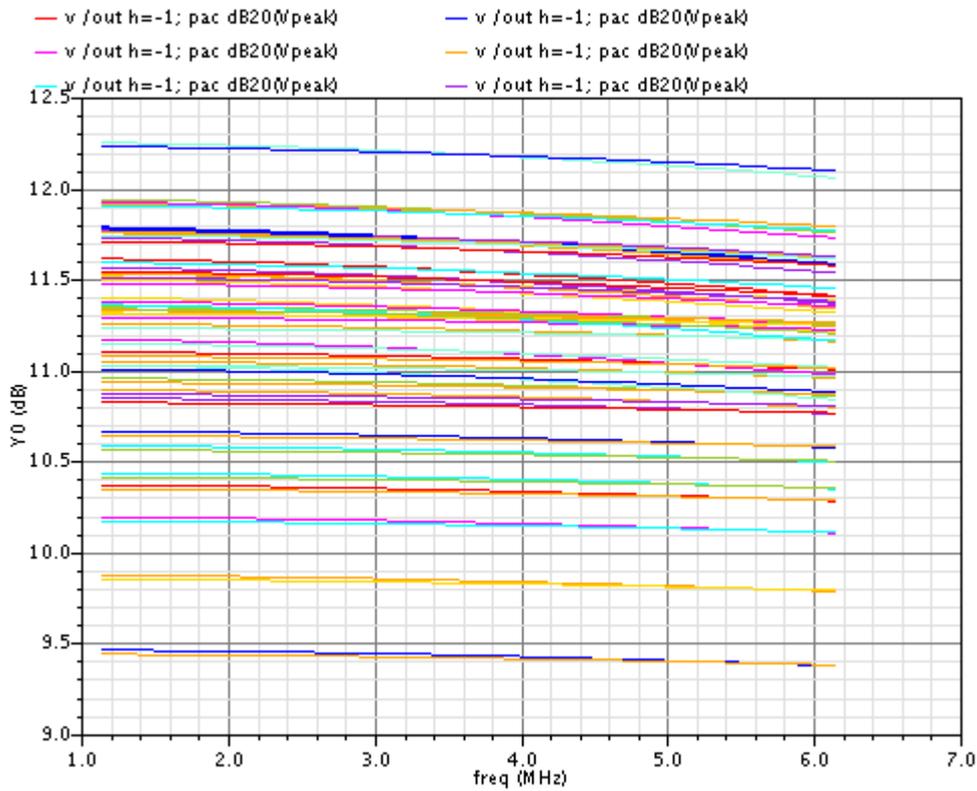


Figure 3.30 Corners gain

Noise Figure:

Also after using modified constant gm-cell Nf variation has decreased from to about 3dB as gain variation decreased

- Maximum value=**14.3 @corner** Mosfet “SS” Capacitance “FF” ,Resistance “SS” and Temperature 85° Supply 1.1
- Minimum value=**11.8 @corner** Mosfet “FF” Capacitance “SS” ,Resistance “FF” and Temperature 0° Supply 1.1

Noise Figure across corners is shown below

In appendix you will find a table contains all corners Noise Figure

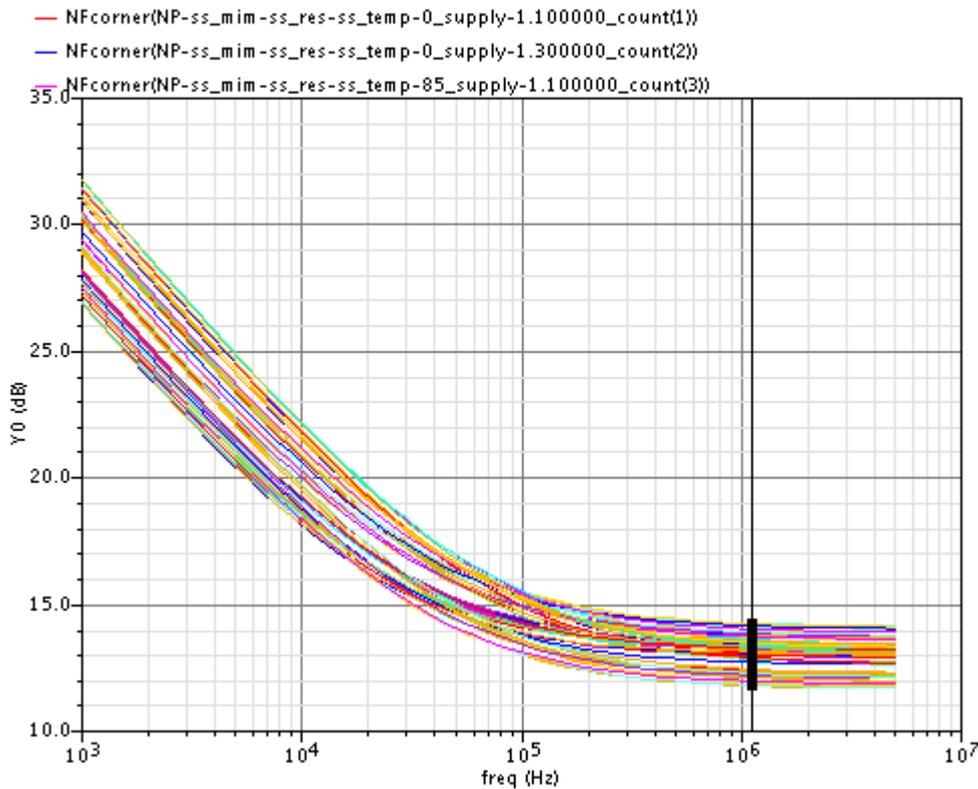


Figure 3.31 Noise Figure across corners

IIP3:

IIP3 was simulated across corners and was found to be having

- Maximum value=**5.2 @corner** Mosfet “FS” Capacitance “FF”, Resistance “FF” and Temperature 85° Supply 1.3
- Minimum value=**0.7 @corner** Mosfet “FF” Capacitance “SS” ,Resistance “SS” and Temperature 0° Supply 1.1

IIP3 plot across corners are not shown as IIP3 has variation the figure not appear clear so we are sufficient in showing the table across corners in Appendix

Summary:

In this table, summary of specifications are shown where each value written in certain corner. The corner condition written in order Mosfet, Capacitance, Resistance, Temperature and Supply

	Min	Typical	Max	Spec. required	Unit
Gain	9.4	11.46	12.3	12	dB
Condition	FF,SS,SS, 0,1.3	Typical	SS,FF,FF 85,1.1		
NF (SSB)	11.8	12.55	14.3	15(SSB)	dB
Condition	FF,SS,FF 0,1.1	Typical	SS,FF,SS 85,1.1		
IIP3	0.7	3.07	5.2	2.5	dBm
Condition	FF,SS,SS 0,1.1	Typical	FS,FF,FF 85,1.3		

Gain Sheet

Counter	NMOS	PMOS	Mimcap	Resistance	Temp	Vdd	gain
1	ss		ss	ss	0	1.1	11.6
2	ss		ss	ss	0	1.3	11.8
3	ss		ss	ss	85	1.1	11.2
4	ss		ss	ss	85	1.3	11.5
5	ss		ff	ss	0	1.1	11.6
6	ss		ff	ss	0	1.3	11.8
7	ss		ff	ss	85	1.1	11.1
8	ss		ff	ss	85	1.3	11.5
9	ss		ss	ff	0	1.1	11
10	ss		ss	ff	0	1.3	11.1
11	ss		ss	ff	85	1.1	9.5
12	ss		ss	ff	85	1.3	10.2
13	ss		ff	ff	0	1.1	10.9
14	ss		ff	ff	0	1.3	11.1
15	ss		ff	ff	85	1.1	9.4
16	ss		ff	ff	85	1.3	10.2
17	ff		ss	ss	0	1.1	11.6
18	ff		ss	ss	0	1.3	12.3
19	ff		ss	ss	85	1.1	11.4
20	ff		ss	ss	85	1.3	11.9
21	ff		ff	ss	0	1.1	11.5
22	ff		ff	ss	0	1.3	12.2
23	ff		ff	ss	85	1.1	11.4
24	ff		ff	ss	85	1.3	11.9
25	ff		ss	ff	0	1.1	11.3
26	ff		ss	ff	0	1.3	11.3
27	ff		ss	ff	85	1.1	10.6
28	ff		ss	ff	85	1.3	10.8
29	ff		ff	ff	0	1.1	11.2
30	ff		ff	ff	0	1.3	11.3
31	ff		ff	ff	85	1.1	10.6
32	ff		ff	ff	85	1.3	10.8
33	sf		ss	ss	0	1.1	11.8
34	sf		ss	ss	0	1.3	11.9
35	sf		ss	ss	85	1.1	11.5
36	sf		ss	ss	85	1.3	11.8
37	sf		ff	ss	0	1.1	11.8
38	sf		ff	ss	0	1.3	11.9
39	sf		ff	ss	85	1.1	11.5
40	sf		ff	ss	85	1.3	11.7
41	sf		ss	ff	0	1.1	11.3
42	sf		ss	ff	0	1.3	11.4
43	sf		ss	ff	85	1.1	10.4
44	sf		ss	ff	85	1.3	10.7
45	sf		ff	ff	0	1.1	11.3
46	sf		ff	ff	0	1.3	11.3
47	sf		ff	ff	85	1.1	10.3
48	sf		ff	ff	85	1.3	10.6
49	fs		ss	ss	0	1.1	11.4
50	fs		ss	ss	0	1.3	11.7

51 fs	ss	ss	85	1.1	11
52 fs	ss	ss	85	1.3	11.5
53 fs	ff	ss	0	1.1	11.3
54 fs	ff	ss	0	1.3	11.7
55 fs	ff	ss	85	1.1	11
56 fs	ff	ss	85	1.3	11.5
57 fs	ss	ff	0	1.1	10.9
58 fs	ss	ff	0	1.3	11
59 fs	ss	ff	85	1.1	9.9
60 fs	ss	ff	85	1.3	10.4
61 fs	ff	ff	0	1.1	10.9
62 fs	ff	ff	0	1.3	11
63 fs	ff	ff	85	1.1	9.9
64 fs	ff	ff	85	1.3	10.4

Noise Figure Sheet

Counter	NMOS	PMOS	Mimcap	Resistance	Temp	Vdd	NF
1	ss		ss	ss	0	1.1	13.4
2	ss		ss	ss	0	1.3	13.5
3	ss		ss	ss	85	1.1	14.2
4	ss		ss	ss	85	1.3	14.2
5	ss		ff	ss	0	1.1	13.4
6	ss		ff	ss	0	1.3	13.5
7	ss		ff	ss	85	1.1	14.3
8	ss		ff	ss	85	1.3	14.2
9	ss		ss	ff	0	1.1	12.3
10	ss		ss	ff	0	1.3	12.4
11	ss		ss	ff	85	1.1	13.8
12	ss		ss	ff	85	1.3	13.6
13	ss		ff	ff	0	1.1	12.3
14	ss		ff	ff	0	1.3	12.4
15	ss		ff	ff	85	1.1	13.8
16	ss		ff	ff	85	1.3	13.7
17	ff		ss	ss	0	1.1	13
18	ff		ss	ss	0	1.3	12.7
19	ff		ss	ss	85	1.1	13.7
20	ff		ss	ss	85	1.3	13.6
21	ff		ff	ss	0	1.1	13
22	ff		ff	ss	0	1.3	12.7
23	ff		ff	ss	85	1.1	13.7
24	ff		ff	ss	85	1.3	13.6
25	ff		ss	ff	0	1.1	11.8
26	ff		ss	ff	0	1.3	11.9
27	ff		ss	ff	85	1.1	12.9
28	ff		ss	ff	85	1.3	12.9
29	ff		ff	ff	0	1.1	11.8
30	ff		ff	ff	0	1.3	12

31	ff	ff	ff	85	1.1	12.9
32	ff	ff	ff	85	1.3	12.9
33	sf	ss	ss	0	1.1	13.1
34	sf	ss	ss	0	1.3	13.2
35	sf	ss	ss	85	1.1	13.8
36	sf	ss	ss	85	1.3	13.9
37	sf	ff	ss	0	1.1	13.1
38	sf	ff	ss	0	1.3	13.2
39	sf	ff	ss	85	1.1	13.9
40	sf	ff	ss	85	1.3	13.9
41	sf	ss	ff	0	1.1	12
42	sf	ss	ff	0	1.3	12.1
43	sf	ss	ff	85	1.1	13.2
44	sf	ss	ff	85	1.3	13.2
45	sf	ff	ff	0	1.1	12
46	sf	ff	ff	0	1.3	12.1
47	sf	ff	ff	85	1.1	13.2
48	sf	ff	ff	85	1.3	13.2
49	fs	ss	ss	0	1.1	13.3
50	fs	ss	ss	0	1.3	13.3
51	fs	ss	ss	85	1.1	14.1
52	fs	ss	ss	85	1.3	14
53	fs	ff	ss	0	1.1	13.3
54	fs	ff	ss	0	1.3	13.3
55	fs	ff	ss	85	1.1	14.2
56	fs	ff	ss	85	1.3	14.1
57	fs	ss	ff	0	1.1	12.2
58	fs	ss	ff	0	1.3	12.3
59	fs	ss	ff	85	1.1	13.5
60	fs	ss	ff	85	1.3	13.3
61	fs	ff	ff	0	1.1	12.2
62	fs	ff	ff	0	1.3	12.3
63	fs	ff	ff	85	1.1	13.5
64	fs	ff	ff	85	1.3	13.4

IIP3 sheet

Counter	NMOS	PMOS	Mimcap	Resistance	Temp	Vdd	IIP3
1	ss	ss	ss	ss	0	1.1	1.1
2	ss	ss	ss	ss	0	1.3	1.8
3	ss	ss	ss	ss	85	1.1	2
4	ss	ss	ss	ss	85	1.3	3.3
5	ss	ff	ss	ss	0	1.1	1.4
6	ss	ff	ss	ss	0	1.3	2.1
7	ss	ff	ss	ss	85	1.1	2.2
8	ss	ff	ss	ss	85	1.3	3.5
9	ss	ss	ff	ss	0	1.1	3.2
10	ss	ss	ff	ss	0	1.3	4.6

11	ss	ss	ff	85	1.1	1.2
12	ss	ss	ff	85	1.3	3.9
13	ss	ff	ff	0	1.1	3.3
14	ss	ff	ff	0	1.3	4.8
15	ss	ff	ff	85	1.1	1.2
16	ss	ff	ff	85	1.3	4
17	ff	ss	ss	0	1.1	0.7
18	ff	ss	ss	0	1.3	1.4
19	ff	ss	ss	85	1.1	2
20	ff	ss	ss	85	1.3	2.9
21	ff	ff	ss	0	1.1	0.9
22	ff	ff	ss	0	1.3	1.6
23	ff	ff	ss	85	1.1	2.2
24	ff	ff	ss	85	1.3	3
25	ff	ss	ff	0	1.1	3.1
26	ff	ss	ff	0	1.3	4
27	ff	ss	ff	85	1.1	3.8
28	ff	ss	ff	85	1.3	4.9
29	ff	ff	ff	0	1.1	3.2
30	ff	ff	ff	0	1.3	4.1
31	ff	ff	ff	85	1.1	3.8
32	ff	ff	ff	85	1.3	4.9
33	sf	ss	ss	0	1.1	0.9
34	sf	ss	ss	0	1.3	1.5
35	sf	ss	ss	85	1.1	2.1
36	sf	ss	ss	85	1.3	3.1
37	sf	ff	ss	0	1.1	1.2
38	sf	ff	ss	0	1.3	1.8
39	sf	ff	ss	85	1.1	2.2
40	sf	ff	ss	85	1.3	3.3
41	sf	ss	ff	0	1.1	3.2
42	sf	ss	ff	0	1.3	4.4
43	sf	ss	ff	85	1.1	2.7
44	sf	ss	ff	85	1.3	4.6
45	sf	ff	ff	0	1.1	3.3
46	sf	ff	ff	0	1.3	4.5
47	sf	ff	ff	85	1.1	2.8
48	sf	ff	ff	85	1.3	4.6
49	fs	ss	ss	0	1.1	0.9
50	fs	ss	ss	0	1.3	1.6
51	fs	ss	ss	85	1.1	2.1
52	fs	ss	ss	85	1.3	3.2
53	fs	ff	ss	0	1.1	1.2
54	fs	ff	ss	0	1.3	1.8
55	fs	ff	ss	85	1.1	2.3
56	fs	ff	ss	85	1.3	3.3
57	fs	ss	ff	0	1.1	3.4
58	fs	ss	ff	0	1.3	4.4
59	fs	ss	ff	85	1.1	3.7
60	fs	ss	ff	85	1.3	5.1
61	fs	ff	ff	0	1.1	3.5
62	fs	ff	ff	0	1.3	4.5
63	fs	ff	ff	85	1.1	3.8
64	fs	ff	ff	85	1.3	5.2

References:

1. RF Microelectronics ,Second Edition ,Behzad Razavi
2. Noise in RF-CMOS Mixers: A Simple Physical Model Hooman Darabi and Asad A. Abidi, Fellow, IEEE, IEEE TRANSACTIONS ON SOLID STATE CIRCUITS, VOL. 35, NO. 1, JANUARY 2000
3. A Low-Voltage, Low-Power, and Low-Noise UWB Mixer Using Bulk-Injection and Switched Biasing Techniques ,Myoung-Gyun Kim, Hee-Woo An, Yun-Mo Kang, Ji-Young Lee, and Tae-Yeoul Yun, Member, IEEE , IEEE TRANSACTIONS ON MICROWAVE THEORY AND TECHNIQUES, VOL. 60, NO. 8, AUGUST 2012
4. Improvements in Biasing and Compensation of CMOS Opamps ,SEAN NICOLSON ANDKHOAN PHANG, Analog integrated Circuits and Signal Processing 43.237-245,2005 @2005 Springer Science +Business Media,Inc Manufactured in The Netherlands
5. Process and Temperature Compensation for RF Low-Noise Amplifiers and Mixers ,Didac Gomez ,Milosz Sroka and Jose Lui Gonzalez Jimenez ,Member IEEE,

4 IF Filter

4.1 Introduction

The IF filter is one of the most important blocks in the chain of the RX in any SOC transceiver , it is the block which introduces one of the main difficulties in the RX chain design , and represents the backbone of the performance of the all receiver as it is usually in the middle of the chain , that is , it is required from it BOTH good IIP3 performance and good NF results as well .

One must admit that the operation of designing such a filter was a very great CHALLENGE , and the amount of the knowledge gained in this part of my life was incredible in reality , and the number of the trade-offs that I met in this block affected me not only in the level of technical skills , but also in the level of methodology I think with in my life .

The selected topology for the filter here is the Switched Capacitor filter topology , one of the somewhat new topologies to be introduced for graduation projects , it gives some great relaxations in specs in many aspects , but in the same time it makes it difficult for the other specs , that is , for example ; the specs of the core OTA involved in the design is highly relaxed in linearity and gain specs , but it must give me the best SR and linear settling time and GBW specs in the same time .

The general requirements and specs for this filter is that , the gain is 8 dB and can Not exceed 10 dB for any corner , and the ACRR is >25 dB with a channel offset of 750 KHz , and the Rejection ratio at offset of 2MHz must exceed 60 dB , and that after 10MHz must exceed 70dB , these rejection ratios all was given by the BLOCKER PROFILE given to the system designer and for the IRR , it was required to be > 25 dB also .

All of these required rejections was exceeded by the filter done , and it is maintained across the corners as well , for the BW requirements , a MATLAB model was made for the modulation scheme involved in our system [GFSK] , and it was found that 300 KHz will contain approx. a 97% of the signal power for the channel passes through the filter , the BW achieved by the filter done was 280 KHz which contains approx. 95% of the signal power , which is satisfactory for us to a far limit (for the MATLAB code , see Appendix A) .

IF center frequency is maintained to be close to 1MHz so as to be far enough from the FLICKER noise , and to use a moderate values of capacitors required to EMULATE the resistors by the means of switched capacitors .

The IIP3 required from me was 10dBm for a blockers at 2MHz and 4MHz offsets from the IF freq , which was a challenge , but the great rejections achieved at these frequencies and the large linearity of the S.C. filters in general made it somewhat easy to achieve.

Eventually , the NF needed from me was 30 dB , it is a normal NF for a block like S.C. filter as it contains a large number of OTAs , and as CLOCKED block , the folded noise will make it more challenge issue for getting such a number .

4.2 Topology survey and selection

There are an approximately endless topologies of the filters , But here we will introduce the most famous ones , and the ones that I get my topology from , and then a brief summary of them will be given for clarifying the advantages and disadvantages of them .

4.2.1 Main topologies and general features of filters

The main topologies to be introduced here are

- ▶ The Active RC and MOS-C filters .
- ▶ The Gm-C filters .
- ▶ The Switched capacitor filters .
- ▶ The Switched Current filters .

The main features of the VLSI on chip filters are

- ▶ Compatible with VLSI process .
- ▶ Immunity to parasitic effects, fabrication tolerances, and environment variations .
- ▶ Large dynamic range .
- ▶ Good power supply rejection .
- ▶ Low power and small chip area .

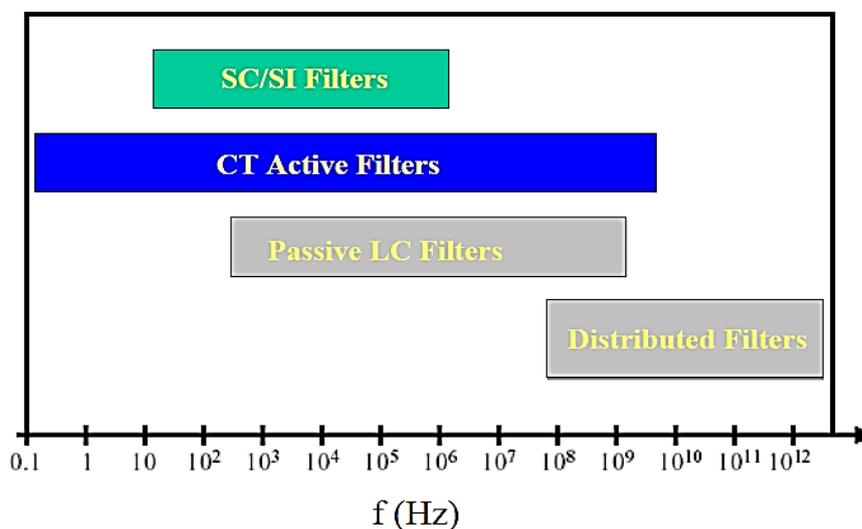


Figure 4.4.1 : operating regions for various filters topologies

For the filters in general ,.... we can classify it by various ways , the most famous models used for the filters are the Butterworth , Chebyshev 1&2 , elliptic , Bessel , they are summarized as follows

- ▶ Butterworth filters .
 - Maximum pass band flatness.

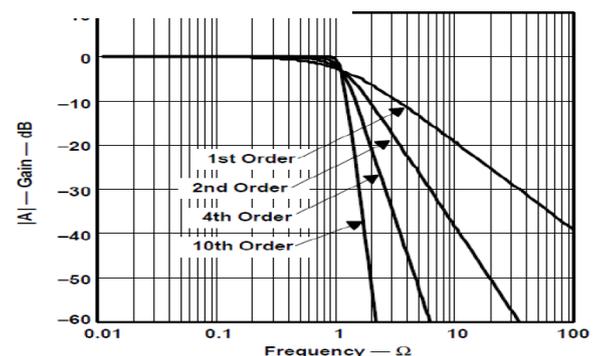


Figure 4.4.2 : the Butterworth responses

- All poles and there's no zeros.
- Number of poles equal to filter order.
- Poles located on the unit circle with equal angles.
- Its main responses is shown clearly in the figure above .

► Chebyshev1 filters .

- Higher gain roll-off above f_c ; Sharper transition band compared to Butterworth (for the same number of poles).
- The higher the pass band ripples, the higher the filter's roll-off.
- Each ripple accounts for one second-order filter stage.
- Filters with even order numbers generate ripples above the 0-dB line.
- Filters with odd order numbers create ripples below 0 dB.
- Poorer group delay compared to Butterworth.
- More ripple in pass band → poorer phase response.

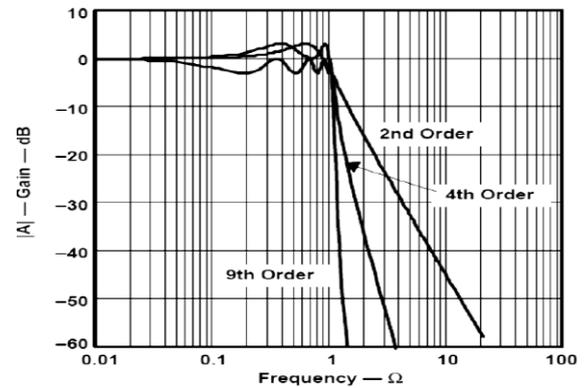


Figure 4.4.3 : the responses of Chybeshev 1 filters

► Chybeshev2 filters .

- This type is less common because it does not roll off as fast as Chebychev I, and requires more components.
- No ripple in pass band.
- Nulls or notches in stop band.
- Sharper transition band compared to Butterworth .
- Pass band phase more linear compared to Chebychev I .

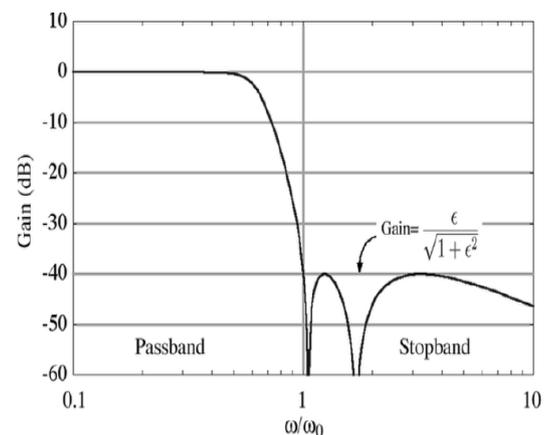


Figure 4.4.4 : the Chebyshev2 responses

► Elliptic filters .

- Ripple in pass band.
- Nulls in the stop band.
- As the ripple in the stop band approaches zero, the filter becomes a Chebychev I.
- As the ripple in the pass band approaches zero, the filter becomes a Chebychev II.
- As both ripple values approach zero, the filter becomes a Butterworth filter.
- Sharper transition band compared to Butterworth & both Chebychevs.
- Poorest phase response.

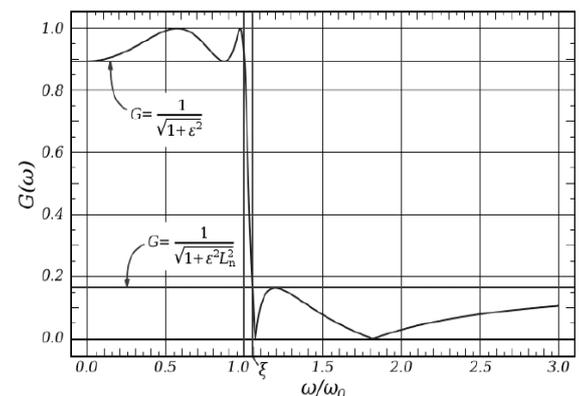


Figure 4.4.5 : the Elliptic responses

And the main parameters that may define any filter are the ripples ratio, roll-off , the corner (cut-off) frequencies ,...etc. , they will be summarized clearly in the following points

► Amplitude Response

It's defined as the ratio of the output amplitude to the input amplitude versus frequency.

► Phase Response

All non-ideal filters introduce a time delay between the filter input and output terminals. This delay can be represented as a phase shift.

► Transition band

It's the narrow band of frequencies between a pass band and stop band.

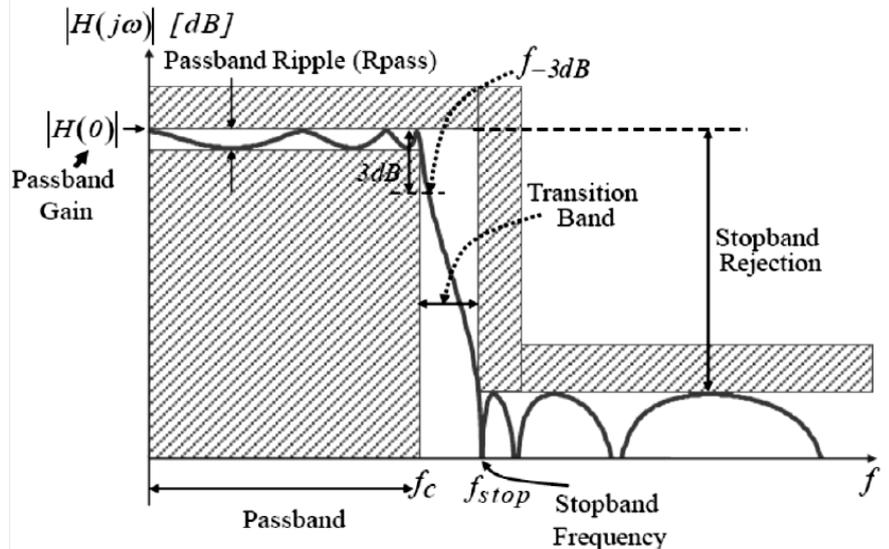


Figure 4.4.6 : this figure summarizes all filters parameters .

► Corner frequency (f_c)

It's the frequency where attenuation reaches -3 dB below the pass band amplitude or exactly when the phase reaches -90° .

► Roll-off

It's the rate at which attenuation increases beyond the cut-off frequency. Higher is better but requires more stages which means more complexity.

This is a table summarizes the filters types and its features

Table 4.3 : comparison between main filters models

Alignment	Pass-Band Description	Stop Band Description	Comments
Butterworth	Monotonic	Monotonic	All-pole; maximally flat
Chebyshev	Rippled	Monotonic	All-pole
Bessel	Monotonic	Monotonic	All-pole; constant phase shift
Inverse Chebyshev	Monotonic	Rippled	
Elliptic (or Cauer)	Rippled	Rippled	

Let us now to mention the main topologies that you can find for filters , and then summarize their features , advantages and disadvantages

4.2.2 The Active-RC and MOS-C filters

The main points that summarize the Active RC filters idea are

- ▶ It is direct mapping of discrete active RC filters to VLSI active RC filters with CMOS compatible R, C, and Op-amp realizations.
- ▶ For MOS-C filters, resistors are mapped to MOS VCRs [voltage controlled resistors].
- ▶ Active-RC filter is superior for the relatively large dynamic range comparing with Gm-C filter.
- ▶ Tuning is usually required due to parameter variation of the VLSI components.[tuning is the most critical part of any filter design except for S.C. filters]

Then , the main idea is to have a VLSI compatible OP-AMPS , Resistors , and capacitors to have such a filter , first we can see here a verity of the available OP-AMPS that can be employed for this type of filters

. . .

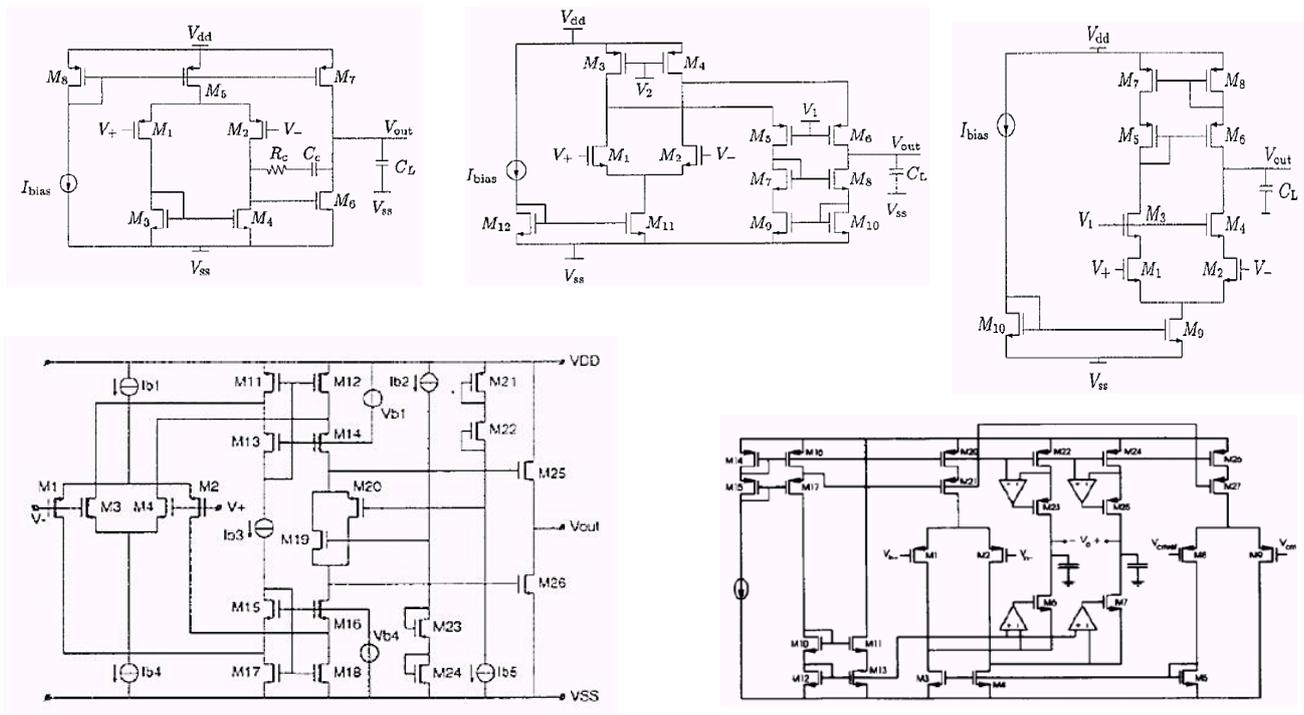


Figure 4.4.7 : Various OP-AMPS that can be used for Active-RC and MOS-C filters

Then , we are in need for a VLSI compatible CAPACITORS to be employed for this filter , there are mainly three types of capacitors that can be used , that is

- ▶ Gate CAPs : easy to fabricate with high density , but it is non linear
- ▶ Junction CAPs : its main drawback is the large nonlinearity .

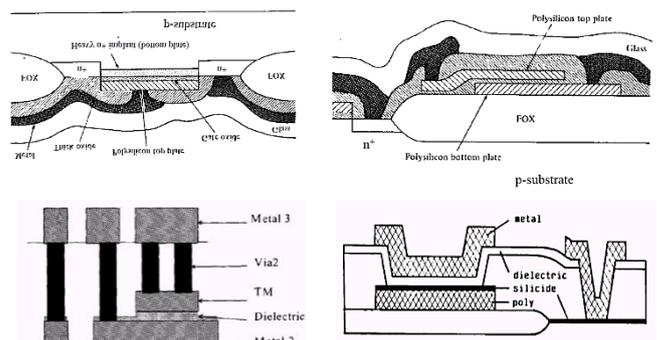


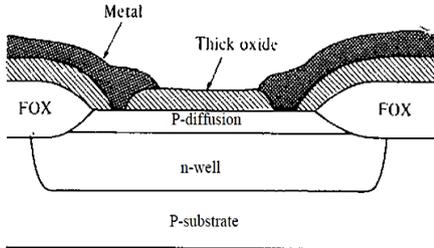
Figure 4.4.8 : the various types of VLSI CAPS

- ▶ MIM or Poly-Poly CAPs : it provides good linearity and high Q in the same time

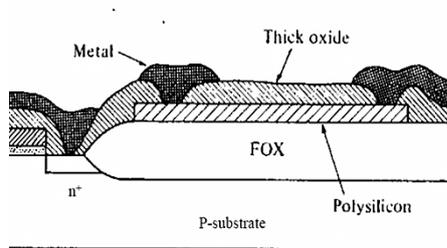
Then , the final thing needed for this filter is to get a VLSI compatible RESISTORS , this can be achieved by using one of the famous three RESs types , that is

- ▶ Diffusion resistors .
- ▶ Poly resistors .
- ▶ Well resistors .

Diffusion Resistor



Poly Resistor



Well Resistor

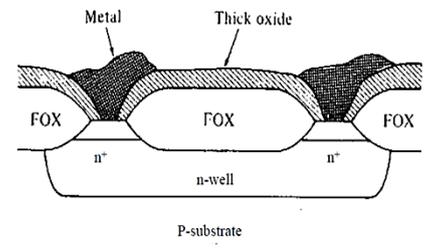
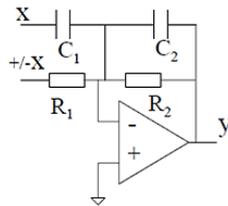


Figure 4.4.9 : the Various VLSI resistors can be used .

Then , after having a VLSI compatible OP-AMPS , resistors and capacitors , one can now build the Active-RC and MOS-C filters easily , you may see the following circuits as an example for this filter and for wide range of filter types and orders

Single-ended

$$H(s) = -\frac{C_1}{C_2} \frac{s \pm \frac{1}{R_1 C_1}}{s + \frac{1}{R_2 C_2}}$$



Fully differential

$$H(s) = \frac{C_1}{C_2} \frac{s \pm \frac{1}{R_1 C_1}}{s + \frac{1}{R_2 C_2}}$$

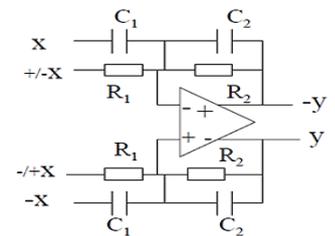
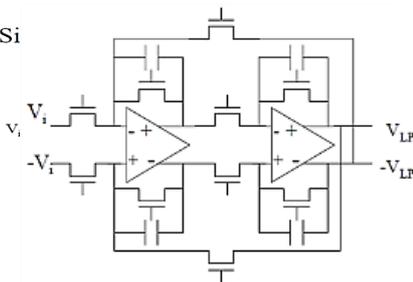
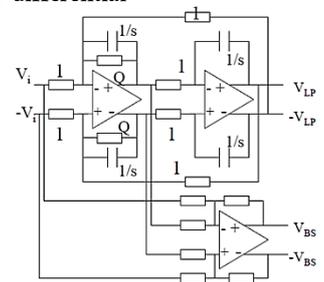


Figure 4.4.10 : the first order Active-RC filters .

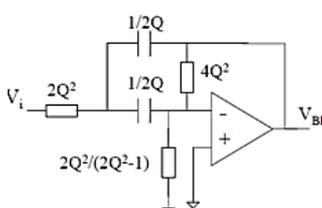
Si



Fully differential

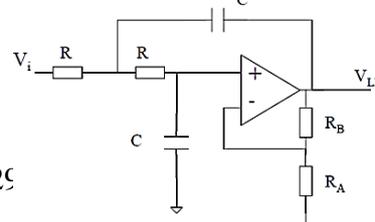


Delyiannis-Friend Circuits



$$H_{BP}(s) = \frac{s}{s^2 + \frac{1}{Q}s + 1}$$

Sallen-Key Circuits(I)



$$H_{LP}(s) = \frac{K}{\left(\frac{s}{\omega_o}\right)^2 + \frac{1}{Q}\left(\frac{s}{\omega_o}\right) + 1}$$

$$\begin{aligned} \omega_o &= 1/RC \\ K &= 1 + R_B/R_A \\ Q &= 1/(3-K) \end{aligned}$$

The main advantages of the Active-RC and MOS-C filters are

- ▶ Since resistors are quite linear, linearity only a function of OPAMP linearity.
- ▶ Good linearity.
- ▶ High dynamic range.

While the main disadvantages are

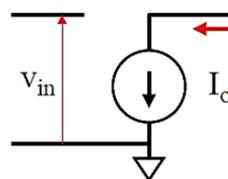
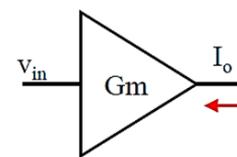
- ▶ OPAMP have to drive resistive load, low output impedance is required.
- ▶ High power consumption.
- ▶ Continuous tuning not possible-tuning only in discrete steps.
- ▶ Tuning requires programmable R's and/or C's.

4.2.3 The Gm-C filters

The idea of the Gm-C filters can be summarized in the following points

- ▶ The Gm here stands for the transconductor , which is involved in the all of the circuits of the Gm-C filters .
- ▶ The main idea here is to EMULATE the resistors , inductors , adders ,and integrators using only the Gm-C cells , then using the SFD (signal flow diagram) ... you will be able to implement any transfer function you need (including filters of course) .
- ▶ The integrator itself can be of course considered as a first order low pass filter .
- ▶ Here , we will introduce the ways to EMULATE these parts and the filter circuits itself .

Here we can see the main definition of the transconductors in the figure here , it is ideal one , of course the real one will have some imperfections .



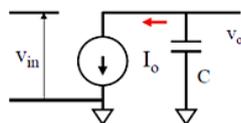
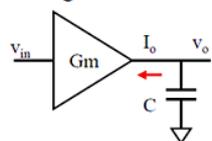
$$i_o = g_m v_{in}$$

Figure 4.4.12 : the transconductors definition

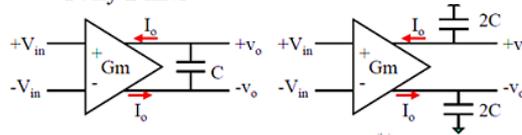
And in the figure below , we can see clearly

the circuits of the integrators , single ended and versions are seen

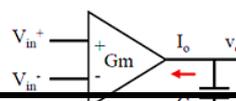
• Single-ended



• Fully Differ



• Differential inputs



$$H(s) = \frac{V_o(s)}{V_{in}(s)} = g_m$$

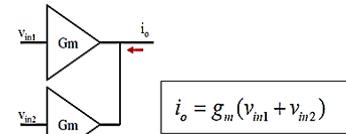
differential ended clearly below . . .

Figure 4.4.13 : the different integrators circuits using Gm-C cells

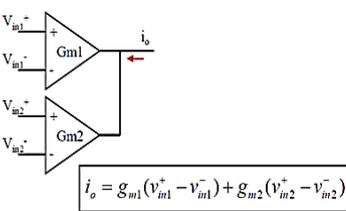
In the figure here below , we will see the various ways to EMULATE resistors , inductors , adders and scalars (amplifiers) using the Gm-C cells only

Adders (circuits)

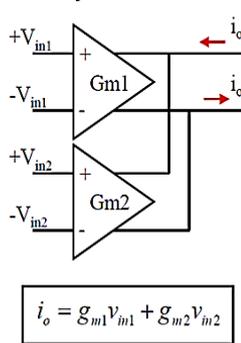
- Single-ended inputs



- Differential inputs

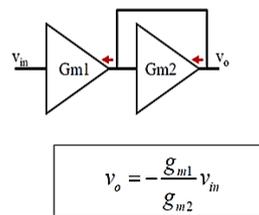


- Fully Differential

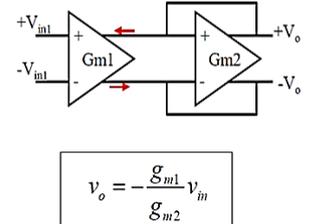


Scalars [Amplifiers] (circuits)

- Single-ended

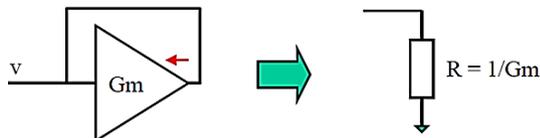


- Fully Differential

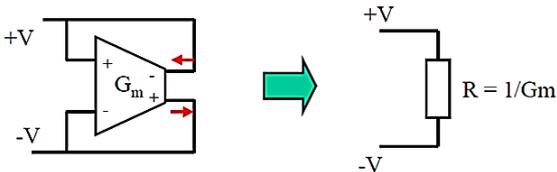


Resistors (circuits)

- Grounded resistor



- Floating resistor



Inductors (circuits)

- Grounded Inductor

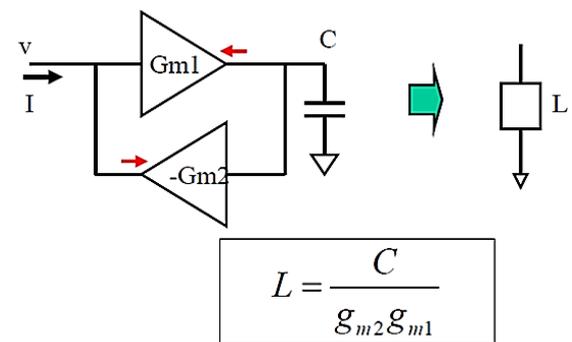
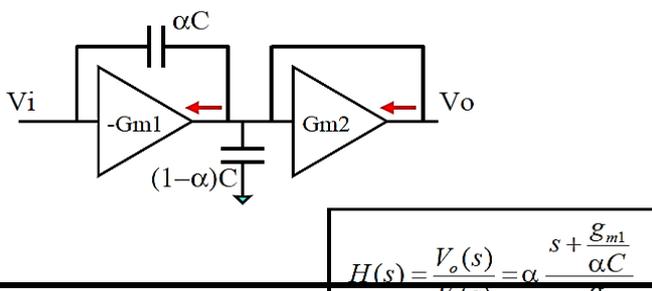


Figure 4.4.14 : EMULATING resistors , inductors , adders and amplifiers using Gm-C cells only

Then we now can introduce the filter circuits that is based on the Gm-c cells , they may be just

- Single-ended



- Fully Differential

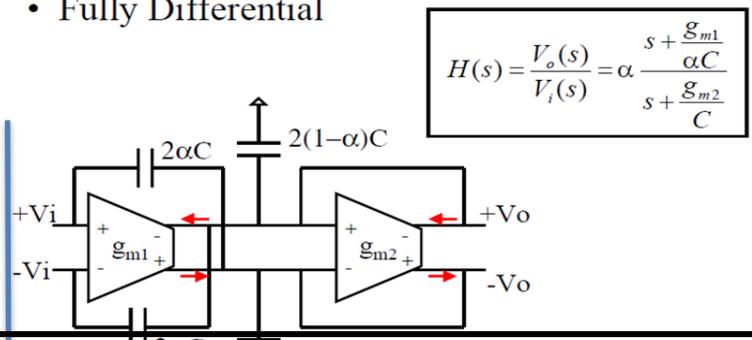


Figure 4.4.15 : the first order Gm-C filters

integrators , or mapping from a corresponding Active-RC filters , or an implementation of a certain SFD that gives a certain transfer function , let us first introduce the first order filters here below

And now , we can see the second order one in the clear figure shown below

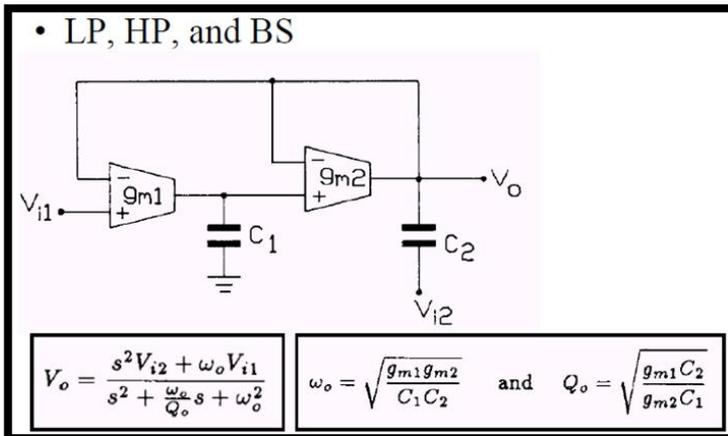
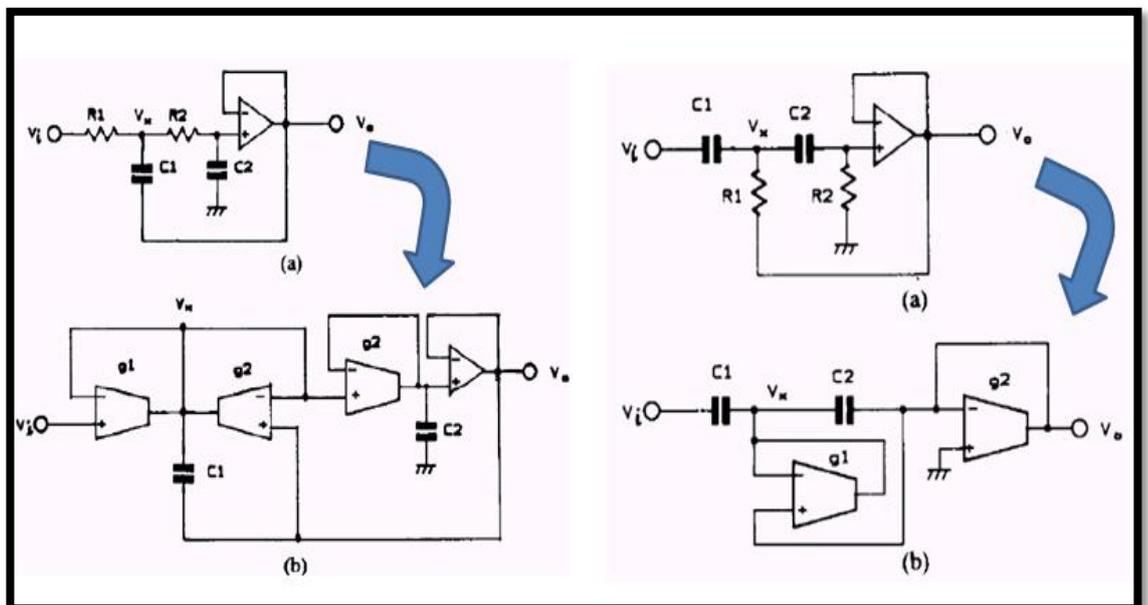


Figure 4.4.16 : second order Gm-C filter that can be employed as low , high and band pass filters depending on the values of Vi1 and Vi2

Figure 4.4.17 : second order Gm-C filters using direct mapping from the active-RC filters



Now , we must introduce the advantages and the disadvantages of the Gm-C filters , firstly The advantages are . . .

- ▶ Highest frequency performance -at least an order of magnitude higher compared to other integrator-based active filters (<100MHz).
- ▶ Faster than active-RC filters, since they use open-loop stages, and (*usually*) no OPAMPS
- ▶ Lower power, since the active blocks drive only capacitive loads (not resistive like OPAMPS).
- ▶ High frequency Operation and simple.
- ▶ Uses a single capacitor between differential outputs.

And its disadvantages are

- ▶ More difficult to achieve linear operation (as there's NO feedback).
- ▶ Gm must be linear and tunable.
- ▶ Sensitive to parasitic capacitances at the output nodes.

- ▶ Differential output requires common mode feedback.
- ▶ Requires some sort of common-mode feedback to set output common-mode voltage.
- ▶ Needs some extra capacitors for compensating the common-mode feedback loop.
- ▶ Typically, dynamic range not as high as OPAMP-RC but better than OPAMP-MOSFET-C.

4.2.4 The Switched-Capacitor filters

This is the topology that I used here in my filter , and it is one of the most popular topologies now in the literature for many reasons that will be introduced later , let us now summarize its main idea . . .

- ▶ A SC circuit operates as a DT (discrete time) signal processor.
- ▶ Most easily analyzed using the z-transform.
- ▶ The main idea of its operation is to EMULATE a resistor by using only switches and capacitors , the way that is used to do so will be explained later here .
- ▶ Then having resistors done by S.C. , you can implement your filter using simply a direct mapping from the Active-RC filters .

The main idea of the S.C filters as mentioned before is to EMULATE resistances by using only a switches and capacitors , this can be illustrated by the simple example shown here in the figure , in this figure we can see how the charges transfer from one side to the other side can be equated the amount of the charges that will pass from a resistor having an equivalent value of (T/C) .

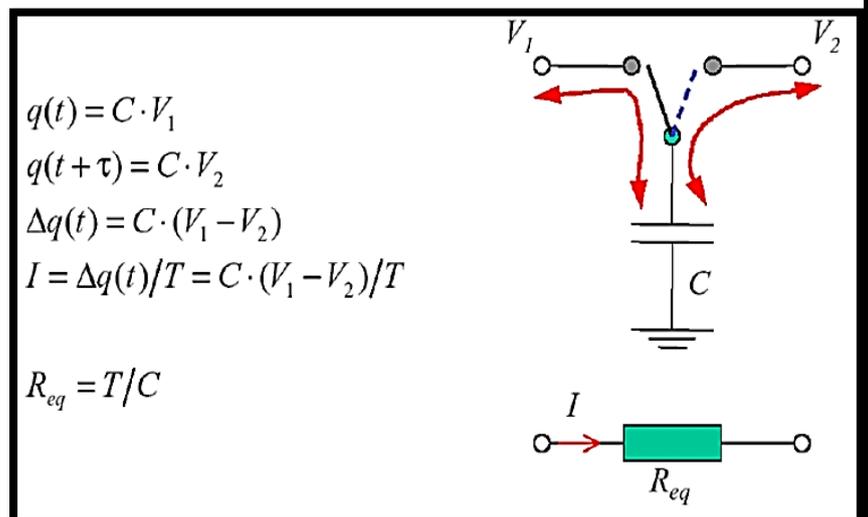
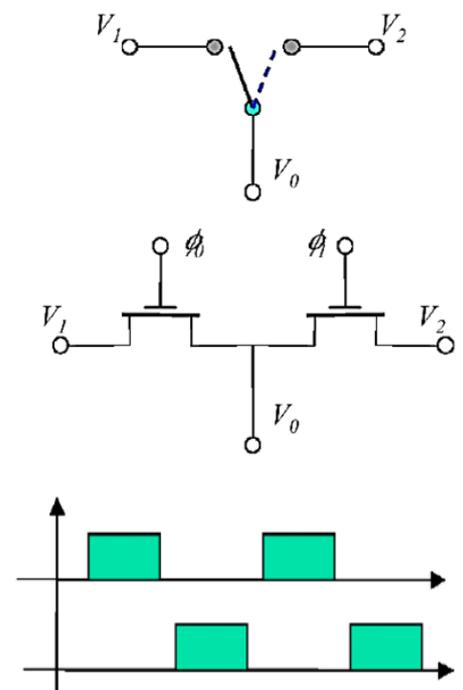


Figure 4.4.18 : how the S.C. can emulate a resistor

The switch that is shown in the previous figure can be implemented using a simple MOSFET as a switch , as the one shown in the figure here , But . . . in my final circuit I used a TRANSMISSION GATE as a switch for many reasons that will be illustrated in DETAILS later we see also the the clocks that drive the switches are Non-overlapping ones due to guarantee that the two NMOS transistors will not conduct in the same time , and that no charge leakage will occur

There are many problems that can be encountered in this type of filters , there are for example the Clock feed through , the charge injection , the Gain errors , ...etc.



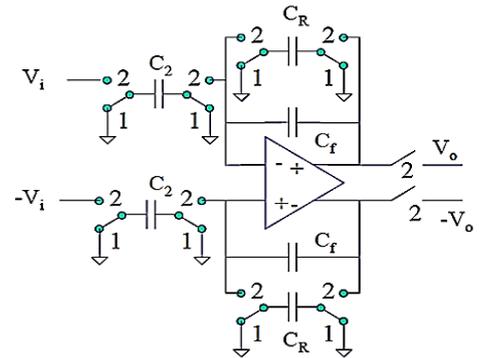
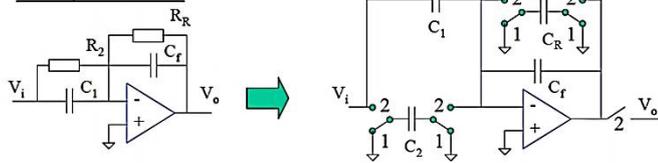
all of these problems will be discussed in detail in the section of the “ selected topology ” , so I will not mention it here and let it to be mentioned there .

Then , having a resistors emulated be a S.C. , you can do your filter using two methods , the first one is to get a certain Z-domain transfer function , then implement it using the S.C. and the CHARGE TRANSFER ANALYSIS , or if – and only if – the switching frequency is roughly large , you can do direct mapping from the Active-RC filters with just replacing the resistances by the S.C. cells here we can see in the figure below some examples of these filters

Figure 4.4.19 : the switches implementation in general (conceptual)

Low pass fully diff. filter

Low pass filter



$$\frac{V_o(s)}{V_i(s)} = -\frac{C_1}{C_f} \frac{1}{R_2 C_1 + s} \rightarrow \frac{V_o(z)}{V_i(z)} = -\frac{C_1(1-z^{-1}) + C_2}{C_f(1-z^{-1}) + C_R}$$

Eventually , we can summarize the advantages of

Figure 4.4.21 : the first order S.C. filters examples

the S.C. filters as follows . . .

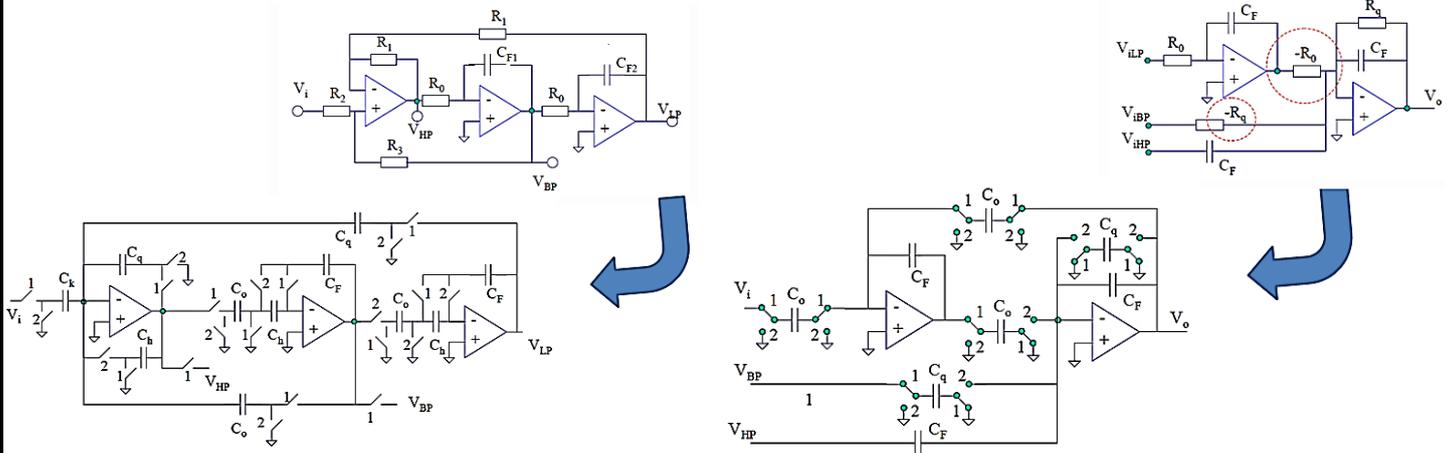


Figure 4.4.20 : the second order S.C. filters examples

- ▶ Accurate frequency response (determined by capacitance ratios that can be set with an accuracy of 0.1% or better)
- ▶ Good linearity and dynamic range
- ▶ Can be manufactured with low cost CMOS Processes
- ▶ Compatible with VLSI digital Circuits .
- ▶ Realization of high accurate resistor with small area.

And the disadvantages are

- ▶ It requires an anti-aliasing filter before it which may be tough in its design .

- ▶ It requires a reconstruction filter after it to smooth the signal at the output .
- ▶ The OTAs involved in its design will require a large SR and linear settling time .
- ▶ There will be a need for designing a clocking circuitry for such a filter .
- ▶ the Clock feed through , the charge injection , the Gain errors , ...etc. , will be an issues .

4.2.5 The switched current filters

Its main idea is clearly summarized in the following points . . .

- ▶ Its main idea is roughly similar to that of the S.C. filters , but here the CURRENT will be sampled rather than the VOLTAGE .
- ▶ It depends as shown here on the current mirrors to sample the value of the input current , and this is done by the means of the capacitor put at the gate , that see a practical *infinite* input impedance , then its voltage will be maintained and the CURRENT is said to be sampled .
- ▶ Using this methodology , we and make integrators , adders , scalars, . . .etc. then using this cells to make the needed transfer function with circuits that will be illustrated later here .

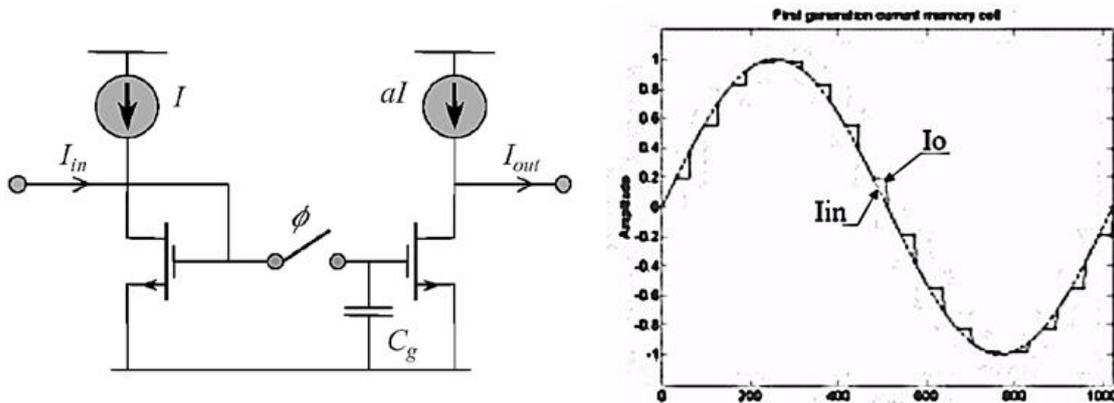
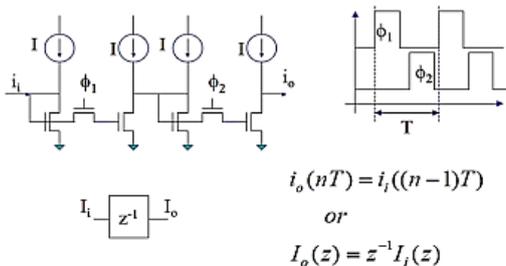


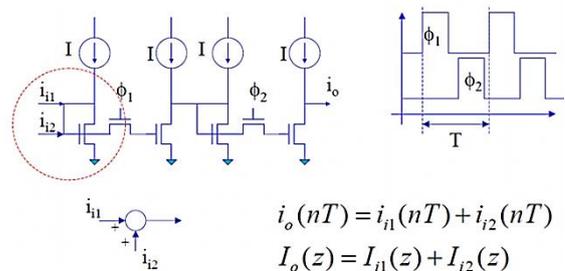
Figure 4.4.22 : the main idea of the switched current filters

Then using this idea , we can make the cells that may be used in making any transfer function by the means of SFD , here is some cells that is made using the SI (switched current) cells.

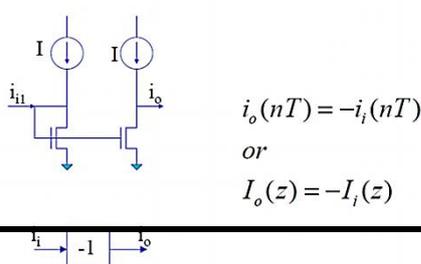
Delay



Adder



Inverter



integrator

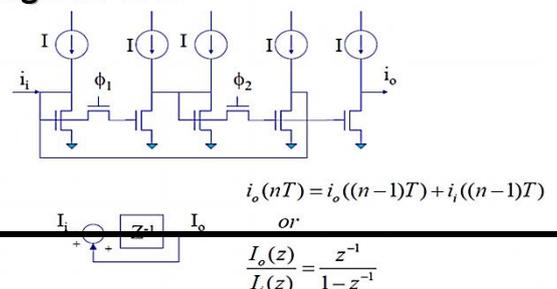


Figure 4.4.23 : the main cells made by the SI circuits

Then , using these cells we can do the needed transfer functions including the filters , here below I will introduce the different circuits made by the SI principals and the SFD used in these circuits . . .

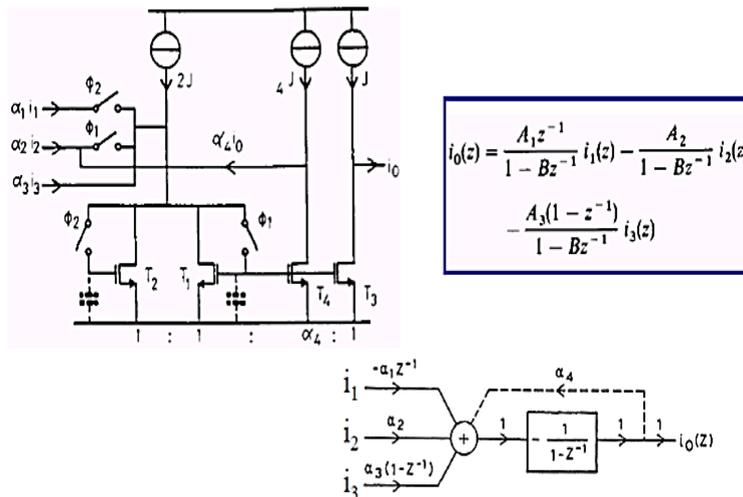


Figure 4.4.24 : the first order SI filter example

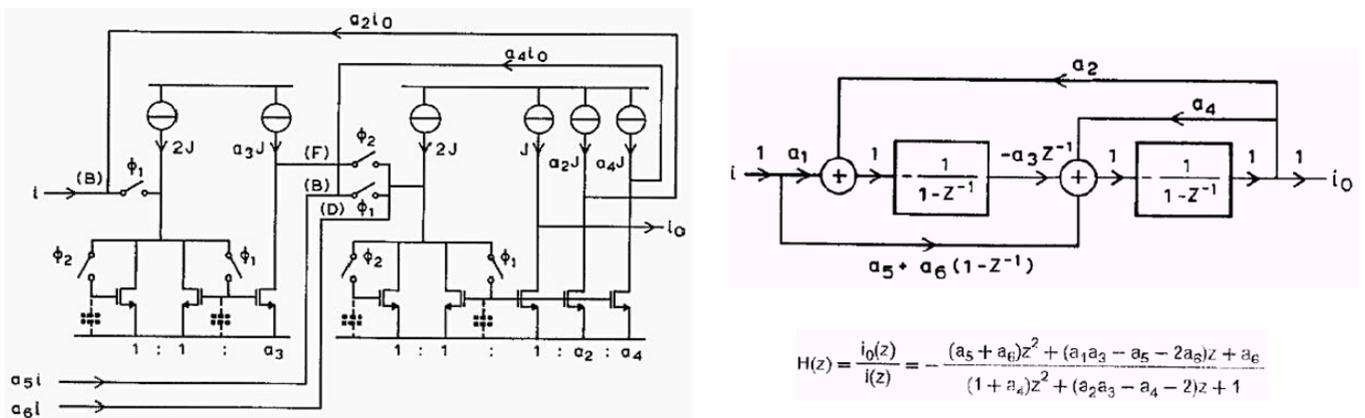


Figure 4.4.25 : the second order SI filter example

Eventually , we can summarize the advantages of this filter as follows

- ▶ It have the ability to adapt the VLSI digital trends .
- ▶ Low supply voltage is applicable .
- ▶ Low linearity can be accepted .
- ▶ Potentially high speed and high regularity in the structures involved.
- ▶ No impact of voltage scaling on the operation as the signals here is CURRENTS .
- ▶ Possible to increase DR while keeping the speed .
- ▶ 0.1% matching is achievable .

And the disadvantages in these filters are

- ▶ There are many issues to be solved like .
 - Device mismatches
 - Channel length modulation
 - Charge Injection
 - Noise
 - Settling and Leakage
 - Power Line drop and bias mismatch

4.2.6 The selected topology (Switched Capacitor filter)

The selected topology for the filter here is the S.C. filter , it is one of the best filters you may find in the SOC field due to many reasons , its idea is really intelligent and its design is really relaxed in many aspects , and its performance combines both of the digital systems accuracy and the analog systems capabilities .. I really loved this filter .

This filter today becomes one of the most usable ones in market , its compatibility with CMOS fabrications makes it one of the most desired filters in the literature , and many advances and tricks are adopted for it now . . .

We may –for short- summarize the benefits gained from using this filter in the following points here below ,

- ▶ Implementation is fully compatible with modern digital CMOS, requiring just :
 - Amplifiers (whose only requirement is to reach end-values between clock transitions, so that non-linearities are tolerable: DC gain and bandwidth are the key parameters);
 - Capacitors (metal interconnect capacitors are sufficient in most cases);
 - Clocked switches.
- ▶ Accuracies of key parameters depend on a stable clock frequency (primary parameters) as well as capacitor *ratios* (secondary parameters) and remain accurate with temperature and aging.
- ▶ Easy migration to the latest CMOS processes with only limited small signal parameters and matching data necessary.
- ▶ No tuning required (one of the most imp. advantages).
- ▶ Re-configurability and re-programmability which can co-function with re-configurable logic.

For all of this regions , and others . . . the selected topology is the S.C. filter , for which the design steps and the trade-offs and the challenges that was seen in the way of constructing such a filter will be illustrated clearly in the following sections

4.3 General issues

4.3.1 What is the complex filters , and how complexity generated ?

Wireless systems often make use of the quadrature relationship between pairs of signals to effectively cancel out-of-band and interfering in-band signal components. The understanding of these systems is often simplified by considering both the signals and system transfer-functions as ‘complex’ quantities.

Complex filters is a two input , two output network that can operate on these complex signals , it uses cross-coupling between the real and imaginary signal paths to realize asymmetrical filters (in the frequency domain)having transfer functions that do not have the conjugate symmetry of real filters. This implies their transfer functions have complex coefficients.

A complex transfer function can be realized using four real filters as shown in Figure below . for analog filters. An identical realization is possible for digital filters where the transfer functions are functions of Z-domain . . .

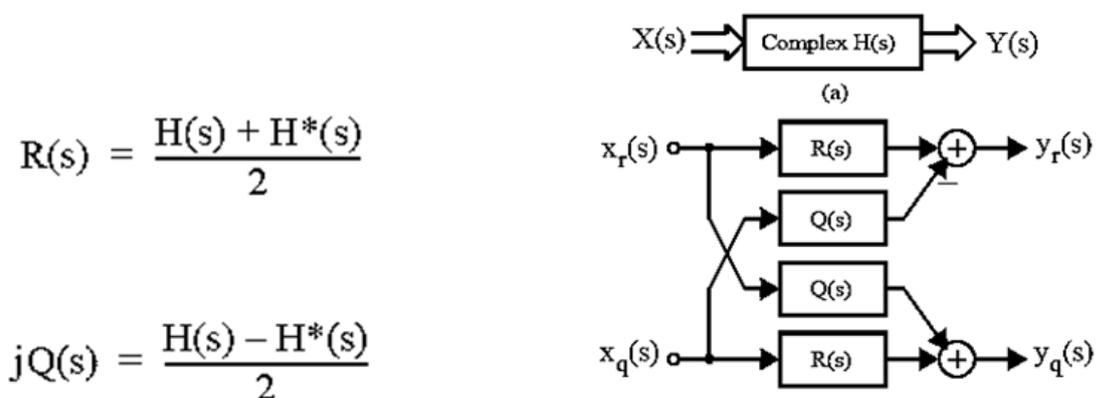


Figure 4.4.26 : the most general idea of the complex filters

We can see clearly that the real filters have a *cross coupling* from the output of a one filter to the input of the other filter , this is the main reason of the *complexity* in filters .

In the frequency band point of view , this can be see as a *translation* of the response of the LOW PASS FILTER to the

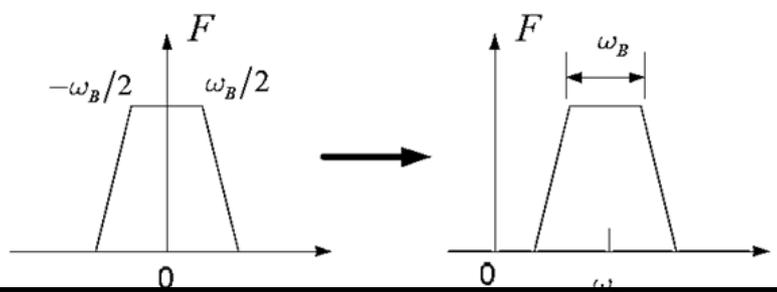


Figure 4.4.27 : transferring the low pass filter into COMPLEX filter

right (or left) as shown in the figure here , these is represented mathematically by replacing the S variable shown in the all of the transfer functions by $S-j\omega$.

This transfer of the response is done to make a Positive Pass Filter (PPF) , this is the main idea behind the image rejection , that is . . . the image in the *low IF* receivers is usually down converted with a 90 degree phase with the signal to be extracted , and it is usually in the negative side if the frequency domain , the complex filter will be able easily to get the signal only and suppress the image in the same time .

4.3.2 The selected circuit for the S.C. filter

In general , we can do easily an integrators and adders by the S.C. circuits , and using the previously shown concept of the *cross coupling* to generate the complexity of the filter , we can see that the show block diagram here will give us the SHIFTD response of the complex filter using only *integrators and adders* , you can verify it easily by following the mathematical equations of such a system .

Then now , we are in need for a S.C. integrator and adder and then we will be able to do our S.C. complex filter , the integrator is done simply by replacing the *resistors* of the corresponding Active-RC by the S.C. cells as shown simply in the figures here below .

We will notice here two tricks , The first trick done in the in this schematic shown here is that the two switches that sharing the same point and do the same job can be combined in only one switch for area and power optimizations .

The second trick that can can be shown in the second integrator in the figure 4.29 is that . . . as the C2 cap is always the min. cap. and all of the other capacitors are only a certain multiples of it (as the value of the R2 in the corresponding Active-RC is the largest between the all resistors values) , then reducing its value will be efficient for the area saving , But , How to do that by the same value of the Cap. C2 ?? . .

The trick is to use two of it in SERIES , with considering there switching , then . . . the equivalent value of the C2 is halved , and so the values of C1 and C3 .

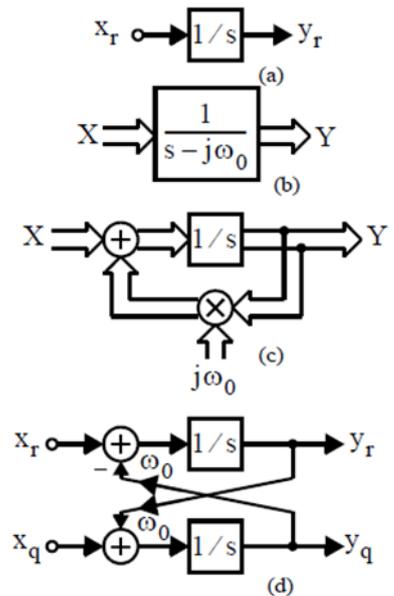
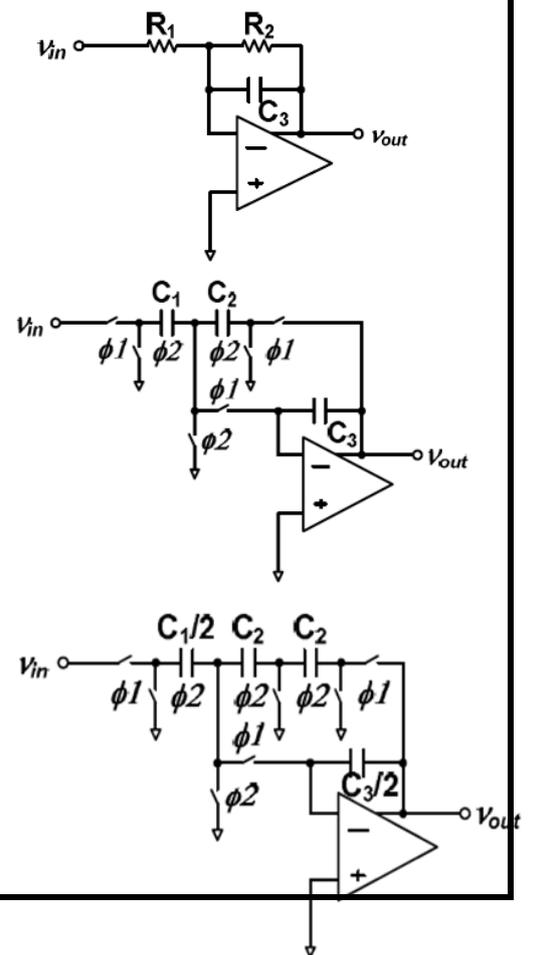


Figure 4.4.28 : making complex filter using only integrators and adders



Then using the last proposed integrator with the tricks shown before one can introduce the final complex integrator cell that will be the main building block stage of the filter , it is shown here below clearly in the circuit level , we can see that , we really have integrators , but we don't have adders (at this ease) , but we know that for the active integrators . . . If the inputs is coming from multiple sources through resistors (or S.C. cells here) the integrator will response to the sum of these inputs , it will do the job of the adder in reality inherently as it is done with active components ,so we can do this by this shown circuit here .

Figure 4.4.29 : the single S.C. integrator cell used in the filter

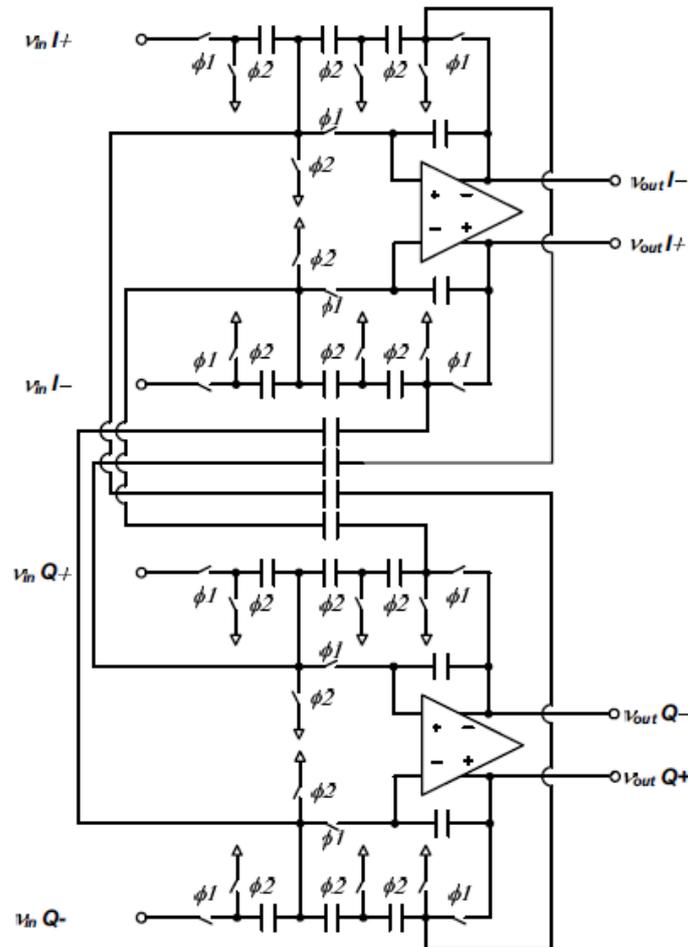


Figure 4.4.30 : the main circuit of the single stage complex integrator used in our filter in reality

4.3.3 The filter's main switch design

For any S.C. filter , the needed specs on its switch will be as follows . . .

- ▶ MIN. R_{on} through operation .
- ▶ Constant value for R_{on} w.r.t. the voltage applied .
- ▶ MIN. charge injection for certain error .
- ▶ MIN. clock feed through during operation .

Let's examine these specs and tradeoffs involved in the design if the main filter's switch

The selected topology for the switch is the *Transmission gate* switch , why ?? ... because this topology will introduce to me the following benefits ...

- ▶ It solves the problem of the variability of the R_{on} with the variation of the input voltage inherently as the combination of the parallel NMOS and PMOS transistors make it have a low value R_{on} in the both low and high input voltages as shown in the figure here ...
- ▶ It solves also the problem of the *clock feed through* as the clock is feed differentially to the switch at the same time then the error the will be introduced at the output due to the clock feed through will cancel each other .

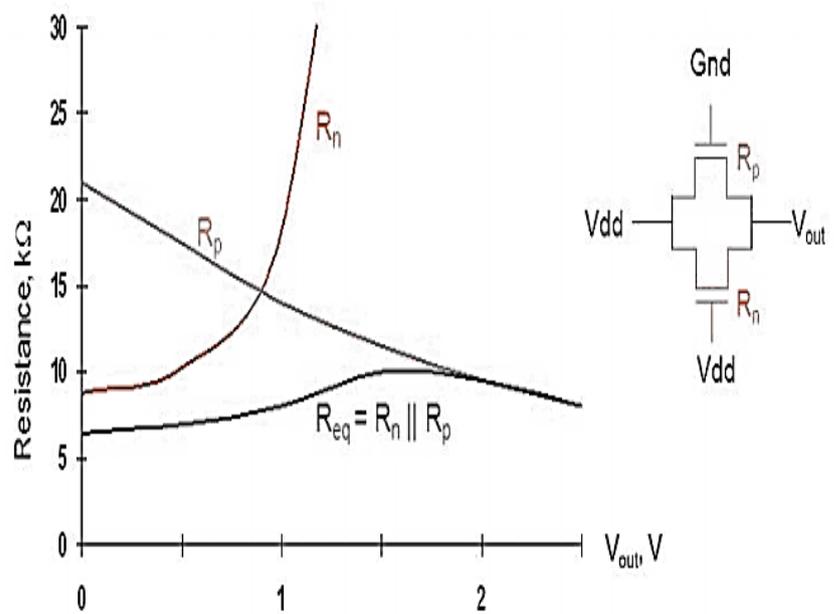


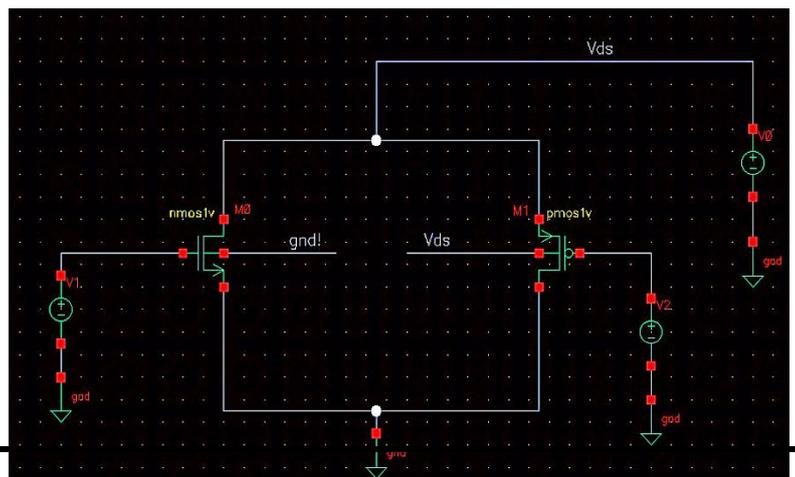
Figure 4.4.31 : the transmission gate's resistance Vs. input voltage

- ▶ It is guaranteed that it must pass corners , as the transmission gate is the main switch used in the digital part , and the all technology is driven in reality by the digital progress , then the FAB itself will not introduce a new technology until all of the main digital parts will pass corners safely .

Then it will remain the problems of the *charge injection* and the *min Ron* , both must be satisfied in the same time , but you must notice that we have a contradiction here , that is , ... the min R_{on} condition will need us to ENLARGE the width of the transistor as this resistance is inversely proportional to the transistor's width , and the charge injection is directly proportional to the area of the transistor , then it is directly proportion to the width of the transistor .

Then for such a case , the optimization must be done to get the optimum value for the two cases to be fulfilled satisfactorily ... to do so , two test benches are constructed to measure these two parameters as a function of the width , then we will be able to get the optimum value ...

The shown schematic here is the test bench of measuring the R_{on} as a



function of the width , we do DC sweep with the design variable W , then this sweep is done again versus various values of the length L to know the effect of the length on the value of the R_{on} , the results if this test is shown clearly in the following page . . .

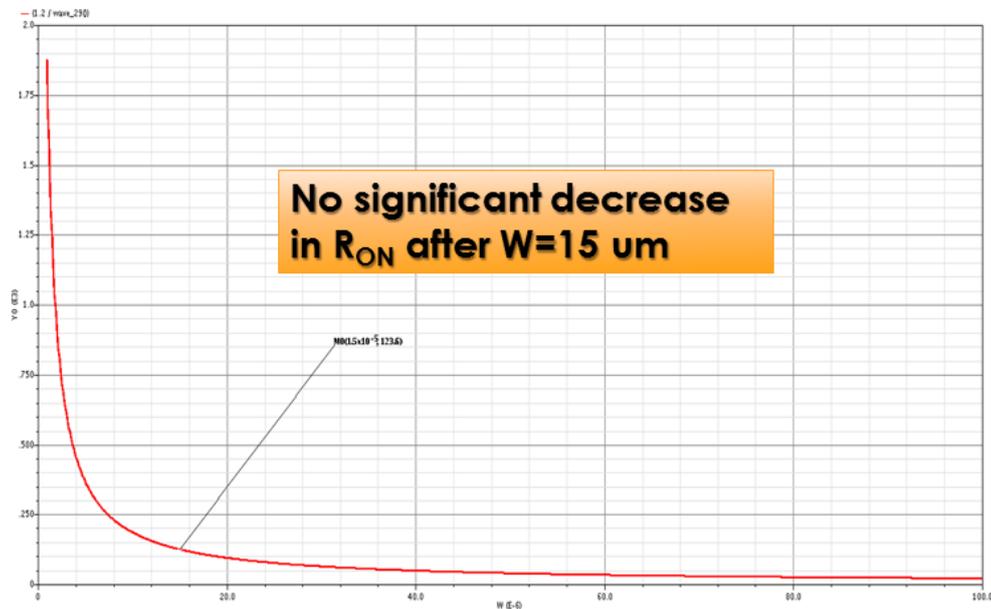


Figure 4.4.32 : R_{on} versus the transistor width

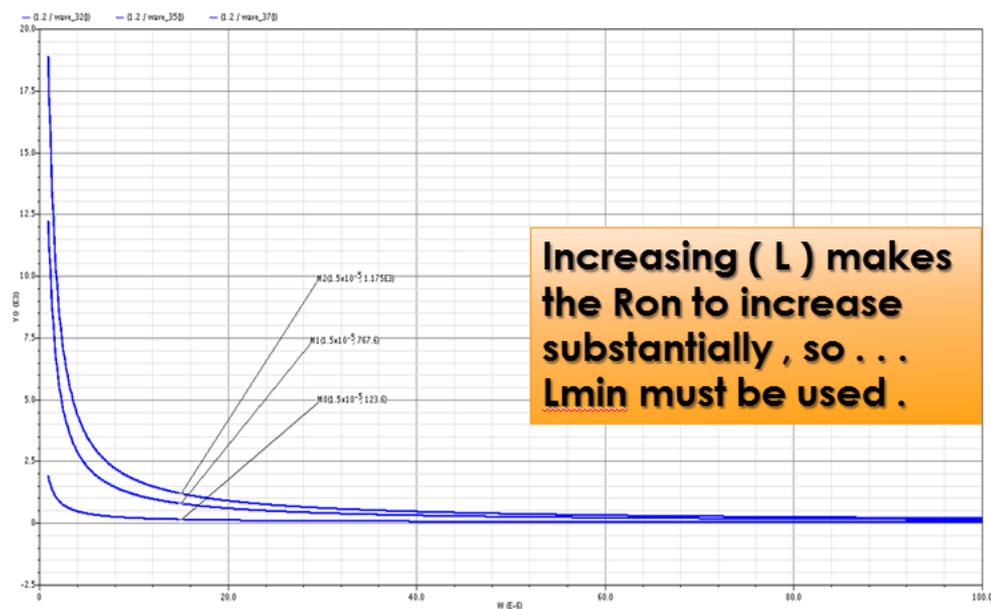
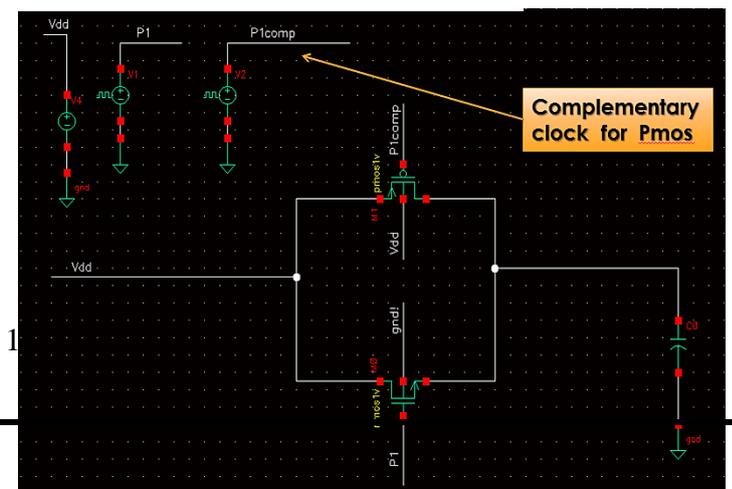


Figure 4.4.33 : the variation of the R_{on} with the transistor length

Then from the figures above , and the comments shown on them we see that the value of $W=15\mu$ is a good one to get an R_{on} of a value of 120 ohm , but what will be the charge injection



effect at this point , for knowing this , we will do the test bench seen here . . .

In the previously seen test bench , the switch is to SMAPLE the Vdd value on the capacitor , with the ideal clock applied on it to be the switching clock , if the switch was ideal , the wave form on the capacitor must be CONTANT and equal to the Vdd , but due to the charge injection , a *Glitches* in the output voltage will be found as the seen ones in the figure below

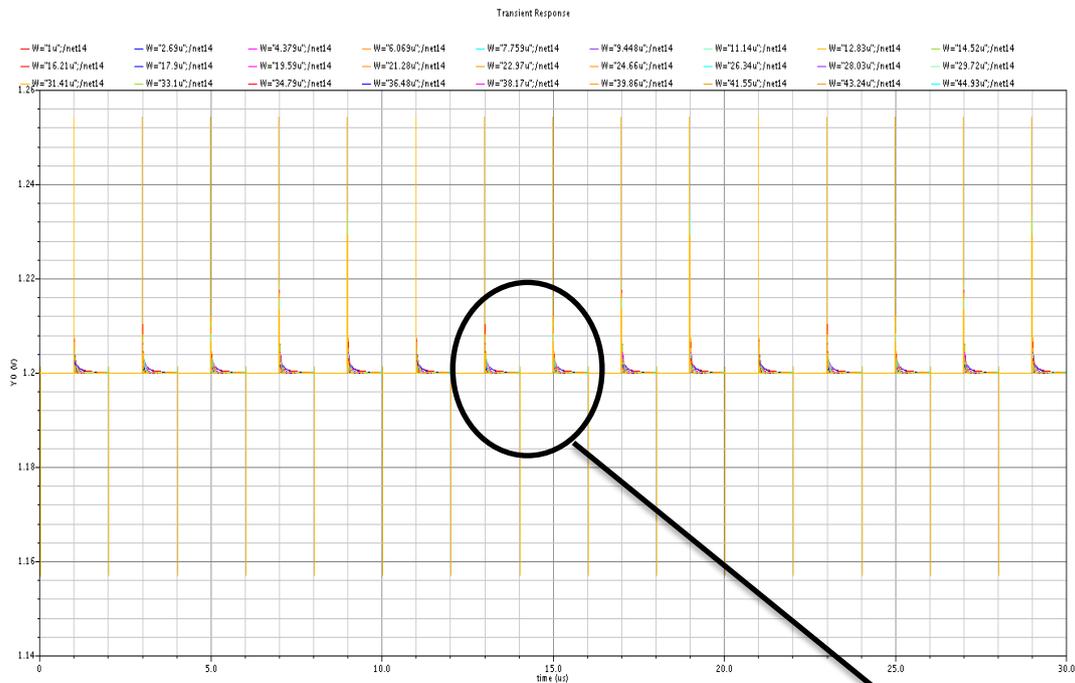


Figure 4.4.34 : the output of the charge injection test bench

Here , we can see that the error due to the charge injection is roughly acceptable , and hence it is now clear that the value of $W=15\mu$ and the $L = L_{min}$, are the optimum value of the switch's transistors size .

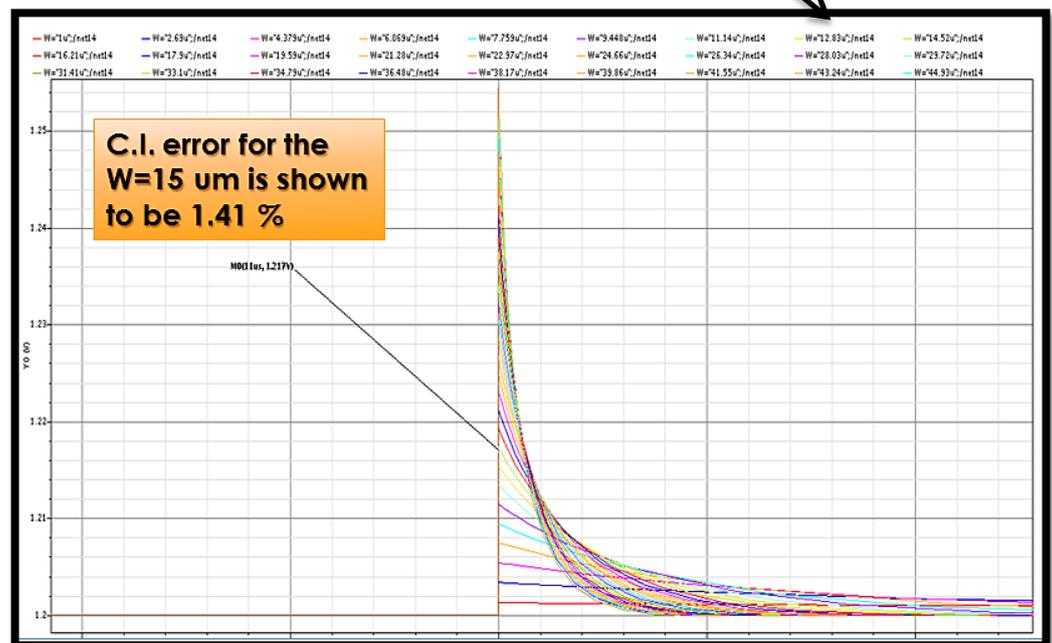


Figure 4.4.35 : the error due to charge injection

We noticed in the circuits of the *integrators* involved in the complex filter was usually containing two switches sharing the same point, then doing one symbol for these two switches will ease the operation of drawing the schematics.

Then for this purpose, I combine two transmission gate in one symbol as shown in the figure here, so as to get the *dual way switch*, this switch is ROUGHLY clarified in the proceeding figure...

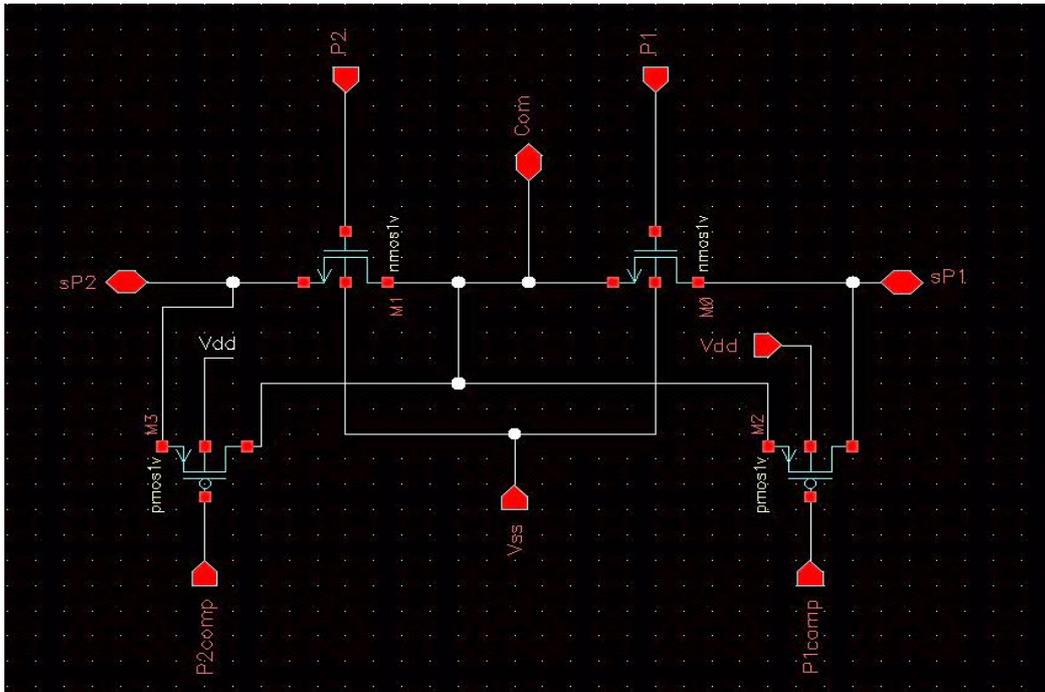


Figure 4.4.36 : the main switch schematic used in the filter

Eventually, you must know that this will be the symbol used in the ALL of the filter circuits in ALL of the proceeding section, so it is important to know the origin and the interior circuit of the switch used in the filter, it can be seen that the symbol used is done professionally and is dramatically expressive for the function that it performs.

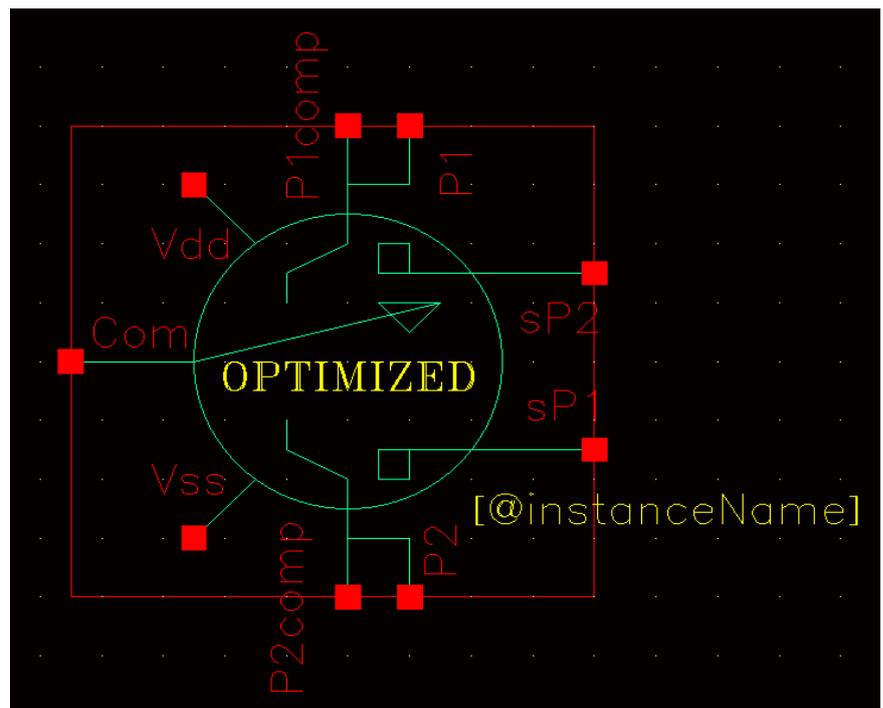


Figure 4.4.37 : the main symbol used in ALL filter circuits

4.3.4 The schematic and symbol of the single complex integrator stage

Before introducing the iterations done and the filter itself , we must know the symbol that is used in the single stage of the filter and what is inside it , we remember the complex integrator introduced in the section 1.3.2 , it is containing only OTAs (called the core OTA) and the switches that can be done by the one optimized in the section 1.3.3

The circuit of the OTA will be explained later in detail in the sections containing the filter simulations and circuits , the figure shown here is the schematic of the *single stage complex integrator* that will be used in all of the filter circuits introduced later .

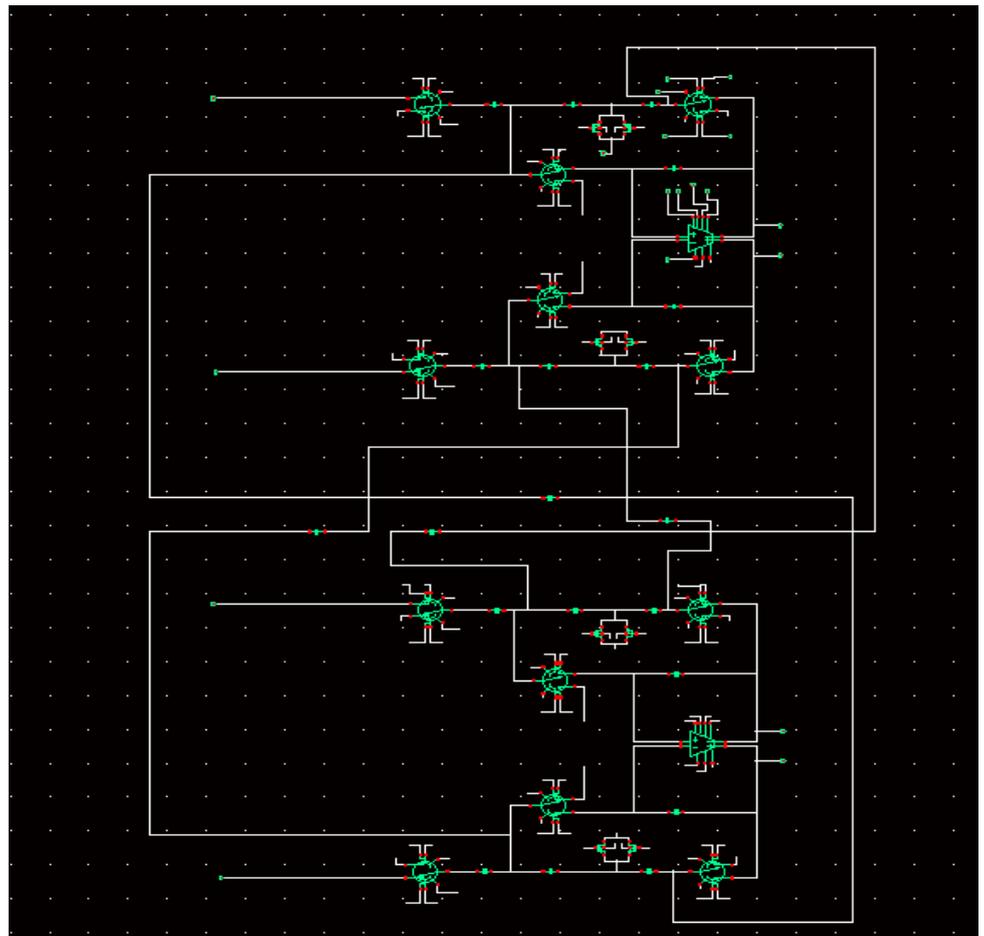
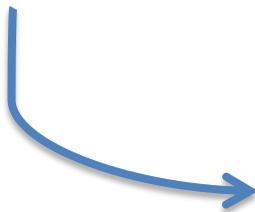
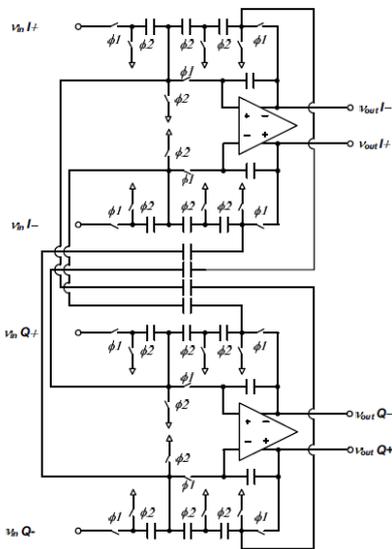
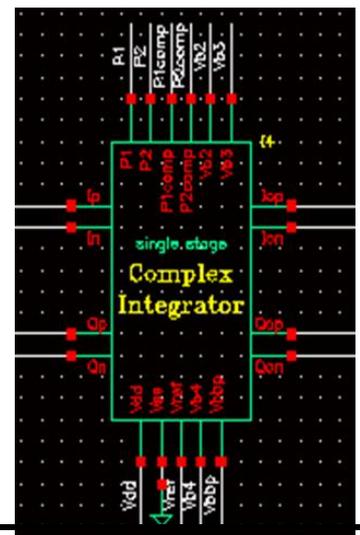


Figure 4.4.38 : the schematic of the single stage complex integrators

And in the figure here , the symbol used for this schematic is shown here , it have the differential ended input and output for both I and Q channels , and the clocks inputs also is shown clearly , this symbol will be used for all of the filter schematic shown later , so it is very important .

Figure 4.4.39 : the symbol of the single stage complex integrators



4.3.5 How to measure the response of the complex filters in CADENCE

The complex filter offers a transfer function that is “complex” in reality , that is , the positive side in the frequency domain is totally different from the negative side . . . but in CADENCE , if you wants to plot the transfer function in the LOG scale , the negative side will not be seen in the same plot with the positive side , then we make the following trick to overcome this problem , the AC sources is introduced at the both channels of the filter , but the phase difference in between will determine the side that i draw for which the transfer function .

This methodology is dramatically illustrated here in the figure here , and this way will be used in all of the simulations for our filter .

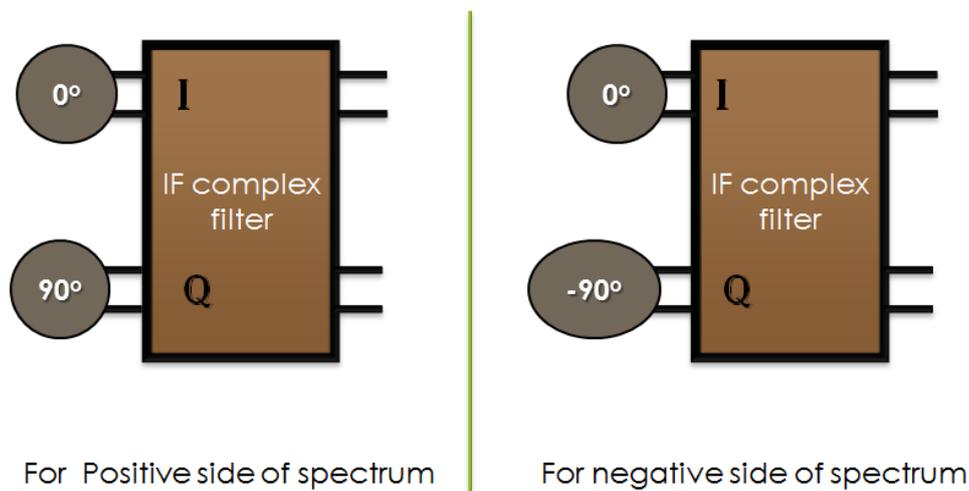


Figure 4.4.40 : how to measure the response of the complex filters .

4.4 Failed iterations , and tradeoffs solved throughout it

4.4.1 The first iteration

This iteration was a one for the idea verification and for excavating for the main tradeoffs that I will met in the design of this filter , the OTA done here is not the one that is assigned in the final filter , so The details of this OTA is not to be given here . only a summary of its parameters is given here in this table shown here .

The circuit of the iteration is shown here below clearly , it is done using the symbols introduced before in the previous sections , and we can notice also the input sources that is explained before , and it is explained how it is used for measuring the transfer function of the complex filter .

The analyses done to measure this transfer function is the PSS and the PAC analyses , this due to the fact that there are a clock in my circuit , which is a large signal that will vary the DC operating points of the all circuit , for such a circuits , we can't do a simple DC and AC analyses but we must instead do the PSS and PAC analyses .

Property	Value
Vdd	1.2
Current	80 uA
Gain	68.14 dB
BW	65.24 KHz
PM	56 °
GBW	166.53 MHz
SR	10.23 nsec
CMRR	265 dB
PSRR	263 dB
CMFB loop gain	55.53 dB
CMFB loop PM	78°

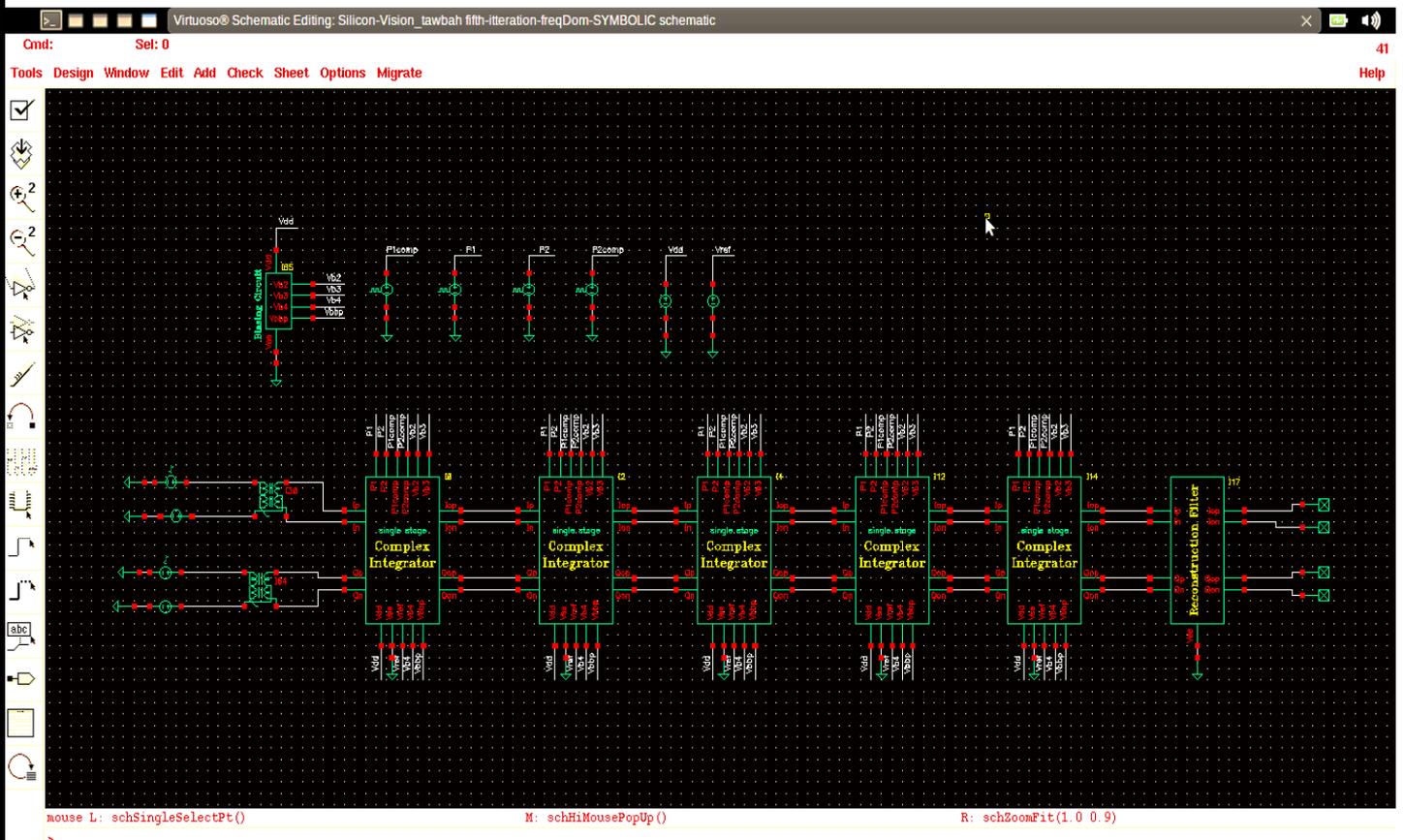


Figure 4.41: the circuit of the first iteration

The results from this circuit is shown here . . .

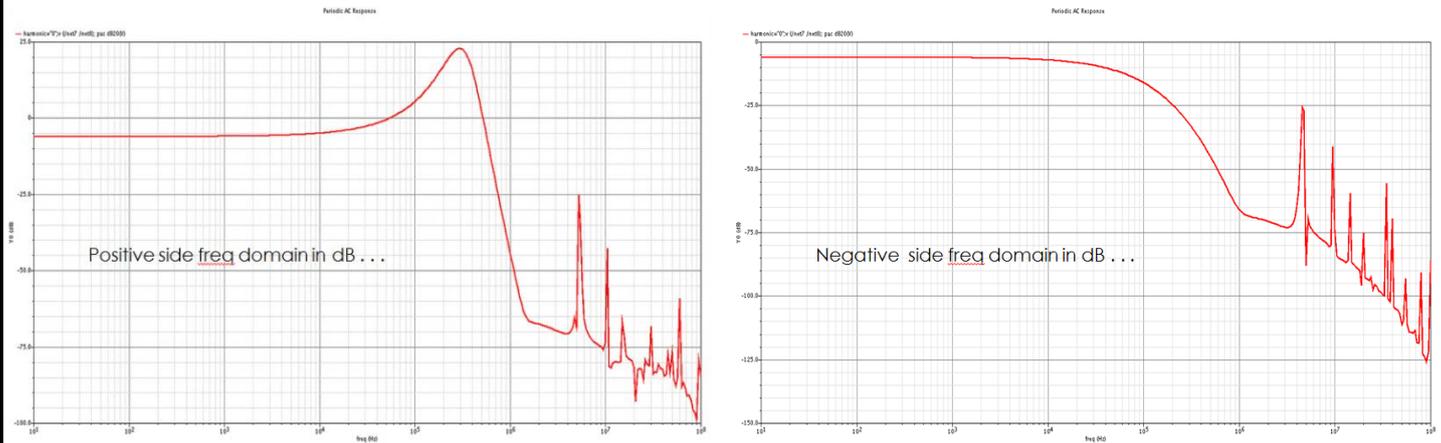


Figure 4.4.42 : the response of the first iteration filter

The main problems encountered in this iteration are

- ▶ there are a Tradeoff that can't be easily treated , that is ,, the specs that was given by the system designer (which he got from the Datasheet) was that for the positive side of the complex filter will be a band pass (of course) with a center frequency of 304 KHz , and a width of 540 KHz
- ▶ this means that the Q needed for this filter must be $= f / \Delta f = 304 / 540 = \text{less than one}$, this Q value when made in reality the specs of the ACRR and the IMRR were not met absolutely , they were ACRR = 25 dB and IMRR= 25dB also , these specs found to need a relatively high Q in reality (when the Q is increased and a high Q complex filter is done , these specs was met) . . .
- ▶ But when we increase the Q value , the band width was lowered , and the spec of 540 KHz band width becomes impossible , then we find that the main solution for this problem is to increase the IF center frequency so that for the high Q used we can also get a high band width (remember that $Q = f / \Delta f$) . . .
- ▶ then we suggest an IF freq of 1.2 or 1.2 MHz , and for the ACRR to be met , we make the filter order to be 6th (instead of 5th) , all of the specs was met by these new IF frequency . .

4.4.2 the second iteration

The modifications that are done in this new iteration are . . .

- ▶ Firstly, the order increased to be [6 order] after consulting the company in the matter , to be able to get a reasonable ACRR with this BW needed.
- ▶ The Aliasing filter and the reconstruction filter is done also , each one of them essential for the proper operation .
- ▶ Both of these filters is done easily by an R-C sections , they are indicated clearly in the next page
- ▶ The values of the capacitors is increased for two reasons (lowering KT/C noise , Kit limitations)



Figure 4.4.43 : the AA and the reconstruction filters of the second iterations

These are the modifications done , the circuit of the A.A. and the reconstruction filters done are shown here

The circuit of the second iterations is shown here below . . .

This are the results obtained in this iteration (in brief) . . .

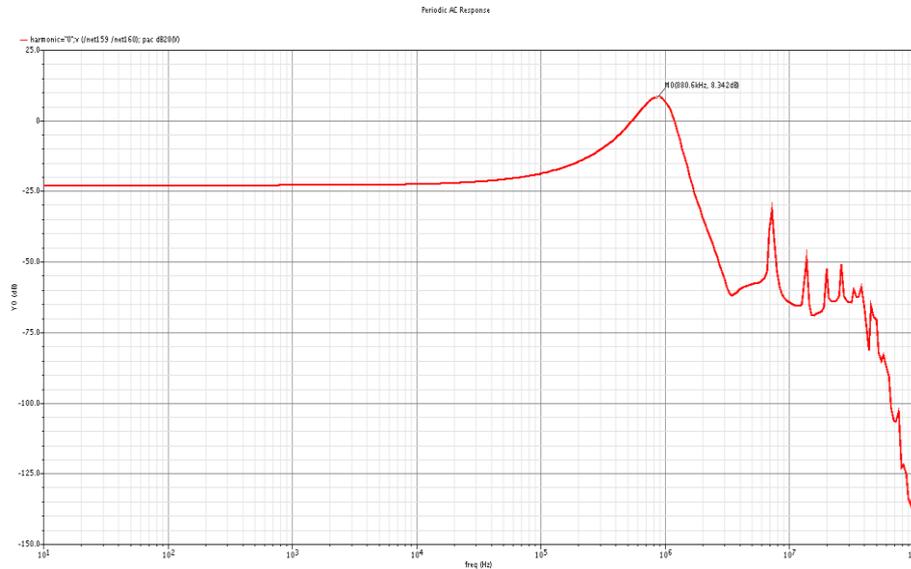
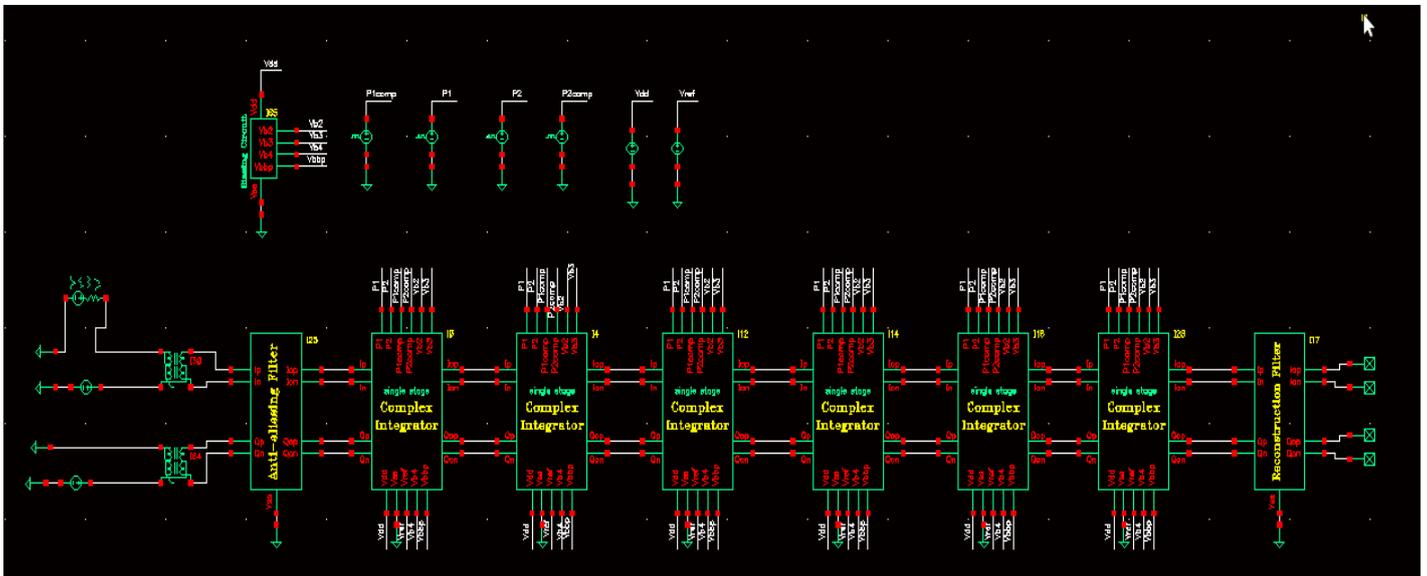


Figure 4.4.45 : the response of the second iteration



The problems encountered in this iteration are

- ▶ The main problem was the power consumption , the power now become 1.08 mA , the power wall is broken now . . .
- ▶ And it will be broken further , as we must do the output stage that may draw large current for driving the resistive load of the PGA .

- ▶ The specs is achieved but at the EDGE , any corner may do more . . .
- ▶ The noise planning of such a circuit is really bad and if the nosie analysis is done , it will be very noise in reality .

4.4.3 The third iteration

The new modifications done in tis iteration are . . .

- ▶ An input stage is done , and output stage also .
- ▶ A tough noise planning is done in this iteration to enhance the overall noise performance , that is , the gain is drifted into this done input stage to enhance the noise , this input stage was made to be low noise as far as possible , it actually acts like the LNA for the all system .
- ▶ Also , for the noise planning , I make a CURRENT REDISTRIBUTION , that is , the OTAs in the last stages is made to have lower current , and this of the input stage are given this gained excess current to be less noisy .
- ▶ In this iteration also , it is recognized that the filter order in reality is not simply the No of this cascaded stages , that is . . . the A.A. filter is responsible also for the roll-off of the overall filter .
- ▶ But why the ACRR and Attenuations were so hard to achieve ?? , all was due to the low SWITCHING FREQUENCY that was used , which makes the alias bands to affect the main pass band .
- ▶ Then , the SWITCHING FREQ . Is raised to be 15 MHz , then we gain many things
 - The ACRR and the attenuation needed after 2 MHz and 10 MHZ can be done now with LOWER number of stages .
 - Then lower power is achieved now .
 - The IF freq. can be lowered also , then relaxing the ADC specs .
 - The specs of the REC filt. and the AA filt. Is more relaxed
- ▶ This step was not to be done without a test that was done in the time domain which ensures that the OTAs done will support this Switching freq .
- ▶ Also the output stage is done to have a low output resistance , to be capable of deriving the resistive load of the PGA .
- ▶ Also the values of the used capacitors is more raised to reduce the mismatch problem to a limit that makes it not to be an issue In reality .
- ▶ The AA filt. Is highly improved by making the input stage not only to give a gain , but also to make its dominant pole to be at the same freq as the one of the original AA filt. Done before , it becomes like a Gm-C filt. Followed bu an RC- section , which gives more attenuation for the alias bands .
- ▶ A new BW analysis is done to know the real needed BW for this filter , a MATLAB model is done for a GFSK signal and the BW is calculated from it (see appendix A)
- ▶ It is found that a 320 KHz Band width will contain more than 97 % of the power , we make it these tough to consider also the uncertainty of freq. of pll as the channel may be shifted that depend on freq tolerance of pll .

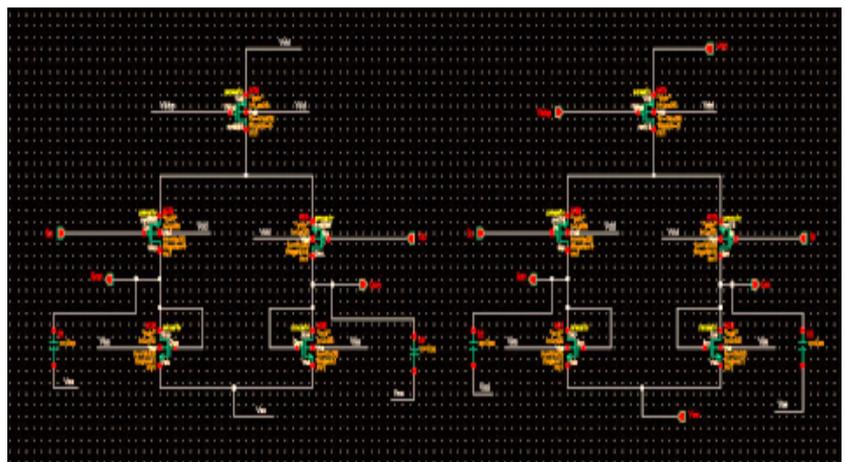


Figure 4.4.16 : the input stage of the third iteration

The circuits of the input third iteration are shown in the figure here . .

And the response of the all A.A. filter in the third iteration is shown here ...

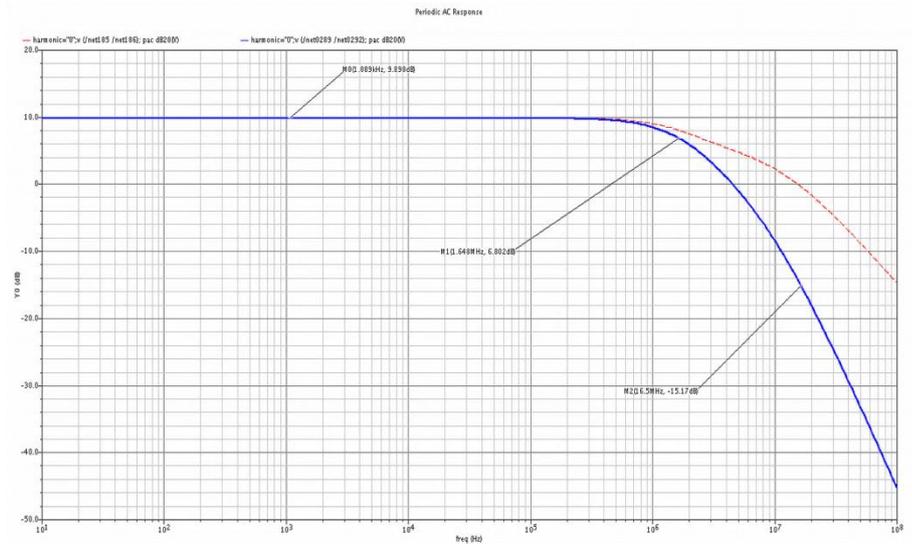


Figure 4.4.47 : the response of the A.A. of the third iteration

And the circuit of the output stage in the third iteration is shown here . . .

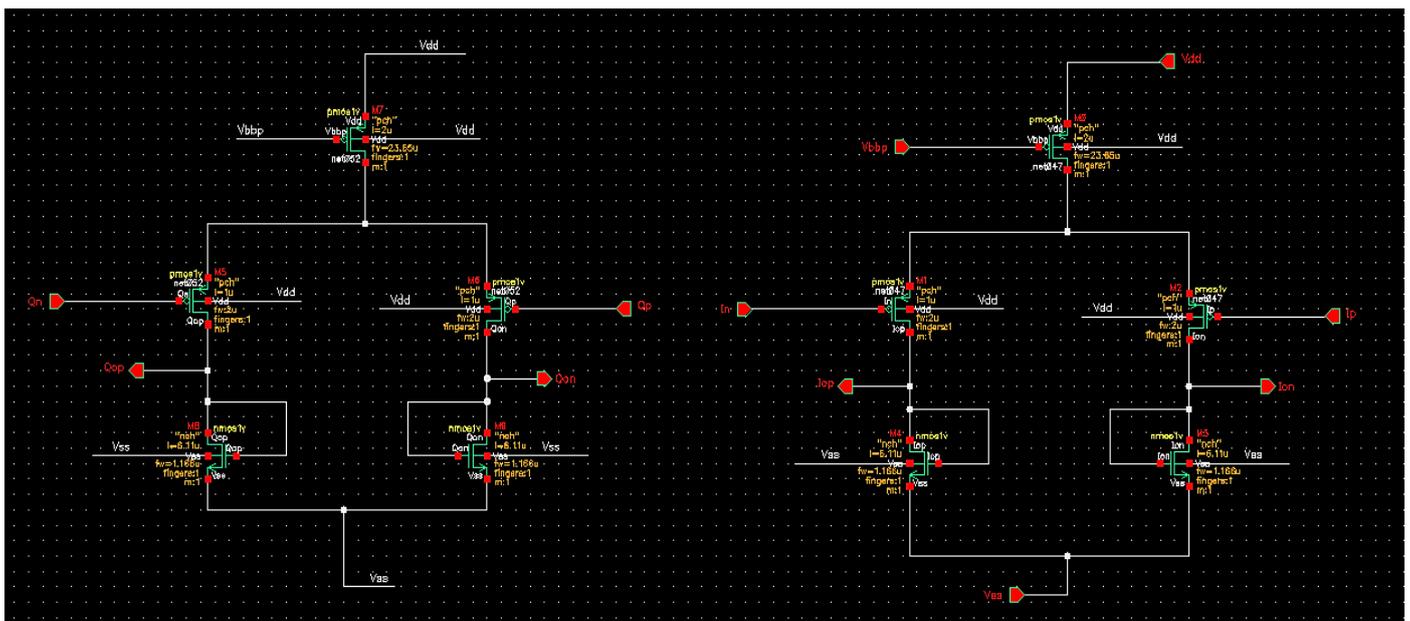


Figure 4.4.48 : the output stage of the third iteration

This is the circuit of this iteration

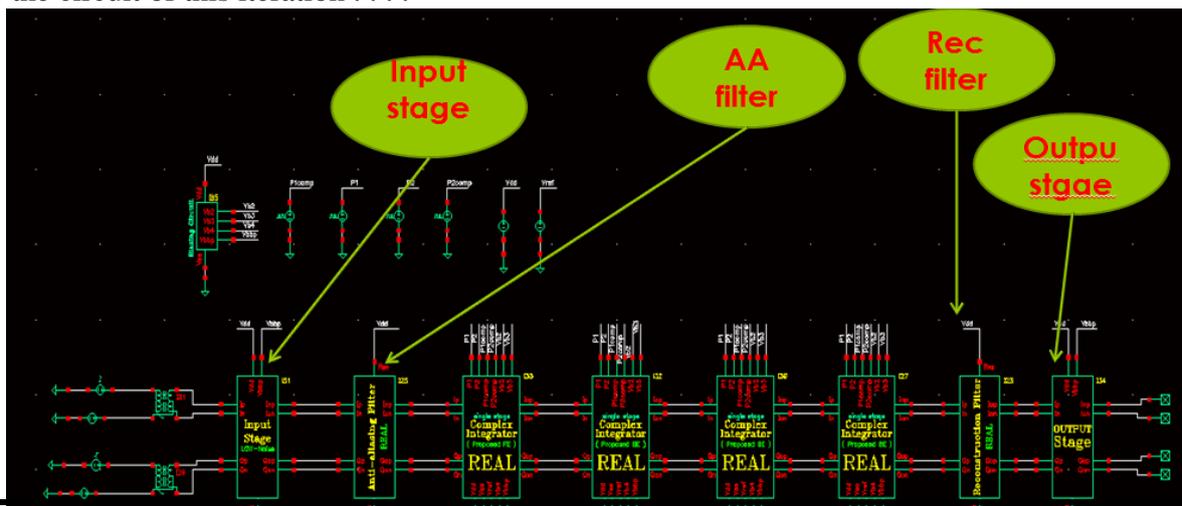
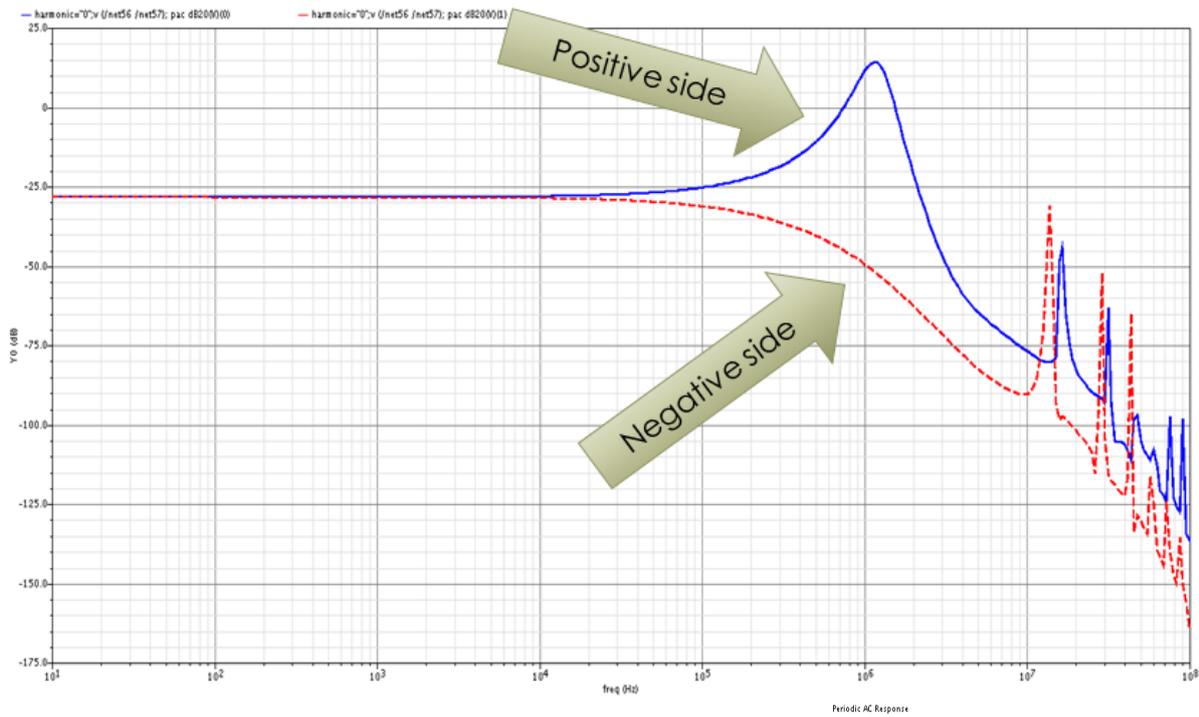


Figure 4.4.49 : the circuit of the third iteration

The results obtained for the third iteration are shown here clearly . . .



And

Figure 4.4.50 : the response of the third iteration

the phase reponse of the third iteration is shown here clearly

. . .

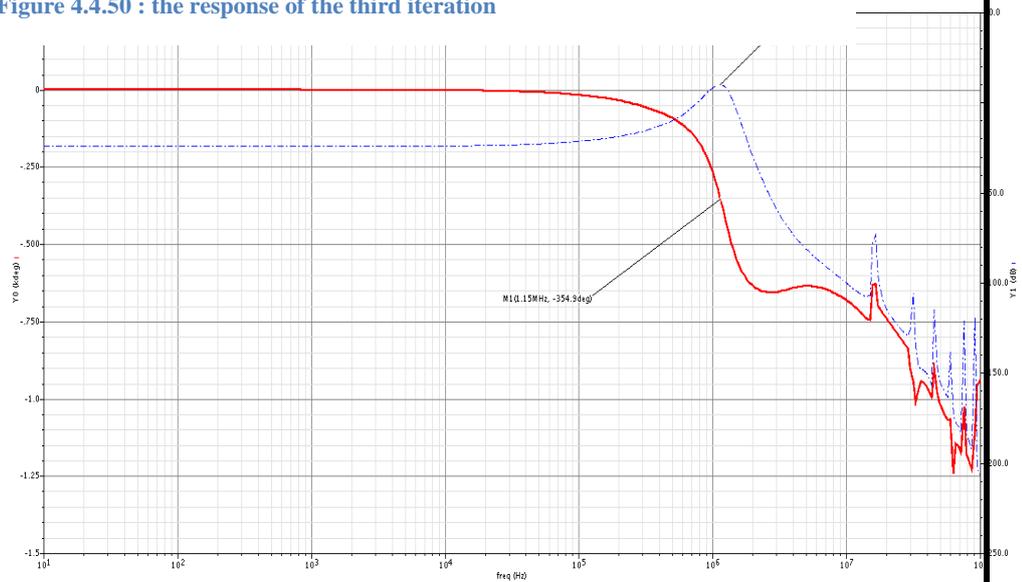


Figure 4.4.51 : the phase response of the third iteration

The problems encountered in this iteration are

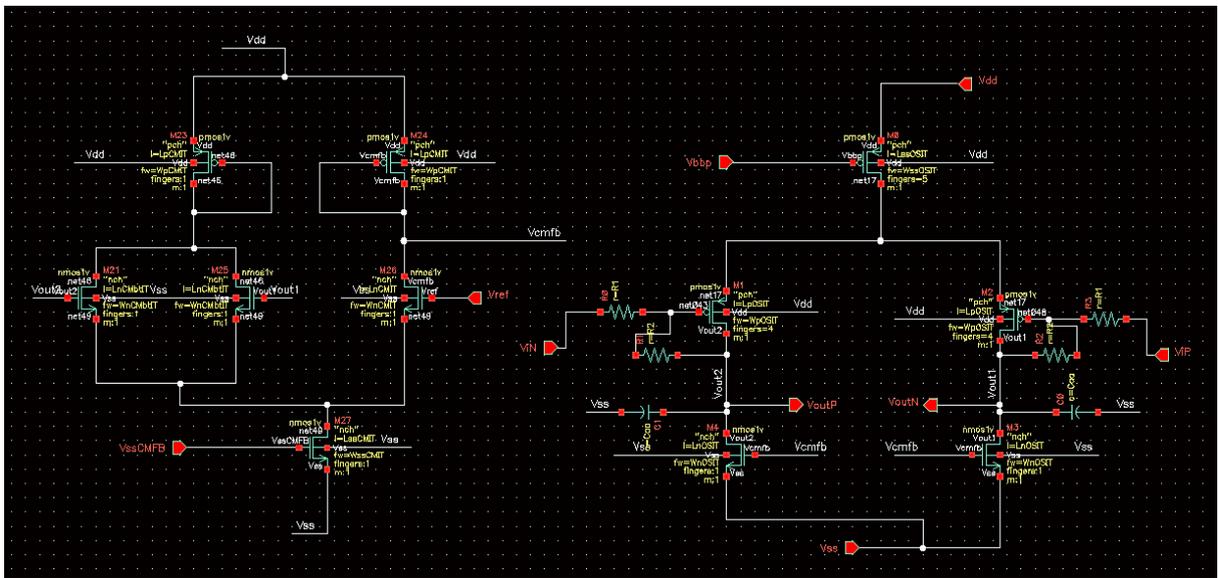
- ▶ The first problem is the this bad injection of the alias frequency band which makes the BLOCKERS in this band to pass easily from the filter . . .
- ▶ The other , and very critical problem is that the input and output stages is designed to have a low R_{out} , then they contain diode connected transistors for this purpose , but this makes its gain to be dependent on gm/gm value , that is , it will be highly VARIANT with corners , PVT , ...etc.
- ▶ The last problem is that the $iip3$ will be bad as the gain is only CONCENTRATED In the front end of the filter .

4.4.4 The fourth iteration

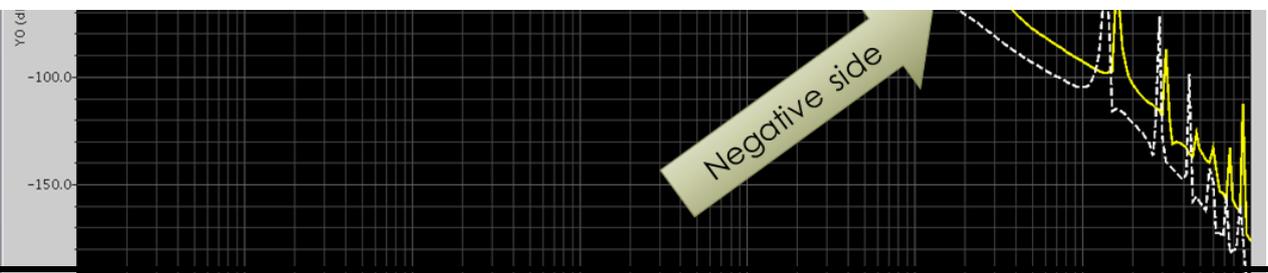
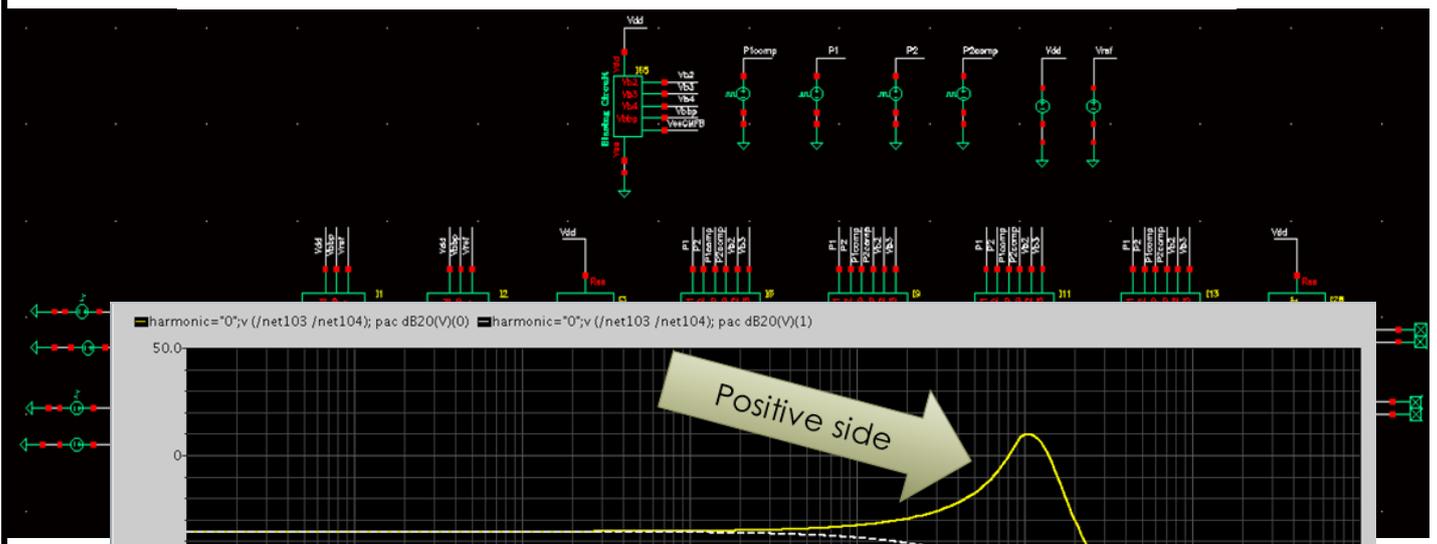
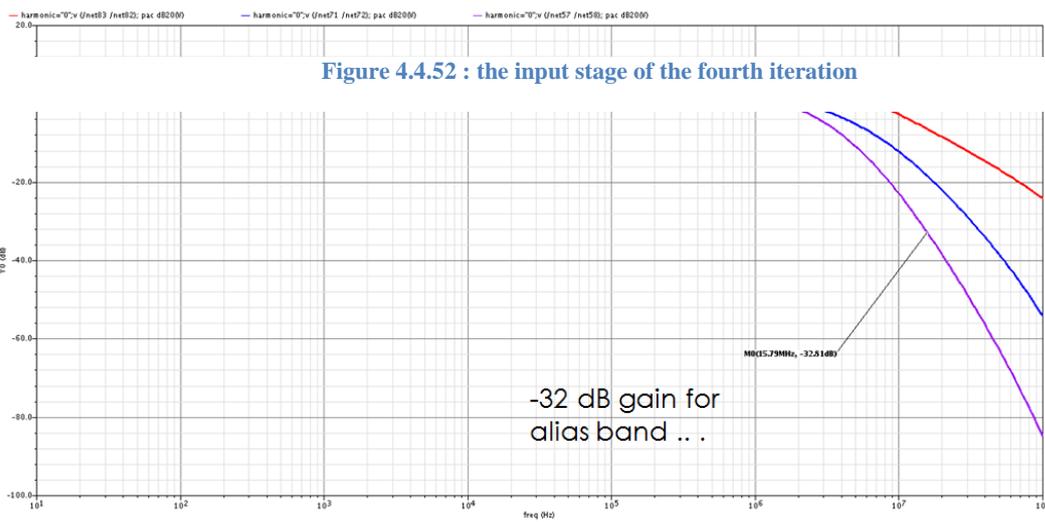
In this iteration , there many modifications done to solve the problems encountered before , like...

- ▶ The input stage is done in a dramatically new methodology to eliminate the problems appeared in the previous iteration , that is . . .
 - To make a gain DESENSETIZATION , i.e. to make the gain not to depend on gm/gm , we must use a FEEDBACK ,
 - To use feedback , we must do an OTA that have a relatively large gain to make $(A / (1 + AB))$ to be as close as possible to $(1/B)$, a 40 dB is far enough .
 - Then this OTA is done and its CMFB too , it have a gain of 38 dB and BW of approx 6 MHz and draws only a 40 uA . (See next page)
 - Also , a large cap in its output is to make it as an Gm-C filter , giving more attenuation for the alias band .
- ▶ For the problem of the alias frequency bands , the input stage is made to have two stages of this new OTA done with its feed back and the large Cap in it output (Gm-C filter , and also the AA filt. Made before , these three levels input stage gives more attenuation for the alias band (up to - 30 dB gain)
- ▶ For the noise , this input stage is not very noisy . .
- ▶ For the $iip3$ planning , the gain is not concentrated in the first stage but distributed on the first two stages and the filter stages also . . .
- ▶ The output stage is needed by OMAR ABD EL-RAHMAN to make it tit to its system , he asked to take my previous design and to make the needed modifications .
- ▶ For the power consumption issue , as the noise problem is found to be not the dominant one , a new OTAs is used in the filter , they all made to consume a 60 uA current .
- ▶ This new OTA is made to have a 68 dB gain with BW of 60 KHz and a PM of 72 degree , its CMFB is stable with Loop PM of 78 degree .
- ▶ These values shown to be good enough for the operation in the SC filter .
- ▶ Then , the consumed current now is $70*8 + 50*4 = 760$ uA .
- ▶ I have a budget of 140 uA , that may be used for the output stage (as the power budget is 900u)

The new made input stage is shown clearly here below



And the response of this input stage and the all A.A. filter is shown clearly below . . .



And the results obtained from this iteration are shown here clearly . . .

The main problems encountered in this iteration are . . .

- ▶ The attenuation given by the A.A. filter was not enough to get the required one by the blocker profile , so more stages are needed .
- ▶ The clock frequency used here was 15 MHz , which is not as the one of the PLL (26 MHz) or even a number of its divisible , and then you will not able to get this frequency in the chip , and now it a must to use on of the two frequencies (26 or 13 MHz) .
- ▶ The main problem also appear when corners were run , the core OTA designed in this iteration didn't pass corners and then a new methodology in designing the OTAs are employed to get an OTA that is capable of passing corners (explained in detail in the next section) .

These are the failed iterations , I presented it here in a brief for the purpose of listing the main tradeoffs and the solutions that I made to solve it , I didn't mention the detailed analysis or the all results in the time domain of the other parameters of these iterations , so as to make it possible to do so in the final (adopted) iteration , which is the one of our project in reality (next section) .

4.5 The filter design

4.5.1 New modifications from last iteration

There many modifications that must be done from the last iteration so as to solve the problems encountered there , and to enhance the operation of the filter itself , All of the following points will be mentioned in detail in the sections that explain each block of the filter.

- ▶ In the last iteration , the input stage was made by the means of R/R amplifiers , these amplifiers was shown to be very NOISY , due to these two Resistors that spill their noise on the input directly , and we may remember that the main reason for adopting this amplifier was the variation of gm/gm value across the corners and PVT , this problem is solved in this iteration by the means of the CONSTANT GM CELL , that is In the filter , the input stage is done using the gm/gm amplifier (which is not NOISY) , and in the same time , the constant gm cell will solve the problem of the variability of the gain across PVT .
- ▶ This stage is proved to pass all of the possible corners .
- ▶ The A.A. filter is done using a four stages of this new gm/gm done amplifier , so that the aliasing band can take enough rejection to meet with given blocker profile .
- ▶ A new core OTA is done , rather than the one used in the previous iterations , the new one absorbs less current and gives lower noise and good results
- ▶ And the main advantage of this new core OTA is that it is designed carefully to pass the corners .

- ▶ Most of the available current is concentrated in the first input stage , so it have the lowest noise , and then the noise of all of the filter will be substantially lowered .
- ▶ The clock of the filter now is 13 MHz , which can be easily taken now from the PLL frequency (26 MHz) after dividing it by two .
- ▶ The values of the capacitors of the filter is further increased so as to have lower KT/C noise , and to be far from the mismatch problem (from Monti-carlo simulations shown later) .

4.5.2 The general structure of the filter .

For all of the S.C. filters , the general block diagram will contain the following blocks , . . .

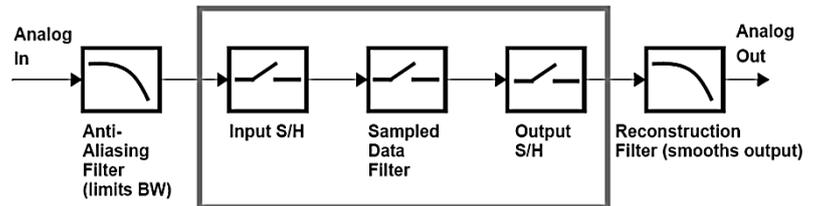


Figure 4.4.56 : the anatomy of the S.C. filters

- ▶ The Anti-aliasing (AA) filter : this is a filter that is responsible for eliminating (or attenuating) the alias bands arises in the S.C. filter , these bands appears due to the fact that the S.C. depend on sampling the signal , then if a higher frequency signal get the values of the lower frequency signal I the moments of SAMPLING , then the filter will be *blind* for these signals and will interpret it as the lower frequency signal .
- ▶ The core S.C. filter : it is the filter circuit itself which contains the S.C. integrators and so on
- ▶ The reconstruction filter : it the fiter responsible for smoothing the signal that emerges from the core filter which is in the form like the stair case wave form due to the *clocked* nature of the S.C. filters .

4.5.3 The AA filter design

The AA filter is done as mentioned before by the gm/gm amplifiers that employs the *constant Gm cells* , then its gain will not differ a lot with the corners , and a relatively large capacitors is introduced at their outputs so as to make it like the Gm-C filters , and then we can make the limited BW of the AA filter needed , the AA filter is done using four stages of the gm/gm amplifiers (Gm-C filters in reality) to get the needed attenuation at the Alias bands as stated by the blocker profile , for the noise considerations , the gm/gm amplifiers involved in the AA design is separated into to ones , the first one at the input of the AA filter has the majority of the gain of the filter, for the noise enhancement , and the other three stages is done by a gm/gm amplifier that has larger noise but no gain , it is contributing in the attenuating of the Alias bands only , let's check these two amplifiers .

4.5.3.1 he input stage of the AA filter.

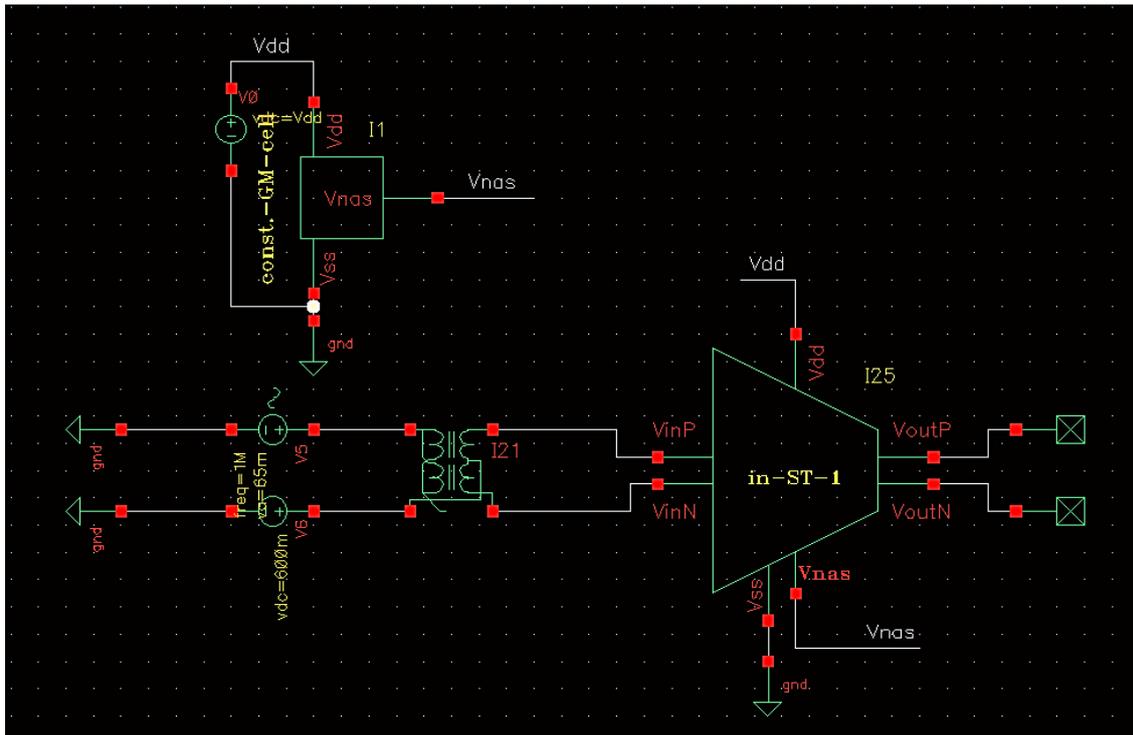


Figure 4.4.59 : the test bench of the input stage of the AA filter

The gain of this amplifier in the typical case and across the corners is shown here below clearly .

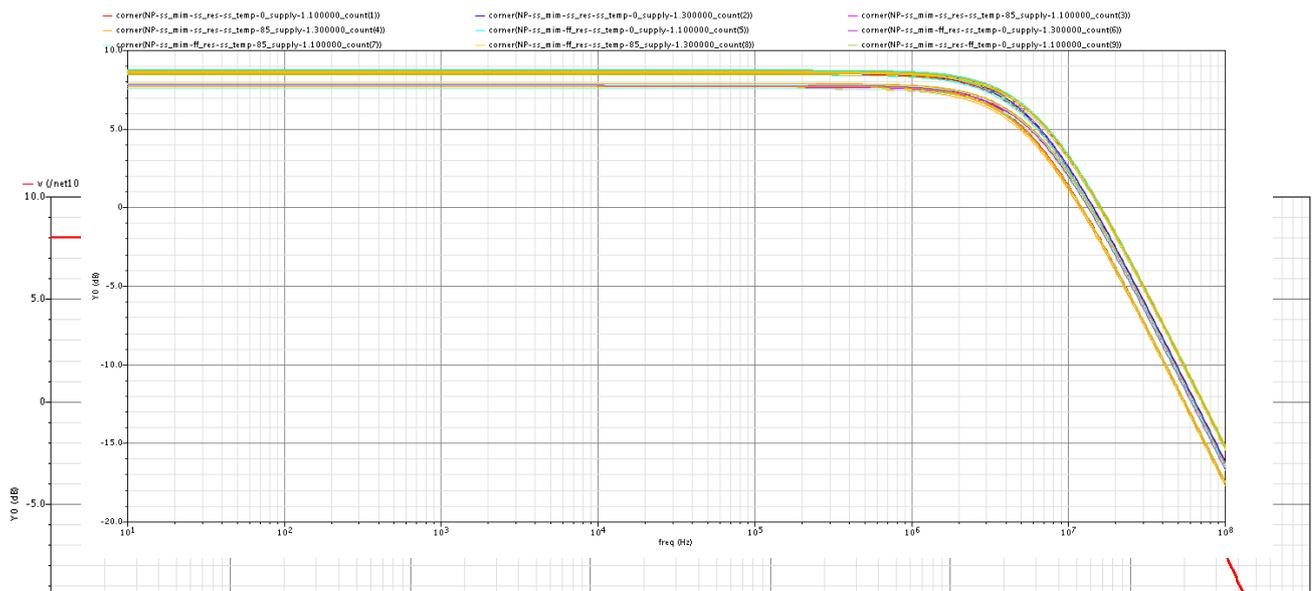


Figure 4.4.60 : the gain of the input stage of the AA filter

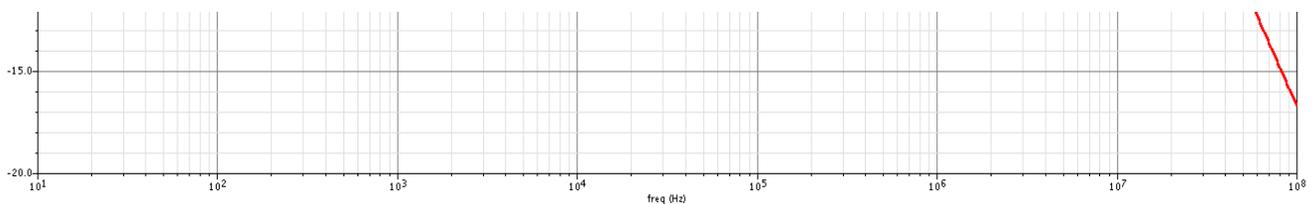
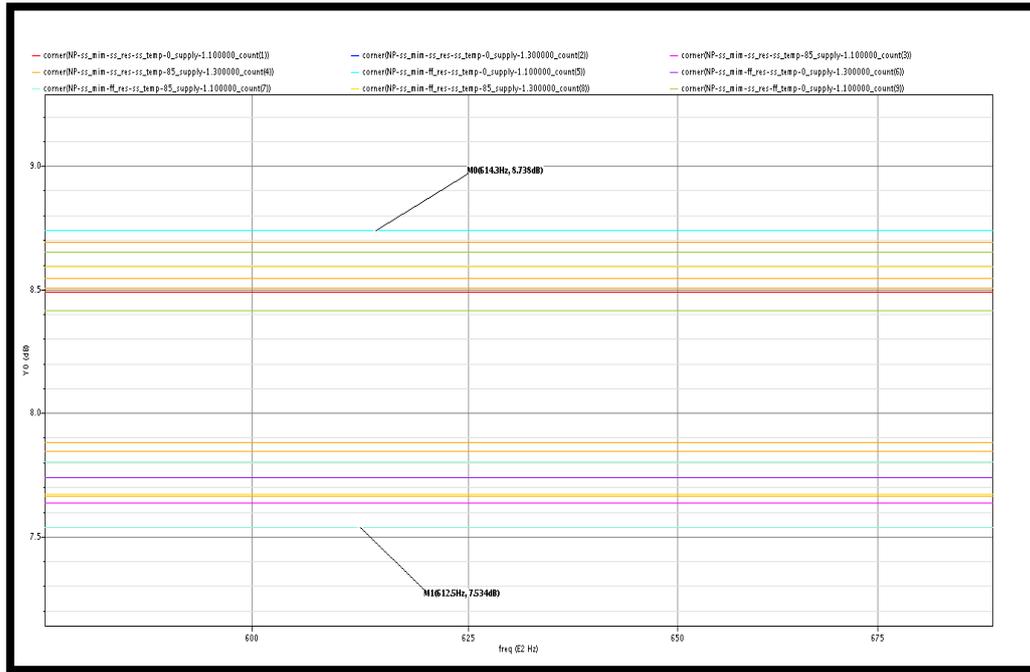


Figure 4.4.61 : gain variation across corners in about 1 dB which is satisfactory enough



The noise achieved by

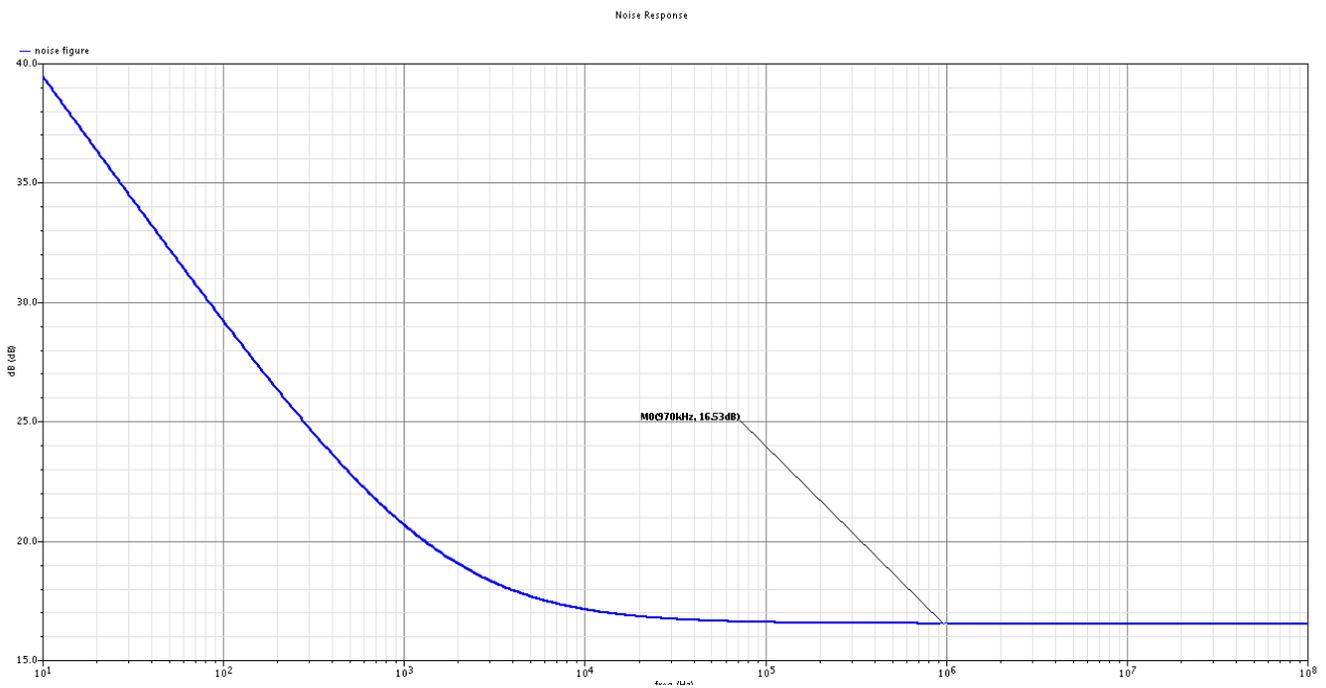


Figure 4.4.62 : the noise figure of the input stage of the AA filter .

this stage is 16.5 dB which is fair enough for the filter system design , noise result is shown in the figure below .

The other stages of the AA filter .

The other stage of the AA filter is approximately identical with the one introduced in the previous section , but the main differences is that the one here consumes less current and have no gain , and of course it will be more noisy , but the gain of the previous stage will *kill* this noise in reality . . . this is the circuit of this amplifier (including its DC solution) . . .

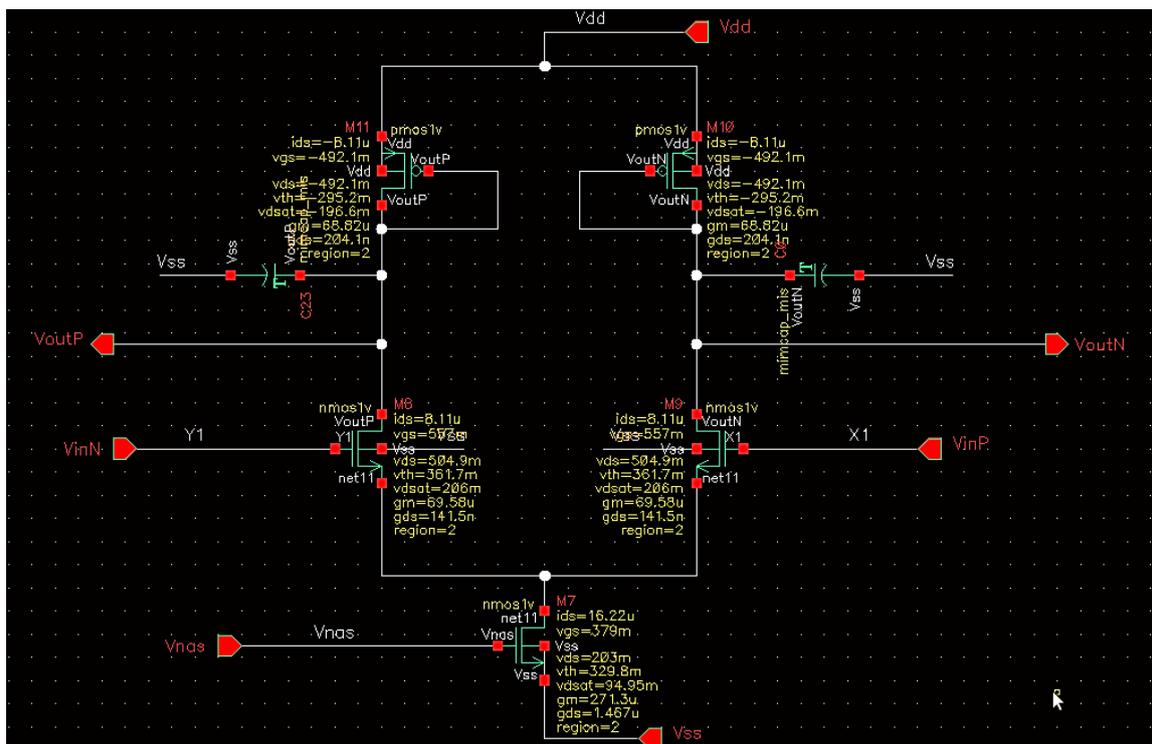


Figure 4.4.63 : the preceding stage of the AA filter

And this its gain in the typical and corners case

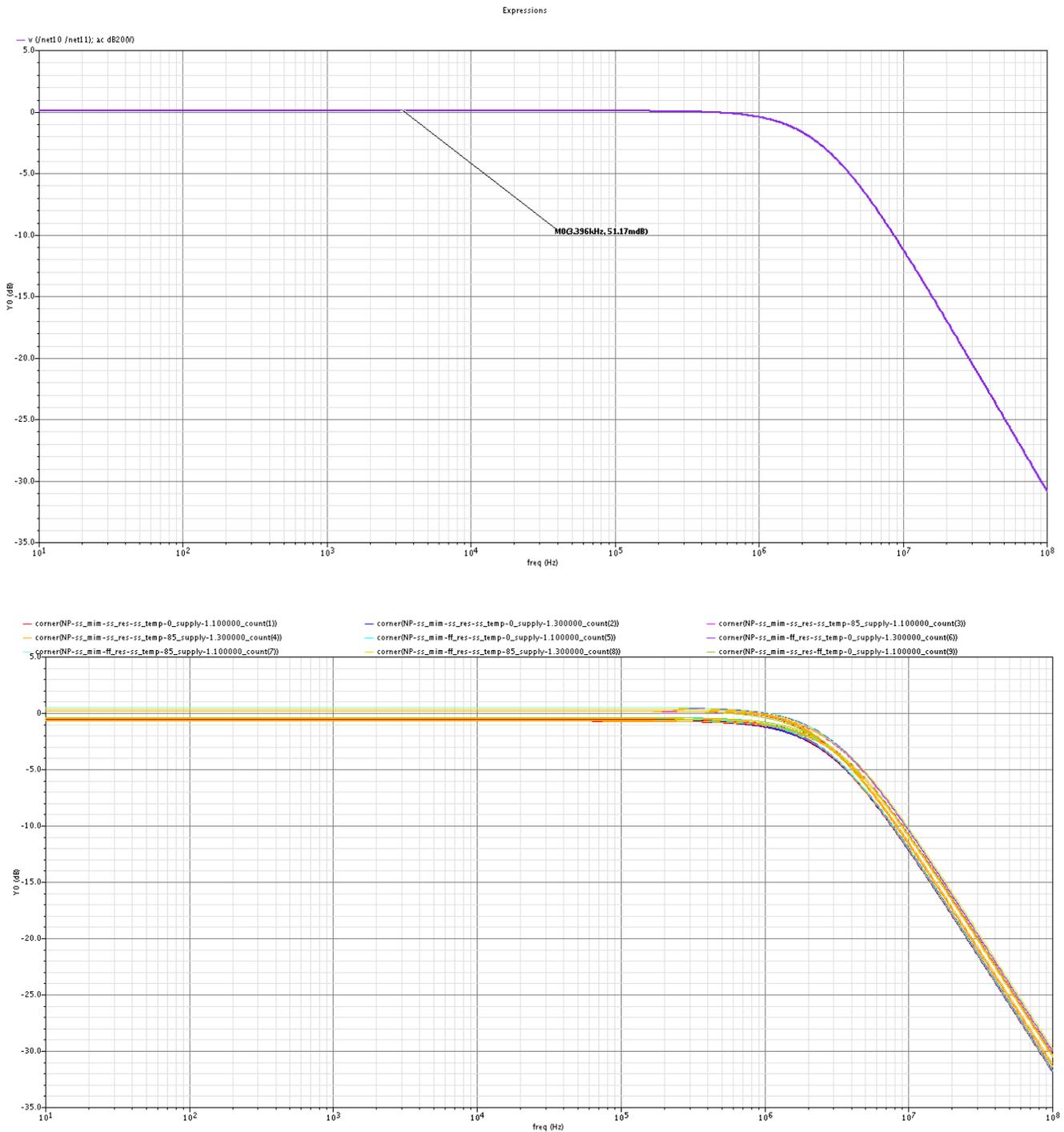


Figure 4.4.64 : the gain in the typical and corners for the proceeding stage of the AA filter

4.5.3.2 The AA filter as a one block .

These amplifiers are gathered in one symbol for being introduced in the filter circuit , and as mentioned before , the first one only is the one that have low noise and absorbs more current , and the

other ones are the ones have more noise but absorbs less current , the schematic of the AA filter is shown in the following figure (the I and Q channels). . . .

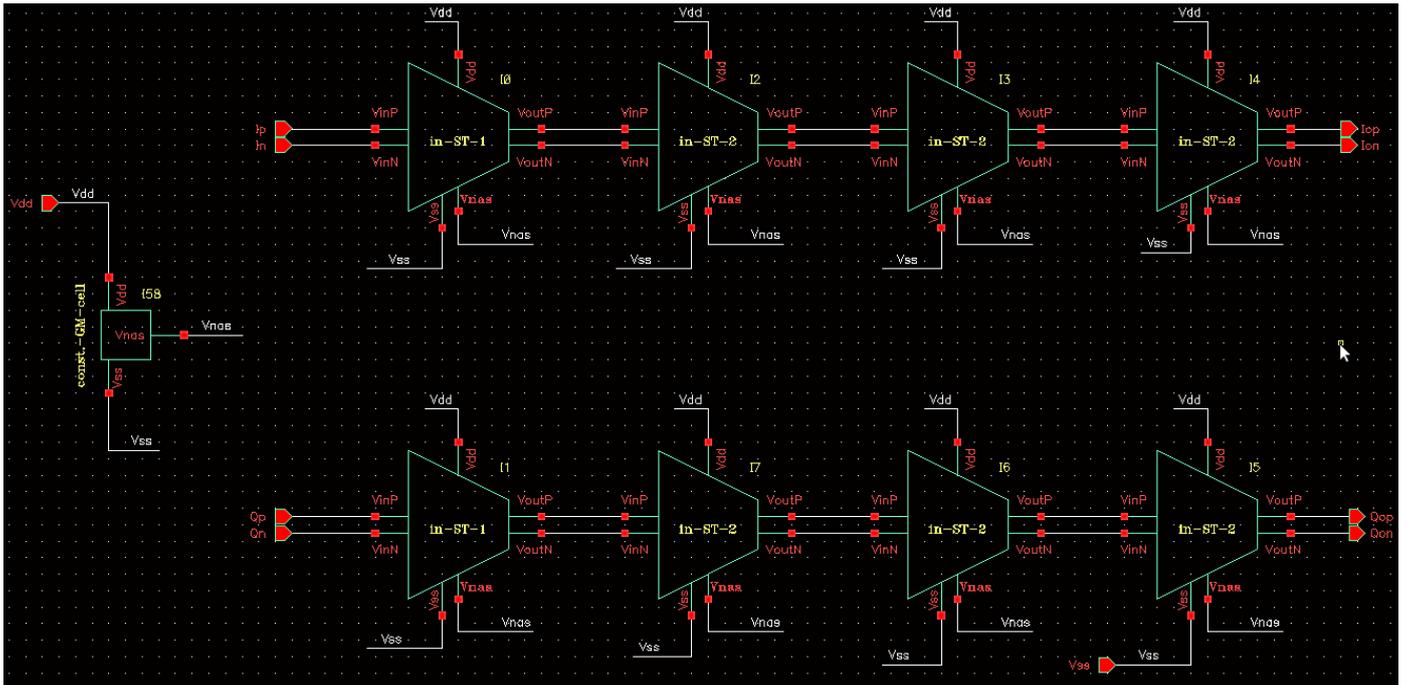


Figure 4.4.65 : the schematic of the AA filter .

the test bench used for testing the AA filter is shown here clearly below (including the symbol used for the AA filter)

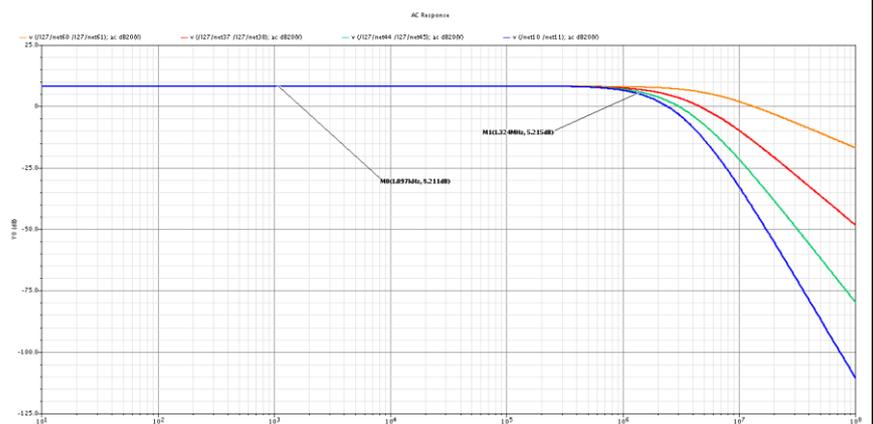
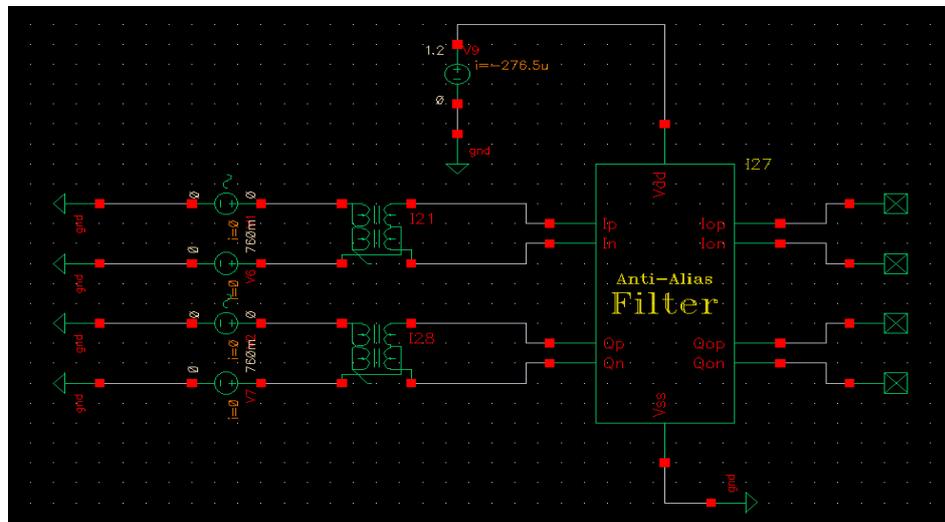


Figure 4.4.66 : the response of the AA filter after each stage of its stages

The response of the AA filter after each stage of its stages is shown in the figure here . . .

4.5.4 The core S.C. filter

The important parts of the core filter are the (1) core OTA , (2) the switch (3) the capacitor itself , the switch design was mentioned before in the section of “general issues” in detail , so I will not mention it again here , the main structure of the core filter is mentioned before in the general issues section , then the main part to be used is the core OTA design

4.5.4.1 The core OTA design .

The first OTA used in the previous failed iterations was the “ folded cascade ” , it gives good results in general , but there was catastrophic problems that was not to be solved easily with this OTA topology , that is ... the large noise , and the larger power consumption , then , I switched my thinking to another topology , it the simple OTA (diff. pair) that will be shown in detail , it really have a dramatically lower noise , and gives better SR and a good GBW and very good results in the time domain and of course it have a lower power consumption .

The circuit of the OTA used is shown here clearly , including its DC solution

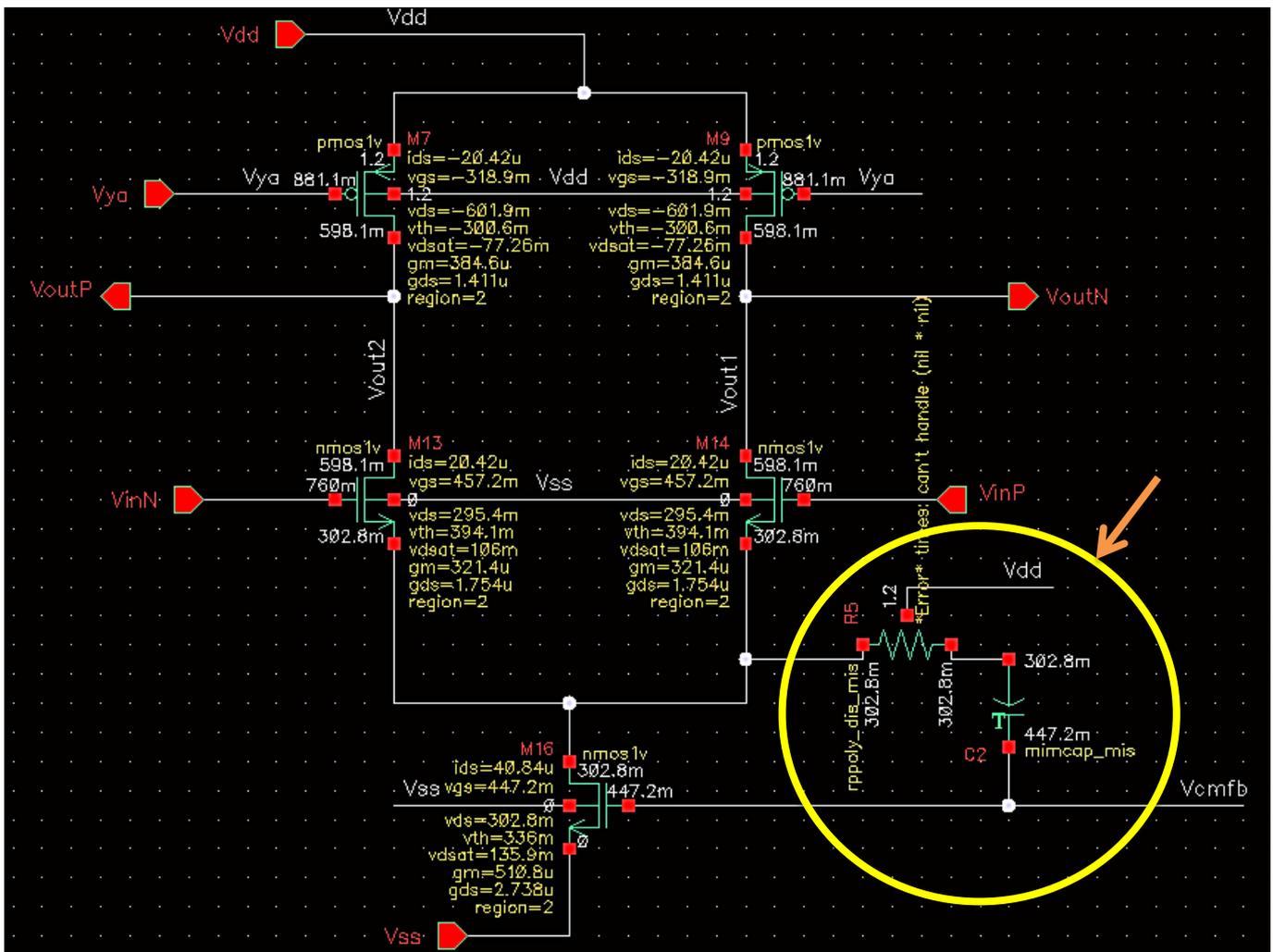


Figure 4.4.67 : the core OTA circuit

And the CMFB circuit used for this OTA is shown in the following figure

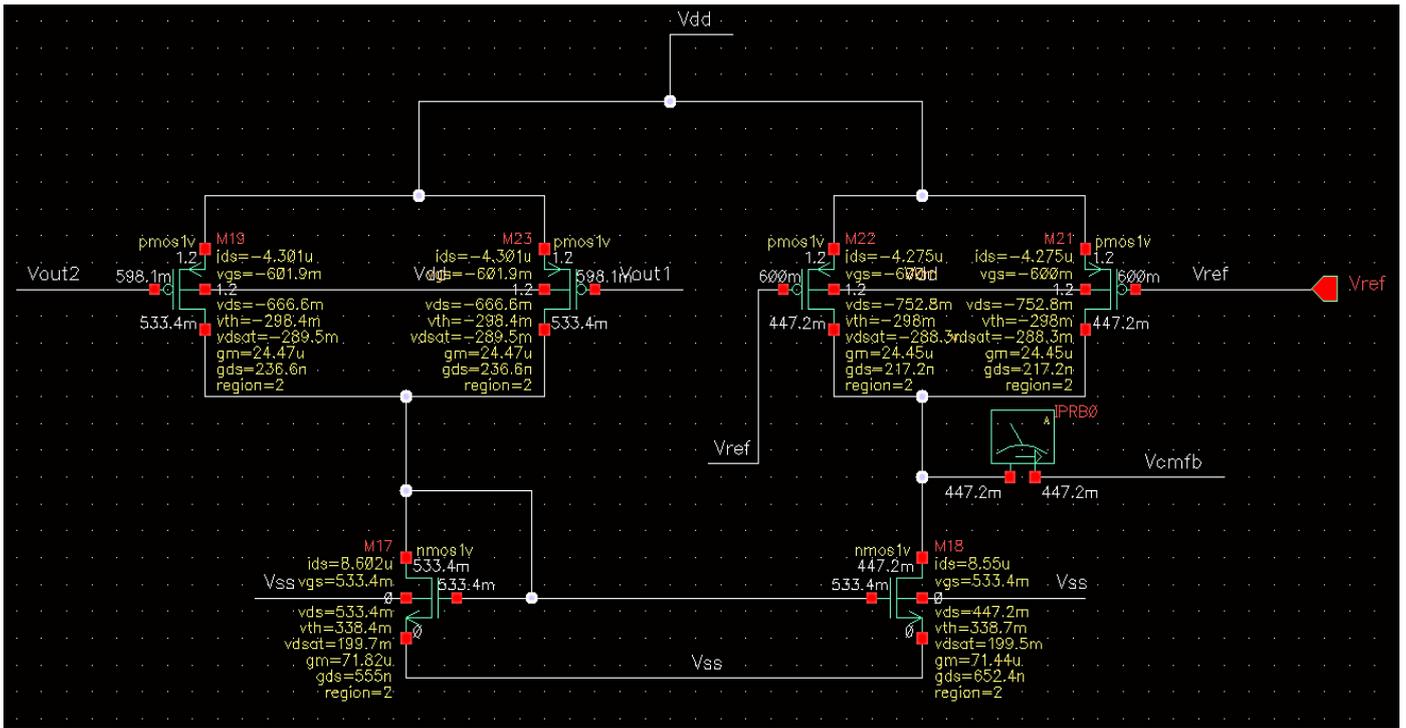


Figure 4.4.69 : the CMFB circuit used in the core OTA

Eventually , the biasing cell of this OTA is shown in the figure here . . .

Before presenting the parameters of this OTA , I must say an important thing firstly , in the circuit of the core OTA , you will see an circled part of it , this is due to the following , if you noticed the loop of the CMFB , you will find that there is two poles very close together , these two poles makes the stability of the CMFB loop to be very bad , then what to do in such a case ??? the best solution in such a cases is the *pole splitting* , this is done in our case by the mains of capacitor and the resistor encircled in the OTA circuit figure

These added cap and resistor is proved not to affect the gain of the OTA , and it enhances the stability of the CMFB even across all of the corners as shown in the figure below (CMFB loop gain and phase across corners after the pole splitting)....

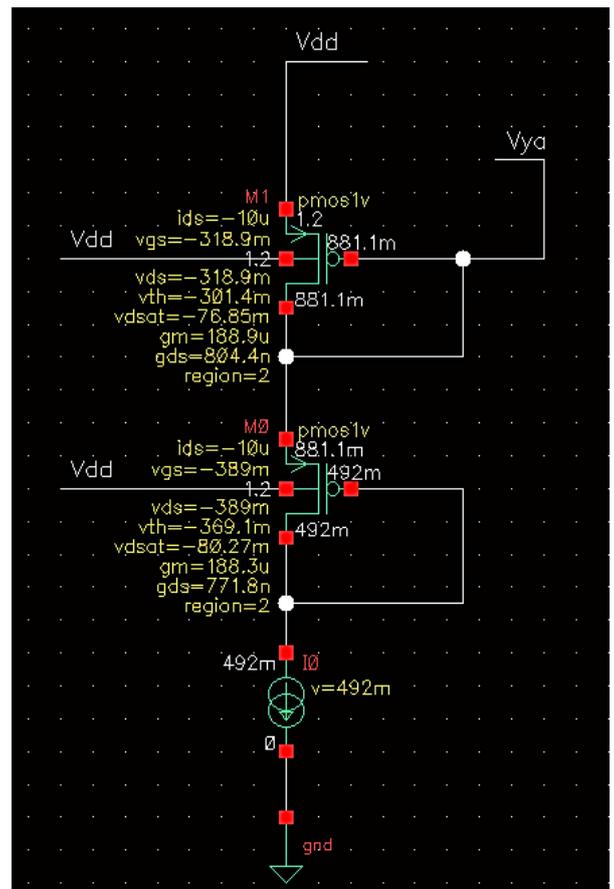
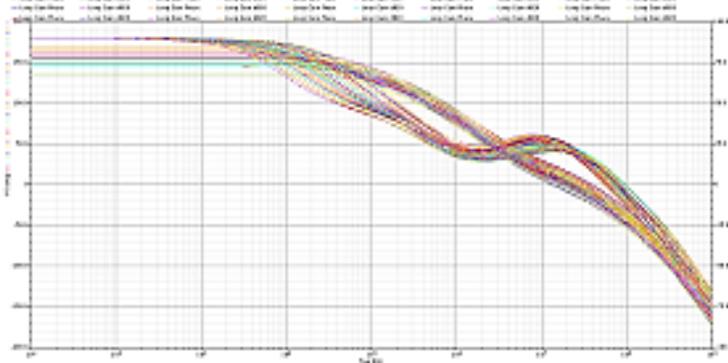


Figure 4.4.68 : the biasing cell of the core OTA



The gain of the core OTA is shown here clearly (typical and corners), we can see that the gain is 40 dB , and the BW is 1.51 MHz , and the GBW is 150 dB . . . these values is good for our purpose

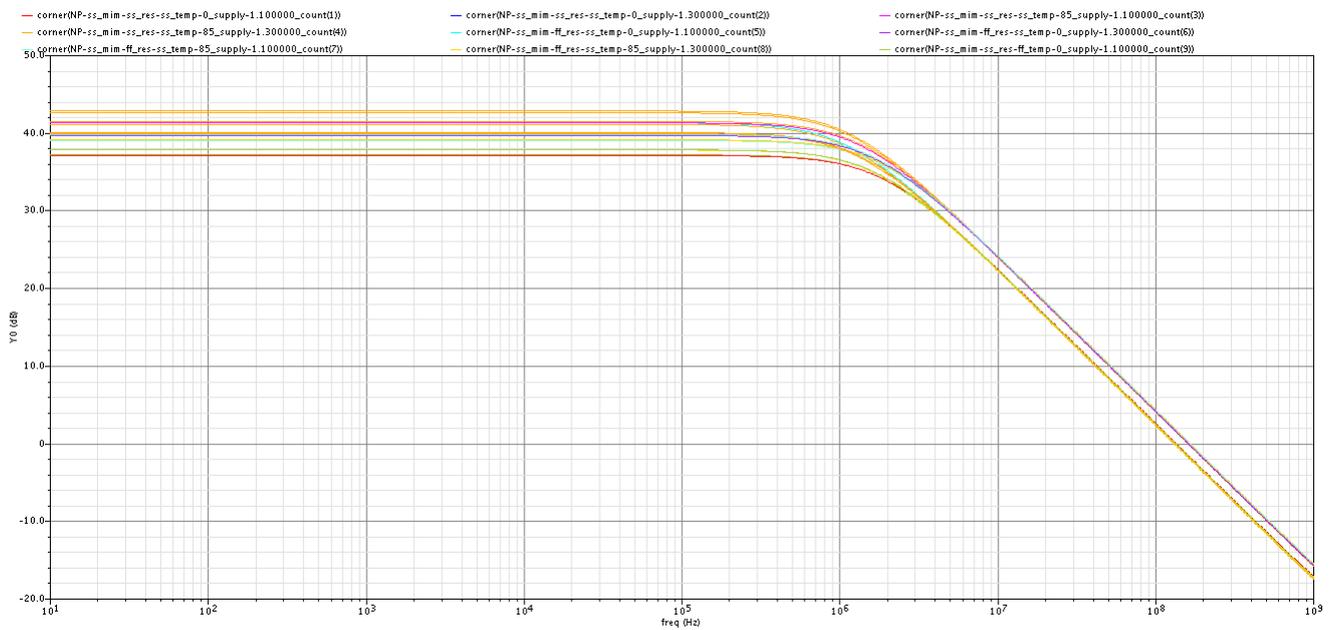
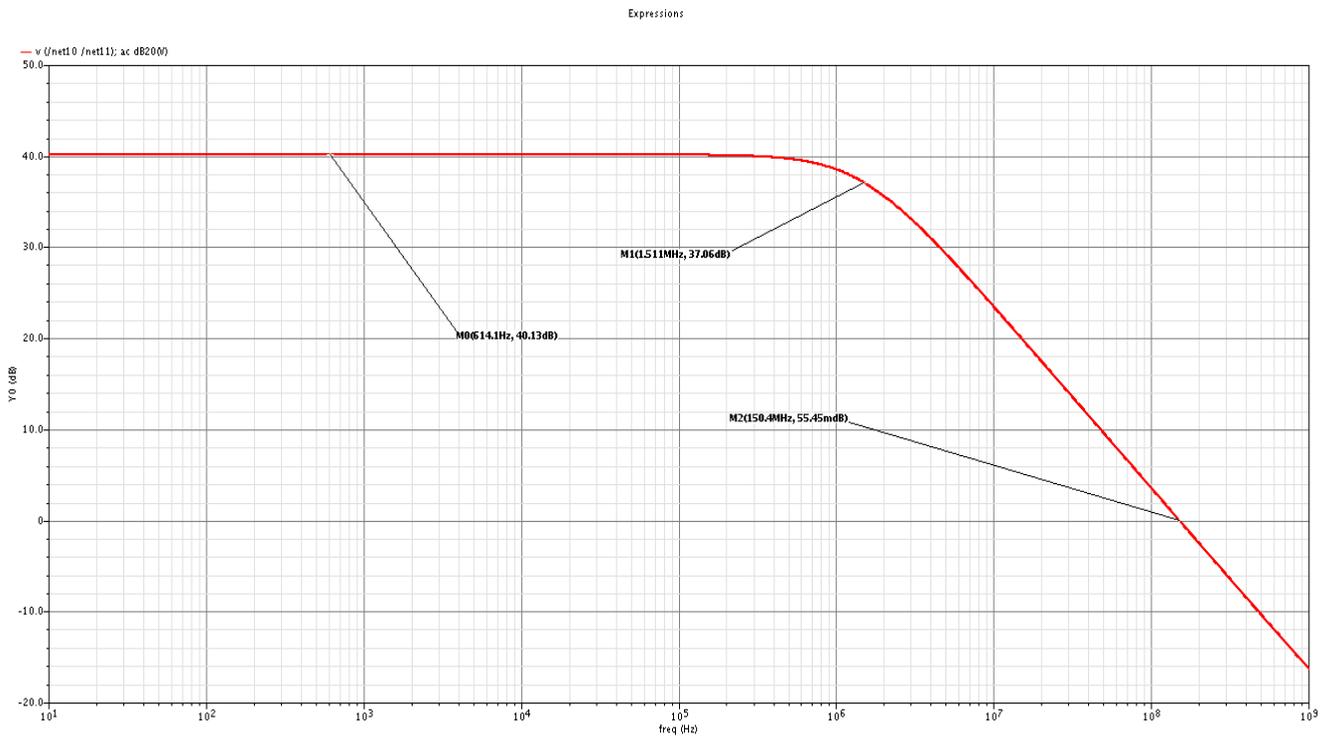


Figure 4.4.70 : the gain of the core OTA (typical and corners)

In the following figures , the CMRR and the PSRR is shown clearly , the values of them is shown to be very well , especially the PSRR which is important for rejecting the noise in the Vdd traces , it is shown that the CMRR is 275 dB , and the PMRR is 280 dB .

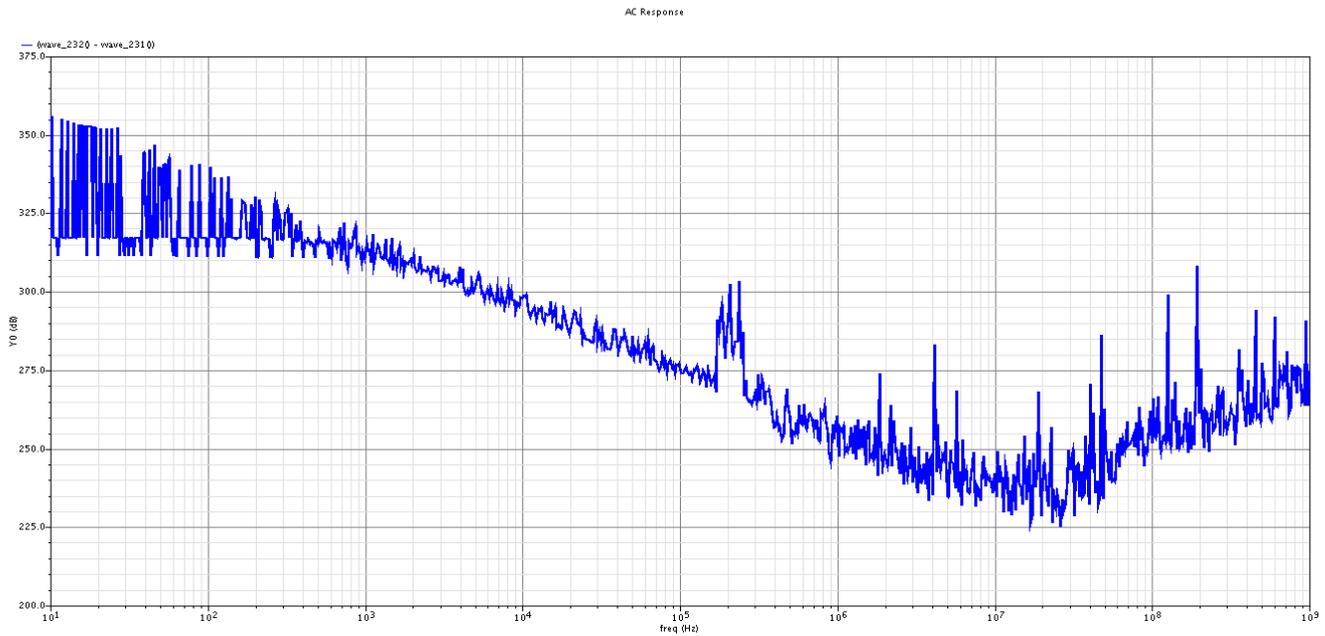


Figure 4.4.73 : the CMRR of the core OTA

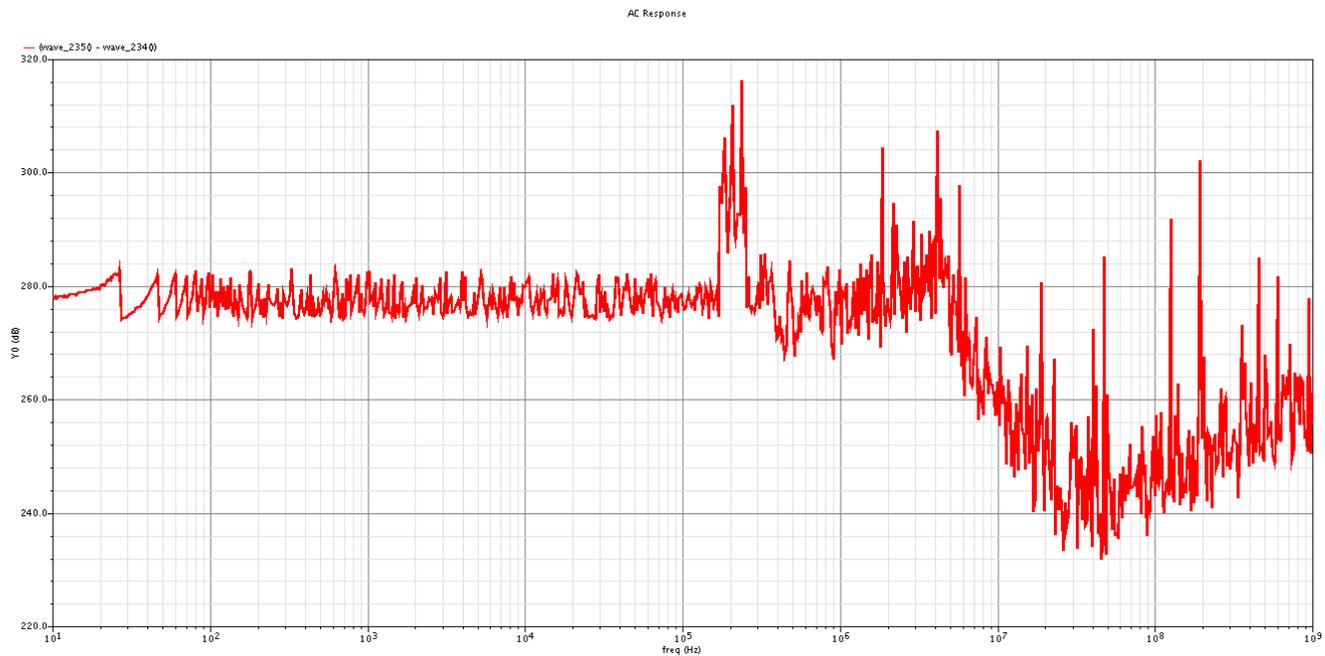
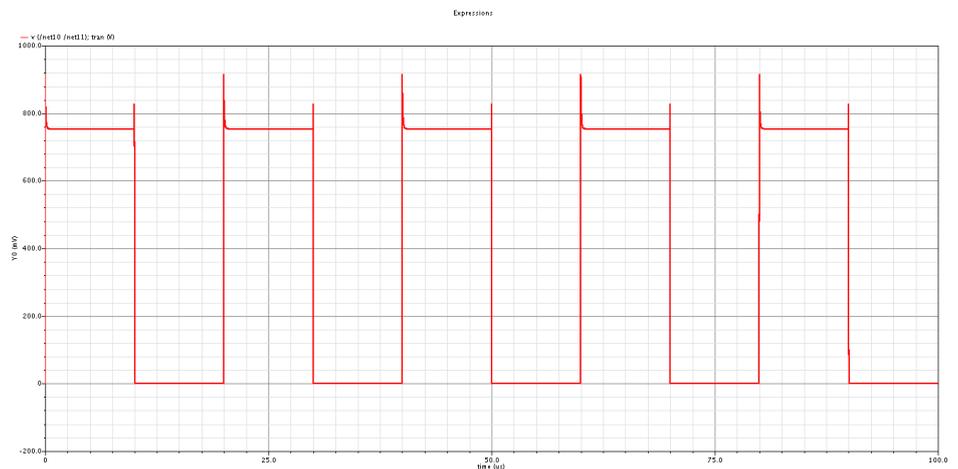


Figure 4.4.72 : the PSRR of the core OTA .

for measuring the SR of the core OTA , we can put a square wave in the input ,and then we can measure the time that is needed , the output of the OTA when the input is the square wave is shown in the figure here . . .



166
Figure 4.4.71 : the output of the square wave

Here is the zoomed picture , on which we can see that the SR is approximately 80 v/usec . . .

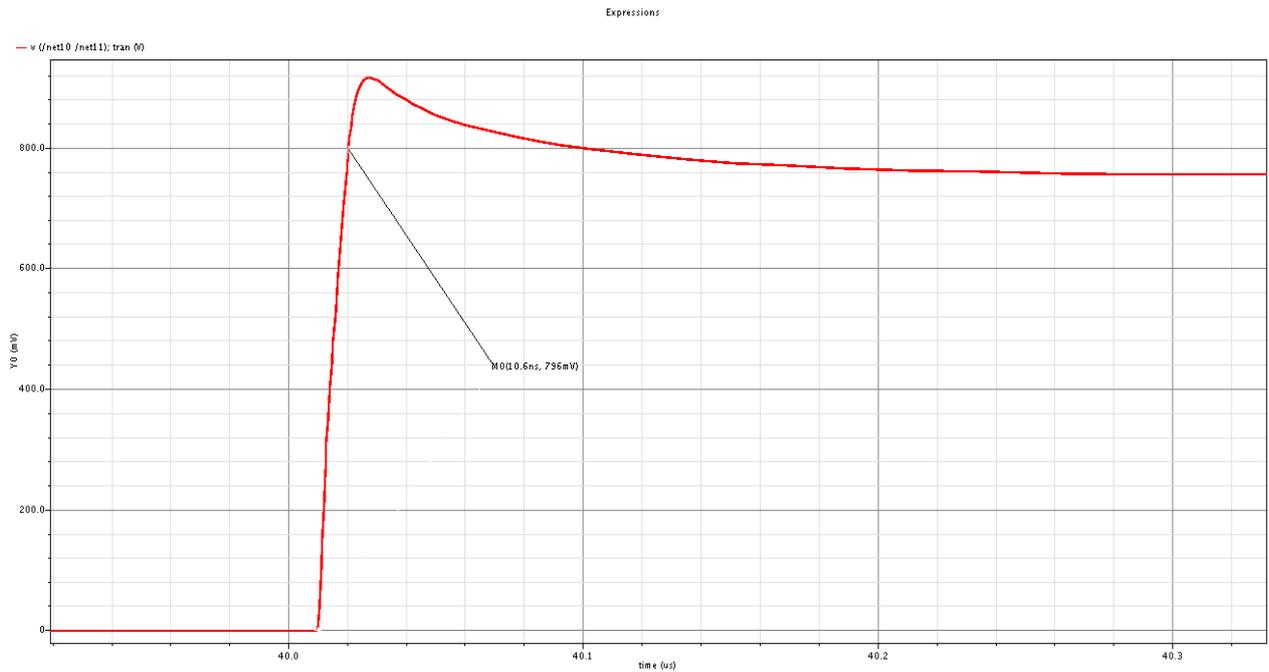


Figure 4.4.74 : measurement of the SR for the core OTA

The main features of this OTA is summarized in the following table

Table 4 : the parameters of the core OTA

Property	Value
Vdd	1.2
Current	40 uA
Gain	400 dB
BW	1.51Mhz
PM	89°
GBW	150 MHz
SR	80 v/usec
CMRR	275 dB
PSRR	280 dB
CMFB loop gain	50 dB
CMFB loop PM	40°

4.5.5 The reconstruction filter

The reconstruction filter used in this filter is a simple RC sections , its cut-off is of course located after the pass band of the filter . . . the circuit of the reconstruction filter is shown here clearly . . .

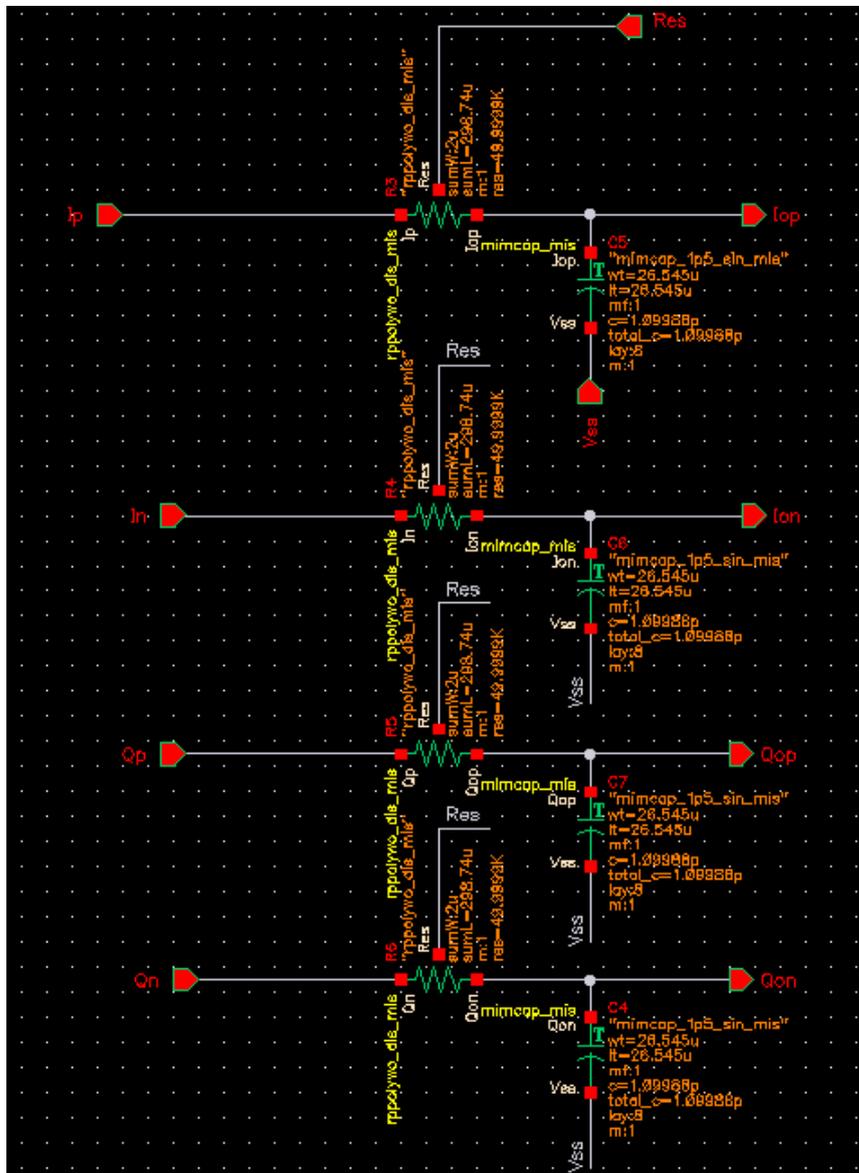


Figure 4.4.75 : the reconstruction filter

The main function of the reconstruction filter is to smooth the signal of course at the output . . .

4.5.6 The circuit of all of the filter

In the figure here , we can see the schematic used in the filter , hwe we can see the AA filter , the core filter , and eventually the reconstruction filter .

4.6 The results of the filter simulation

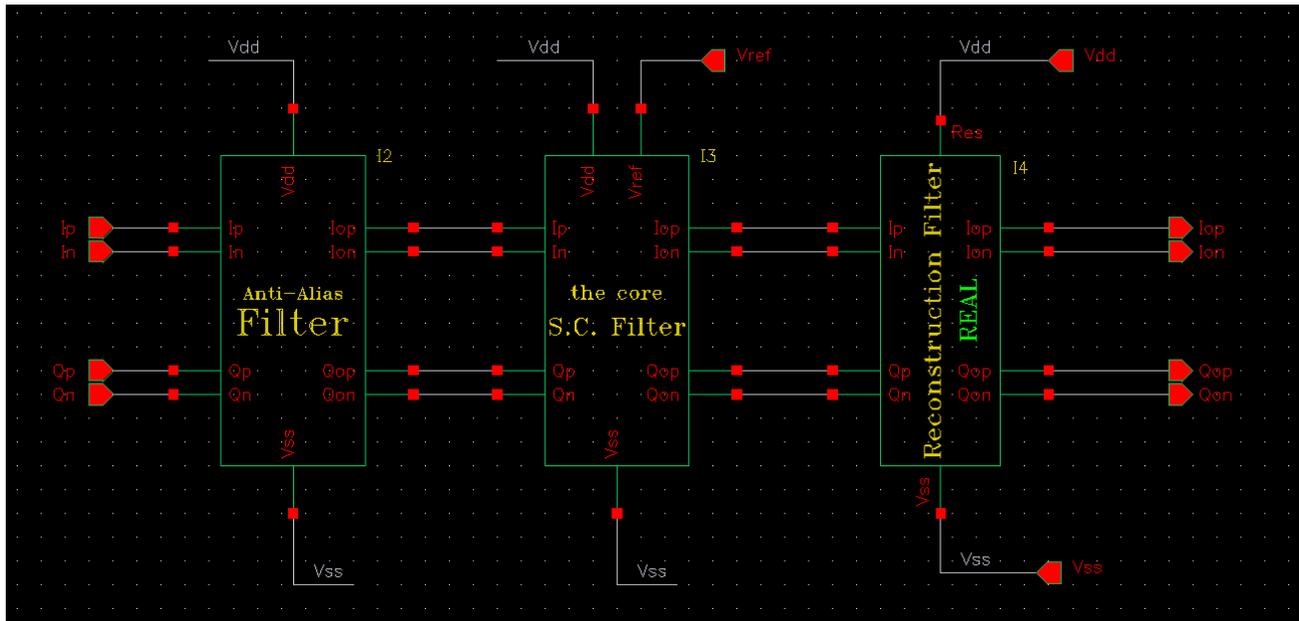


Figure 4.4.76 : the filter circuit (all)

4.6.1 The response (transfer function) of the filter

Here is the transfer function of the filter shown clearly in the figure below

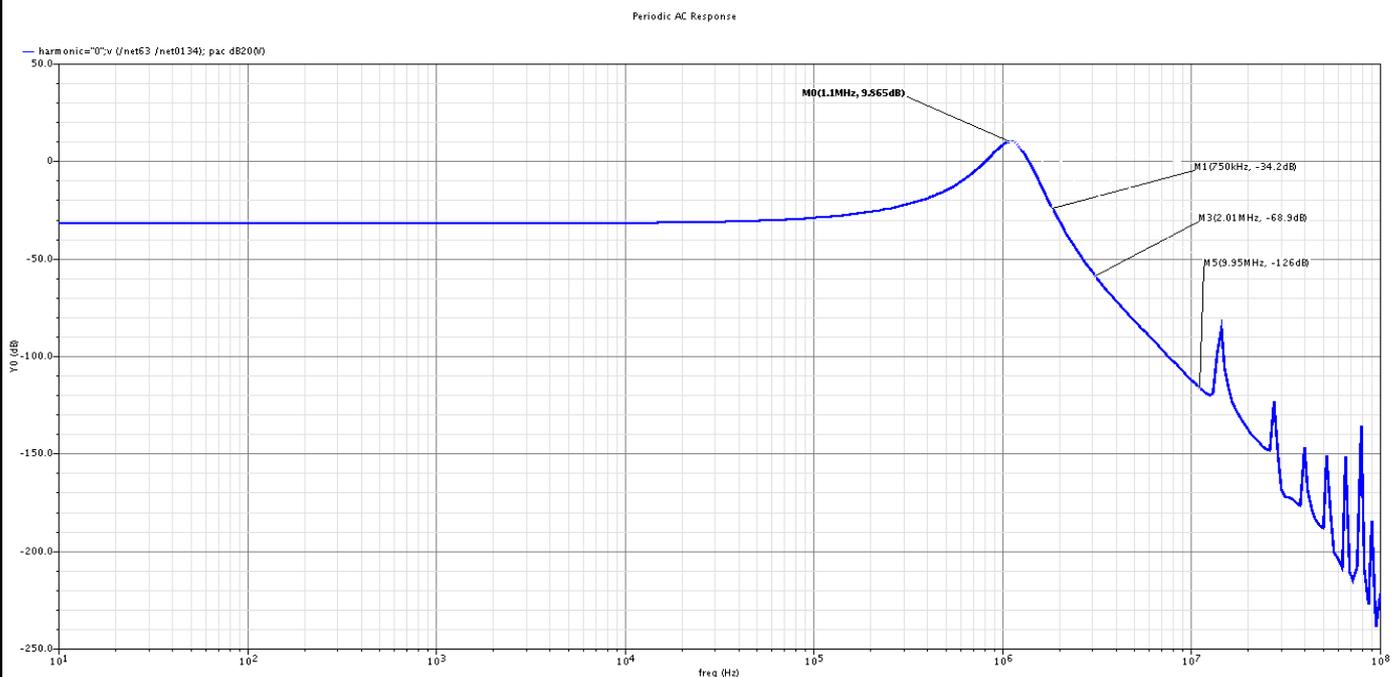


Figure 4.4.77 : the response of the filter and the rejections

In the previous figure , we can see the rejections that is obtained at the offsets determined by the blocker profile , all of the rejections needed are exceeded by this filter (is will be indicated clearly in the summery table) .

And the following figures indicates clearly the BW of the filter (280 khz) .

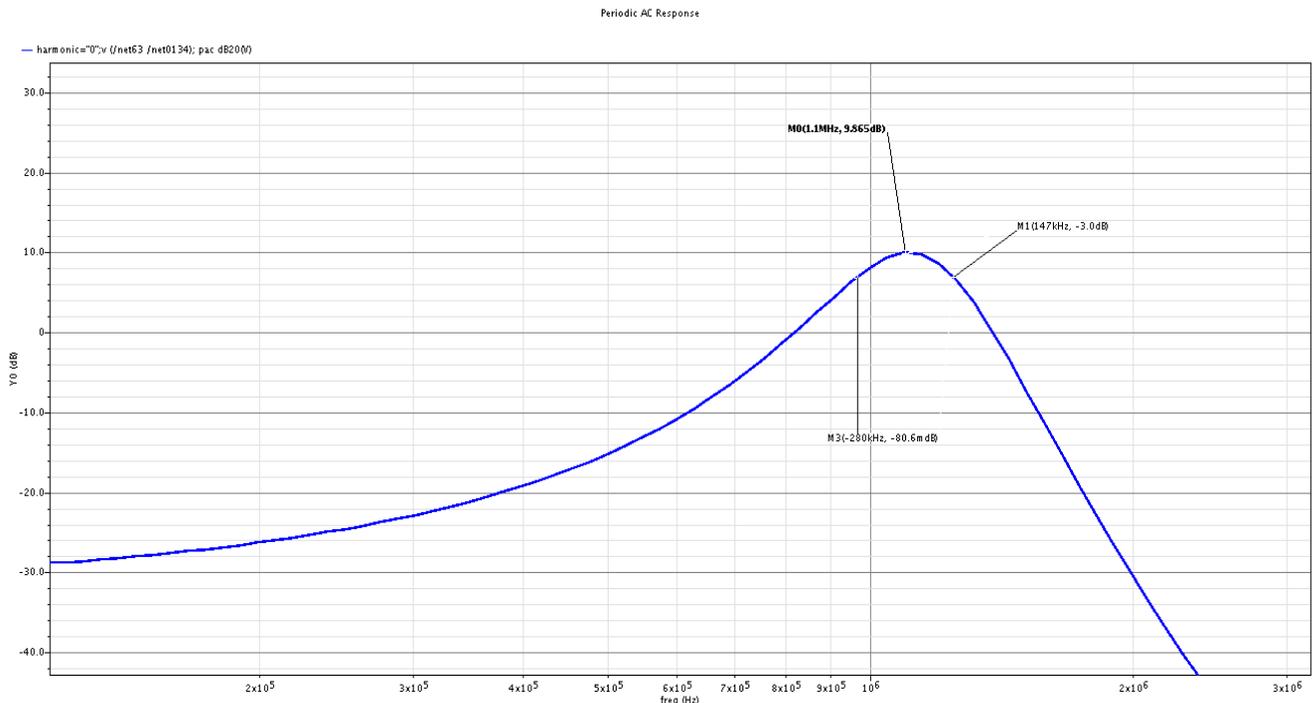


Figure 4.4.78 :the BW of the filter

And the following figure shows the image response of the filter and the IRR (60 dB “it may seem a substantial value but it is verified by the tie domain analysis that the IRR is 50.5 dB ”) . . .

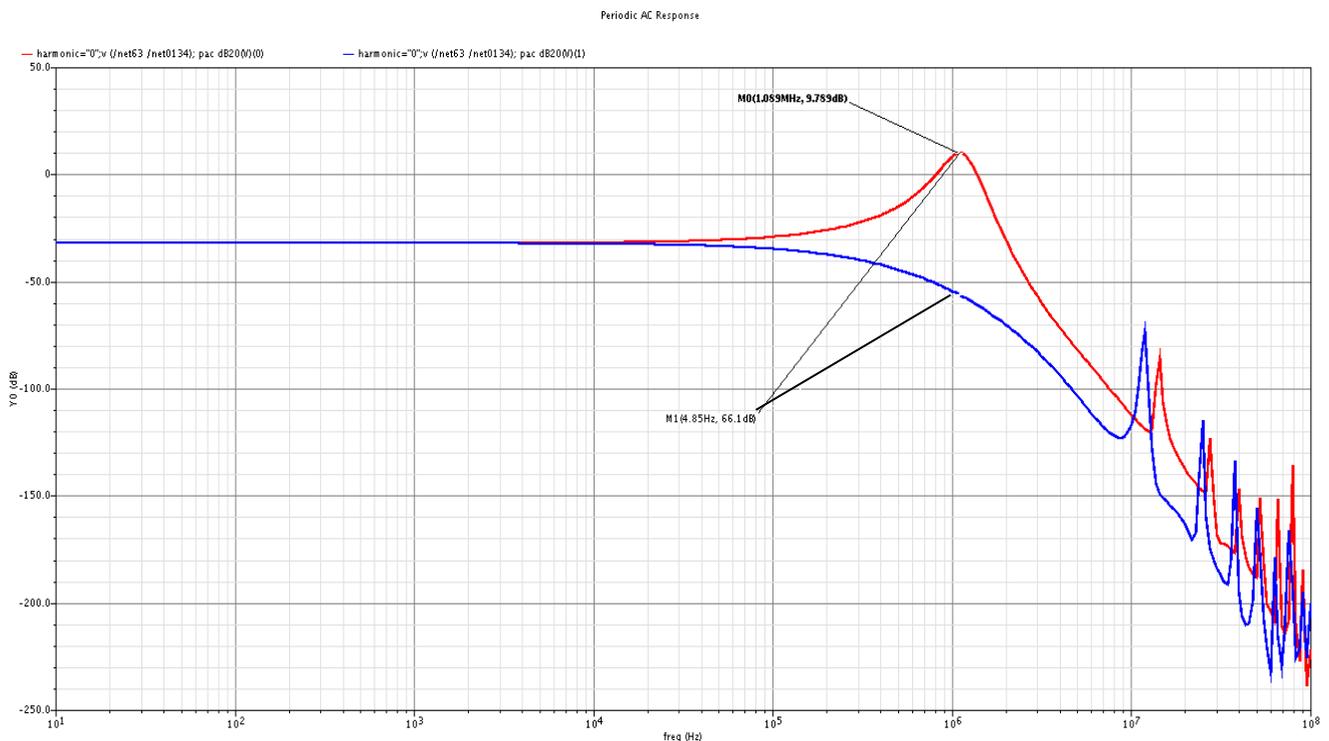


Figure 4.4.79 : the IRR and the image side response of the filte

And here , the figure is the phase response of the filter , is it very clear in this figure that the phase response in the pass band is fairly flat , and this was expected as the filter done in a Butterworth filter .

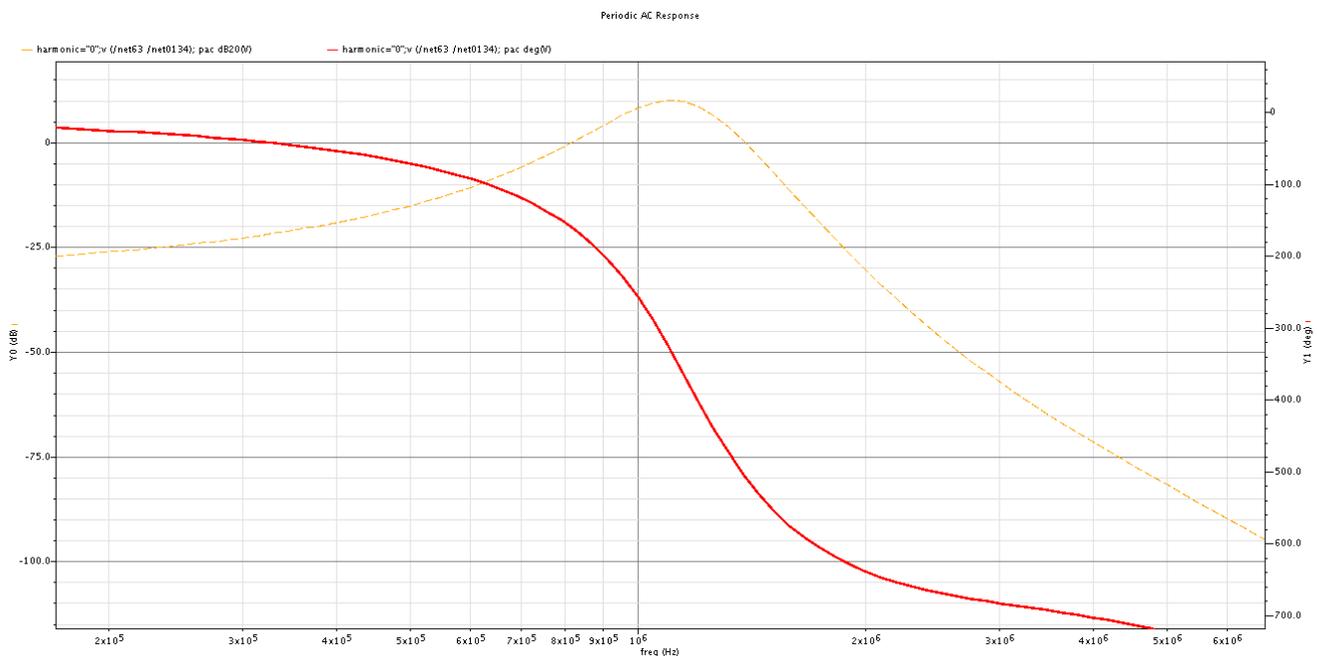
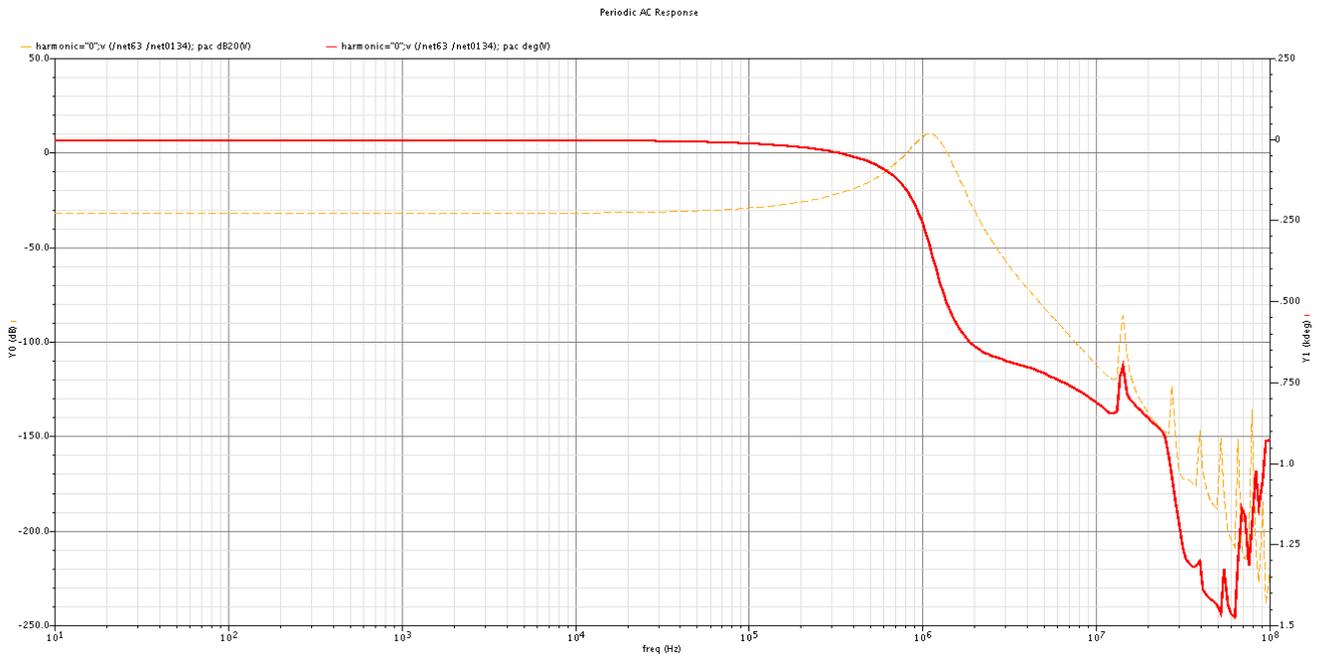


Figure 4.4.80 : the phase response of the filter

4.6.2 The time domain examples on the filter

In the following figures , I will represent some time domain examples , that will ensure the operation of the filter and make the idea of this filter to be more clear , in all of the following figures , the *dashed* orange curves are the input signal to the filter , and the *bold* curves is are the output of the filter at certain node ,the input signal amplitude in all of the cases is 65mV , which is the max input signal to me (according to the system designer) , to know the frequency of the input signal and whether it is in the positive or negative sides , you may read the description of each figure .

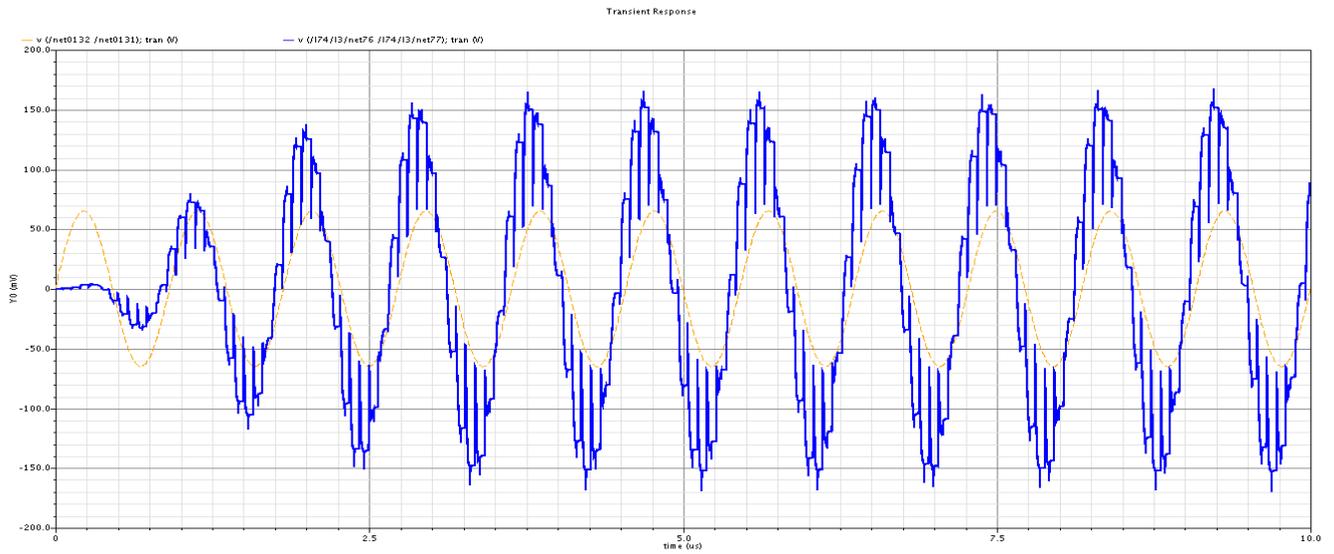


Figure 4.4.81 : the signal at IF frequency (1.1 MHz) at the positive frequency side in the middle of the S.C. core filter

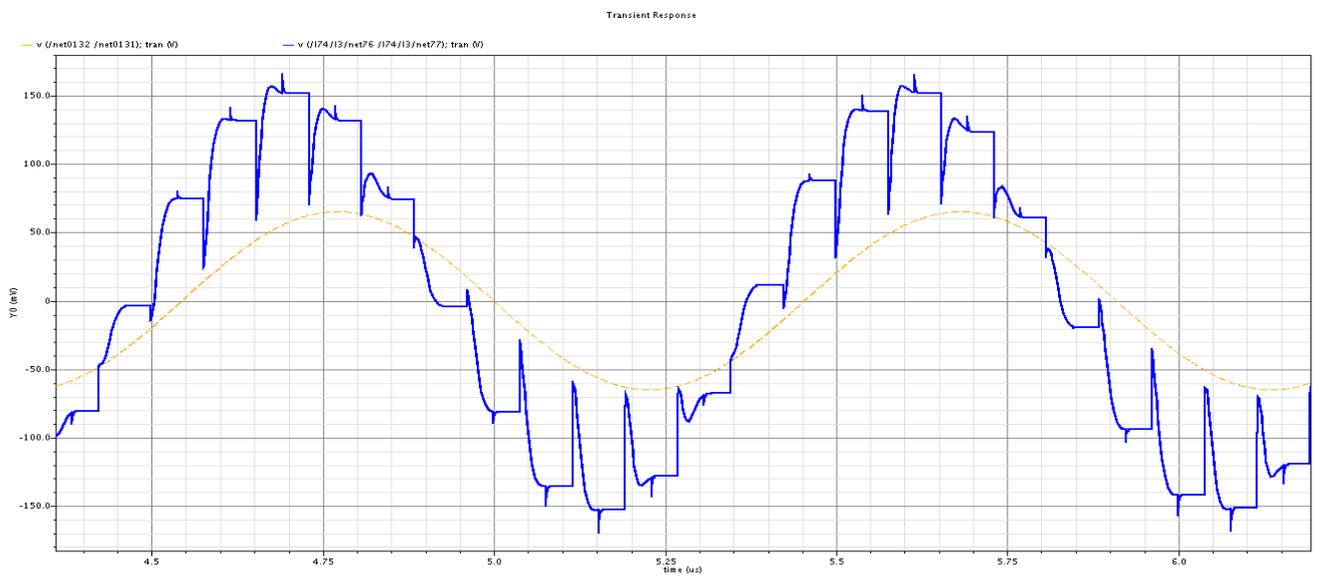


Figure 4.4.83: : the signal at IF frequency (1.1 MHz) at the positive frequency side in the middle of the S.C. core filter "zoomed"

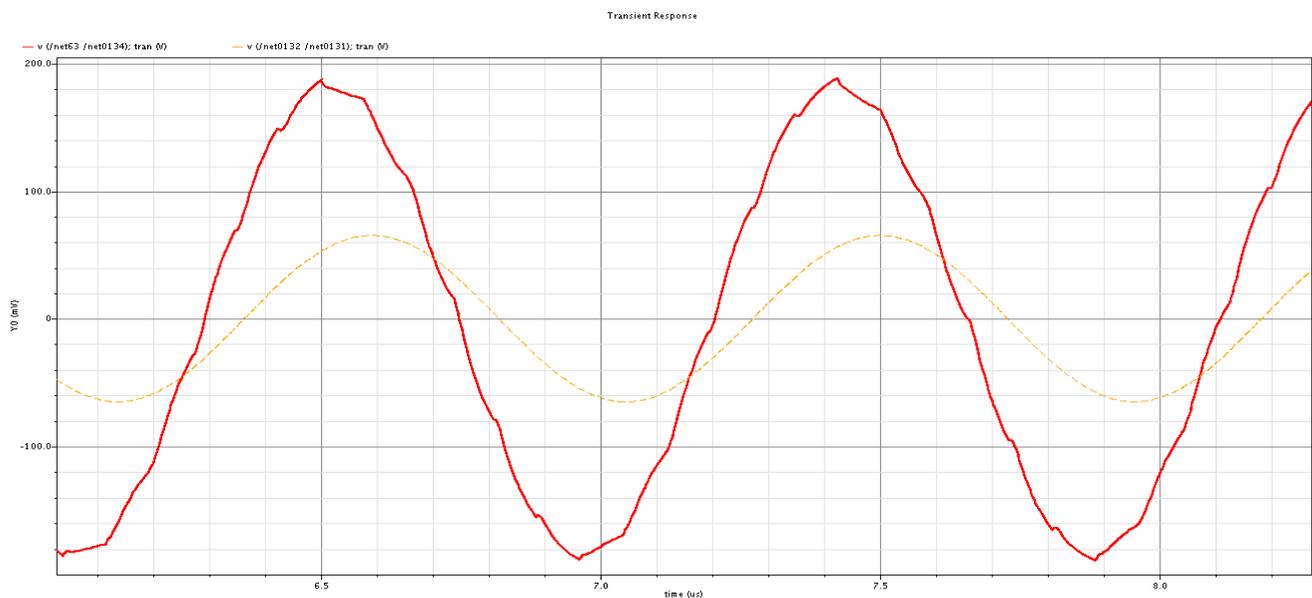


Figure 4.4.82 : : the signal at IF frequency (1.1 MHz) at the positive frequency side after the reconstruction filter

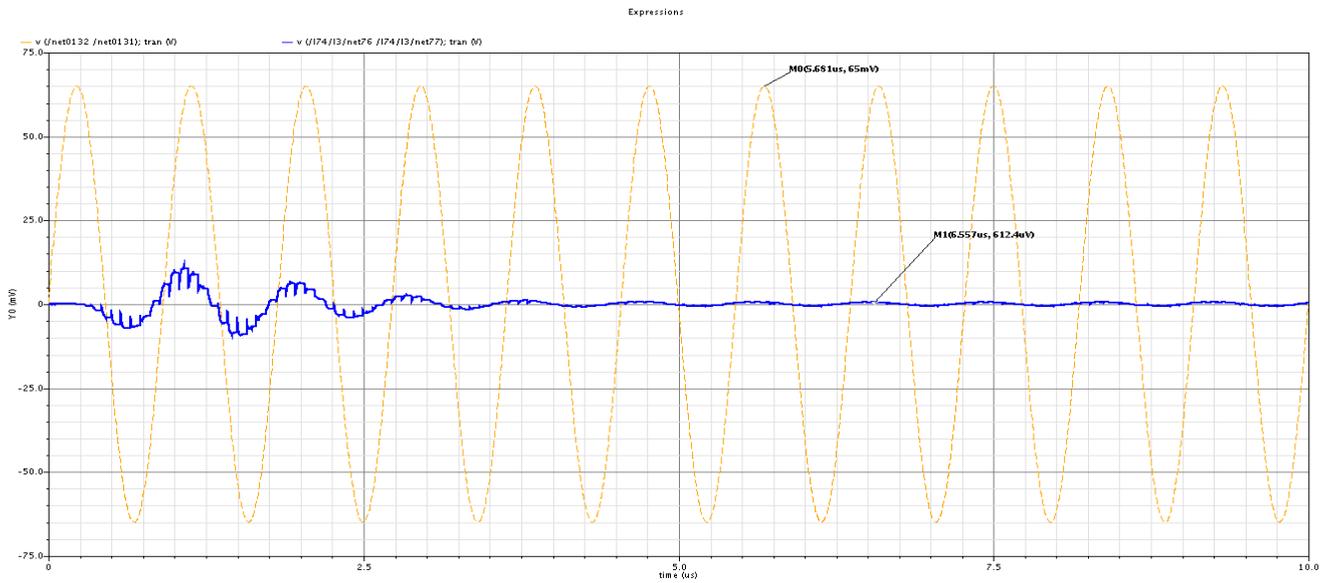


Figure 4.4.86 : the signal at IF frequency (1.1 MHz) at the negative frequency side (image) in the middle of the S.C. core filter “ IRR is shown to be -55 dB form this simulation ”

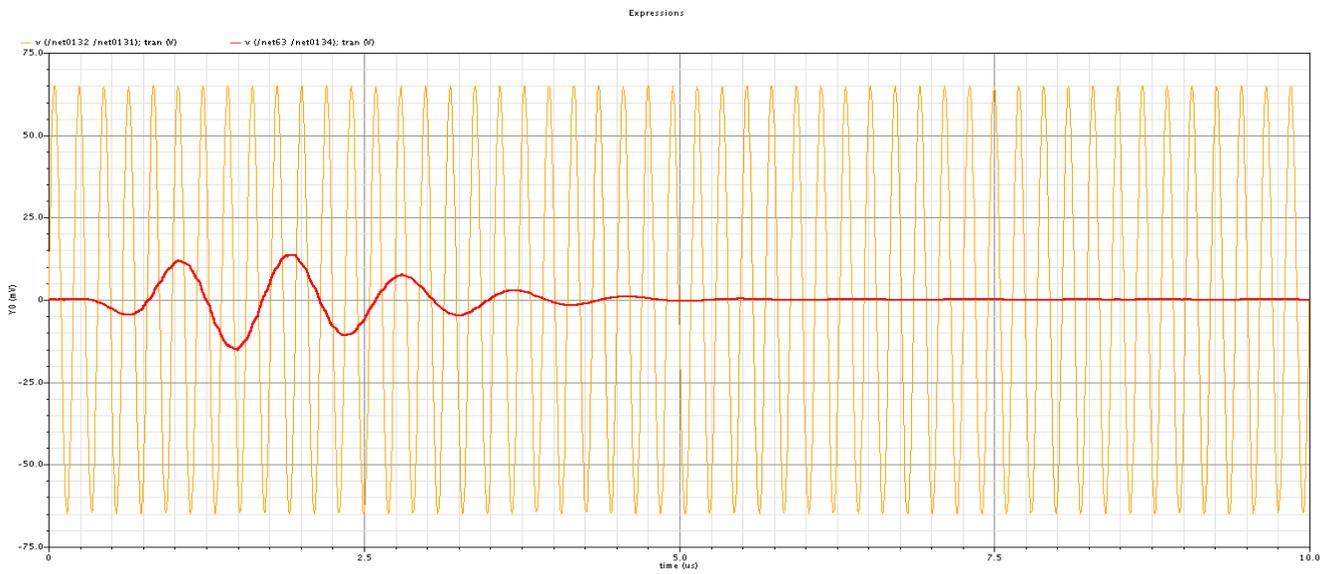


Figure 4.4.85 : : the signal at IF frequency (1.1 MHz) at the negative frequency side (image) after the reconstruction filter

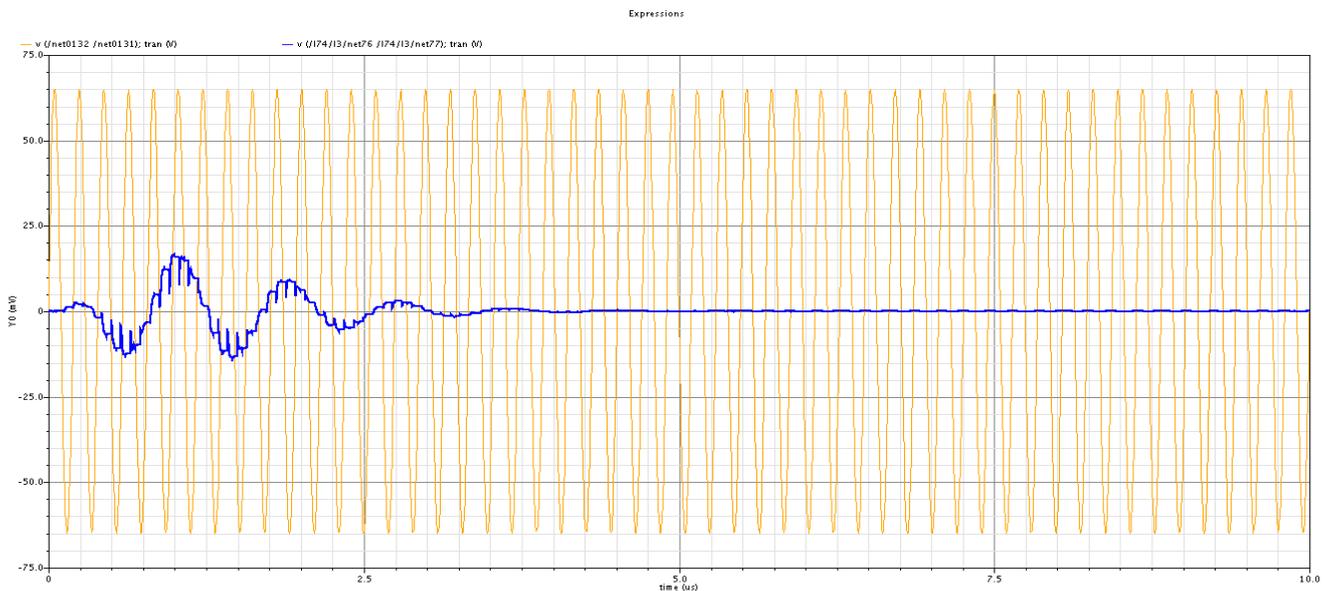


Figure 4.4.84 :the signal at 5.1 MHz frequency at the positive frequency side in the middle of the S.C. core filter (full rejection)

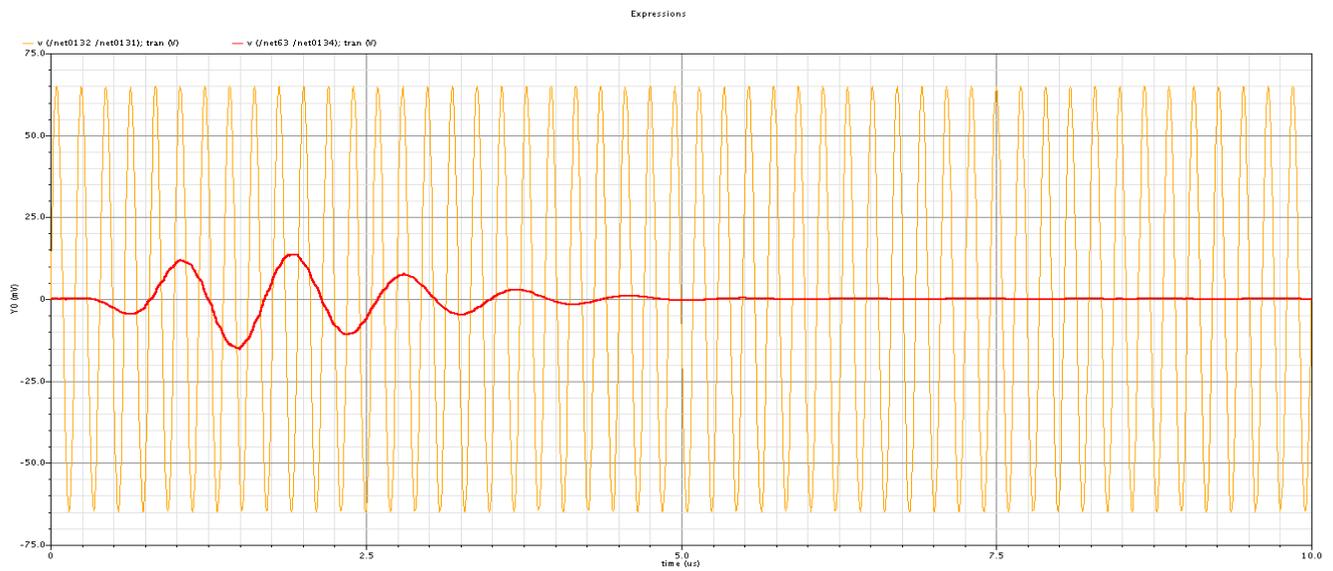


Figure 4.4.89 : the signal at 5.1 MHz frequency at the positive frequency side after the reconstruction filter (full rejection)

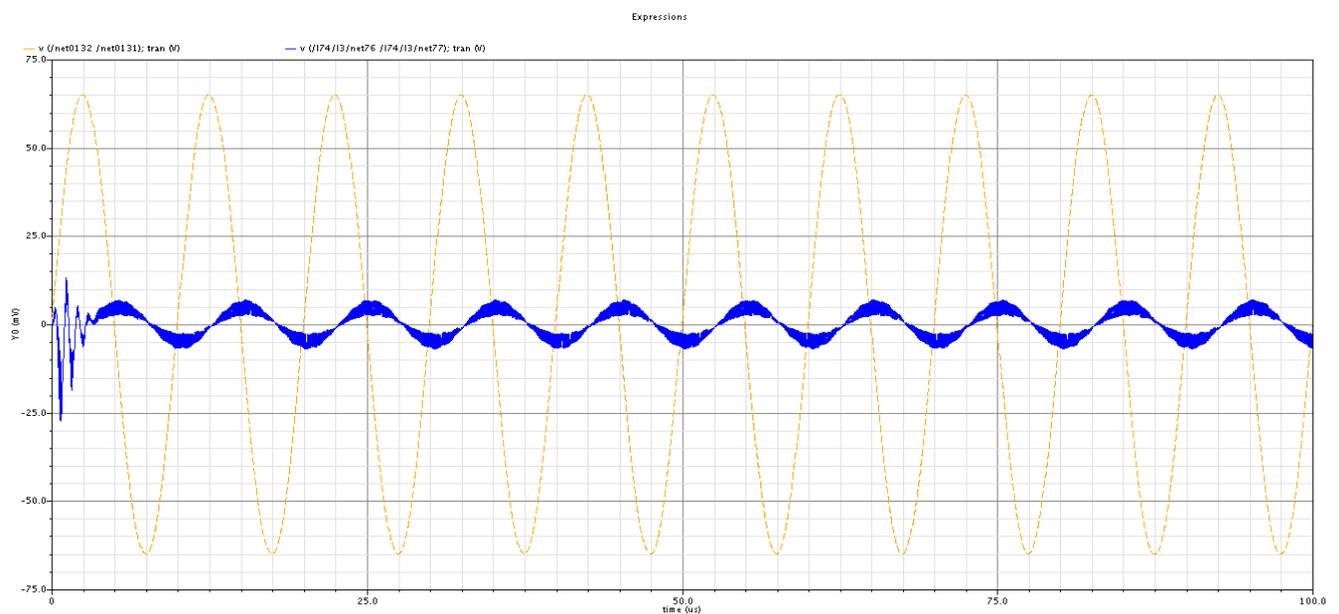


Figure 4.4.88 : the signal at 100 KHz frequency at the positive frequency side in the middle of the S.C. core filter

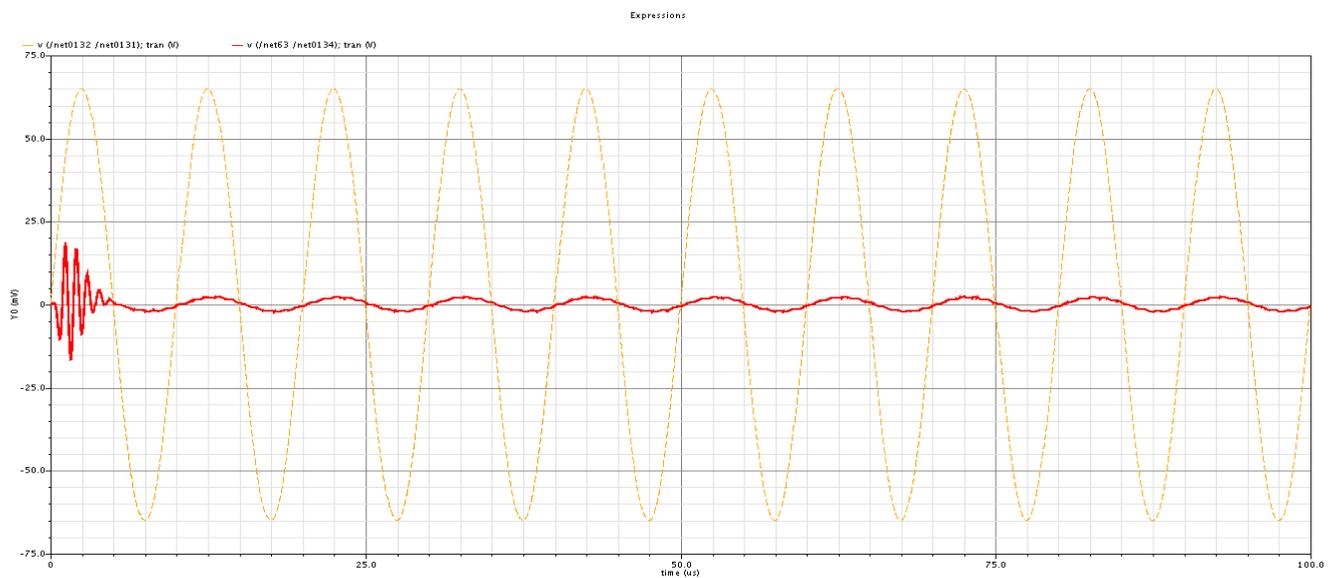


Figure 4.4.87 : the signal at 100 KHz frequency at the positive frequency side after the reconstruction filter

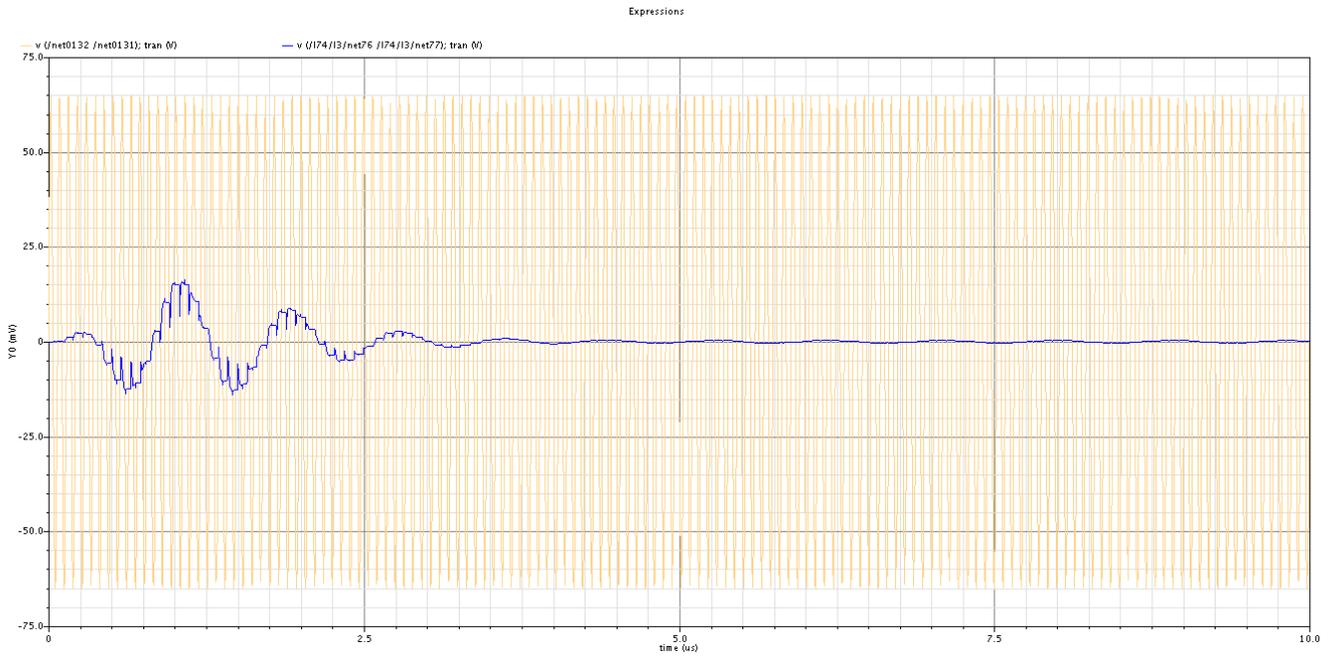


Figure 4.4.90 : the signal at frequency of 14.1 MHz “alias frequency” in the middle of the S.C. core filter ... and we can see clearly how the AA filter eliminates the signal completely

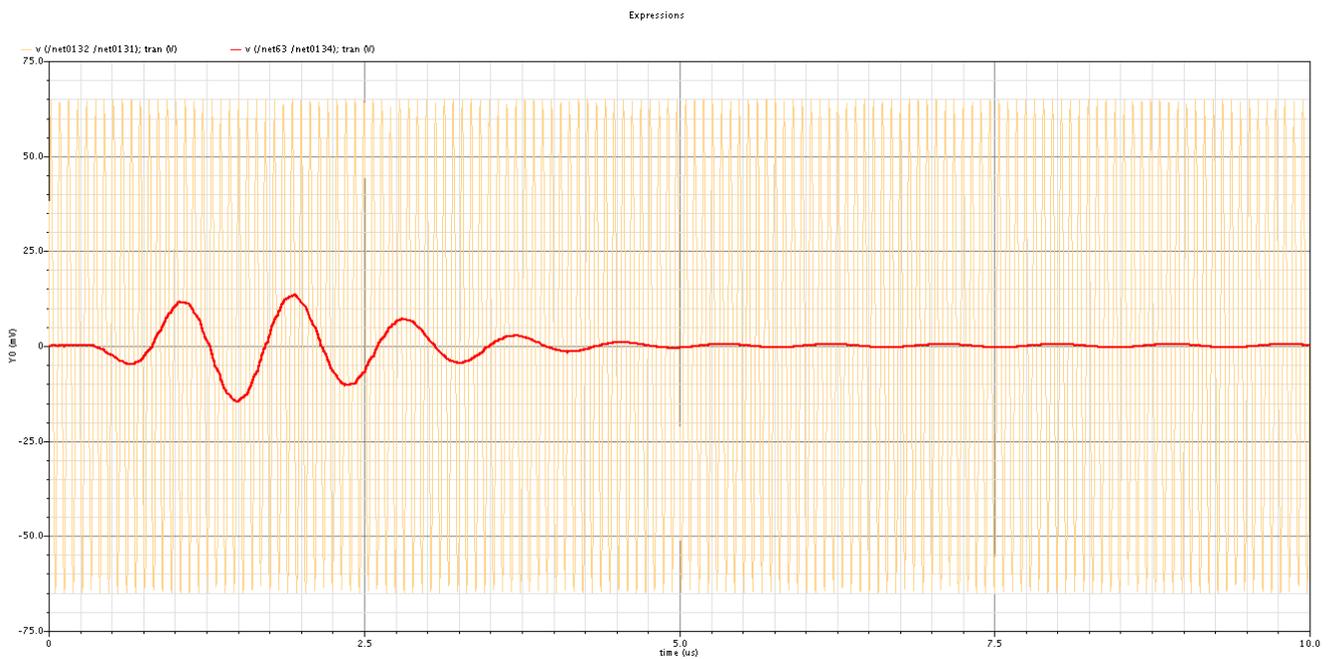


Figure 4.4.91 : the signal at frequency of 14.1 MHz “alias frequency” after the reconstruction filter ... and we can see clearly how the AA filter eliminates the signal completely

4.6.3 The corners simulation results

The corners test is done on the all filter to ensure that the filter will operate will when it is fabricated , the transfer function of the filter is plotted with the various corners , and the following plot show us the results

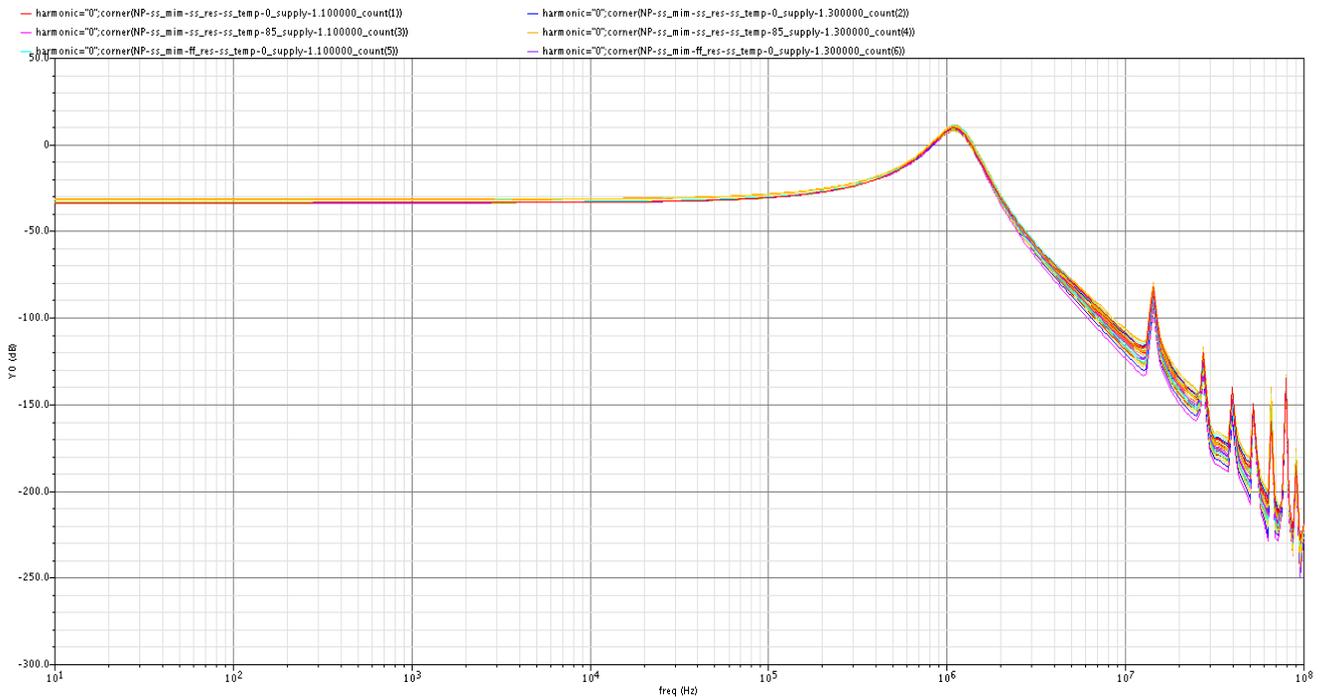


Figure 4.4.93 : the corners test results on the entire filter

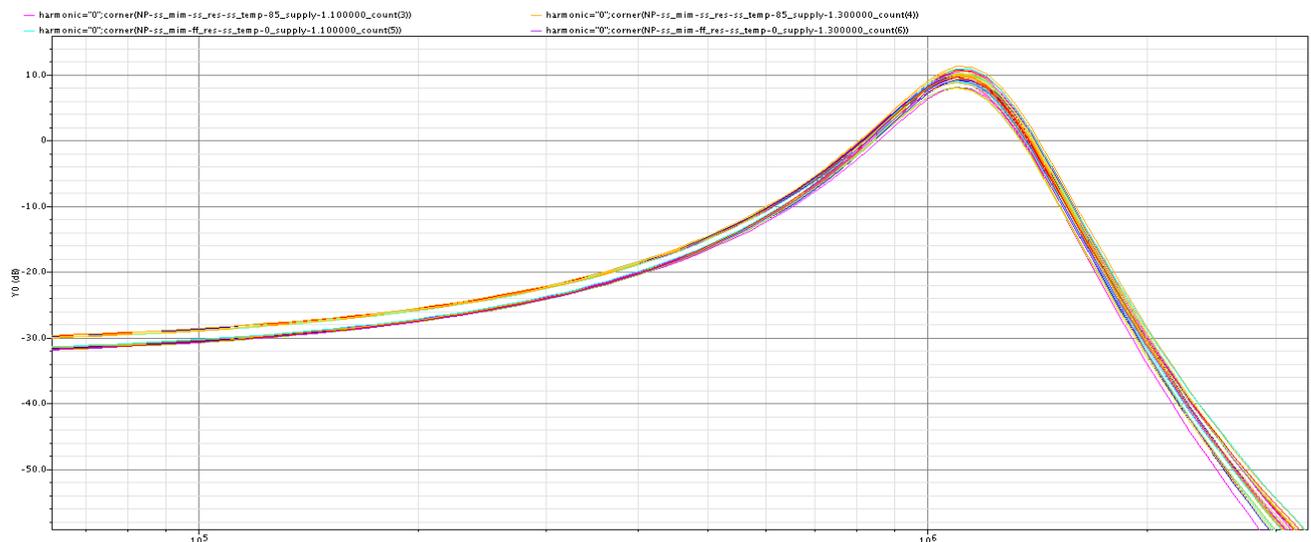


Figure 4.4.92 : a zoomed version of the corners test

We can see clearly that the gain is approximately not affected in the pass band , and that the rejections is maintained exceeding the needed ones , here we can see a zoomed version of the previous figure for more illustration . . .

4.6.4 The NF & IIP3 simulation results

4.6.4.1 The IIP3 results.

It is said before that the SC filter is linear in its nature , and usually have a large linearity , ... the IIP3 test is done for the SC core filter , and is was shown that the value of the IIP3 for this filter is as high as 24 dBm .

4.6.4.2 The NF results

The noise analysis is done for this filter using the test bench shown here below , (in which the all filter is symbolized into one symbol for ease of drawing schematic , and the 90 degree shifter is used to make the input to the filter to be one port only)... .

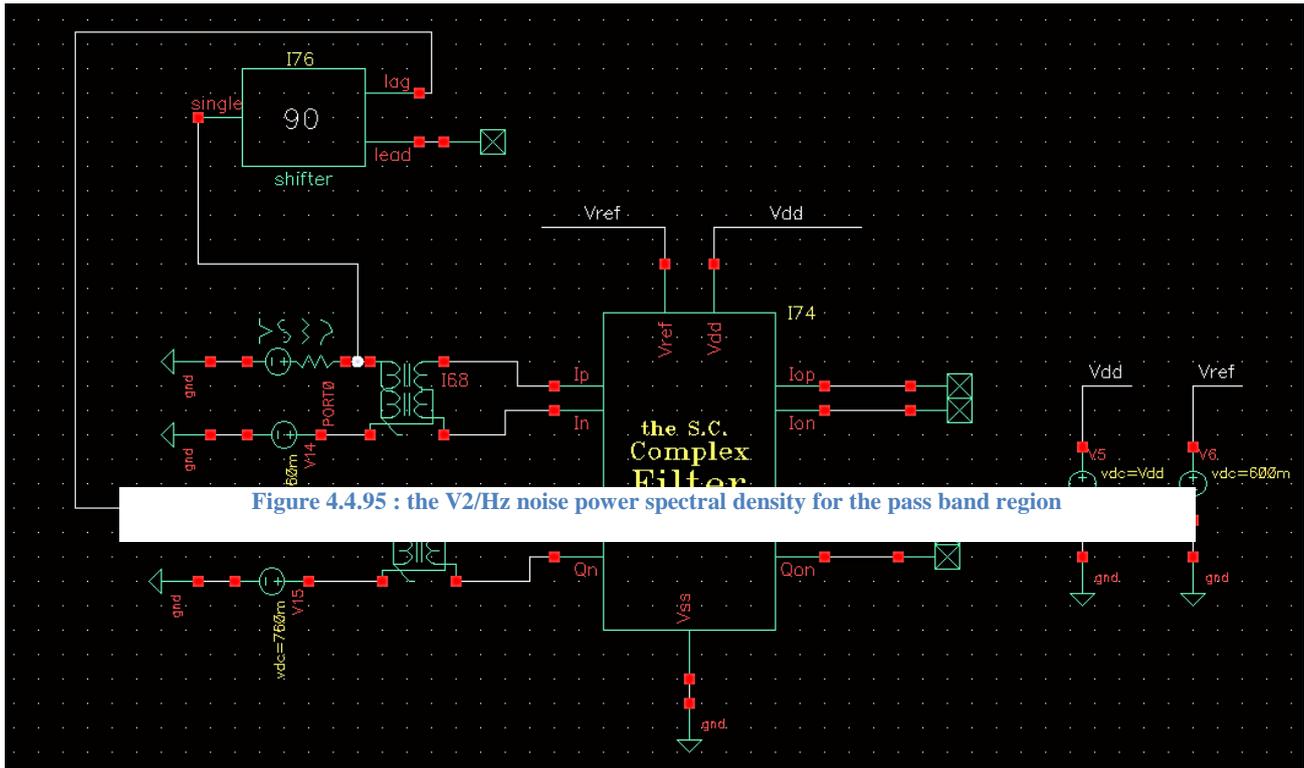
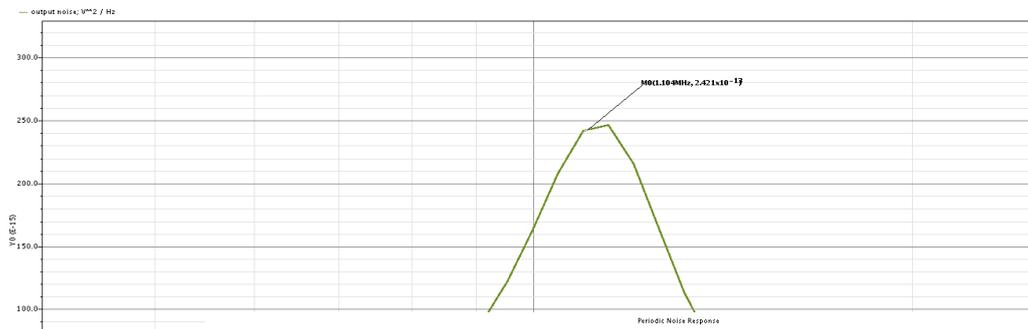


Figure 4.4.95 : the V²/Hz noise power spectral density for the pass band region

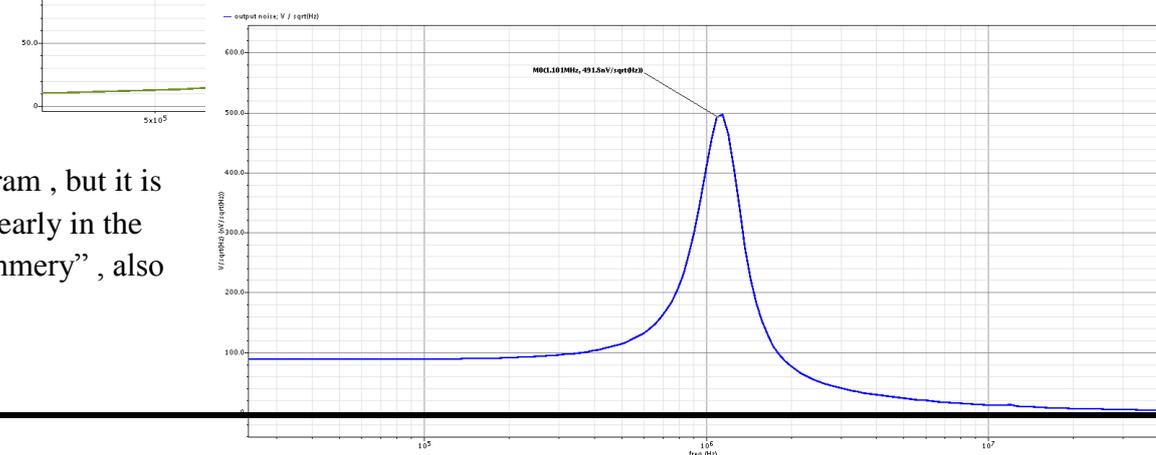
Figure 4.4.94 : the noise measurement test bench

This the the noise spectral for the region . .



curve of V²/Hz power density pass band . .

The value is not clear on the diagram , but it is indicated clearly in the “results summery” , also



the is the curve of the $V/\sqrt{\text{Hz}}$, noise voltage magnitude “RMS” in the pass band region

Figure 4.4.96 : the curve of the $V/\sqrt{\text{Hz}}$, noise voltage magnitude “RMS” in the pass band region

4.6.5 The results SUMMERY

Property	Achieved
Vdd	1.2
Current	657 uA
Gain	8.5 dB
BW	280 kHz
IF freq.	1.1 MHz
ACRR	34 dB
Att. At 2MHz	69 dB
Att. Att 10MHz	126 dB
IMRR	66 dB in Freq , 58 dB in Time
IIP3 of the SC core filter	24 dBm
The spot V^2/Hz noise at IF freq . (output referred)	242 fV^2/Hz
The spot $V/\sqrt{\text{Hz}}$ noise at IF freq (output refer)	491 $\text{nV}/\sqrt{\text{Hz}}$

5 Automatic Gain Control

5.1 Introduction

Automatic gain control Loop (AGC) is base band analog block as shown in Figure 5.1, AGC circuit used in many applications such as hearing aids, disk drives, communication circuits, and signal application. The demand of an AGC loop in wireless systems comes from the fact that all communication systems have an unpredictable received power, to buffer receiver electronics from change of input signal strength by

producing a known output voltage magnitude^[1]. Proposed AGC's function, first detects the strength level of the signal using the peak detector, then its sample the output of peak detector by flash A/D, then digital control take output of flash A/D and adjust gain of Programmable Gain Amplifier (PGA) according to value of digital word.

Proposed AGC is feed forward AGC so there is no loop stability problem existing in this gain-control scheme.

PGA specifications are Gain from 0dB to 54dB in 2dB step, NF = +25dB, 1-dB Comparison point = 0dBm and power consumption = 1.5mA.

Peak detector specifications are settling time <10uSec, input signal level > 100mV, power consumption < 3mA.

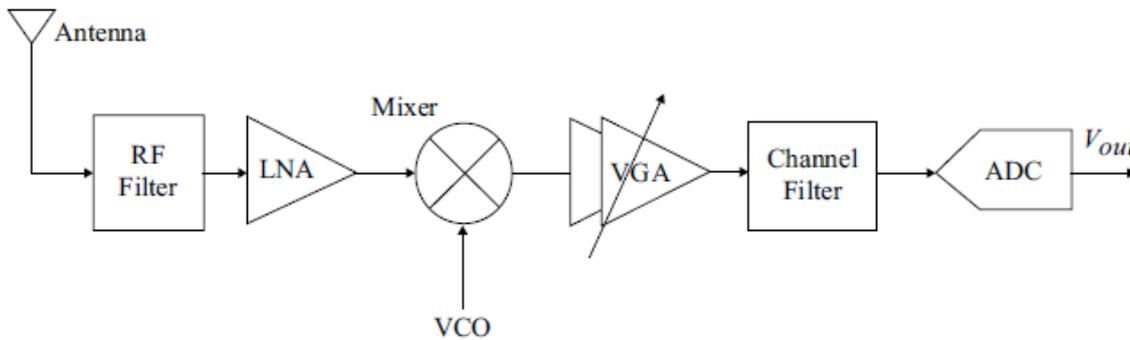


Figure 5.5.1; Simplified Radio Receiver Blocks

5.2 AGC Topologies

AGC Topologies divided into two types as shown in figure 5.2.

Feedback AGC

Feedback take sample from PGA output signal and compare it with reference volt and take decision to amplify more or not depend on comparison.

Feedback AGC Loop has advantages of feedback like high linearity, high BW and low noise, plus peak detector dynamic range required is smaller than in feed forward AGC,

Feedback AGC loop have a maximum boundary bandwidth in order to maintain stability. This maximum bandwidth entails a minimum settling-time of loop, in order to keep the settling-time constant; the feedback configuration requires the use of linear in dB voltage gain control^[2].

Feed Forward AGC

Feed forward take from PGA input signal and compare it with reference volt and take decision to amplify more or not depend on comparison.

Feed Forward AGC Loop offers a time constant that mainly depends on the peak detector response, so this loop is ideally not affected by the minimum settling-time restriction, the disadvantage is peak detector exposed to entire range of the input signal which require high linearity^[2].

Comparison between both types in table 5.1.

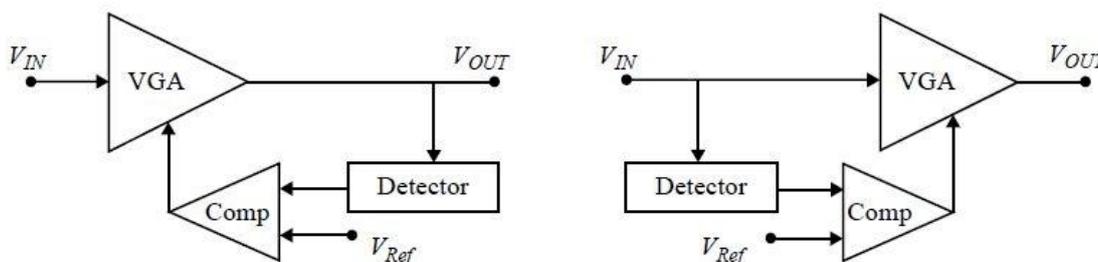


Figure 5.5.2; Feedback AGC

Feed Forward AGC

	Advantages	Disadvantages
Feedback Loop	High BW Low noise High Linearity	Stability analysis required High settling time
Feed forward Loop	No stability analysis required Smaller settling time than FB AGC	High linearity required AGC input dynamic range required by peak detector

Table 5.5; Comparison between Feedback and feed forward AGC Loop

5.3 AGC Main Blocks

Proposed AGC loop consist from four blocks the main block is PGA.

5.4 Programmable Gain Amplifier

5.4.1 Introduction

Programmable gain amplifier (PGA) is a key building block in Automatic Gain Control Loop. PGA is typically employed in a feedback loop to realize an automatic gain control (AGC), to provide constant signal power to baseband analog-to-digital converter (ADC) for unpredictable received signal strengths [3].

5.4.2 Design Parameters

- Gain

Main specification of PGA, gain range calculated according to variation in received signal strength, our target is to deliver constant signal strength to A/D equal 0dBm which equivalent to 316mV, and the sensitivity and saturation equal -95dBm and -17dBm respectively.

- Bandwidth

Bandwidth is the frequency band which signal has this frequency amplified than other signals, PGA should provide bandwidth higher than 1.2MHz for all gain settings.

- Noise Figure

Noise figure is a measure for degradation in SNR, we target NF in our transceiver 8dB, the most affecting blocks in NF is Front End blocks i.e. RF Blocks so we care about decreasing NF for front end blocks, and give small care about NF for back end blocks i.e. base band blocks, using following equation we see than NF of Base band is divide by gain square of RF and filter so NF of PGA may be large and It won't effect on Total NF

$$NF_{RX} = NF_{RF} + \frac{NF_{Filter} - 1}{A_v RF^2} + \frac{NF_{BB} - 1}{A_v RF^2 * A_v filter^2}$$

- 1-dB Comparison Point

Its measure of non-linearity for PGA, The specification for the linearity of the PGA is Generally very high to maintain good overall system linearity, High linearity PGA cell can he designed using techniques including source degeneration, variable current bias, variable triode bias, variable load and high gain with feedback [3].

- Power Consumption

Our receiver target low power consumption, so using enhanced design techniques like g_m/I_D we can achieve small power consumption and achieve required specifications.

5.4.3 PGA Topologies

PGA divided into two main types.

- Linear
 - Gain is proportional with control signal $G(V_c) = a \cdot V_c$
 - Disadvantage is settling time is *function* of input signal
- Exponential
 - Gain proportion exponential with control signal
 - Advantage is settling time is constant and *not depend* on input signal

Two main principle of operation for PGA

- Passive PGA
 - Fixed gain block with resistive array digitally controlled as shown in Figure 5.3.
- Active PGA
 - Change gain through changing trans- conductance of the amplifier by changing current.

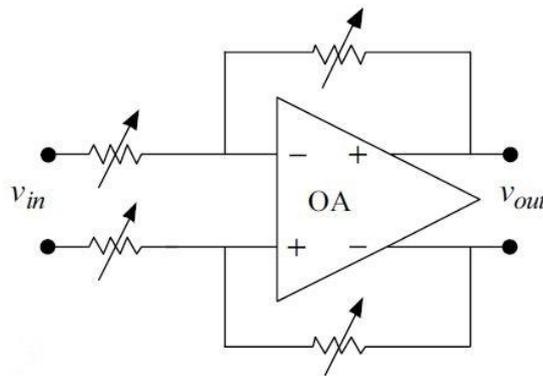


Figure 5.5.3; PGA with resistive feed back

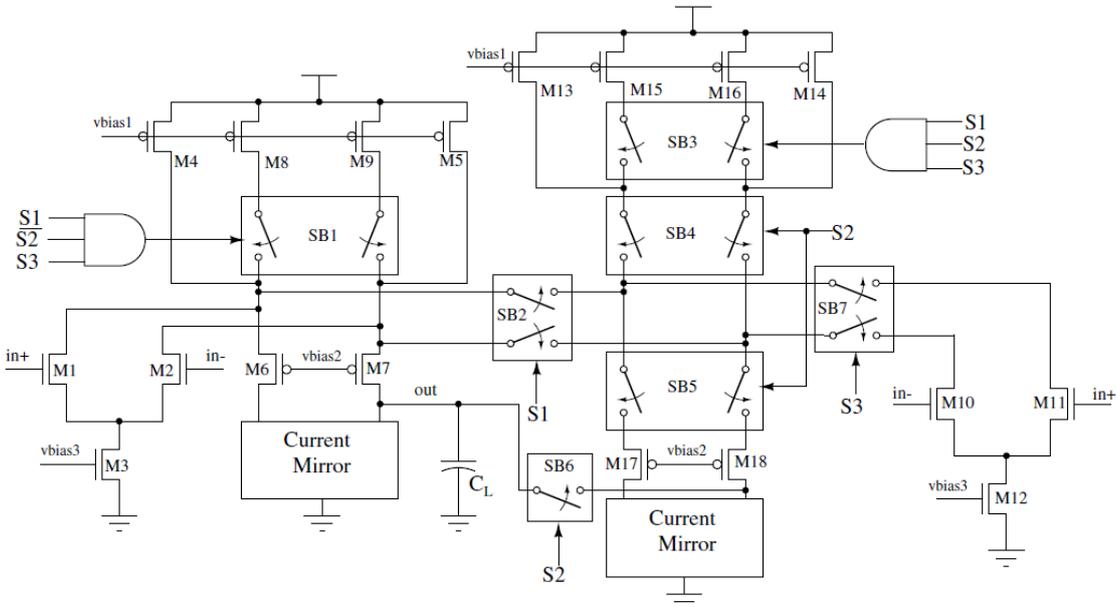


Figure 5.5.6; Proposed Folded Cascode OTA in [4]

5.4.3.3 Variable Gain Amplifier based on the resistive feedback topology

Figure 5.3 shows a high-gain amplifier with resistor-network feedback. Its voltage gain can be varied by changing the ratios of R_{f1}/R_1 and R_{f2}/R_2 . High linearity can be achieved if the loop gain is large and the resistor network is linear. However, if the conventional operational amplifier is used, the variation of the feedback factor results in variations of the bandwidth and the total harmonic distortion. When the circuit is designed to cover the worst-case scenario over the entire gain range, its power consumption is not optimized.

5.4.3.4 Variable Gain Amplifier based on changing Trans conductance of source coupled

The Trans conductance of the source-coupled pair shown in Fig. 5.7 is varied by changing the bias current of the transistors. The circuit's gain is proportional to g_m of the input transistors. When the input signal is weak, the large bias current is needed to obtain high-gain and low-noise performance. On the other hand, when the input signal is large, the low bias current can degrade the linearity as shown in Figure 5.7

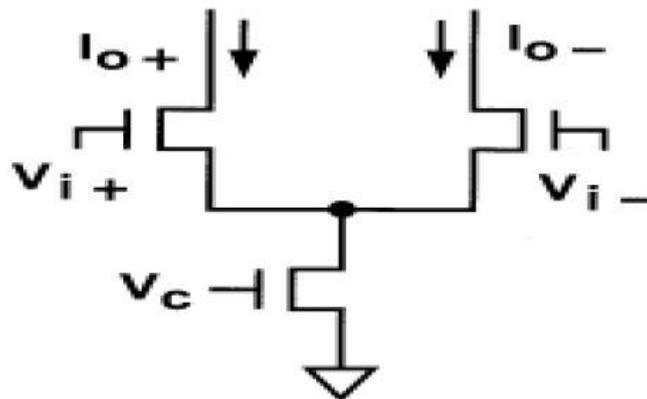


Figure 5.5.7; Variable Gain Amplifier by change Transconductance of source coupled

5.4.4 Required Specifications

Specifications	Required
Gain	0 to 54dB with step 2dB
NF	<25dB
1-dB Comparison Point	>0dBm
Power Consumption	<1.5mA

Table 5.6; Required Specifications for PGA

5.4.5 Design Approach

5.4.5.1 PGA topology used

From required specifications we choose the most suitable topology, as we need at least 28 gains setting plus low power equal 1.5mA, so that the most suitable topology is Operational amplifier with resistive feedback as shown in Figure 5.3.

Advantages	Disadvantages
Gain not function in op amp gain, gain fixed with process variation	Stability Issue
High Linearity, gain function in feedback resistant	Drive resistive load, OTA not used
High Bandwidth due to feedback	Need Buffer at input to drive small resistive loads
Low Power Consumption	
Easy to make 30 gain condition	
Lower error in gain step	

Table 5.7; Advantages and disadvantages of proposed PGA topology

Algorithm used to design PGA with gain range 0 to 54dB with 2dB step.

PGA should provide at least 28 gains setting; PGA design is critical because its input referred noise density determines the overall input referred noise density of the baseband. In order to meet these requirements, first stage should provide maximum gain so that total NF of PGA reduced; the following design approach was adopted. The amplifier was divided in to three design blocks: first stage provides two levels of gain (0 & 36dB), second stage provides three levels of gain (0, 12 & 24dB) and the third stage provides six levels of gain (0, 2, 4, 6, 8 & 10dB) as shown in Figure 5.8.

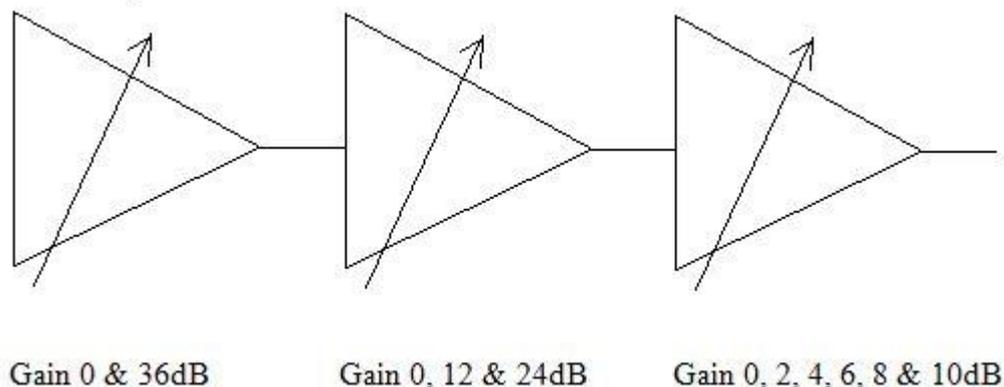


Figure 5.5.8; three stages used in PGA

The advantages of this algorithm is it can provide gain from 0 to 70dB with step 2dB

5.4.5.2 g_m/I_D Design method

g_m/I_D method is used to calculate dimension of transistors, the advantages of this method is that curve is constant for given technology, this method minimize time spent and achieve major accuracy in results, because it's more accurate than square law design method as square law change a bit in modern technology, the transistors achieve maximum efficiency when transistor in triode region and efficiency decrease as transistor go in saturation [5].

The design flow for g_m/I_D method

1. Determine required g_m for transistor from GBW
2. Determine current pass in transistor from power consumption
3. Choose L according to specification, choose min L for high frequency and long L for high output resistant transistor
4. Calculate g_m/I_D ratio
5. Determine bias voltage for current source bias for desired g_m/I_D

Each stage of PGA is OTA with resistive feedback, so using g_m/I_D method we can assume

Design of OTA divided into four parts

1. Folded Cascode OTA
2. CMFB
3. Bias Network
4. Resistant Network

Design Differential Pair of OTA using g_m/I_D Method

Connect transistor as shown in Figure 5.9, through changing V_{gs} we can plot current pass through transistor as shown in Figure 5.10 we need current pass equal 90 μ A, so we calculate required V_{gs} for this current, the next step is to plot g_m/I_D vs. V_{od} as shown in Figure 5.11 we conclude from this to have approximately efficient MOS we have to reduce V_{od} as much as we can so for the other MOSFET's we set min V_{od} which make it work on edge of saturation.

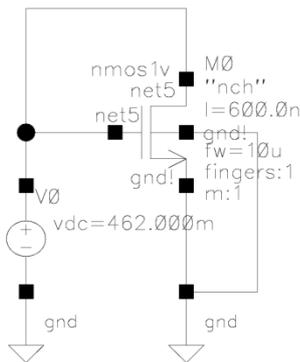


Figure 5.5.9; g_m/I_D method

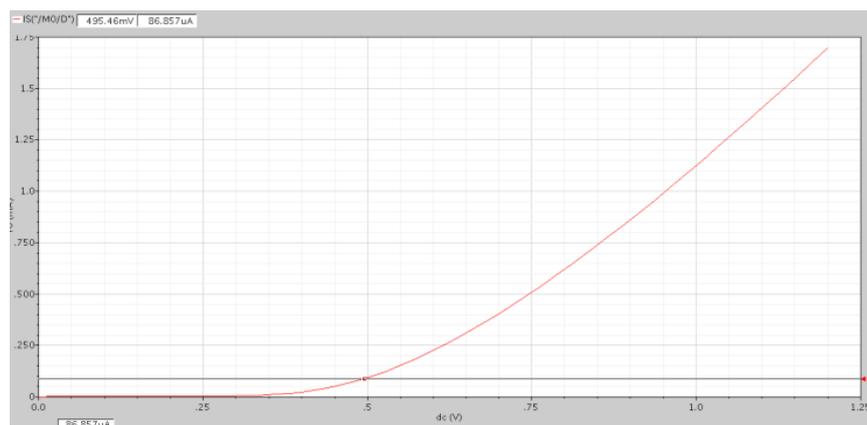


Figure 5.5.10; ID vs. Vgs

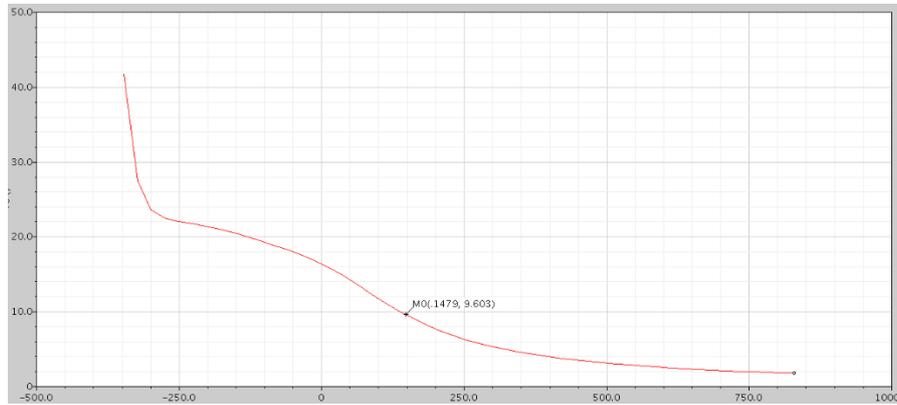


Figure 5.5.11; gm/ID vs. Vod

5.4.6 Simulation Results

Designing three cascading stages need accurate choice for $V_{in,cm}$ and $V_{out,cm}$, using Folded cascode OTA which can provide $V_{in,cm} = V_{out,cm}$, Folded cascode designed on $V_{in,cm} = V_{out,cm} = 650mV$, Open Loop gain $> 60dB$, max gain error $< 1dB$ and phase margin higher than 55° across all corners, proposed Folded Cascode OTA shown in Figure 5.12.

Figure 5.13 shows plot of open loop gain and phase margin of folded cascode using TT-models of tsmc13rf, open loop gain is 66dB and its phase margin around 80° , Figure 5.14 shows AC response of CMFB loop using TT-models of tsmc13rf, CMFB gain is 70dB and phase margin around 60° .

Figure 5.15 shows plot of open loop gain and phase margin of folded cascode using SS-models of tsmc13rf, open loop gain is 53dB and its phase margin around 75° , Figure 5.16 shows AC response of CMFB loop using SS-models of tsmc13rf, CMFB gain is 57dB and phase margin around 63° .

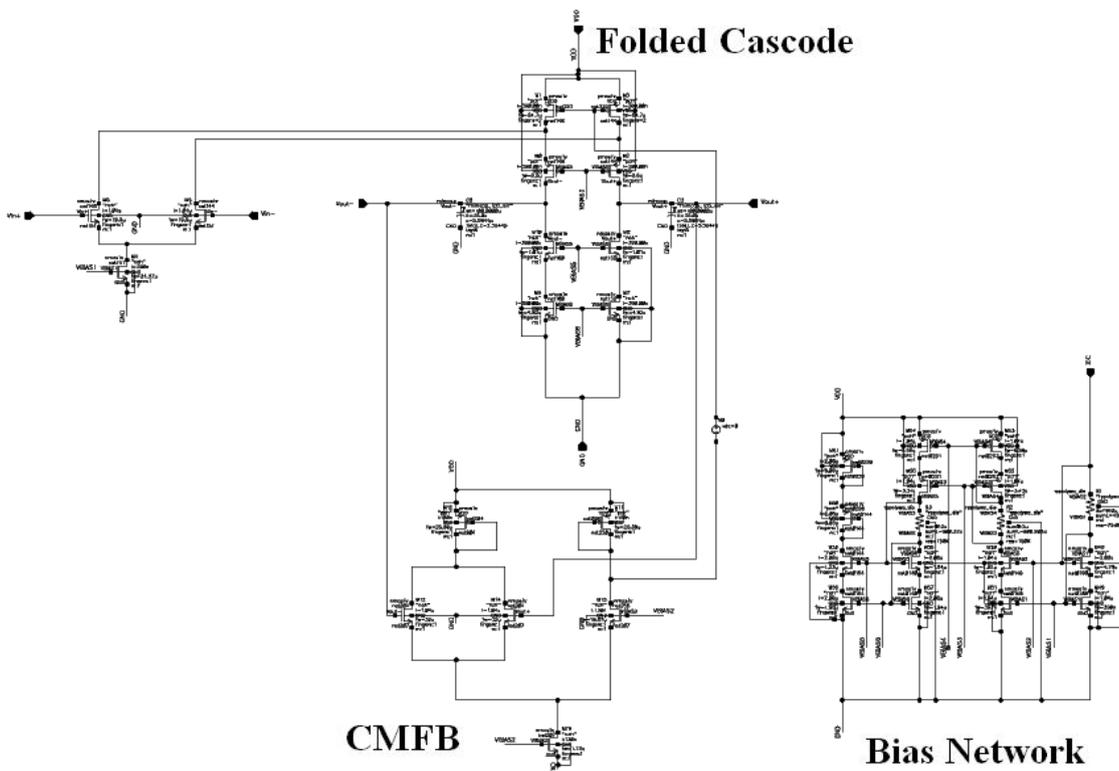


Figure 5.5.12; Proposed Folded Cascode OTA

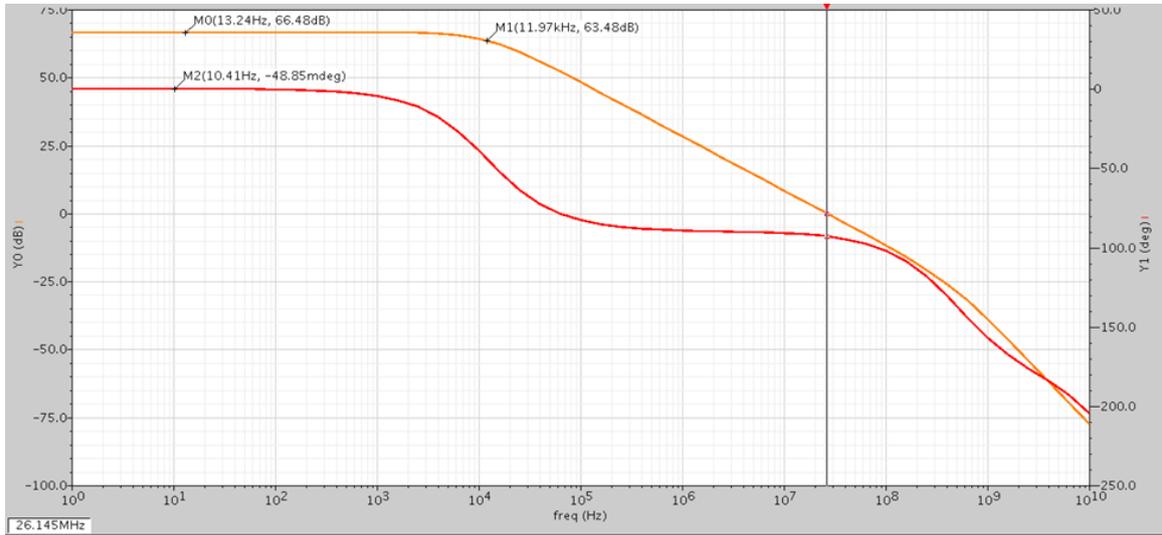


Figure 5.5.13; Open Loop Gain and Phase Margin of Folded Cascode OTA TT

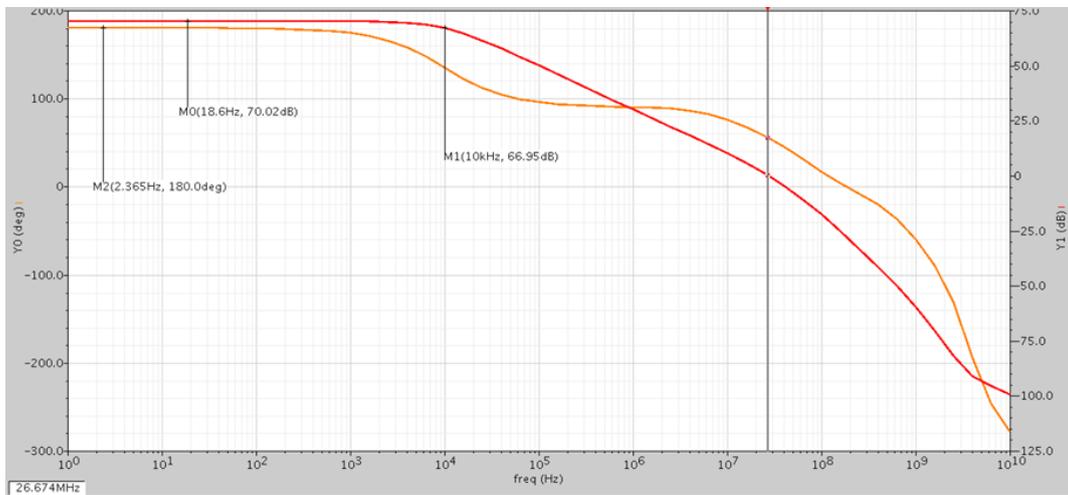


Figure 5.5.14; Figure 14 AC response of CMFB TT

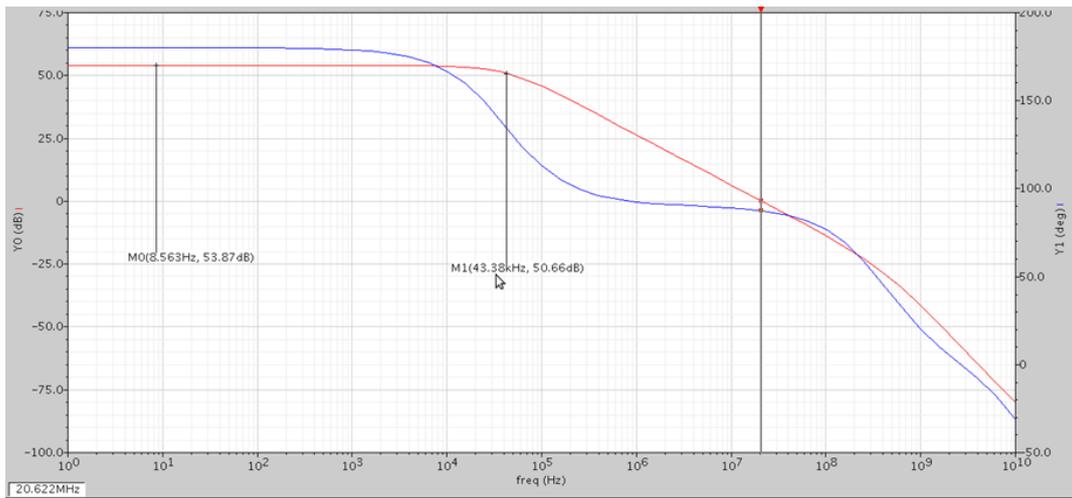


Figure 5.5.15; Open Loop Gain and Phase Margin of Folded Cascode OTA SS

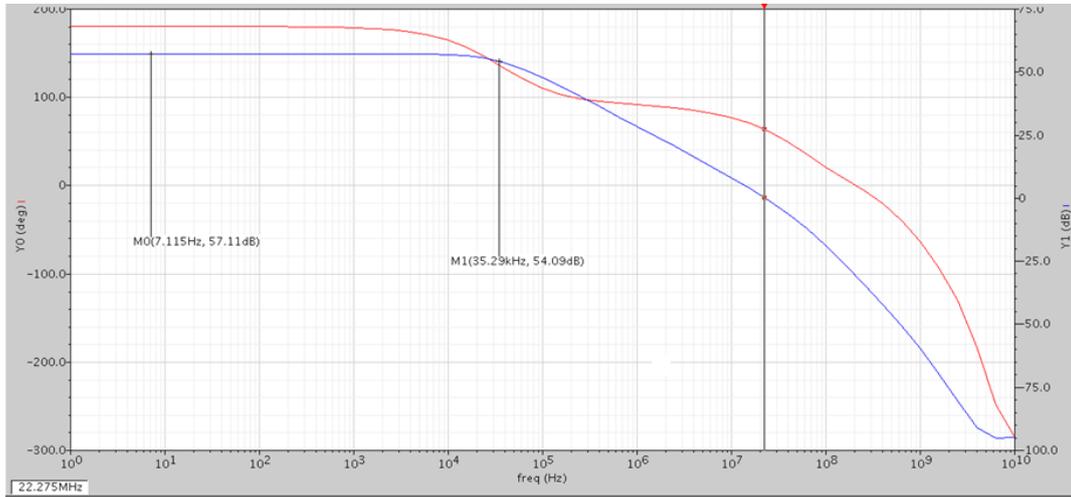


Figure 5.5.16; AC response of CMFB SS

Figure 5.17 shows plot of open loop gain and phase margin of folded cascode using FF-models of tsmc13rf, open loop gain is 66dB and its phase margin around 88° , Figure 5.18 shows AC response of CMFB loop using FF-models of tsmc13rf, CMFB gain is 70dB and phase margin around 54° .

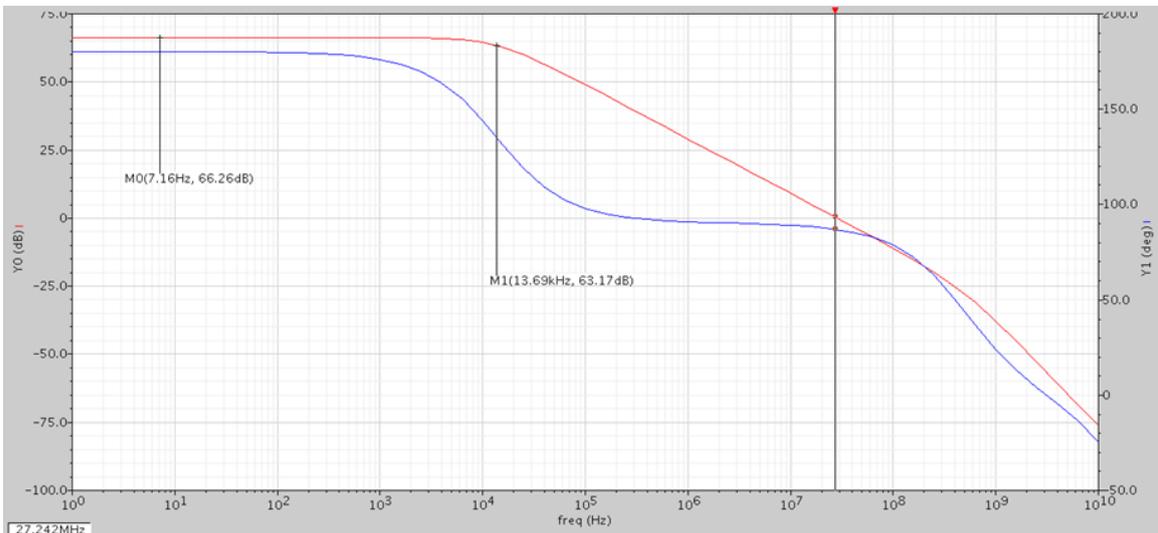


Figure 5.5.17; Open Loop Gain and Phase Margin of Folded Cascode OTA FF

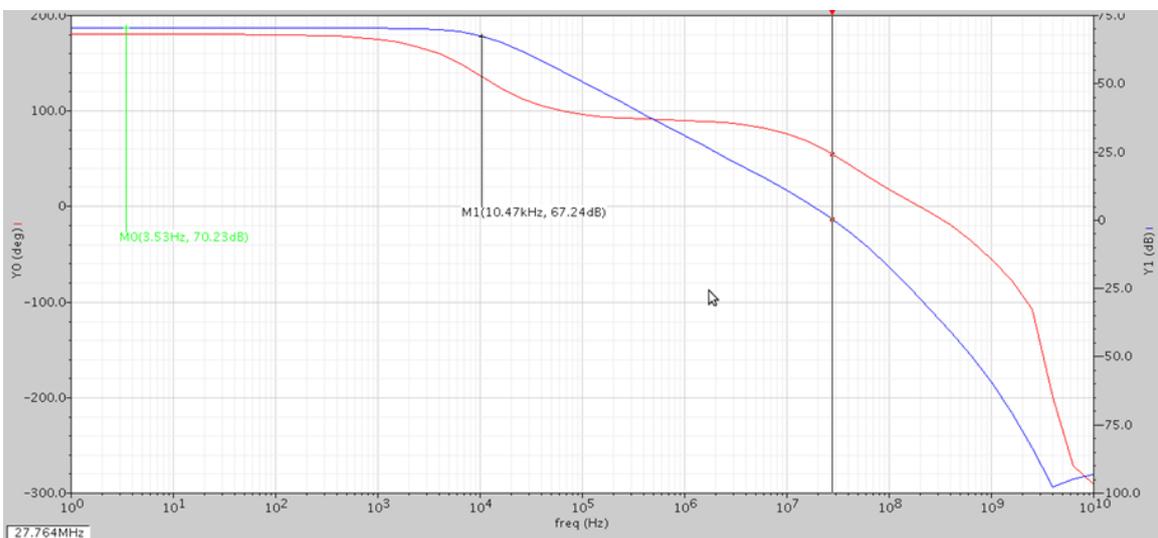


Figure 5.5.18; Figure 18 AC response of CMFB FF

Proposed Folded Cascode OTA is the core of PGA, cascading three OTA stages to achieve gain setting, Figure 5.19 show total PGA system used to provide required gain settings first stage which provide 0 & 36dB, second stage which provide 0, 12 & 24dB, and third stage which provide 0, 2, 4, 6, 8 & 10dB from left to right respectively, control come from eight digital wires which control the 16 latch switches, using latch switch provide more accuracy in gain value, inverted control signal done by simple inverter.

Figure 5.20 shows 1-dB comparison point of PGA its equal 6.47dBm.

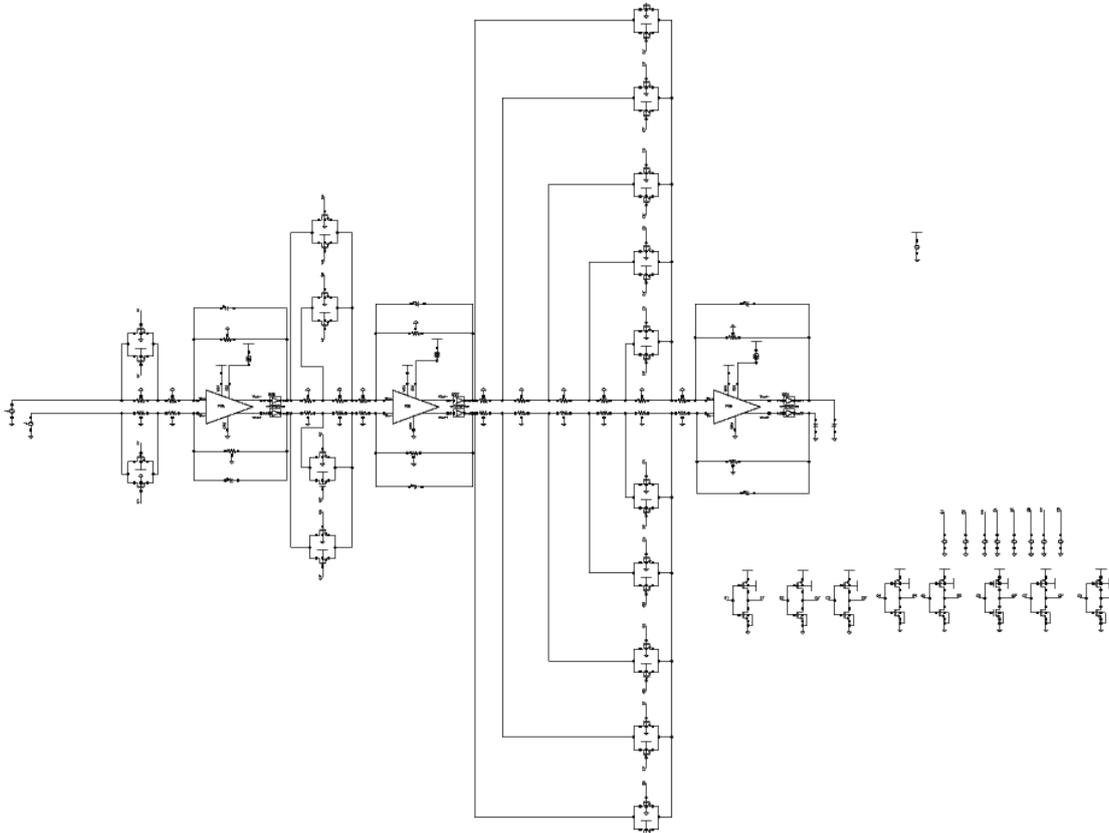


Figure 5.5.19; Proposed PGA

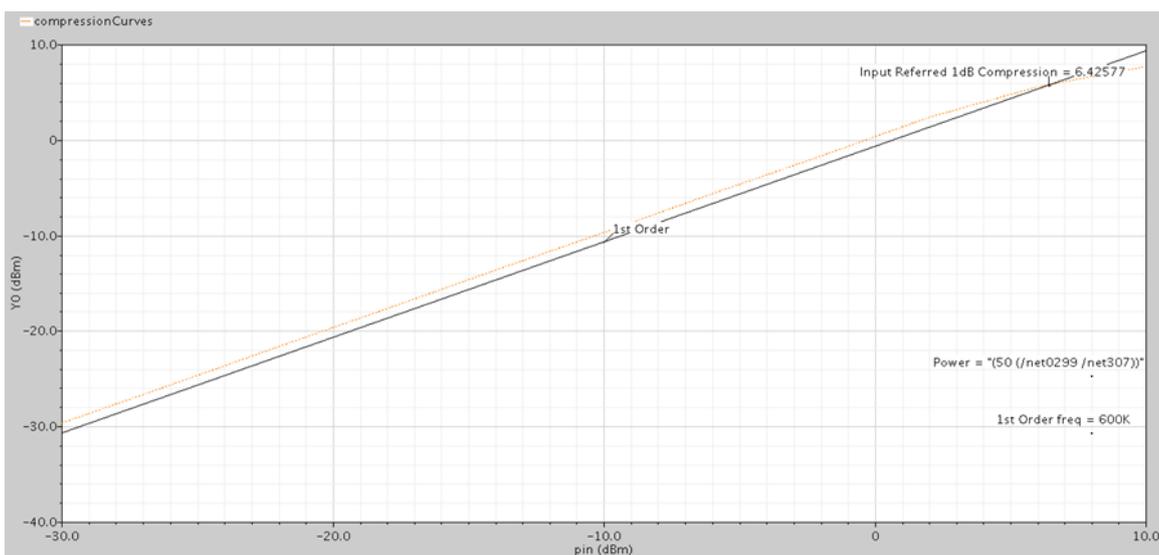


Figure 5.5.20; 1-dB comparison point TT

Figure 5.21 show NF of total PGA at maximum gain setting its equal 20.85dB for our working bandwidth.

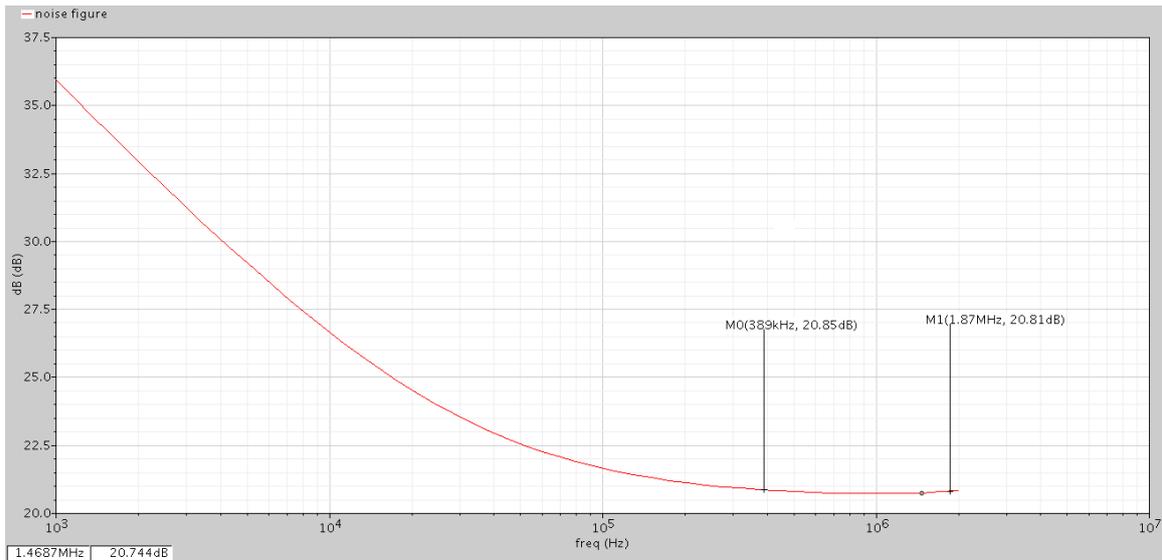


Figure 5.5.21; Figure 21 NF of PGA TT

Figure 5.22 show all gains setting from 0 to 54dB of PGA

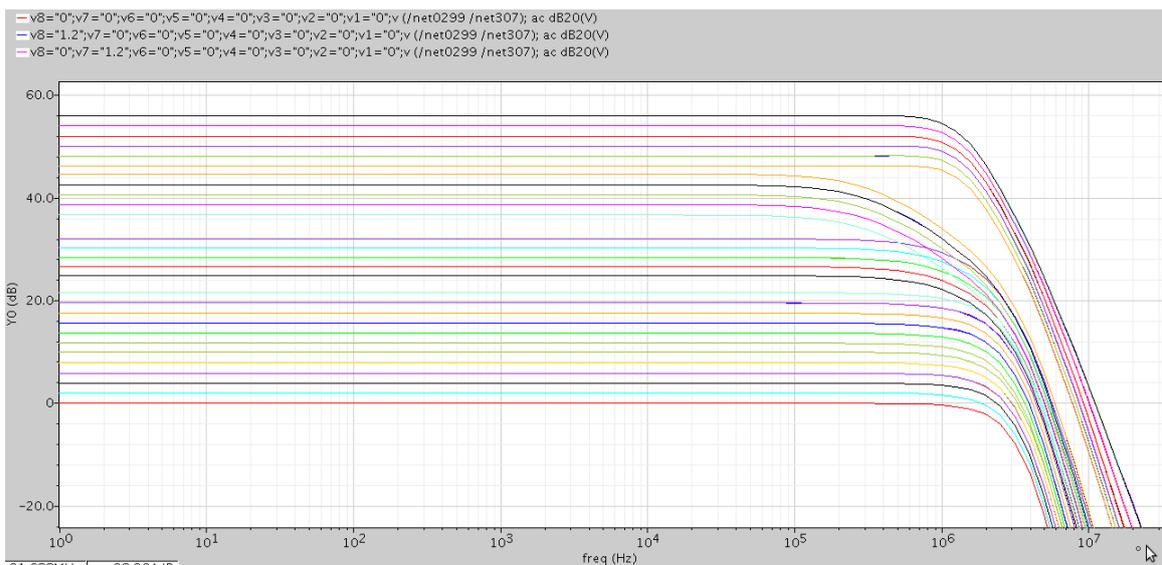


Figure 5.5.22; Proposed PGA AC response

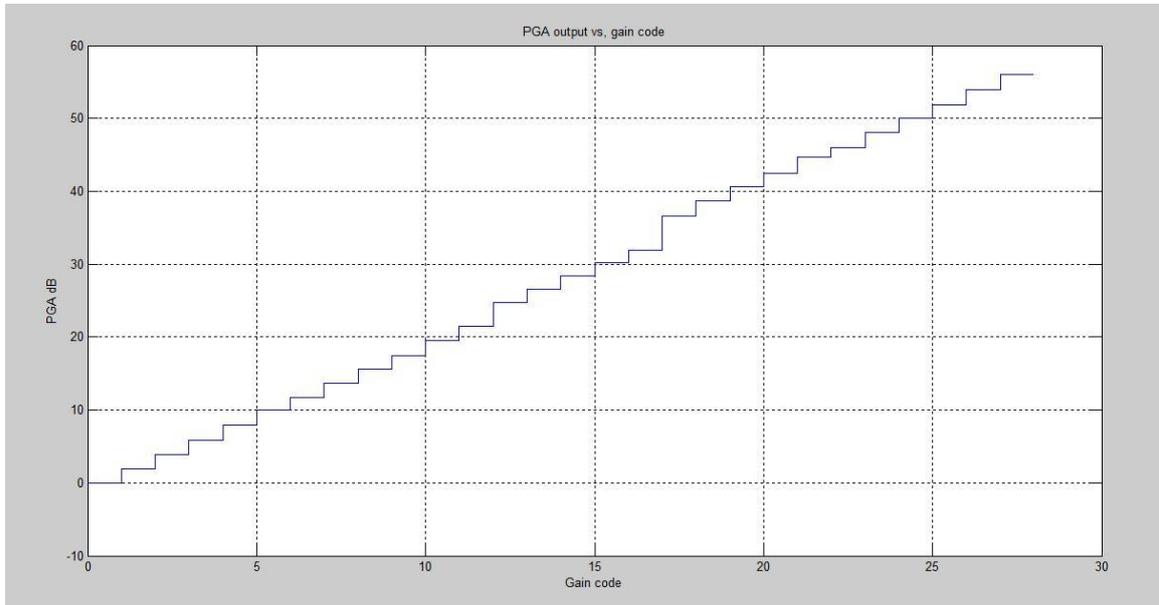


Figure 5.5.23; PGA output gain vs. gain code word

5.4.7 Corners Results

The purpose of Corners simulation is to measure the variation in specifications across PVT, for MOSFET we have four corners than Typical NMOS Typical PMOS, we have Slow NMOS Slow PMOS, Fast NMOS Fast PMOS, Fast NMOS Slow PMOS and Slow NMOS Fast PMOS, for resistor used from tsmc13rf kit which is rppoly_wo_dis, it has two corners Fast Fast & Slow Slow, for capacitor used from tsmc13rf kit which is mimcap, it has two corners Fast Fast & Slow Slow, for temperature we have two corners -40 C° & 125 C° , for voltage supply we have two corners 1.08V & 1.32V, so making all these corners manually will be time consuming, so we use script language called "Ocean Script", Ocean Script is language used for making corners simulation, for PGA we care about PM of CMFB to avoid oscillation across corners, and PGA gain, ocean script codes found in Appendix A, Figure 5.24 shows variations in open loop gain of Folded Cascode OTA across corners, Figure 5.25 shows variations in Folded Cascode CMFB loop gain across corners, Figure 5.26 show variations in PM of CMFB in PGA across corners using Mat lab, Figure 5.27 show variations in maximum gain of PGA across corners, Figure 5.28 shows variations in Noise Figure across corners, due to corners simulation of 1-dB comparison point takes long time so I made worst corners for 1-dB only, Figure 5.29 shows 1-dB comparison point at SS corner, Figure 5.30 shows 1-dB comparison point at FF corner, from all these simulations across corners I can conclude this PGA is ready for layout.

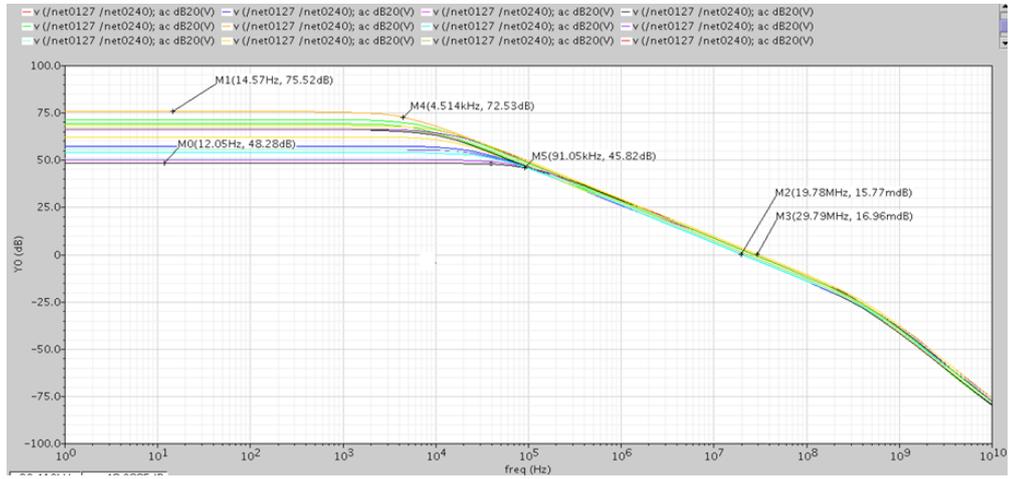


Figure 5.5.24; Open Loop gain of Folded Cascode OTA

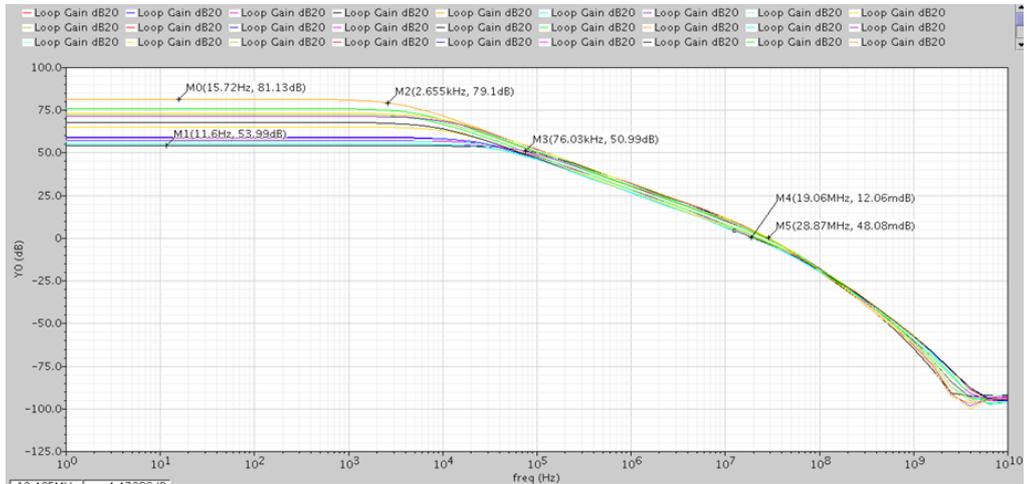


Figure 5.5.25; Loop gain of Folded cascode CMFB

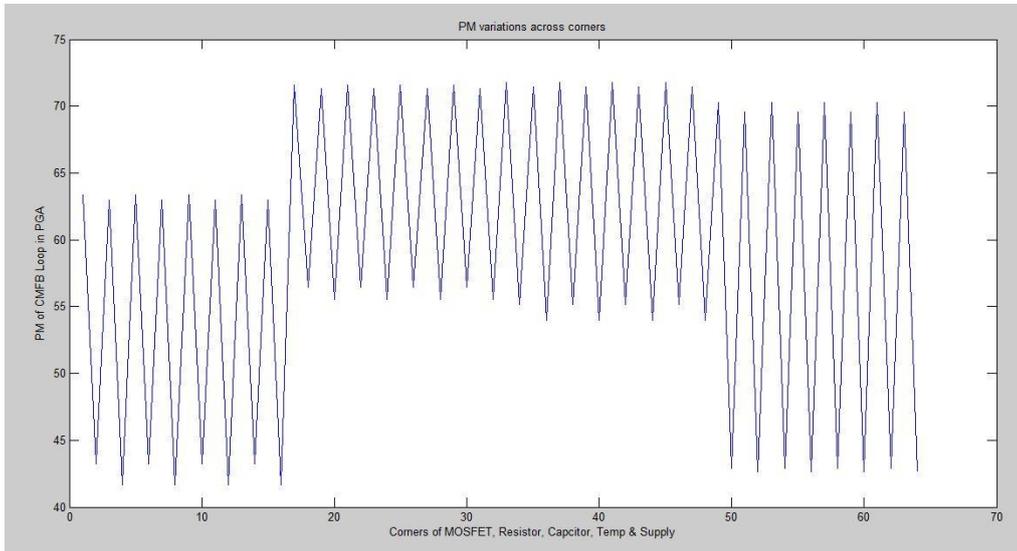


Figure 5.5.26; PM of CMFB across corners

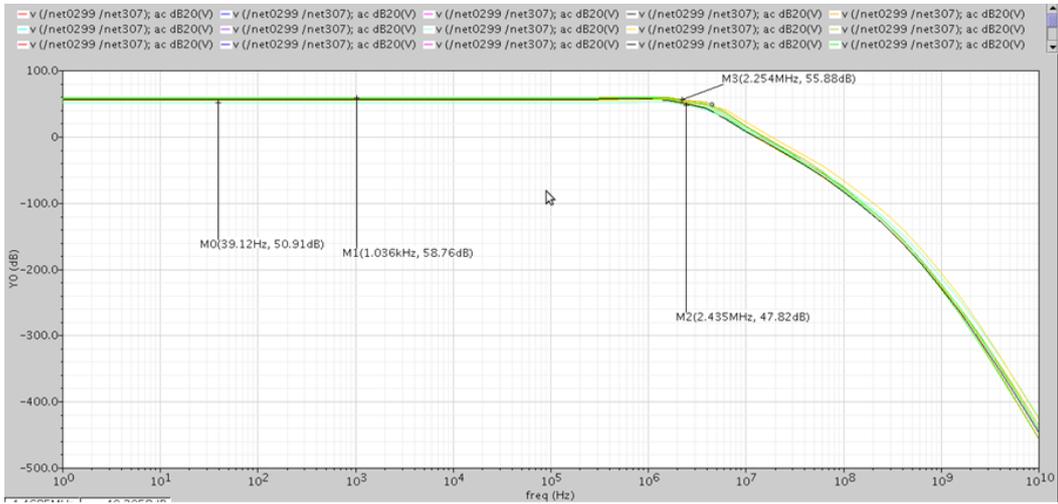


Figure 5.5.27; Max PGA gain across corners

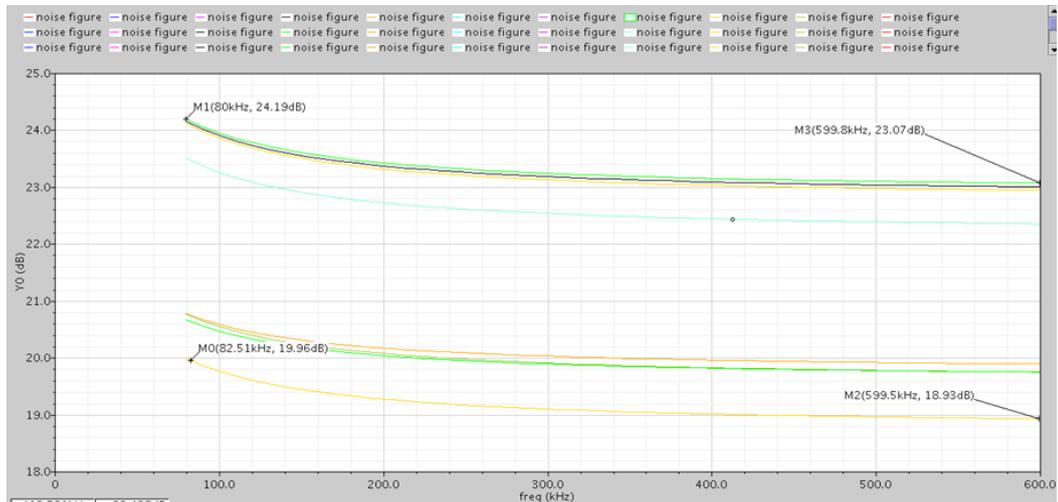


Figure 5.5.28; NF variation across corners

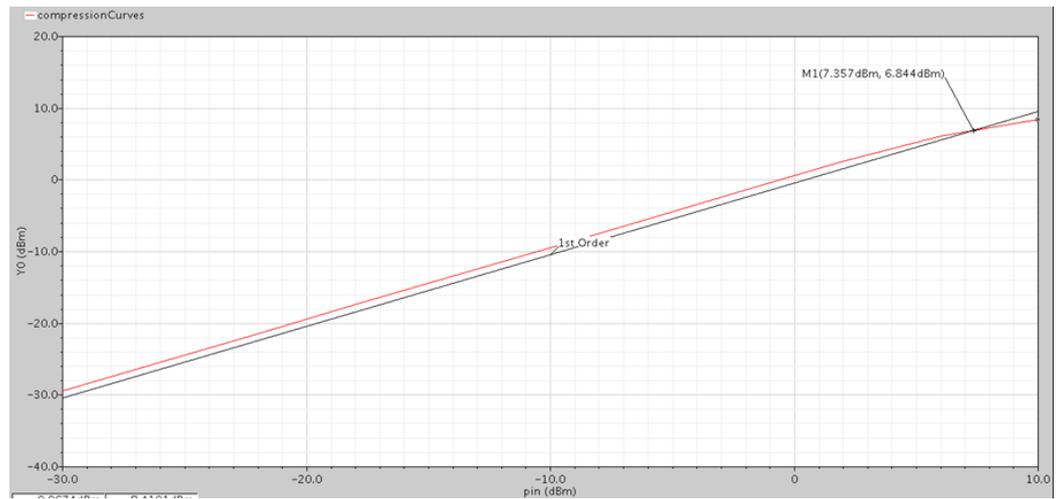


Figure 5.5.29; 1-dB comparison point SS

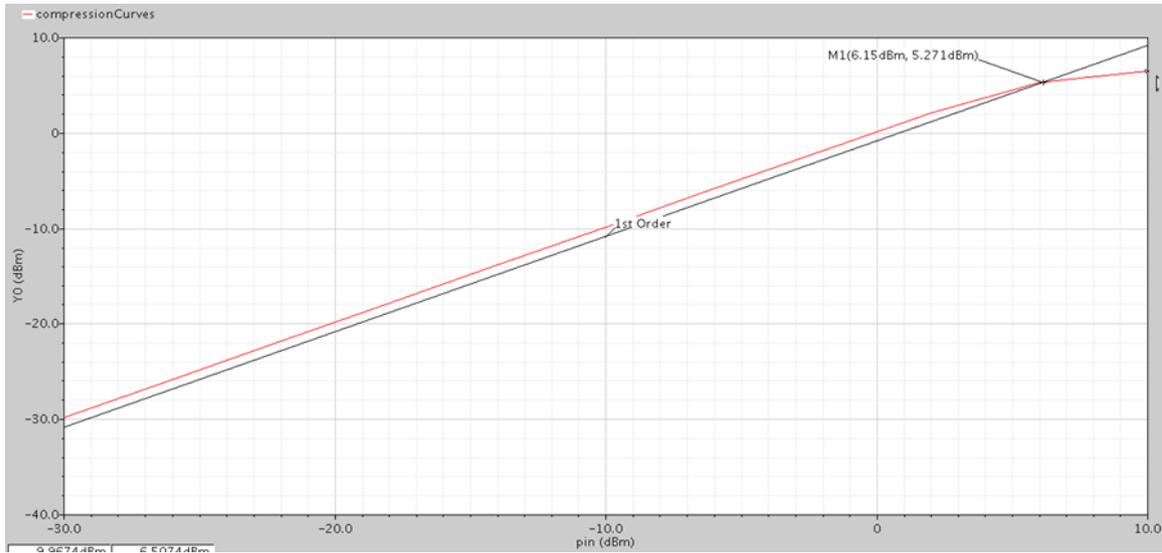


Figure 5.5.30; 1-dB comparison point FF

5.4.8 Layout

After presenting design of PGA and corners simulations, and PGA pass these two stages, PGA now ready for Layout, Layout is a bit harder than schematic design, because we have to take effect of parasitic cap at each node and parasitic resistant of wires which degrade performance of PGA, during this thesis PGA floor planning and layout implement on tsmc13rf kit without verification like DRC nor LVS, because cadence version doesn't support them.

5.4.8.1 Floor Planning

Floor planning is critical step in layout process, because choosing good placing for transistors can reduce effect of parasitic and PVT, Figure 5.31 shows floor planning for Folded Cascode OTA.

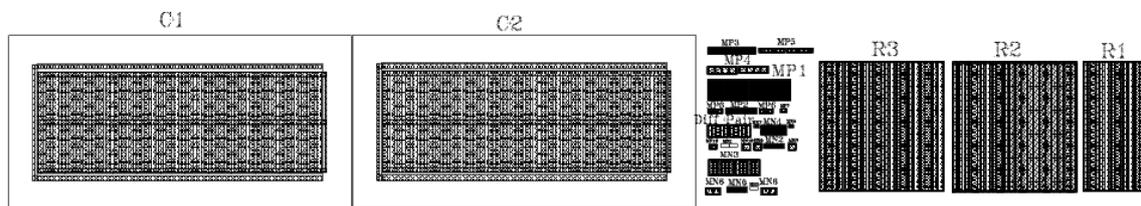


Figure 5.5.31; Floor Planning of OTA

5.4.8.2 Total PGA layout

Layout of PGA divides into two main parts

1. Layout of Folded Cascode OTA
2. Layout of PGA resistive feedback, switches and inverters

Figure 5.32 shows layout of Folded Cascode OTA after implementing layout techniques like common centroid, interdigitated & add dummy, Figure 5.33 shows complete layout of PGA layout of Folded cascode and PGA untested so it may has some errors.

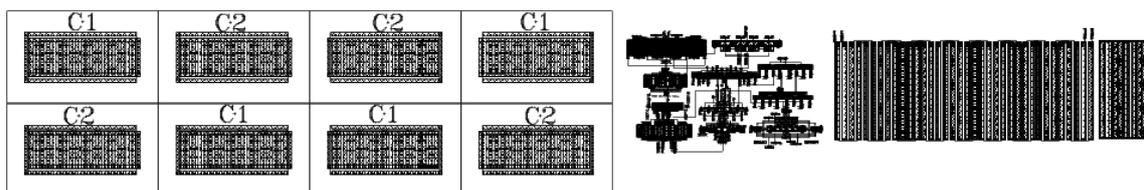


Figure 5.5.32; Layout of Folded Cascode OTA

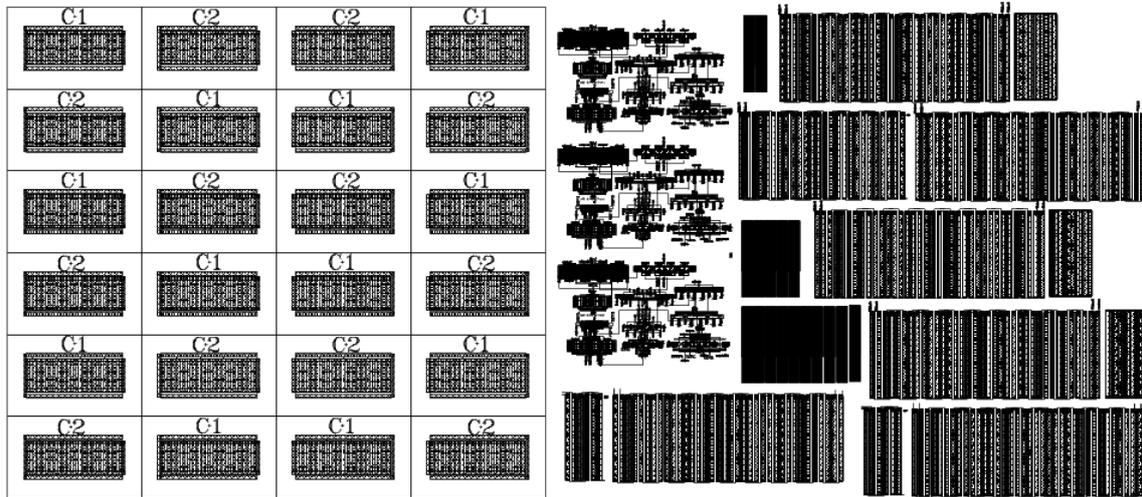


Figure 5.5.33; Layout of PGA

5.5 Peak Detector

5.5.1 Introduction

Peak detectors or, better said in our application field, envelope detectors, are a key block in gain control and spectral energy estimation, High performance envelope detectors are required to obtain the amplitude of the signal in a great variety of circuits, mainly for gain control circuits and spectral energy estimation over a variety of applications, such as hearing aids, cochlear implants and especially in wireless communication receivers.

A new generation of dynamically varying analog circuits needs high performance envelope detectors to optimize signal-to-noise ratio and power dissipation, such as dynamic gain scaling (syllabic companding), dynamic impedance scaling, dynamic biasing and dynamic structure variation ^[6].

The selection of a feed forward control structure considerably increases the envelope detector performance requirements, since its input dynamic range and, consequently, the linearity demand is enlarged. Thus, this basic cell becomes still more essential for the correct performance of feed forward AGCs ^[2].

Both Peak detectors proposed should have settling time < 25uSec, min power consumption < 3mA, and neglected Droop time as we use GFSK modulation so no variation in amplitude in this modulation technique.

5.5.2 Design Parameters

Peak detectors has two main specification

- **Settling Time**

And settling time split into

1. Attack Time

Attack-time, defined as the time required by the circuit to respond to a positive stepwise change in the input signal envelope, is dependent on the slew rate which is an indicator of the speed of the circuit. Higher slew rate offers a higher speed charging the hold capacitor.

2. Release Time

Release-time, defined as the time required to respond to a negative stepwise change in the input signal envelope, depends on the capacitor discharge current and the capacitor size itself. This parameter defines the peak detector capacity to track a decline in the input signal amplitude.

- **Droop Time**

Droop is a slow discharge from the hold capacitor C (Figure 5.34). Discharge can be unintentional, through a leakage current or the path provided by the following stage, or intentional, through a big resistor R or small current source I_b . The droop rate (dV_{peak}/dt) is proportional to $1/RC$ in the case of the big resistor, or to I_b/C for the small current source.

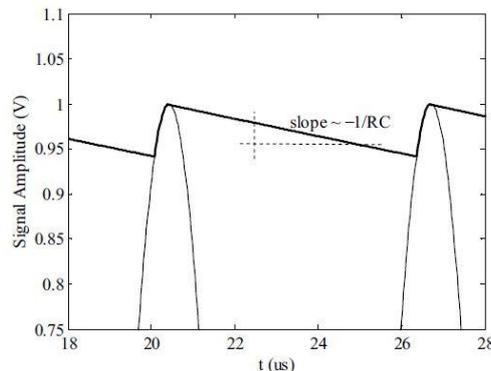


Figure 5.5.34; Peak Detector transient response

5.5.3 PD Topologies

5.5.3.1 Conventional RC Peak Detector

The conventional diode-RC circuit, shown in Figure 5.35, is the simplest structure that can work as a peak detector. In this circuit, when the input signal is above the output signal plus the diode threshold voltage V_{td} , the diode is equivalent to a resistor and the capacitor is charged by the current which flows from the input. On the other hand, when the input signal is smaller, the diode is in cut off operation region and the capacitor load is slowly discharged through the resistance, this structure not used in deep submicron technology.

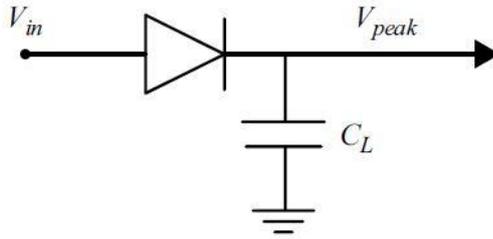


Figure 5.5.35; Conventional RC Peak Detector

5.5.3.2 Op-amp plus diode based peak detector topology ^[7]

To improve the accuracy, an op-amp can be employed in feedback configuration so that the diode output is connected to the op-amp negative input, as shown in Figure 5.36. This op-amp and diode based peak detector reduces the threshold voltage to V_{td}/A_0 , where A_0 is the op-amp DC gain. Hence, this circuit can closely track V_{in} while its value is above the capacitor voltage, V_{peak} . Alternatively, when the input signal is below V_{peak} , the op-amp output goes to negative saturation, the diode goes to the cut-off region and the capacitor voltage is slowly discharged in the same way as previously explained.

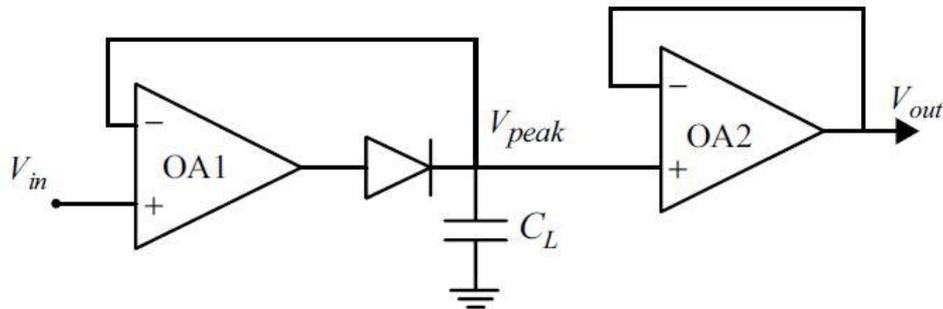


Figure 5.5.36; Op-amp plus diode based peak detector topology

5.5.3.3 Conventional Open-Loop Envelope Detectors

Figure 5.37 shows the cell of the peak detector which is suitable for operation up to frequencies of 100MHz. It consists of a slow source follower composed of M_3 , I_L , C_L and the feedback transistor M_{1a} ^[8]. The transistor M_{1b} outputs a copy of the current in M_{1a} , while transistors $M_{2a,b}$ are introduced to obtain a higher output resistance and thus, to minimize the offset current at the output. The source follower can follow descending signals in the input voltage rapidly because of the exponential dependence of the current of M_3 on its gate voltage. However, the small current I_L is slow in charging capacitor C_L ; as a result, during ascending signals in the input, the output signal is slow to respond as discharge slope is proportional to C_L/I_L . In consequence, the ripple and time constant, τ , are controlled by I_L and C_L .

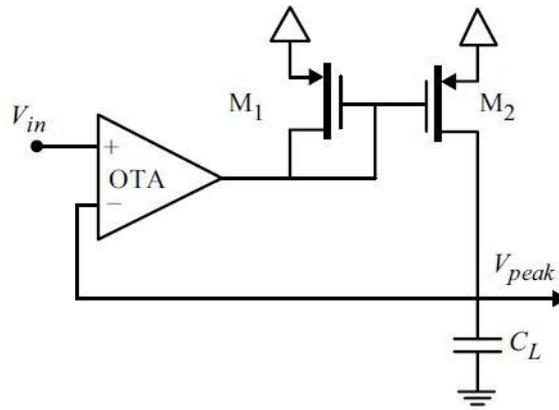


Figure 5.5.38; OTA plus current mirror closed-loop topology

Figure 5.39 show proposed peak detector in [9].

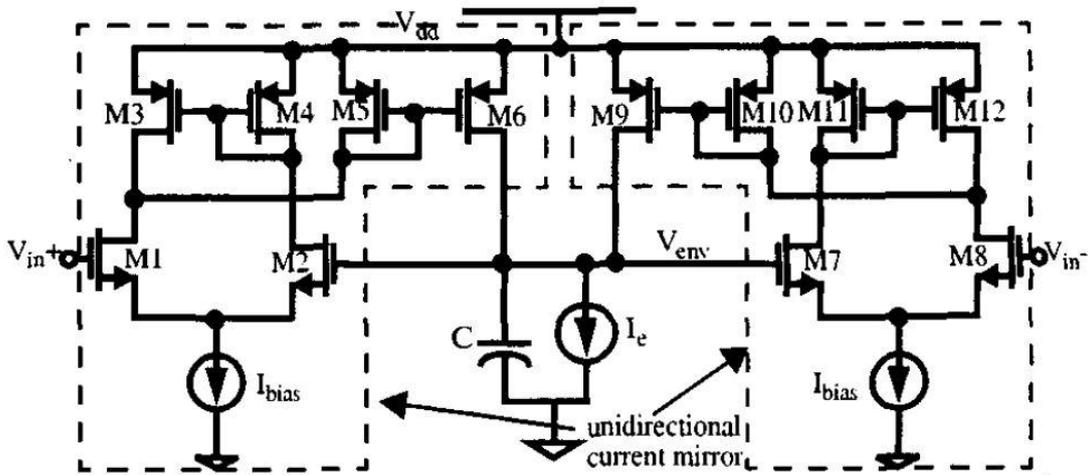


Figure 5.5.39; current mirror closed loop peak detector

5.5.4 Required Specifications

Specification	Required
Settling time	< 25 uSec
Power Consumption	< 3 mA
Input Amplitude	> 300 mV
Operating Frequency	> 1.44MHz
Ripples	< 4mV

Table 5.8; Required Specification for Peak Detector

5.5.5 Design Approach

5.5.5.1 PD topology Number 1

Our receiver architecture is low IF, has center frequency 1.14MHz and BW 300KHz, this IF frequency consider small, so proposed PD1 based on sample RC peak detector, Proposed PD1 is Op-amp plus diode based peak detector topology shown in Figure 5.35, Op-amp used in this topology replaced by Folded Cascode OTA, total circuit shown in Figure 5.40, Diode replaced by diode connected transistor.

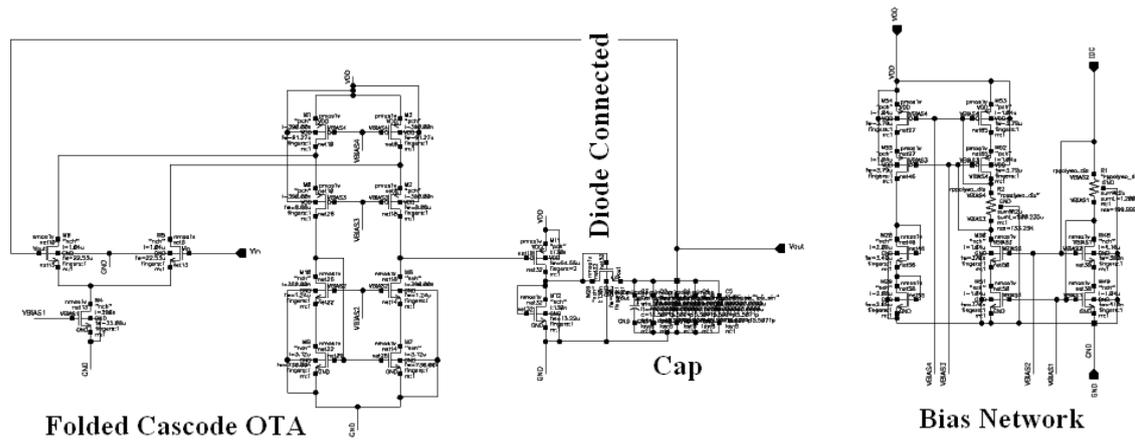


Figure 5.5.40; Folded Cascode OTA used in PD1

5.5.5.2 PD topology number 2

Second topology used for peak detector made for enhancing performance of AGC loop, by reducing power consumption and increasing maximum operating frequency, Proposed PD2 topology is Closed-Loop Envelope Detectors, it's proposed in [9] as shown in Figure 5.39, in paper he used simple Diff amplifier so It's gain is small so performance is low, to enhance this performance I replace Diff amplifier with Folded Cascode OTA, which has gain > 66dB so performance is better than in paper, total circuit used in PD2 shown in Figure 5.41, schematic of Folded cascode shown in Figure 5.42.

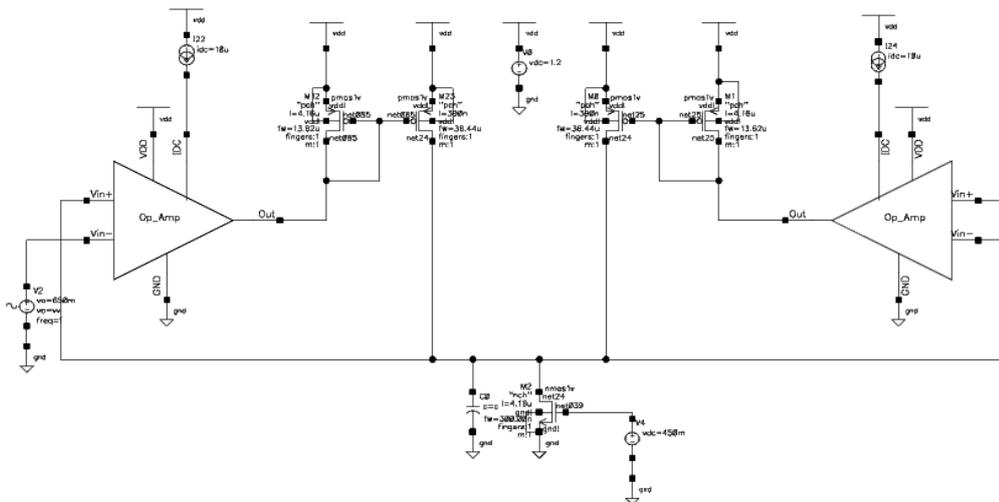


Figure 5.5.41; Proposed PD2 based on closed loop current mirror

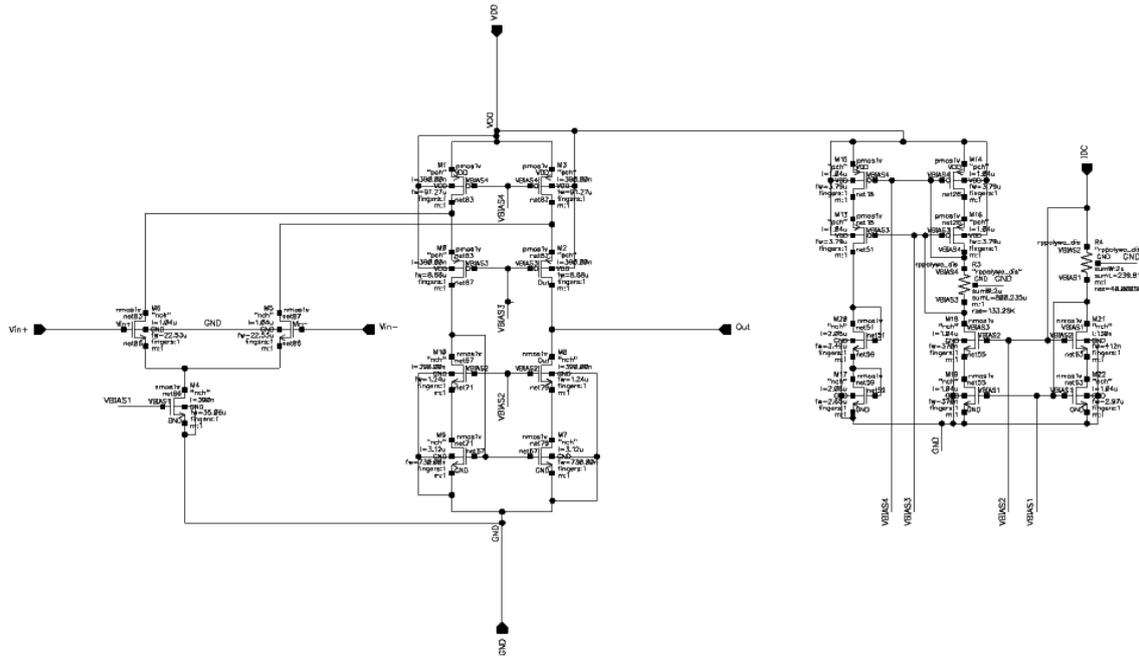


Figure 5.5.42; Folded Cascode OTA used in PD2

5.5.6 Simulation Results

5.5.6.1 PD 1

Figure 5.43 show settling time for PD1 using 431 KHz sine wave with 100mV amplitude as shown settling time is 3uSec.

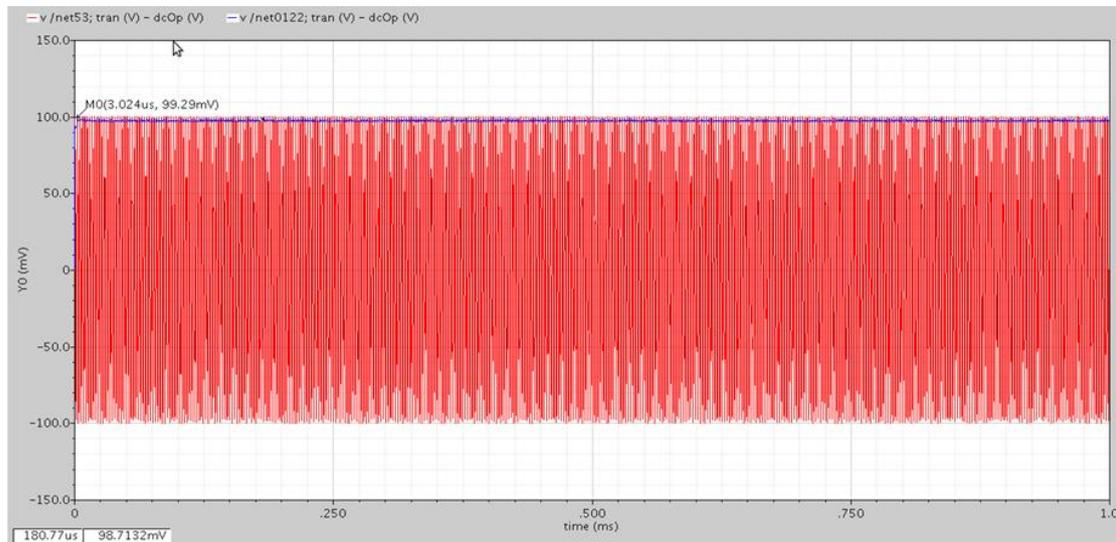


Figure 5.5.43; Settling Time PD1 431KHz

Figure 5.44 show attack time for PD1 using 431 KHz sine wave with 33mV step, as shown attack time is 0.1uSec.

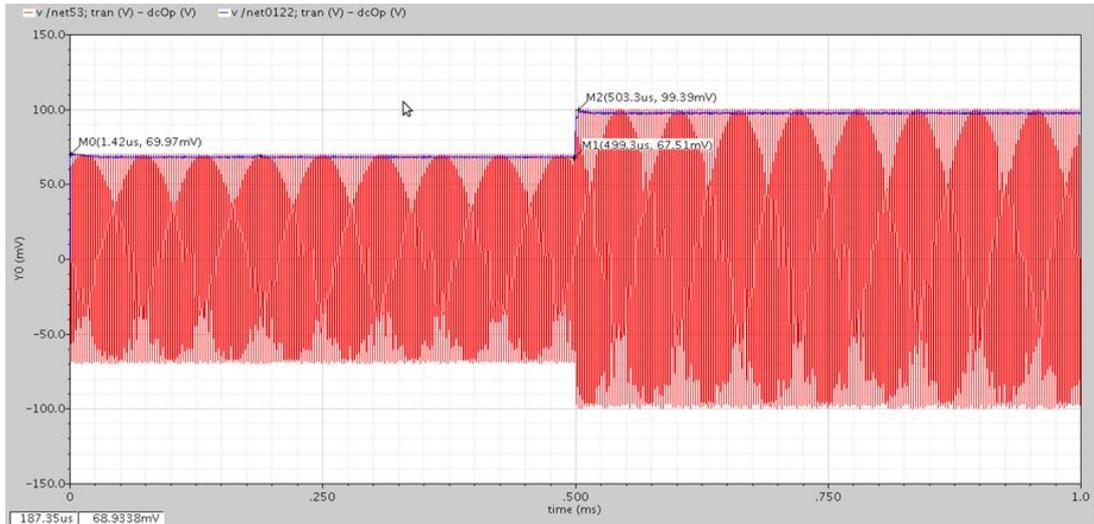


Figure 5.5.44; Attack Time PD1 431 KHz

Figure 5.45 show release time for PD1 using 431 KHz sine wave with 33mV step down, as shown attack time is 370uSec.

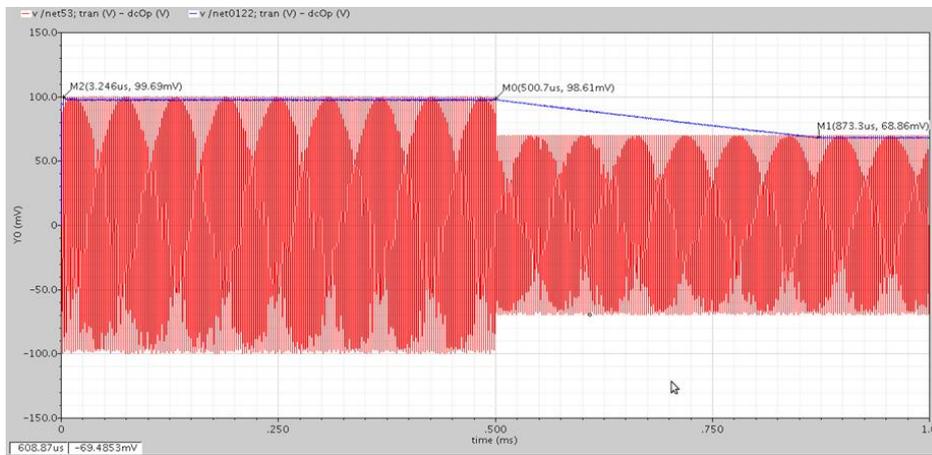


Figure 5.5.45; Release Time PD1 431KHz

Figure 5.46 shows ripples in amplitude for 100mV sine wave, ripples < 1mV.

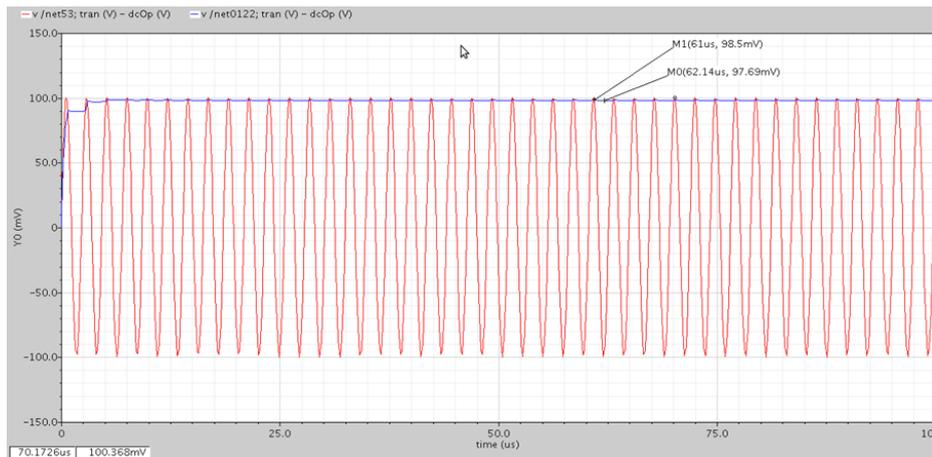


Figure 5.5.46; Ripples in 100mV Amplitude

Figure 5.47 shows Loop gain of CMFB

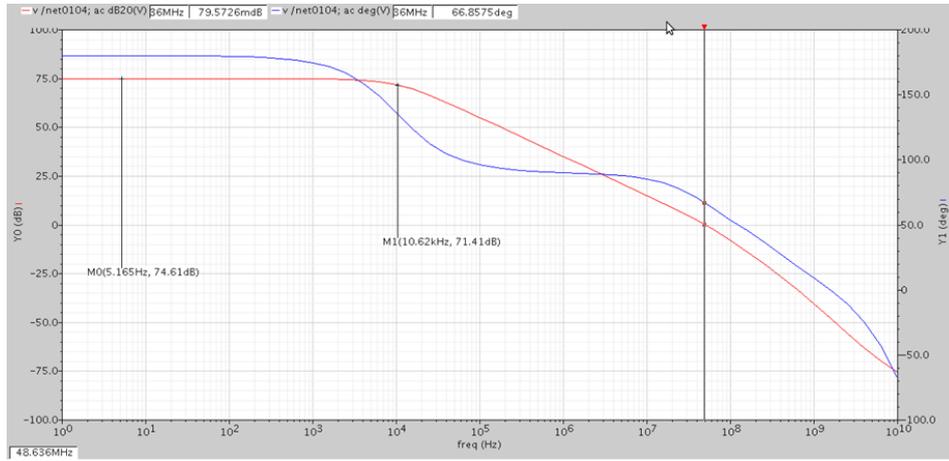


Figure 5.5.47; Loop gain of CMFB

5.5.6.2 PD 2

Figure 5.48 show settling time for PD2 using 40 MHz sine wave with 100mV amplitude as shown settling time is < 1uSec.

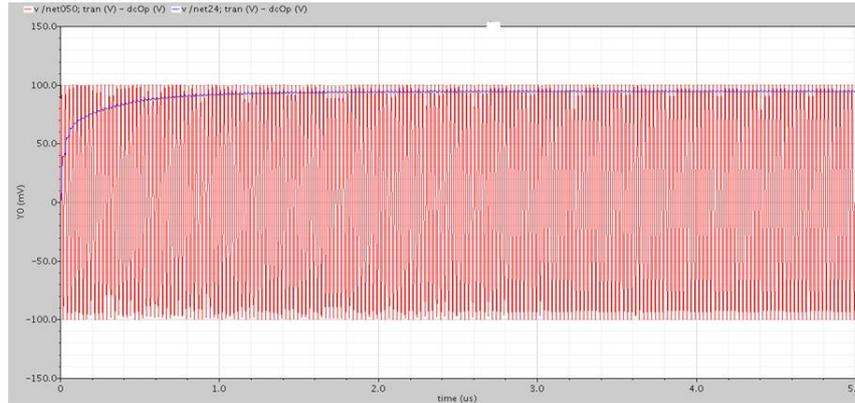


Figure 5.5.48; Settling Time PD2 40MHz

Figure 5.49 show attack time for PD2 using 2 MHz sine wave with 50mV step up, as shown attack time is 0.1uSec.

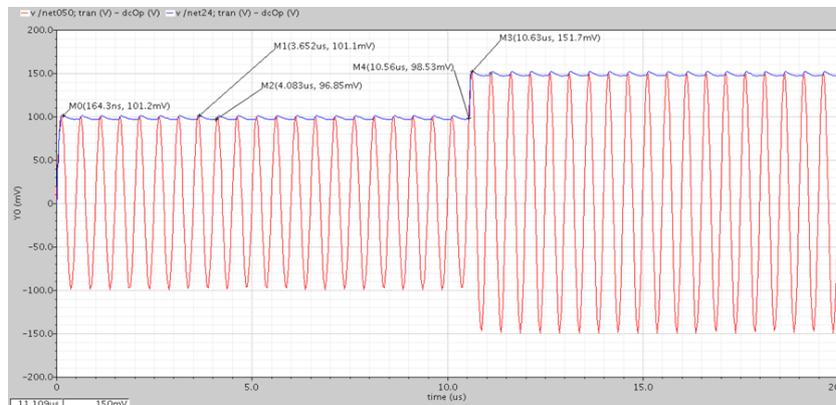


Figure 5.5.49; Attack Time PD2 2MHz

Figure 5.50 shows release time for PD2 using 2 MHz sine wave with 50mV step down, as shown attack time is 5uSec, and ripples equal 4.24mV, PD2 used same Folded Cascode OTA of PD1.

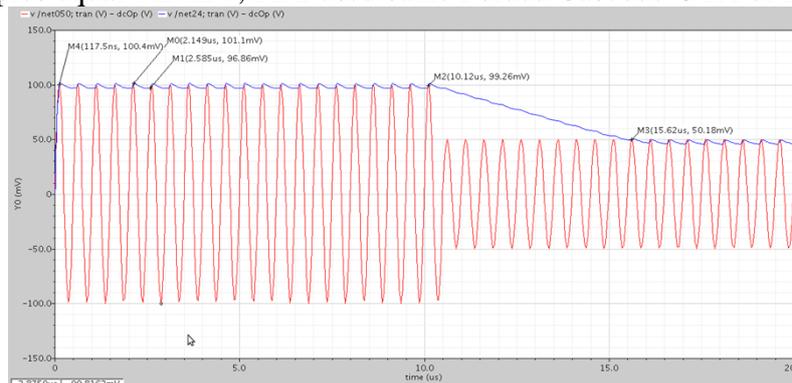


Figure 5.5.50; Release Time PD2 2MHz

Specification	Required	PD1	PD2
Settling time	< 25uSec	3uSec	1uSec
Power	< 3mA	2mA	0.45mA
Consumption			
Input Amplitude	> 100mV	> 100mV	> 100mV
Operating Frequency	> 1.44MHz	2MHz	40MHz
Ripples	< 4mV	1mV	4.5mV

Table 5.9; Comparison between PD1 and PD2

5.5.7 Corners Results

5.5.7.1 PD 1 & PD2

Figure 5.51 shows variations in open loop gain of folded cascode OTA, Figure 5.52 shows variations in loop gain of CMFB of Folded Cascode, Figure 5.53 shows variations in PM of CMFB of Folded cascode OTA using Mat lab, PD1 pass all corners, but PD2 fail at some corners, ocean script codes for corners simulation at Appendix A

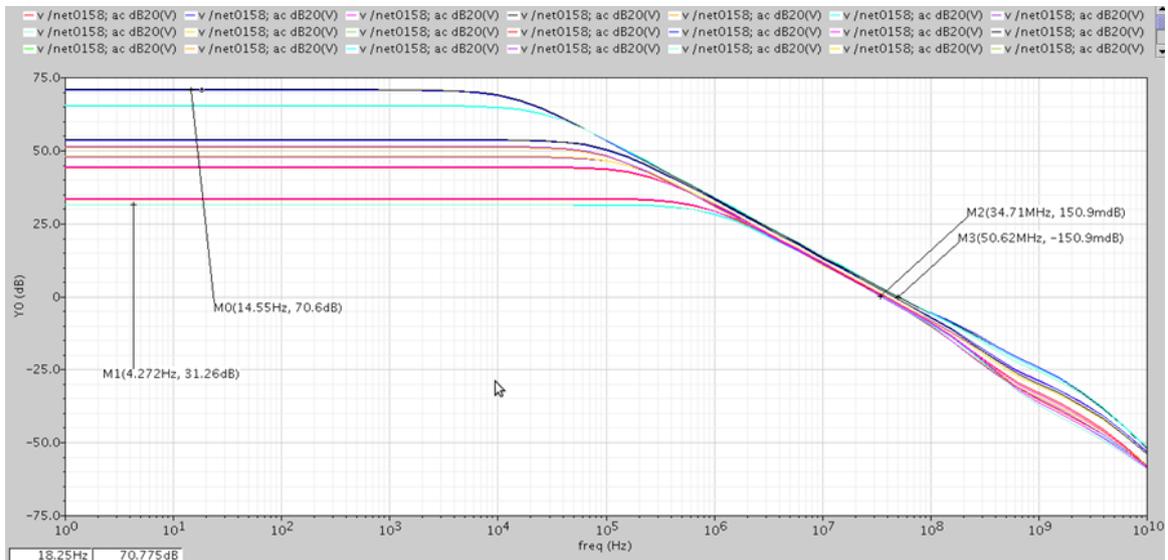


Figure 5.51; Open Loop Variation across corners

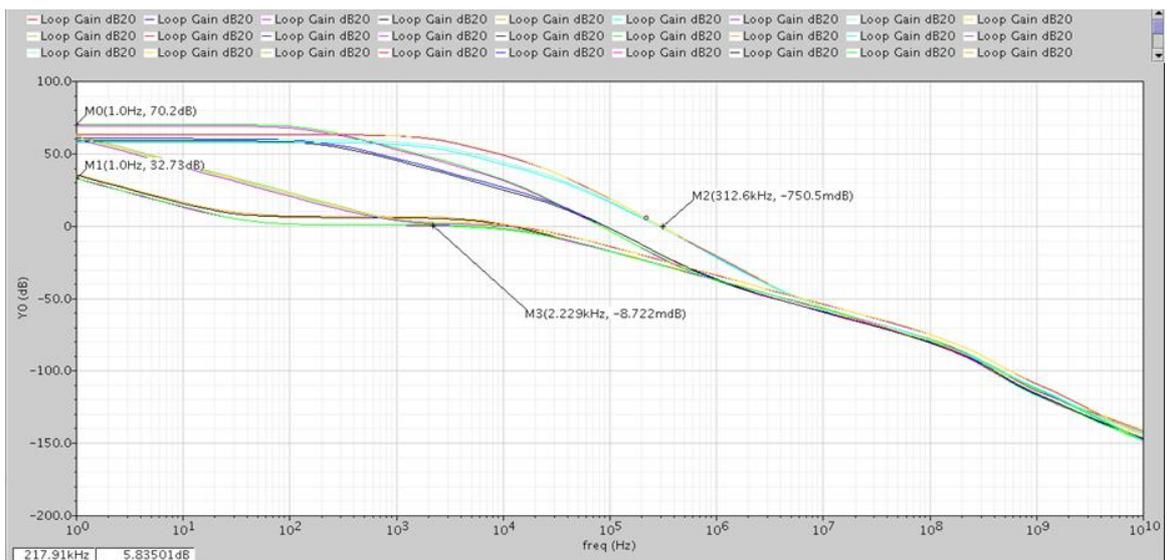


Figure 5.52; Loop gain of CMFB

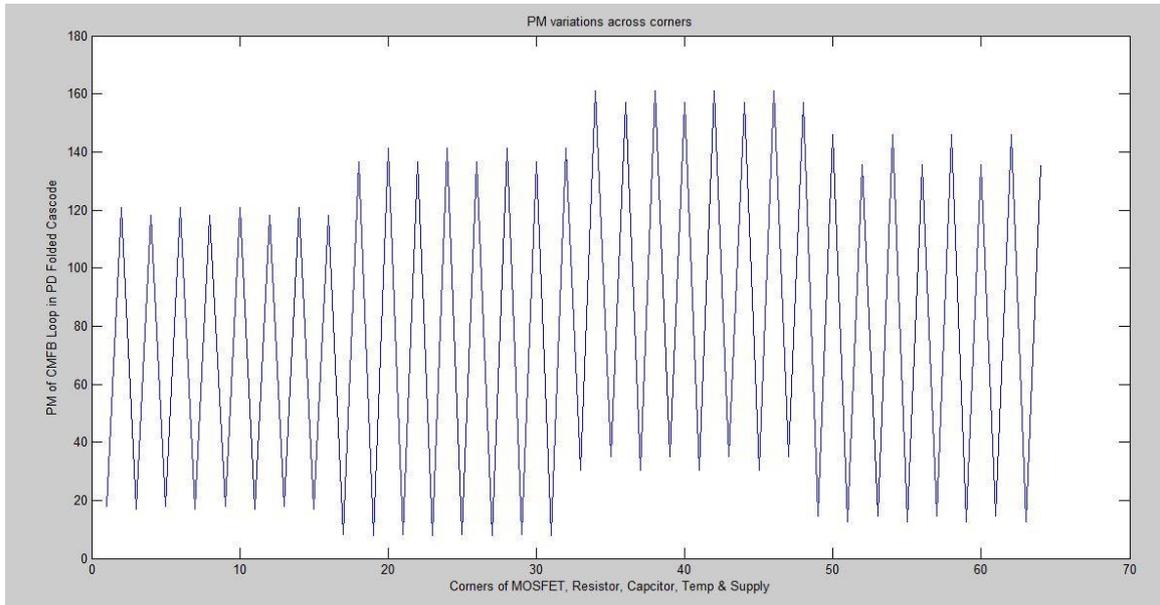


Figure 5.5.53; PM variation across corners of PD

5.5.8 Layout

5.5.8.1 Floor Planning

5.5.8.1.1 PD 1

Floor planning is critical step in layout process, because choosing good placing for transistors can reduce effect of parasitic and PVT, Figure 5.5.4 shows floor planning for PD1.

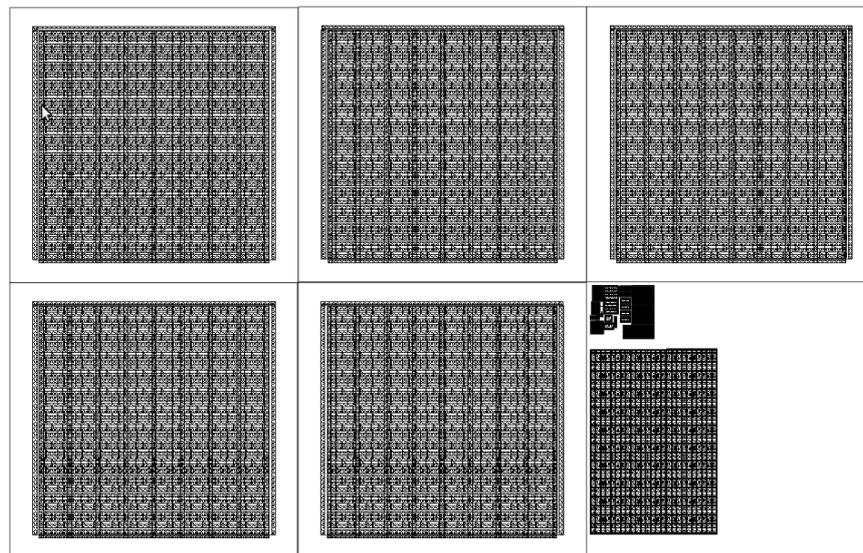


Figure 5.5.54; Floor planning of PD1

5.5.8.1.2 PD 2

Figure 5.55 shows floor planning for PD2.

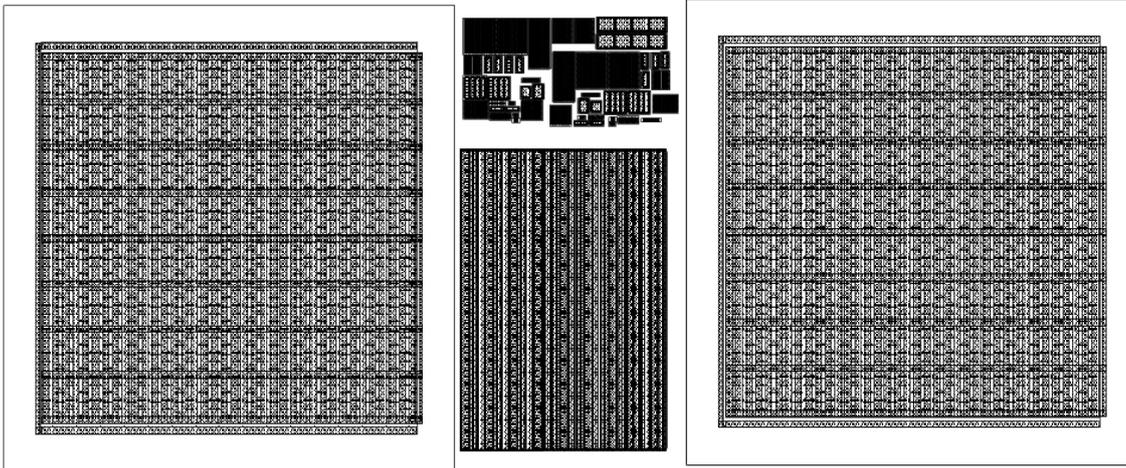


Figure 5.5.55; Floor Planning of PD2

5.5.8.2 Total PD layout

5.5.8.2.1 PD 1

Figure 5.5.6 shows layout for PD1, this layout is untested due to cadence version doesn't support neither DRC nor LVS.

5.5.8.2.2 PD 2

Figure 5.5.7 shows layout for PD2, this layout is untested due to cadence version doesn't support neither DRC nor LVS.

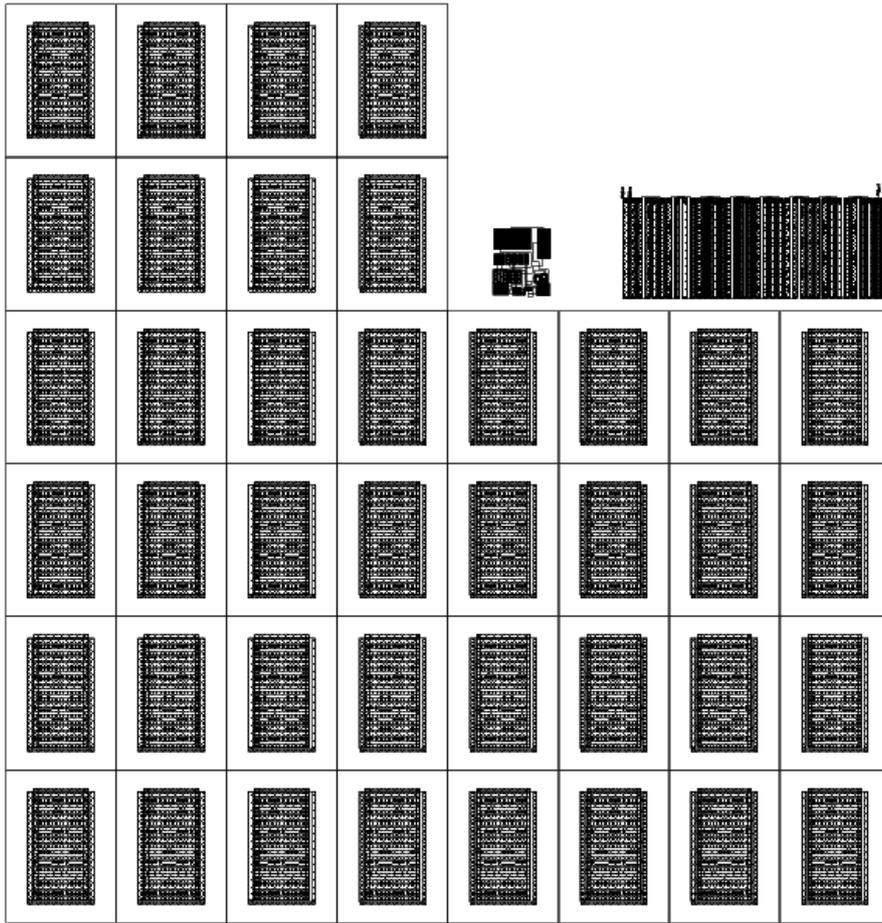


Figure 5.5.56; Layout of PD1

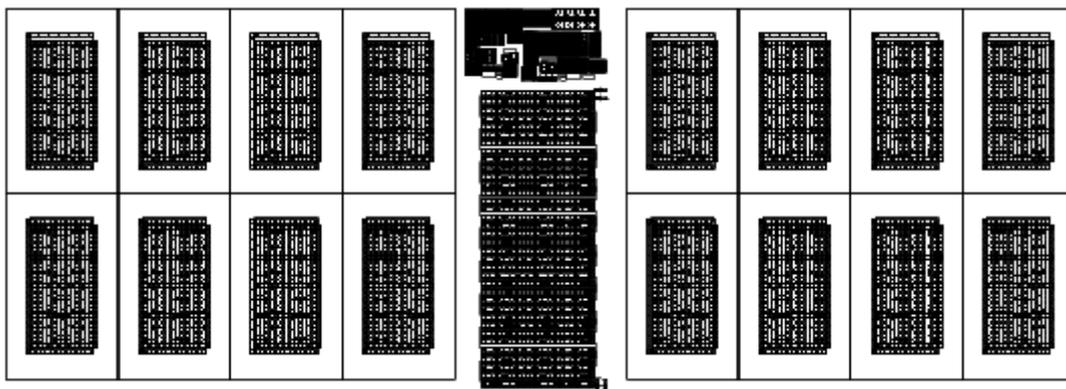


Figure 5.5.57; Layout of PD2

5.6 Flash A/D

5.6.1 Introduction

Digital control approach groups mainly two options. First topology is that where the AGC loop is fully implemented inside the DSP^[10]. This option requires a different type of work completely oriented to digital designers, so it will not be gone into here as we understand it is beyond the aims of this book. The second option however, takes the output of the peak detector and, making use of a simple digital block, generates the digital word required to manage the PGA^[11], during this thesis a new digital control method adopted, using flash A/D after Peak detector we can estimate signal amplitude and give it required gain.

5.6.2 Modeling of Flash A/D

Proposed Flash A/D in this thesis modeled using Verilog-AMS, Flash A/D consist of resistive network then comparators, followed by thermometer, Verilog-AMS code of comparator and thermometer found in Appendix B, total Flash A/D shown in Figure 5.58.

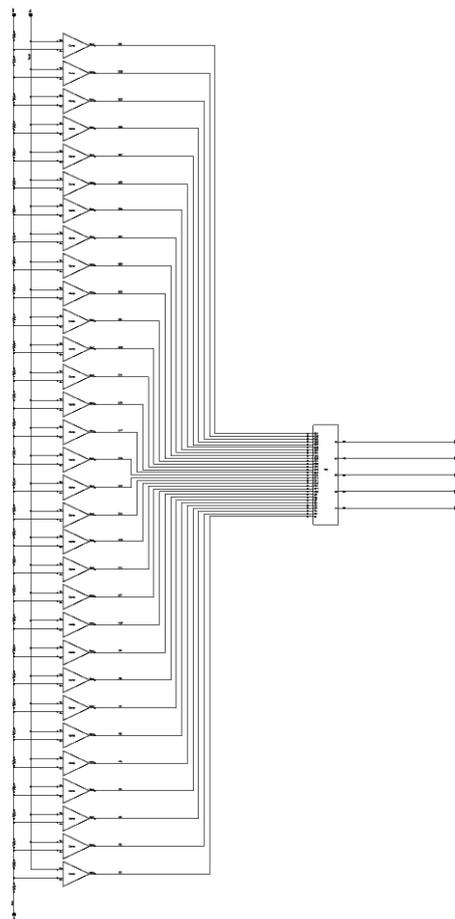


Figure 5.5.58; Flash A/D used in AGC

5.6.3 Design Parameters

In AGC loop the most important specification required from Flash A/D is number of bits, we need gain range from PGA 0 to 54dB with step 2dB, around 28 gains setting, so we set number of bits for flash A/D equal 5 so we have 32 different amplitude we can take decision for it and change can for them.

5.6.4 Simulation Results

Figure 5.59 shows simulation result for flash A/D.

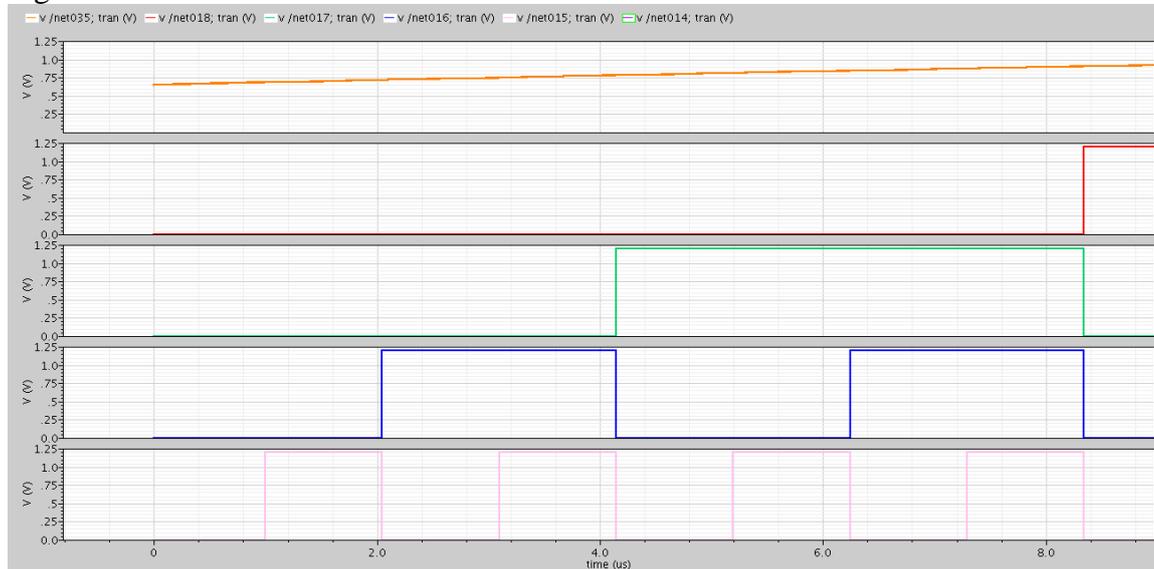


Figure 5.5.59; Flash A/D simulation result

5.7 Digital Control

Digital control in AGC gives more accurate results than analog control, proposed AGC in this thesis use digital control, but not from DSP as made in [10], a new method proposed by generating digital control word from Flash A/D then use small combinational logic block which give digital control word to PGA, this combinational block modeled using Verilog-AMS, Verilog-AMS used found in Appendix B, digital control done by simulating all 32 amplitudes and change gain till output amplitude of PGA be within required range for ADC.

5.8 Simulation Result of all AGC blocks

5.8.1 Control Methodology

Proposed Automatic Gain Control Loop control gain of LNA & PGA, total AGC system shown in Figure 5.60, we have two scenarios

- **PA Leakage through duplexer or Blocker level higher than desired signal**

If coming blocker is very high and desired signal is small, and peak detector sense after Filter only it will see small desired signal, and give high gain control to LNA, so this cause LNA, Mixer & Filter saturated, and desired signal faced high distortion, so solution for this problem by adding another peak detector sense before filter and compare both amplitude if before filter signal is higher than output signal so this mean blocker level is high so we reduce gain of front end receiver, if before filter signal is smaller than output signal so this mean blocker level is high so we increase gain of front end receiver.

- **Desired signal level higher than blocker signal**

This is normal case, peak detector sense after filter and compare it with before filter signal amplitude, and give gain control signal to front end receiver blocks, if signal is very small so digital control increase gain of PGA so that signal level output of PGA be within required range of ADC

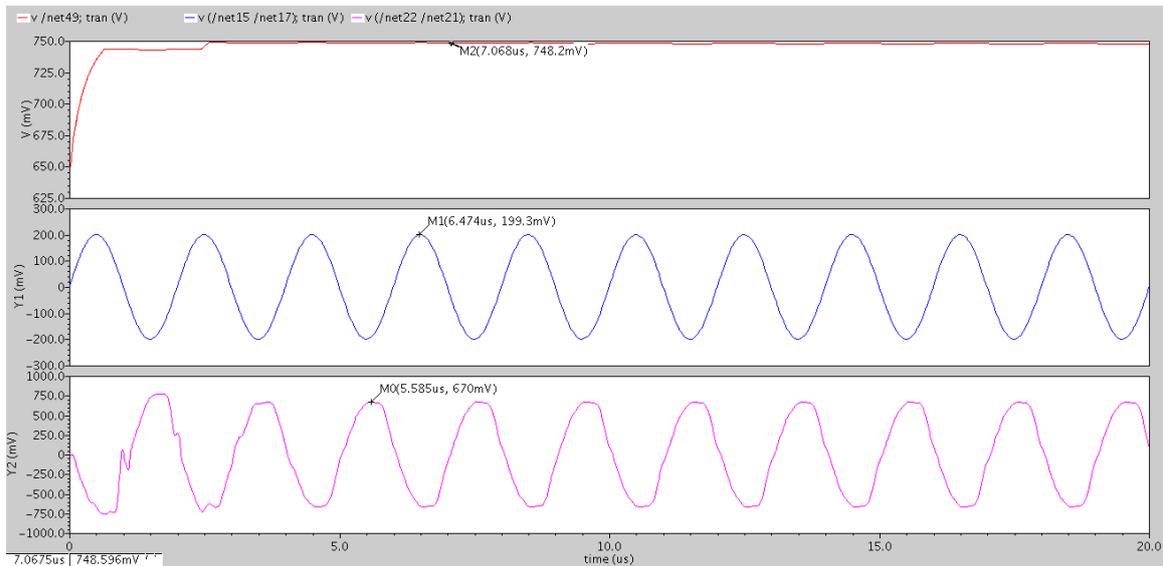


Figure 5.5.62; Max Signal Received

5.9 Conclusion

AGC Loop is important block in receiver, because it's deliver required amplitude to ADC, when received signal strength varying, control methodology take effect of PA power leakage, blocker level higher than received signal, small received signal without blockers, during this thesis a new digital control method proposed, based on flash A/D instead using DSP as control, PGA & PD achieved required specifications, and simulation result of all receiver system found at Receiver system verification.

5.10 Reference

- [1] Hassan O. Elwan, Tuna B. Tar_m2, and Mohammed Ismail "A Digitally Controlled dB-Linear CMOS AGC for Low Voltage Mixed Signal Applications" Circuits and Systems; Publication Year: 1998, Page(s): 423 – 425.
- [2] Automatic gain control Techniques and architectures for RF-Receivers, Mohamed Ismail, First Edition.
- [3] P. Naktongkul I and A. Thanachayanont "A 1-V 25-dB 100-MHz CMOS Variable Gain Amplifier Cell"
- [4] Ashis Kumar Mal Rishi Todani "A Digitally Programmable Folded Cascode OTA with Variable Load Applications"
- [5] ANAND KAMRA "Design of Low power operational amplifier" master thesis Regn. No. 600861005.
- [6] J. P. Alegre, S. Celma, M. T. Sanz and P. Martinez "A Low-Ripple Fast-Settling CMOS Envelope Detector"
- [7] H.-C. Chow and I.-H. Wang; "High performance automatic gain control circuit using a S/H peak detector for ASK receiver"; Electronics, Circuits and Systems, 2002. 9th International Conference on; Vol. 2, pp. 429–432, Sept. 2002.
- [8] Rahul Sarpeshkar, Richard F. Lyon and Carver Mead; "A Low-Power Wide-Dynamic-Range Analog VLSI Cochlea"; Analog Integrated Circuits and Signal Processing; Vol. 16, Issue 3, pp. 245–274, Aug. 1998.
- [9] H.-Y. Cheung, K.S. Cheung, and J. Lau; "A low power monolithic AGC with automatic DC offset cancellation for direct conversion hybrid CDMA transceiver used in telemetering"; Circuits and Systems, 2001. ISCAS '01. Proceedings of the 2001 International Symposium on; Vol. 4, pp. 390–393, May 2001.
- [10] H. Elwan, T.B. Tarim, M. Ismail; "Digitally programmable dB-linear CMOS AGC for mixedsignal applications"; IEEE Circuits and Devices Magazine; Vol. 14, Issue 4, pp. 8–11. Jul. 1998.
- [11] Willy Hioe, Kenji Maio, Takashi Oshima, Yoshiyuki Shibahara, Takeshi Doi, Kiyoharu Ozaki, and Satoshi Arayashiki; "0.18- μ m CMOS Bluetooth Analog Receiver With 88-dBm Sensitivity"; Solid-State Circuits, IEEE Journal of; Vol. 39, Issue 2, pp. 374–377, Feb. 2004.

6 ADC

6.1 Abstract:

New emerging communication systems and applications place very stringent requirements on energy consumption. Such requirements led to exploiting a portion of the analog-to-digital converter design space that has received little attention in the past decade, namely moderate resolution and speed, yet very low power ADCs. Those ADCs are critical components in large-scale wireless sensor networks used in a large set of applications ranging from tracking wildlife populations to measuring and predicting weather patterns. The successive approximation analog-to-digital converter with its minimal analog circuitry emerges as a potential candidate to satisfy their low power specifications.

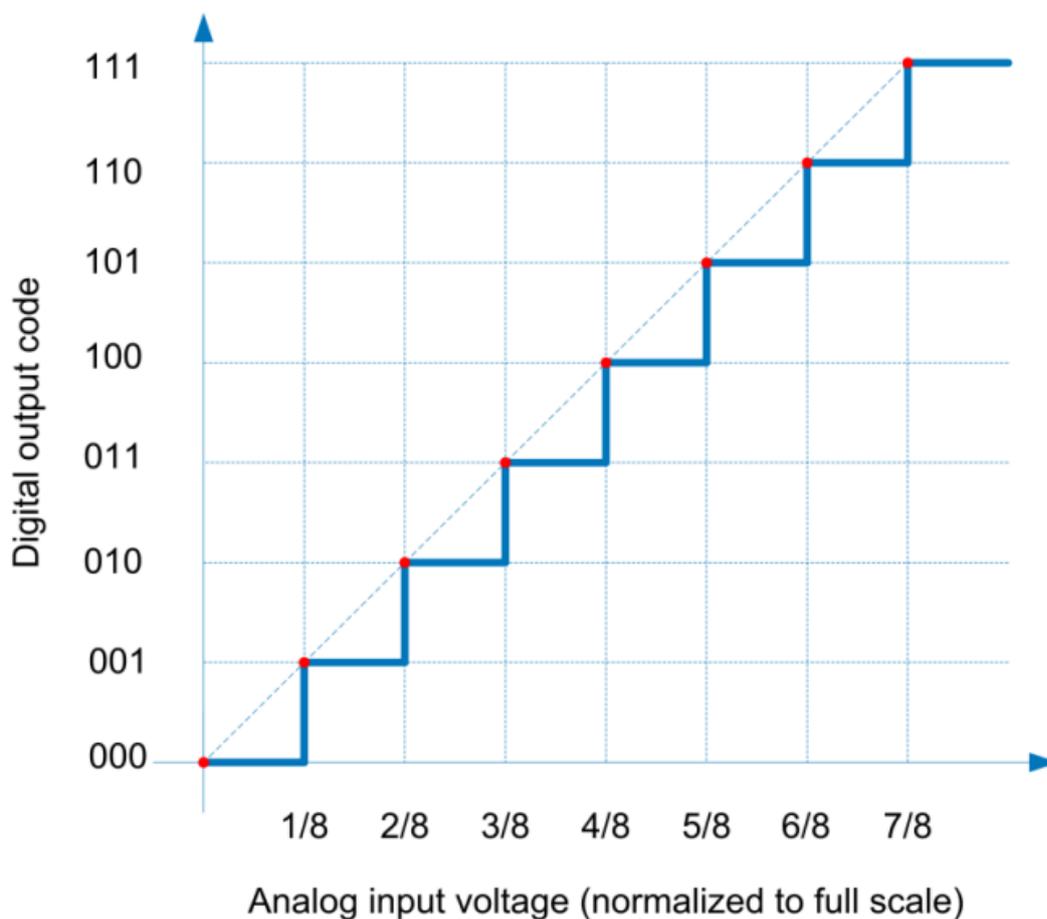


Figure 6.6.1; Ideal transfer function of a 3-bit ADC

6.2 New challenge in ADCs design

Trends in many energy-limited applications and communication systems, such as wireless sensor networks, micro-robotics and software defined radios, add more de-sign challenges to improve flexibility, system integration and bandwidth efficiency, yet with lower power consumption and smaller area to meet cost target.

Typical requirements of these systems architectures are met with a medium-resolution, medium resolution low power consumption analog-to-digital converter (ADC) to extend the duration of the system battery power operation.

Power saving can be achieved at both system architecture and circuit level design. At the architecture level different ADC topologies consume different power for the same specifications. The successive approximation ADC exhibits the lowest power consumption reported in literature due to its minimal active analog circuit requirement.

Decreasing the supply voltage is an effective way to realize a low power design. The power of digital circuits directly benefits from supply voltage reduction. However, the low supply voltage makes the analogy circuit design more difficult. When the sum of the absolute value of the NMOS threshold voltage and that of the PMOS is larger than the supply voltage, conventional analogy switches made of transmission gates may not be fully turned on as in the case of higher supply voltages. In addition, some useful design techniques such as cascoding and gain boosting may not be applicable because of the limited signal swing.

6.3 High speed ADCs topologies:

As far as the system requirements are concerned, a long list of specifications can be prepared. many distinct ADC topologies can be found in the literature such as Flash , folding , pipeline , sigma delta and sar adc.

Resolution, sampling rate, bandwidth, power consumption, noise, area, clock latency, scalability, complexity, technology, etc. determine the most feasible topologies for that particular system.

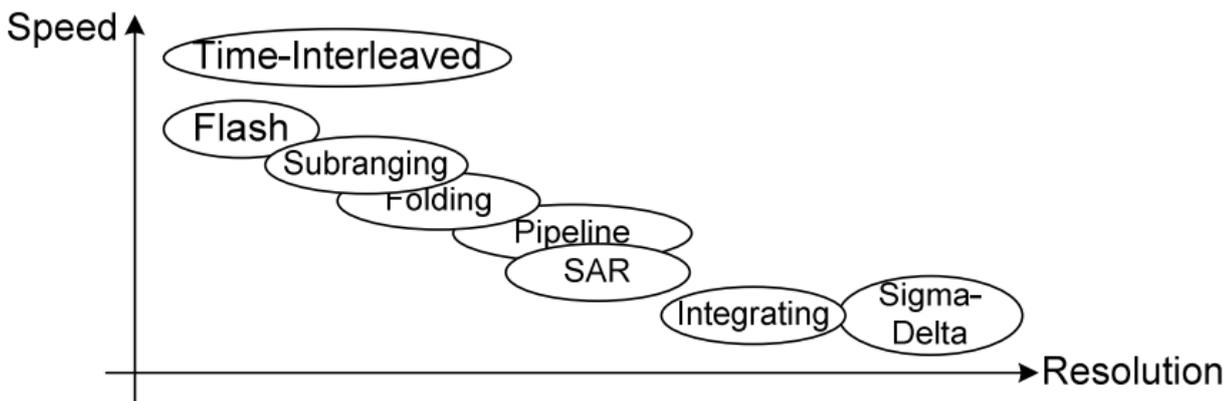


Figure 6.6.2 : ADC classification based on speed and resolution

6.3.1 Flash ADC:

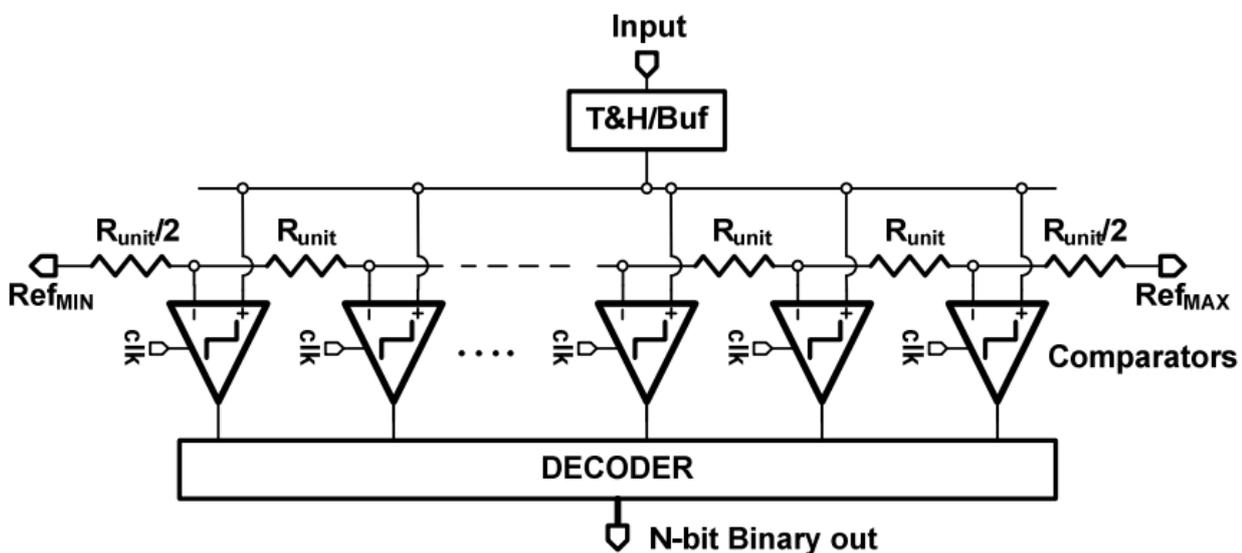


Figure 6.6.3 : Schematic of a flash ADC

Flash ADCs are the fastest single standing ADC topology, whose operation relies on the parallel decision of a number of comparators. Consist of $2^N - 1$ comparator, a resistive ladder, a track & hold (optional), a buffer (optional), and digital logic.

6.3.2 Flash ADC Specifications:

- Low to moderate resolution.
- High frequency.
- High power consumption.
- Large area.
- High delay.
- Lot of analogy blocks.

6.3.3 Pipeline ADC:

Don't use in high speed ADCs. Pipeline is high power consumption because the need of high gain and speed of pumps for good linearity and short settling time.

6.3.4 SAR ADC:

Low power, less area and less number of analog blocks. Used for moderate resolution 8-12 bits, low power application and moderate speed.

6.3.5 Comparison of different types of ADCs:

	ADC Topologies			
	Flash	Folding	Subranging	Time-Inter.
Speed	<4GS/s	1-2 GS/s	1-2GS/s	<25GS/s
Resolution	<7 bits	<12 bits	<12 bits	<10 bits
FoM	1.5-2 pJ/conv.	2 pJ/conv.	1-1.5 pJ/conv.	0.5-1 pJ/conv.
Scalability	Yes	Partially	Partially	Yes
T&H	Required	Required	Required	Required
Calibration	Required	Required	Required	REQUIRED

Figure 6.6.4 : Comparison of high-speed ADCs

6.4 SAR ADC:

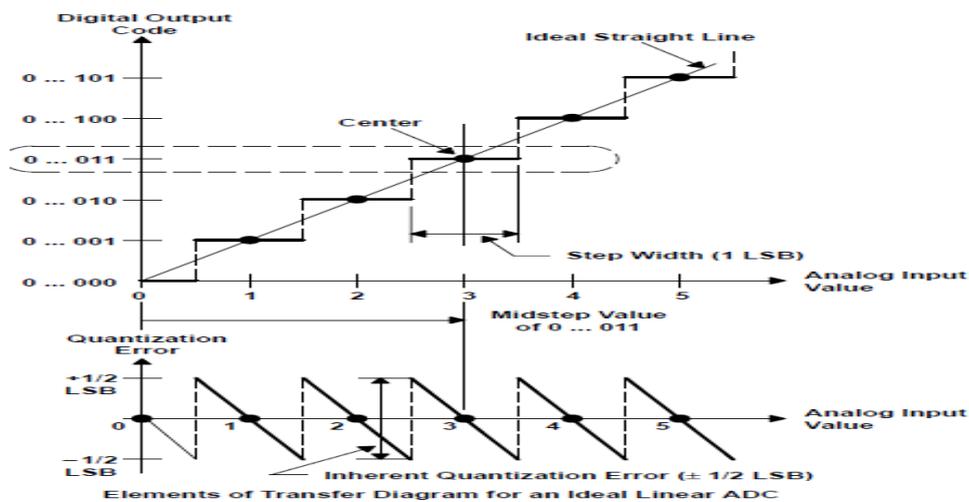


Figure 6.6.5 : Quantization noise

- **Quantization error:** The difference between the ideal straight line (infinity resolution) and the actual characteristics.

- Quantization error = $\Delta/2 = \frac{1}{2}$ LSB

$$= V_{FS} / 2^{N+1}$$

- Quantization error may be +ve or -ve $\frac{1}{2}$ LSB.

6.4.1 Real converters:

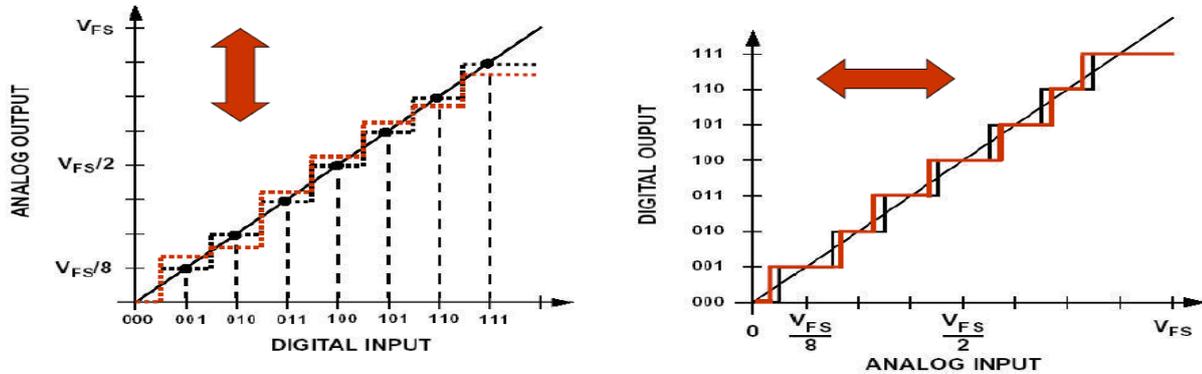


Figure 6.6.6 : ADC and DAC characteristics

- Ideal ADC introduces quantization noise.
- As N increase with a given full scale voltage, error decrease.
- Ideal dac don't introduce and quantization error.

6.4.2 Specifications:

6.4.3 Static performance:

6.4.4 Offset error:

- Offset error is the deviation of code transition from ideal one

6.4.5 DNL (differential non linearity)

- Deviation of code transition width from ideal One.
- In ideal ADC, the code width is always one, so DNL is zero.

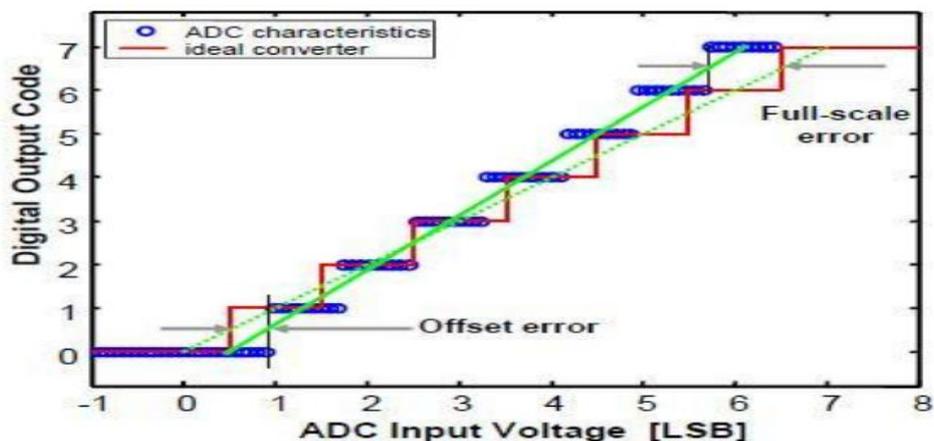


Figure 6.6.7 : offset and full scale error

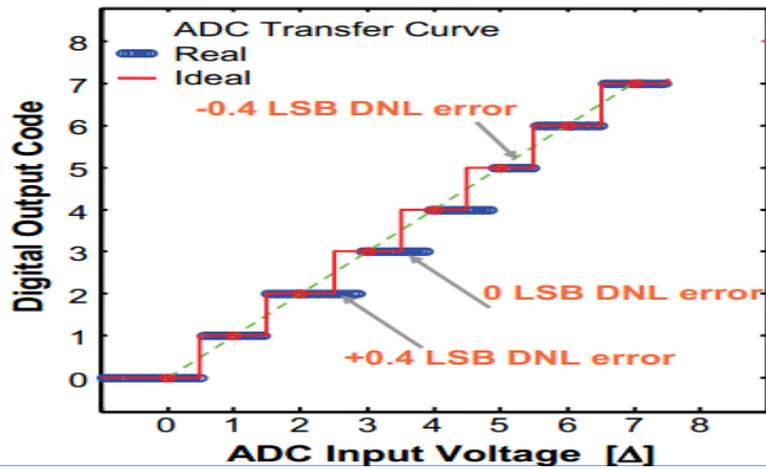


Figure 6.6.8 : DNL

6.4.6 INL (Integral non linearity)

- It is the difference between the code centres from ideal one.

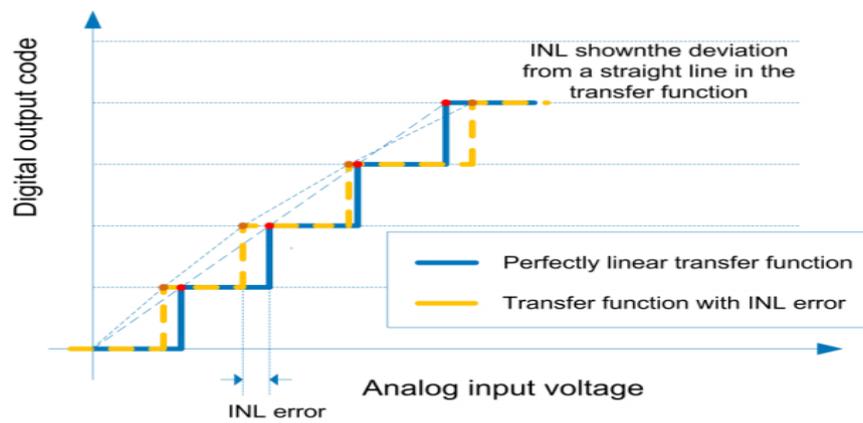


Figure 6.6.9 : INL

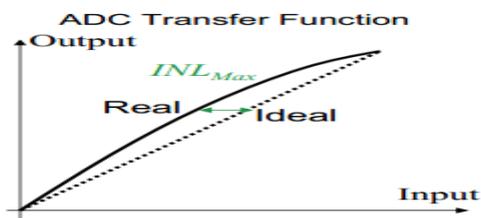


Figure 6.6.10 : INL transfer function

Missing code:

- A digital code at ADC o/p is not produced for the corresponding i/p voltage.
- A missing code result when $DNL > 1 \text{ LSB}$.

6.4.7 Full scale error:

- Difference between highest output code from actual and real transition.

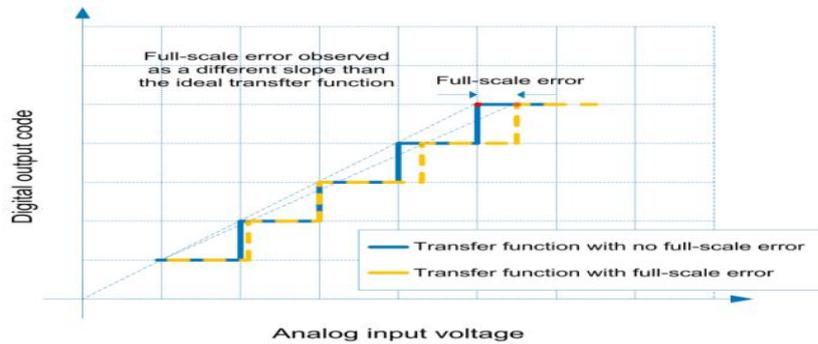


Figure 6.6.11 : Full scale error

6.4.8 Monotonicity:

- A converter is said to be monotonic if the analog amplitude level increases with decreasing the digital code (for DACs).

- For a monotonic converter: $DNL \leq 1$.

6.4.9 Dynamic performance:

6.4.9.1 Signal to Noise Ratio (SNR):

- Ratio between input signal powers to overall noise power.

- $SNR = 6 \cdot N + 1.76$.

6.4.9.2 Spurious free dynamic range (SFDR):

- Ratio of the input signal to the largest peak of spur or harmonic distortion.

- $SFDR = 20 \cdot \log(\text{Signal} / \text{Spurious})$.

6.4.9.3 SNDR:

- Ratio of the input signal to sum of total noise and harmonic component.

- Effective number of bits is directly obtained from SNADR.

- Measure performance of ADCs.

6.4.9.4 Total Harmonic Distortion (THD):

- is the ratio between the total harmonic distortion power and the power of the fundamental within a desired frequency band.

- $THD = 10 \cdot \log\left(\frac{\text{Total harmonics power}}{\text{Signal power}}\right)$ dB.

6.4.9.5 ENOB:

- **Effective Number of Bits (ENOB)** is a measure based on the SNDR of an ADC with a full scale sinusoidal input signal.

$$- \text{ENOB} = \frac{\text{SNDR} - 1.76}{6.02} .$$

6.4.9.6 Dynamic Range:

- **Dynamic Range (DR)** is the ratio between the power of a sinusoidal full-scale input and the least detectable input power (SNDR = 0).

$$- \text{DR} = 10 \log\left(\frac{\text{Maximum input power}}{\text{Minimum input power}}\right) \text{ dB}.$$

6.4.9.7 Effective resolution Bandwidth:

- **Effective Resolution Bandwidth (ERB)** is the input frequency where the SNDR has dropped 3 dB (or ENOB 1/2 bit).

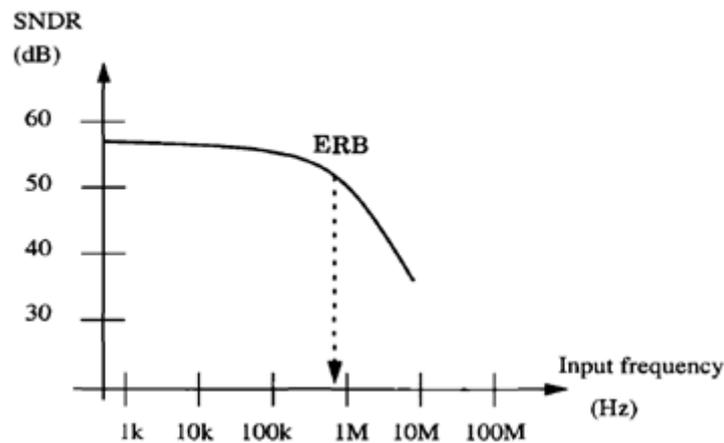


Figure 6.6.12 : ERB

Figure of merit (FOM):

- Used to compare different ADCs.
- Lower FOM means better ADC (less than 1).
- $\text{FOM} = (\text{power consumption}) / ((2^{\text{ENOB}}) * (f_s))$.

6.5 SAR ADC blocks:

6.5.1 Introduction:

- SAR ADC is well suitable for moderate speed, moderate resolution and very low power consumption.
- The primary sources of power consumption are comparator and DAC.
- Sample and hold circuit is high power consumption.
- In new techniques, sampling done in DAC circuit.

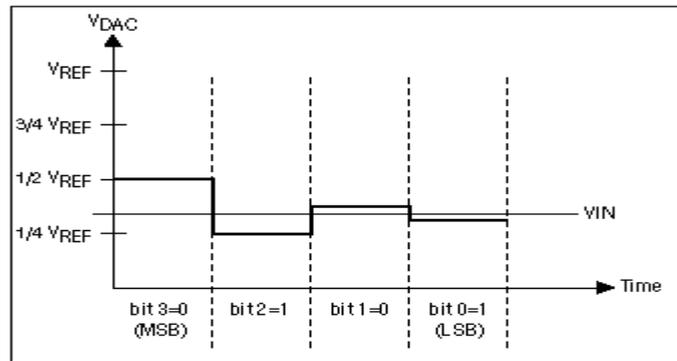


Figure 6.6.13 : SAR algorithm

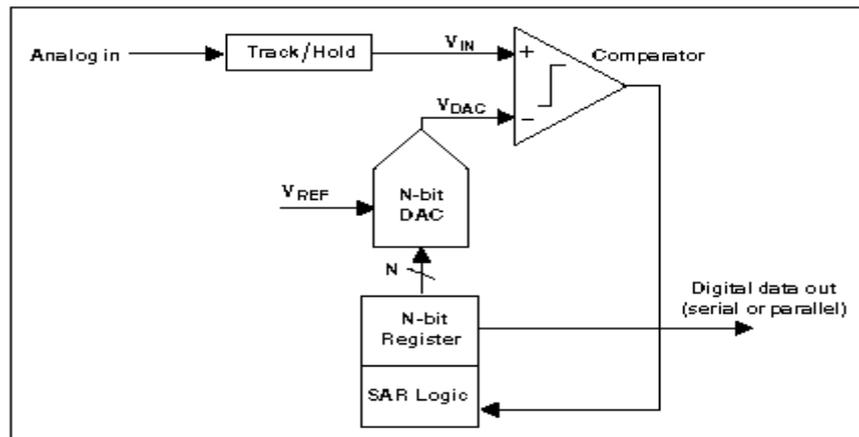


Figure 6.6.14 : SAR ADC

6.5.2 Sample and hold:

- S&H circuit contains switch and capacitor.
- When the sampling signal is high, switch connected and capacitor sample analogy input.
- In holding, capacitor saves its voltage input.

6.5.3 Comparator:

- Compare between input signal and output signal from DAC.
- Accuracy and speed of comparator are two important parameter.
- Its offset lead to some problem in SAR ADC.
- There is much architecture for comparator.

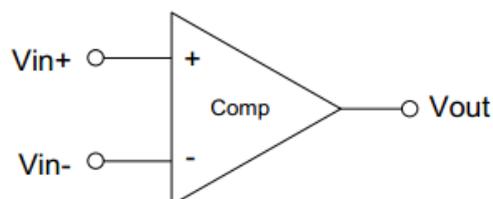


Figure 6.6.15 : comparator

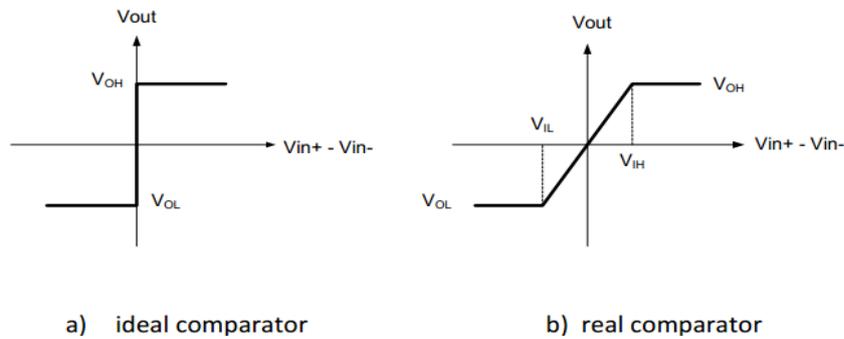


Figure 6.6.16 : comparator characteristics

- A typical comparator consists of latch and pre-amplifier circuits.
- Latch for comparing and pre-amplifier cascode for isolating and mitigate kick back noise (mentioned later).

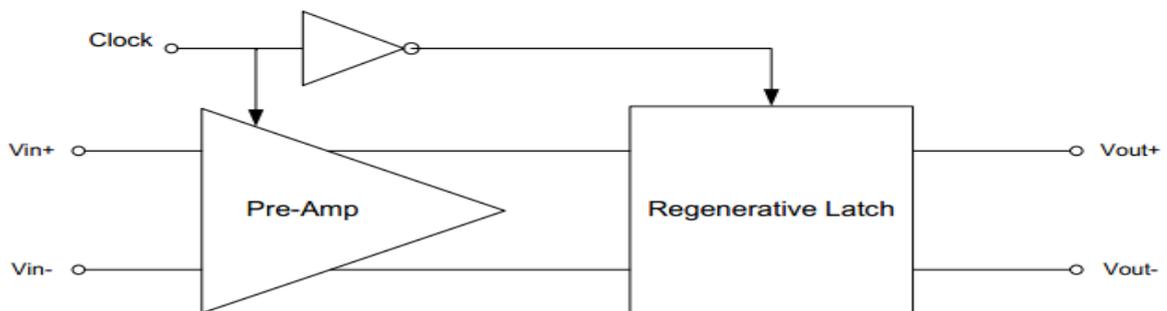


Figure 6.6.17 : comparator component

6.6 Performance parameters:

6.6.1.1 Resolution:

- Minimum input voltage difference which can be detectable by comparator.
- Noise and offset limit the resolution of comparator.
- In ADC minimum required resolution is V_{lsb} , so comparator must be able to detect it.
- $V_{lsb} = 1/2^N$, where N is number of bits.

6.6.1.2 Comparison rate:

- It is the highest frequency in which a comparator results in correct value.
- It depends on speed and also in pre-amplifier and latch circuits.
- It depends also on propagation delay that, the speed of comparator depends on it.

6.6.1.3 Offset:

- Static offset is due to mismatch between two diff-pair, and dynamic offset due to clock feed-through.

- When it's no I/p, there is output.
- Due to mismatch in diff pairs, mismatch in threshold voltage and Trans conductance of misfits.
- Offset can be modelled as voltage source (input offset voltage).
- Output offset voltage is the dc voltage applied in output when two I/p grounded.
- It can inverse the function of comparator.
- It is reduced by pre-amplifier or by using cap to sample the voltage in reset mode, and cancel it during comparison mode.

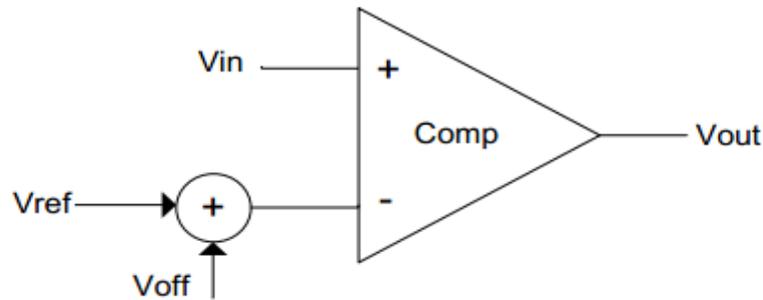


Figure 6.6.18 : comparator offset

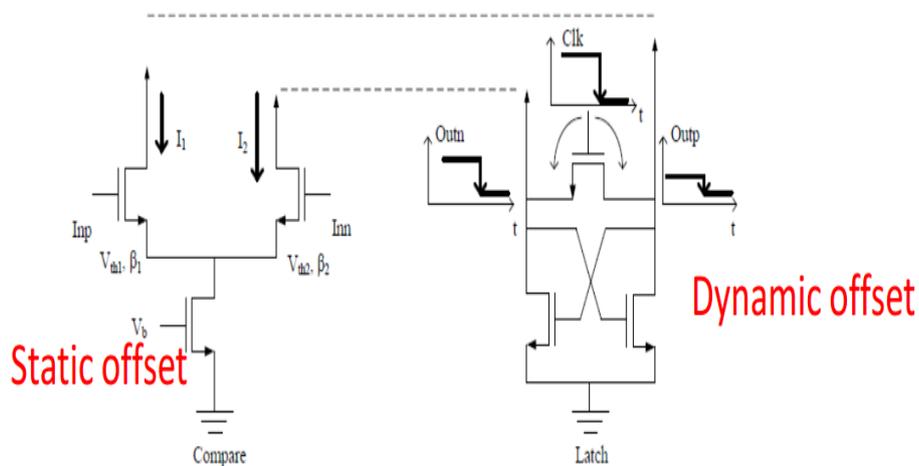


Figure 6.6.19 : static and dynamic offset

6.6.2 Kick back noise:

- Voltage disturbance in input due to large variation at output.
- The gain at output is coupled to input through parasitic capacitance and vary input capacitance.
- Can be reduced by isolation using cascade that isolate input from output) and we can use pre-amplifier.
- Kick back noise can be modelled as in MW, if o/p not matched then some wave reflect again to input.

6.6.3 Metstability:

- It is the failure of the comparator outputs to reach a valid state at the end of the regeneration phase (latch phase).
- It limits the speed and accuracy of comparator.
- We can reduce it by increasing the gain.

6.6.8 Latched comparator following pre-amp:

- The pre-amplifier attenuates the latch offset by its gain.

$$V_{off_T}^2 = V_{off_amp}^2 + \frac{V_{off_latch}^2}{A_{preamp}}$$

- If we have gain of 10 v and latch stage has an offset voltage of 50mv, so input-referred latch offset is 5 mv.
- This architecture can solve offset problem and capable of detecting small voltage differences at its input (high resolution).
- But increasing gain lowers the cut-off frequency (pole) which decreases speed.
- So there is speed-gain trade-off.
- $f_{cutoff} = f_u/A$.

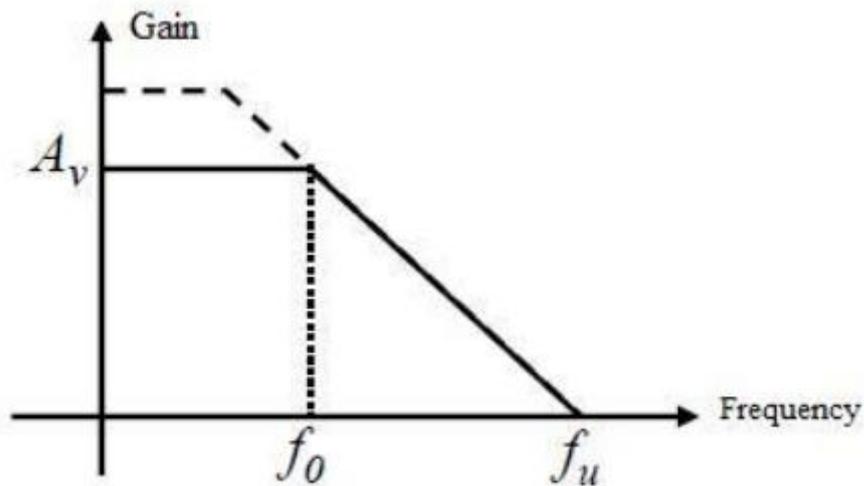


Figure 6.6.21 : open loop gain

6.6.9 Static comparator:

- Its advantages are low kick back noise and high gain and low input referred offset.
- But it is high power due to static power consumption.

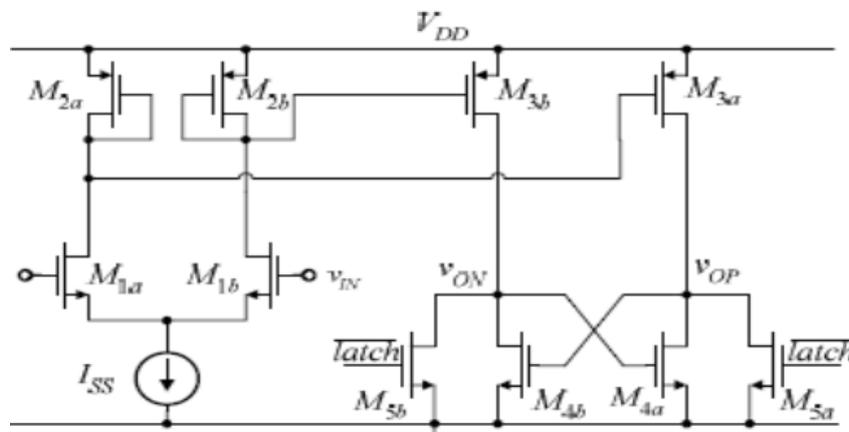


Figure 6.6.22 : static comparator

Dynamic comparator:

- Low power due to no static power.
- In the reset phase, no current pass and drain of input mosfet charge to Vdd.
- In comparison phase, drain of input discharge depend on input voltage, when drain below $V_{dd}-V_{th}$ then n-mos of inverter on and output will discharge, when o/p below $V_{dd}-V_{th}$ then the other pmos of inverter will be on then the o/p will charge till nmos of the other mosfet be on,,, and so on and finally one of them is one and the other is zero.

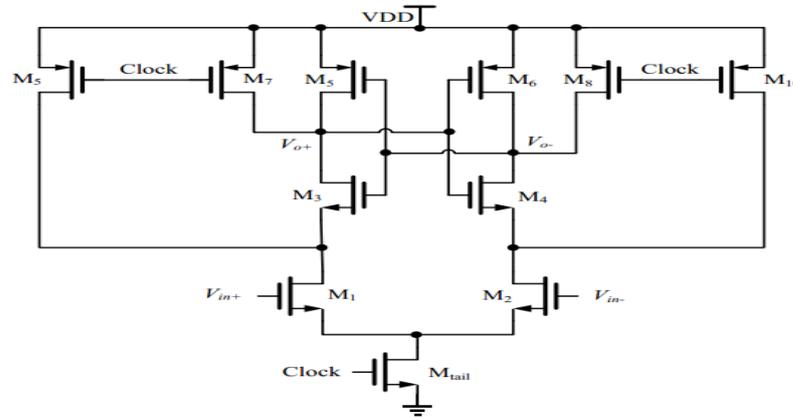


Figure 6.6.23 : dynamic comparator

Fully latched comparator:

- Low power consumption, low kick back noise but it introduce high offset and so low resolution.
- As V_{in+} high \gg V_{di} low \gg V_{gs3} high \gg I_{d3} high \gg V_{out+} low \gg V_{gs4} low \gg V_{out+} high.

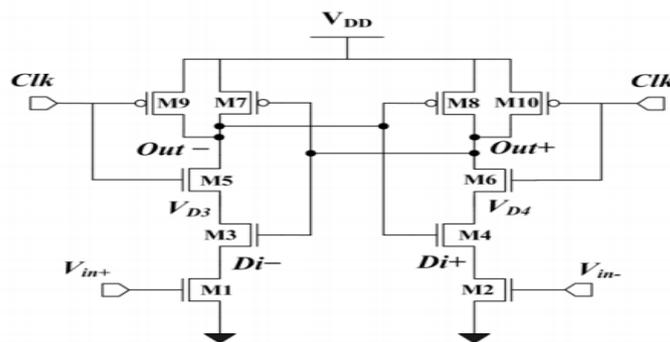


Figure 6.6.24 : fully latched comparator

6.6.10 Rail to rail comparator:

- Low voltage supply reduces the input swing which reduces SNR.
- N-mos differential pair (N1-N2) and p-mos differential pair (P1-P2) are used to extend input swing where n-mos can pull up and p-mos can pull down.
- N3 and N4 is latch to compare between two inputs and store its value.
- N5-N6 is used to mitigate hysteresis effect to speed up the comparator.
- If the comparator toggles in one direction, the comparator may tend to toggle in the same direction the next time, this is called hysteresis Resetting the comparator every clock cycle, will erase the

memory (latch) that comparators may have, eliminating the problem.

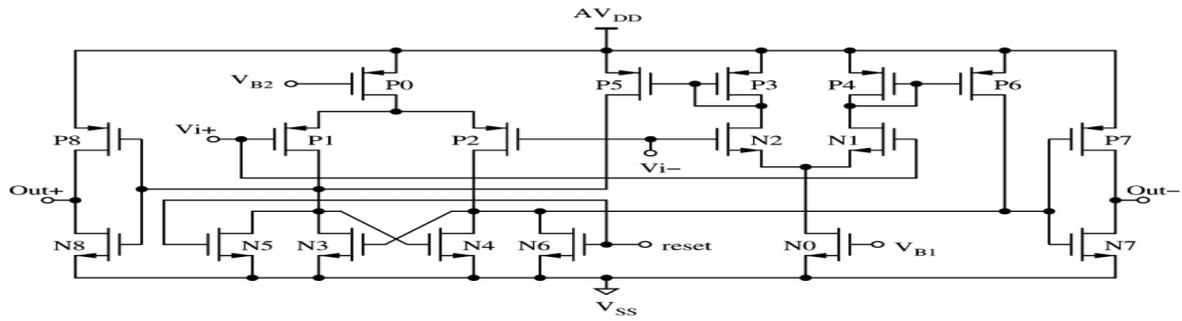


Figure 6.6.25 : rail to rail comparator

6.6.11 Comparison:

Architecture	Advantages	Dis-advantages
Open loop comparator	<ol style="list-style-type: none"> 1) High gain 2) Moderate speed 	<ol style="list-style-type: none"> 3) High offset 4) High power 5) High kick back noise
Latched with pre-amplifier	<ol style="list-style-type: none"> 1) Low kick back noise 2) Low offset 	<ol style="list-style-type: none"> 3) High power 4) Speed-gain trade-off
Latched comparator	<ol style="list-style-type: none"> 1) High speed 2) High gain 	<ol style="list-style-type: none"> 3) High offset 4) High kick back noise
Rail to rail comparator	<ol style="list-style-type: none"> 1) Wide input common mode level 2) High gain 3) High speed 4) Low offset and kick back noise 5) Low hysteresis effect 	Complex structure
Fully latched comparator	<ol style="list-style-type: none"> 1) Low power 2) High gain 	<ol style="list-style-type: none"> 3) High offset 4) Low speed 5) Mismatch

6.6.12 Proposed comparator:

- Dynamic latched comparator based on comparison.

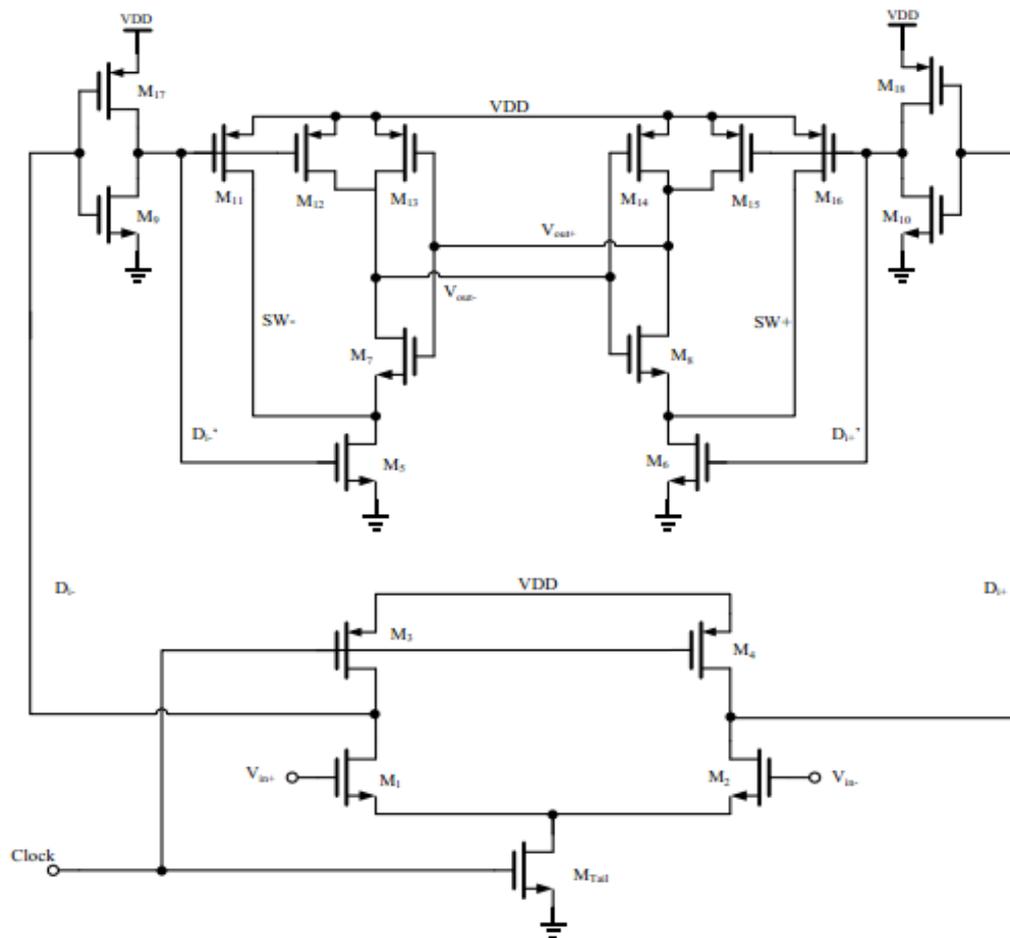


Figure 6.6.26 : proposed comparator

6.6.13 Successive approximation registers:

- SAR is the control unit which takes the output of the comparator and send control signals to the switches in DAC.
- SAR logic block is modelled as finite state machine.
- Described in VHDL or Verilog languages.

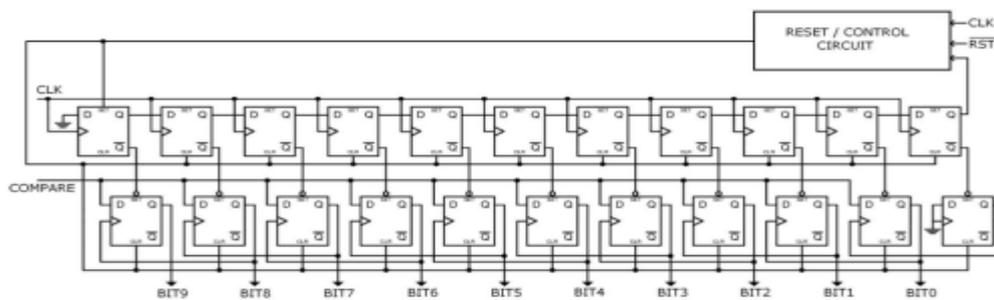


Figure 6.6.27 : SAR logic

6.7 DAC:

- Convert output digital from SAR to analog.
- The sampling operation is done by DAC.
- There are many types of DAC.

6.7.1 Binary-Weighted Capacitor Array:

- It is conventional DAC.
- Cap due to parasitic, it is non-linear because it include comparator's cap which vary with its voltage.
- So, to avoid it C must be very larger than C_p .

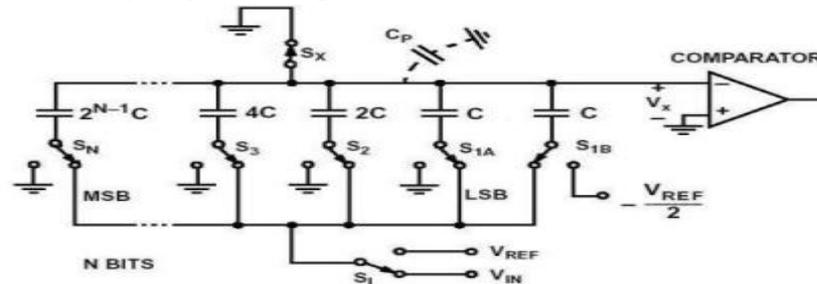


Figure 6.6.28 : binary weighted DAC

Two-Stage Weighted Capacitor Array:

- To mitigate the large capacitance size in BWC array.
- This architecture reduces occupied area and power consumption.
- In this approach the BWC array is divided into two smaller BWC and a coupling capacitor is added between two parts.

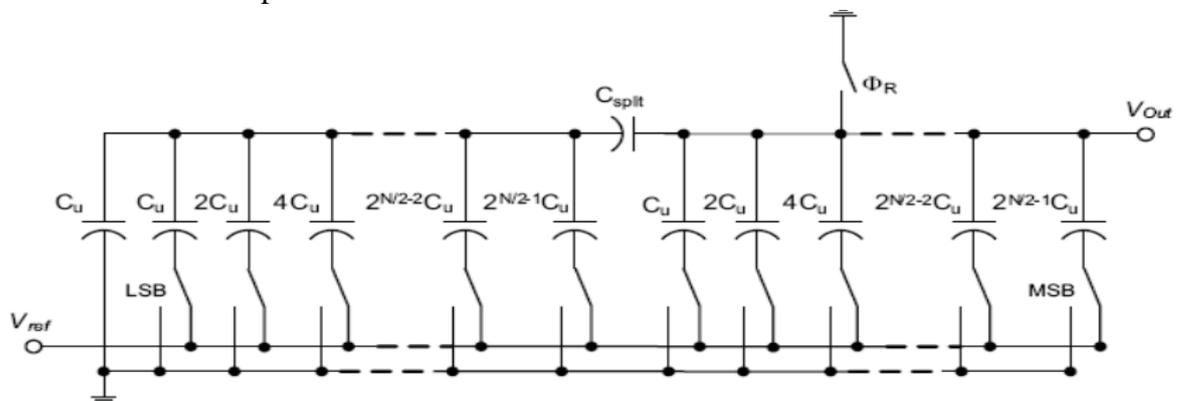


Figure 6.6.29 : two-stage Dac

6.7.2 C-2C Capacitor Array:

- In this configuration, the values of the capacitors are drastically reduced.
- Can achieve higher speed while consuming less power.
- The power consumption increases linearly, in contrast to BWC in which the power rises exponentially.
- The main drawback of this configuration is the degradation of the linearity due to parasitic capacitances.

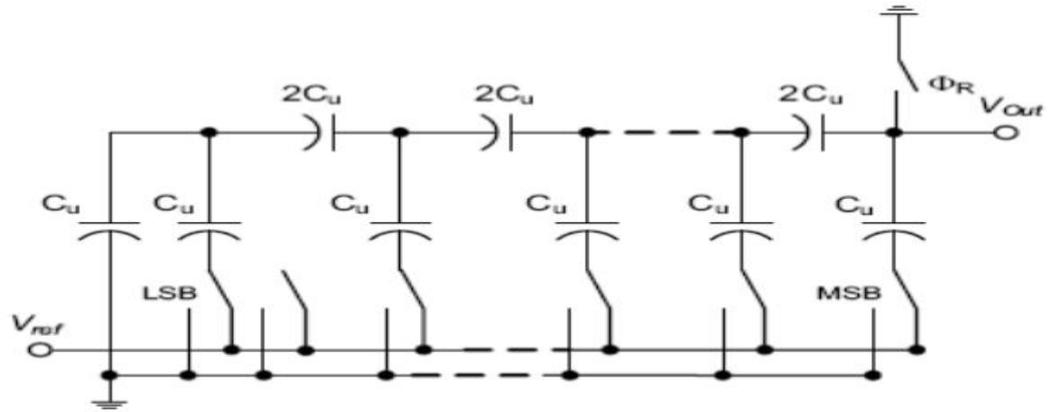


Figure 6.6.30 : C-2C DAC

Resistive Dac:

- For N bit we need 2^N resistor and almost $2 \cdot 2^n$ switches.

Advantages:

- Simple
- High matching advantage in resistor over caps.

Dis-advantages:

- High power consumption due to dc power.
- High settling time.
- We can divide dac to two stages or more (10 bit to 5 bit) by using buffers but have two drawbacks, dc offset and limited BW so reduce speed of dac.
- There is trade off in choosing resistor value between power dissipation and speed.

6.7.3 Comparison:

	Resistive DAC	Capacitive DAC
Static power	Yes	No static power
Dynamic power	No	Yes

linearity	Bad for diffusion type Good for poly type	Good
Area	Less	higher
Matching	Good	Bad
No. of switches	High	Low

6.7.4 Proposed DAC:

- No charging any error.
- Less power consumption.
- Less number of switches.
- Less number of capacitor.

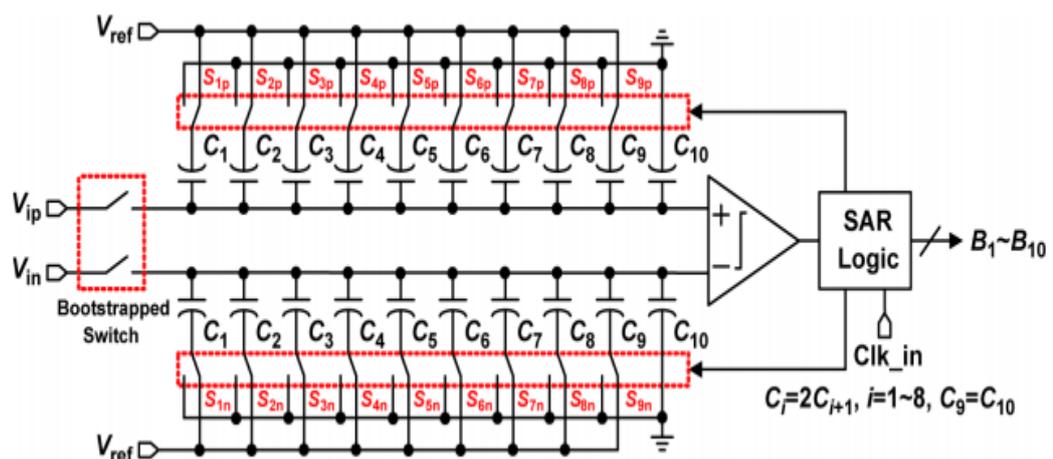


Figure 6.6.31 : proposed dac

A 10-bit 50-MS/s SAR ADC With a Monotonic
Capacitor Switching Procedure
Chun-Cheng Liu , Student Member, IEEE, Soon-Jyh Chang , Member, IEEE ,

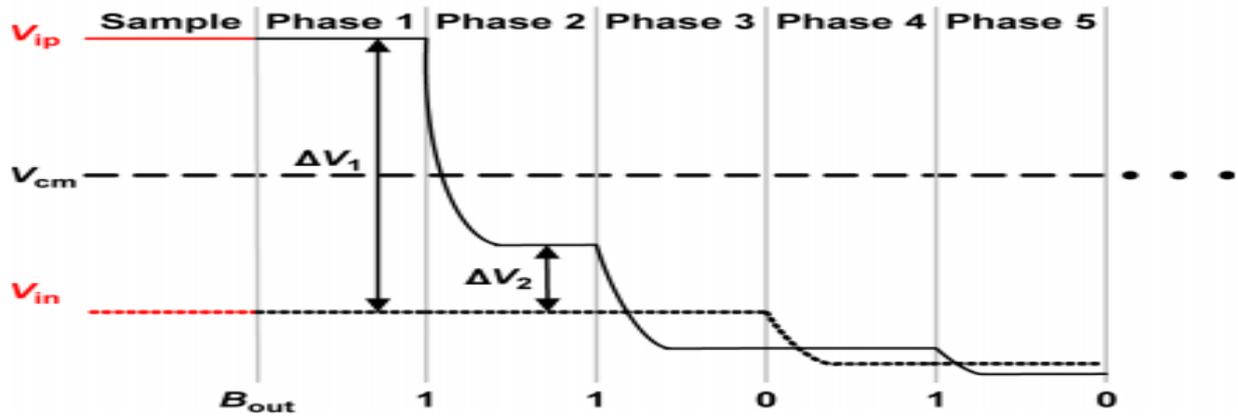


Figure 6.6.32 : algorithm

6.7.5 Capacitor value selection:

6.7.6 KT/C noise:

- Due to sampling switch.
- SNR of thermal noise > SNR of quantization noise.
- $K \cdot T / C_u < \Delta^2 / 12$.
- So value of C_u will be $C_u = 5f F$.

C_{Unit} increases, thermal noise decreases

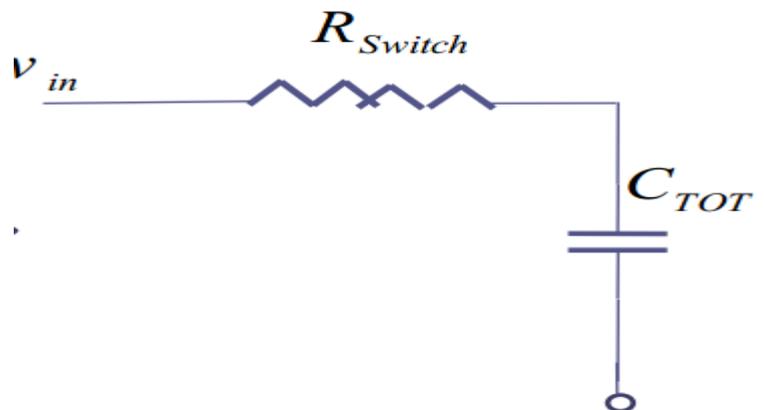


Figure 6.6.33 : KT/C noise

6.7.7 Parasitic capacitors:

Mainly two type of parasitic:

- Parasitic due to routing of capacitor.
- Input capacitance of comparator.

Routing parasitic isn't harmful because it is constant and it is virtually grounded.

Input capacitance is harmful because it is non-linear and cause non-linearity and C_u must be very large than it.

For MIM cap, parasitic capacitance.

	cpp capacitor	cmm capacitor
Area capacitance	0.85 fF/ μm^2	1.25 fF/ μm^2
Perimeter capacitance	0.021 fF/ μm	0.111 fF/ μm
Parasitic area capacitance	105 aF/ μm^2	12 aF/ μm^2
Parasitic perimeter capacitance	57 aF/ μm	36 aF/ μm
Minimum area	17.24 μm^2	25 μm^2
Minimum unit capacitance	15 fF	33.47 fF
Matching parameter A_C	1.25% μm	0.65% μm

Figure 6.6.34 : table of mime-cap.

6.7.8 Mismatch of capacitor:

- It is mainly due to fabrication like etching, under cut.....etc.
- $C = C_u + \text{segma}$.
- To lower the mismatch decrease ratio P/A to make capacitor larger and lower the mismatch.
- We can also use dummy capacitors to uniform etching and decrease mismatch.
- Mainly value of capacitor determined using its mismatch.
- We want $\text{DNL} < 0.5 \text{ LSB}$ to avoid monotonicity.
- We can simulate mismatch using Monte Carlo simulation.

C_{Unit} increases, mismatch effect decreases

6.8 Design methodology:

6.8.1 Capacitor value selection:

By Matlab simulation and monte-carlo simulation we can estimate value of capacitor, the value is according to DNL and INL required from system or in another word the acceptable mismatch.

According to these simulations, unit capacitor is 30 ft.

Capacitor value(f)	Mismatch value(sigma)(a)
30	40.1669
50	51.7865
70	61.006
90	69.1677
110	76.3455
130	82.8787
150	88.778
170	94.6064
190	100
210	104.859
230	109.863

Table 6.10 : monte-carlo simulation

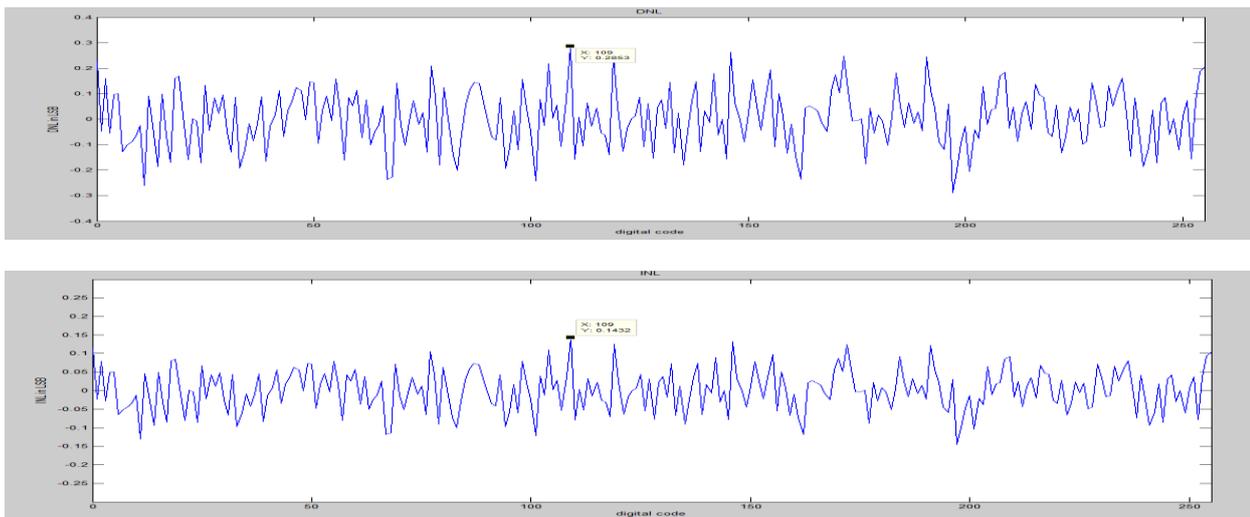


Figure 6.6.35 : DNL, INL Matlab simulation

SAR logic control:

Verilog-A and vhdl with synthesis attached in appendix.

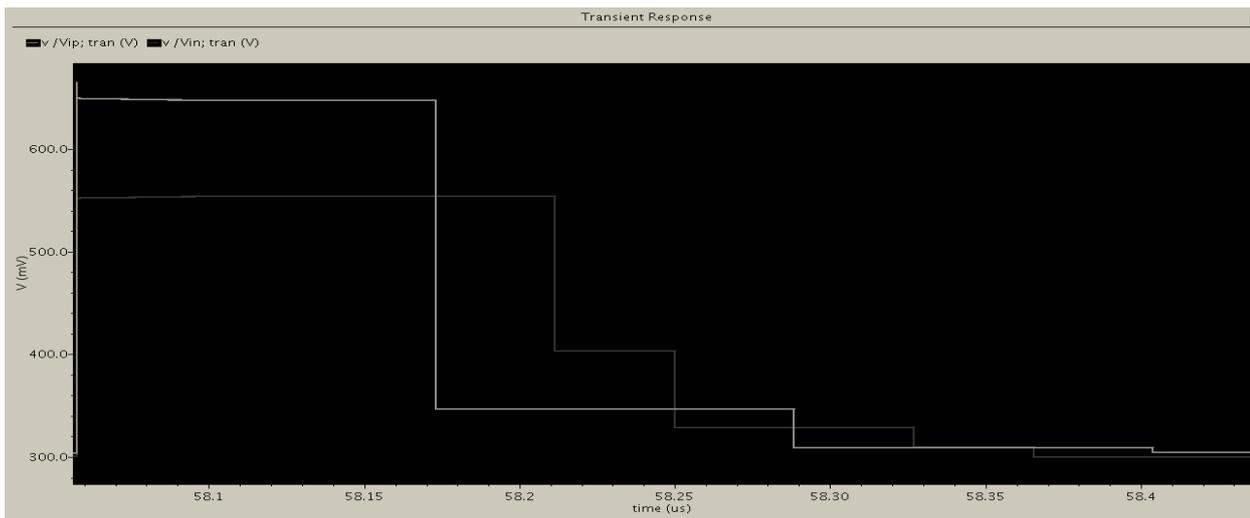


Figure 6.6.36 : SAR algorithm

6.8.2 Sampling switch:

6.8.3 Ideal switch:

Vout track input when S1 is closed.

Grab exact value from input to output when S1 is open (hold).

Ideal switch has zero resistance

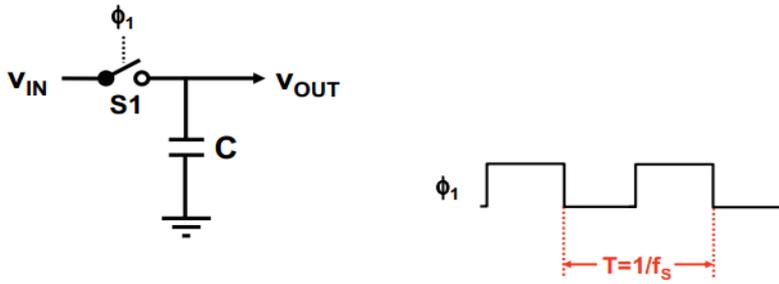


Figure 6.6.37 : ideal switch

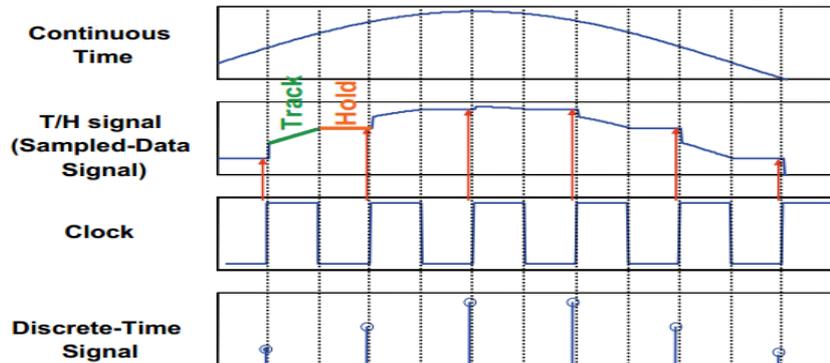


Figure 6.6.38 : ideal switch diagrams

6.9 Practical sampling issues:

Thermal noise due to switch resistance.

- Delay through R_{on} .
- Limited BW.
- Switch charge injection & clock feed through.
- Clock jitter.
- R_{on} of switch cause thermal noise.
- In higher resolution ADC, thermal noise dominates quantization noise.
- To minimize thermal noise under quantization, increase unit capacitor used in dac (power consideration).

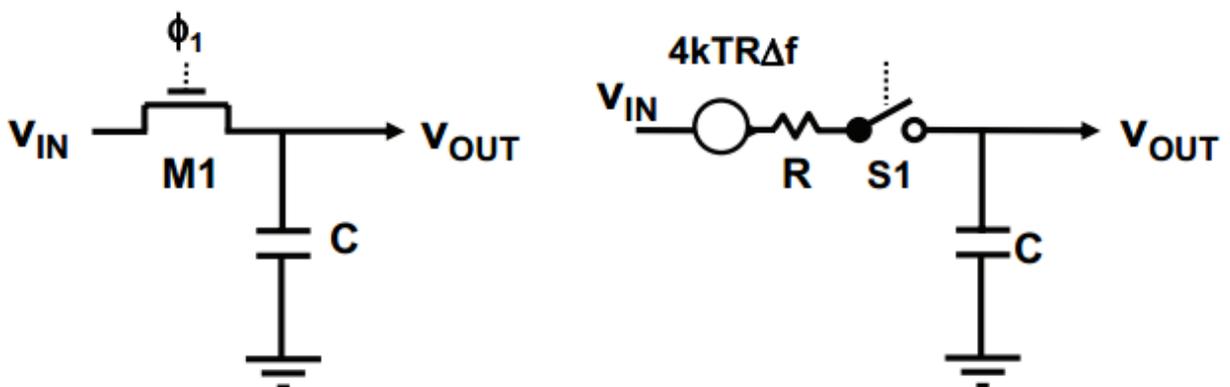


Figure 6.6.39 : thermal noise

6.9.1 1.3.4.3.3 Charge injection:

- Channel: distributed RC network formed between G, S, and D.
- Channel to substrate junction capacitance: voltage dependent.
- Drain/Source junction capacitors to substrate: voltage dependent.
- Over-lap capacitance.
- So, charge injected through these capacitance and lowers the charge to sampling capacitance.
- So there is trade-off between speed (R_{on} and τ_{sw}) and accuracy (charge injection).

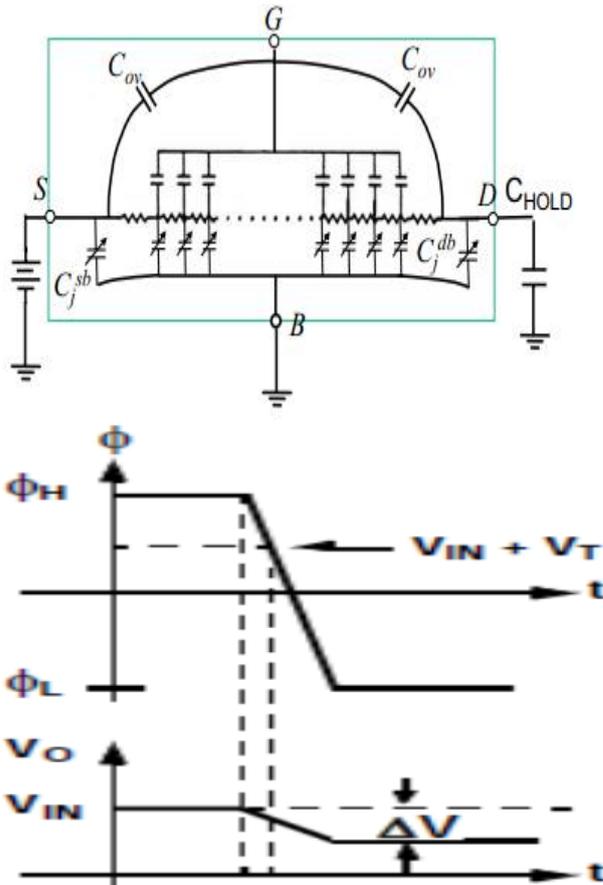


Figure 6.6.40 : charge injection

6.10 Proposed sampling switch:

Because transmission gate fails across corners because sum of n-mosfet and p-mosfet threshold voltage can reach above supply voltage then it fails to work well.

Best solution is to use bootstrapped switch, there are lot of bootstrapped switches in literature

	N-mos	T.G	T.G with dummy	ABO switch	Boot strapped	Modified boot-strapped switch
Power	Less	Low	Low	moderate	high	high
Linearity	Low	Moderate	Better TG	High	Higher	Highest
Complexity	Less	Less	Less	Moderate	More	More
Ron	High	Moderate	Moderate	Low	Low	Low
Effects on my system	Bad	Good	Better	Best	Best but don't need	Best but don't need

6.10.1 Simulation results:

6.10.2 Dessouky switch:

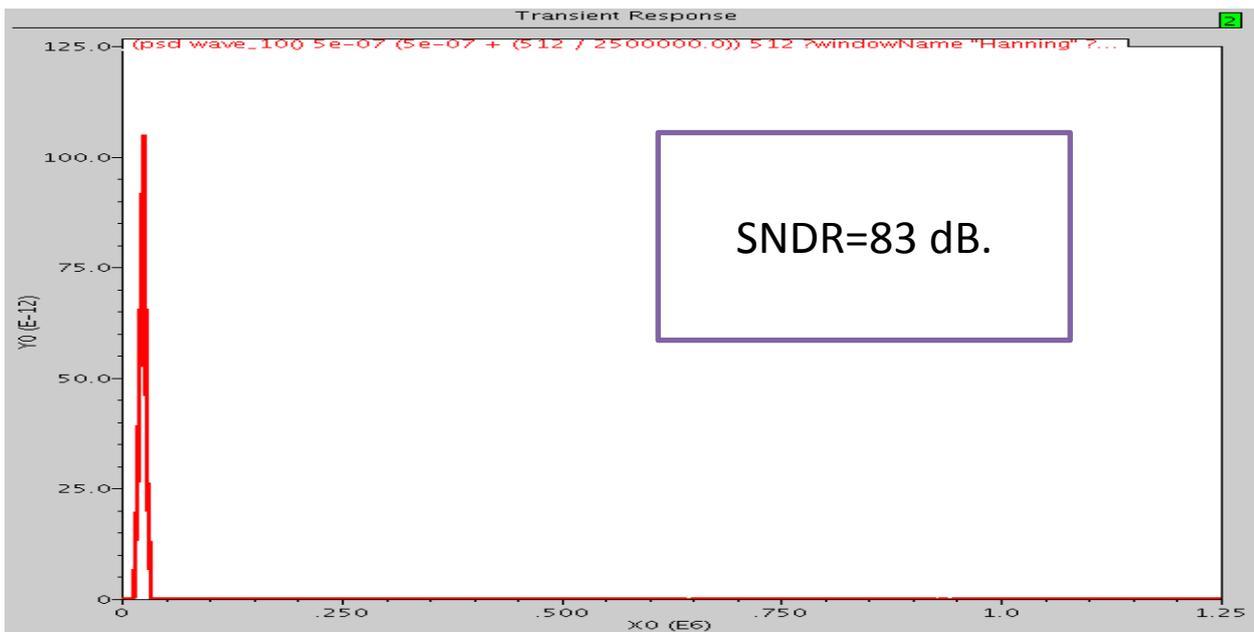
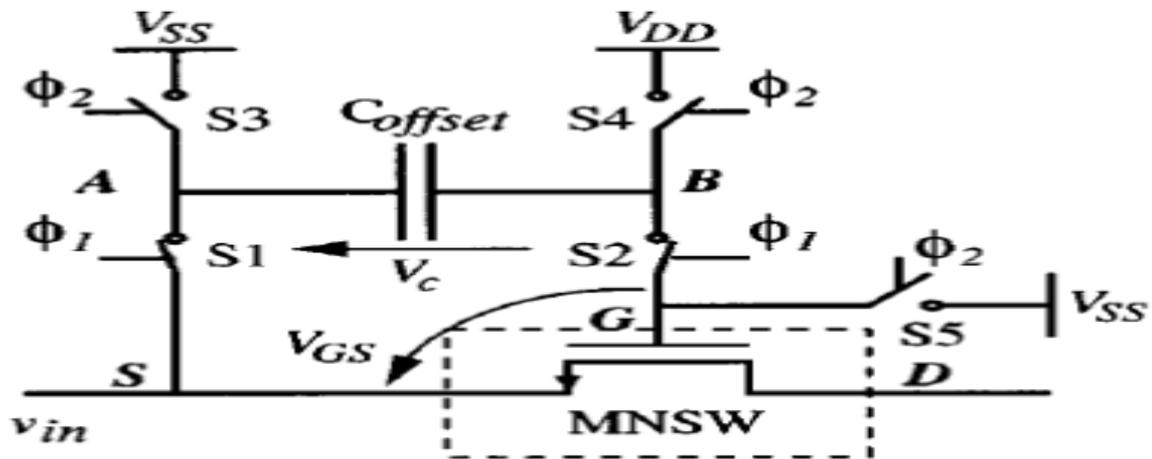


Figure 6.6.41 : dessouky switch

6.10.3 Sadollahy switches [2]:

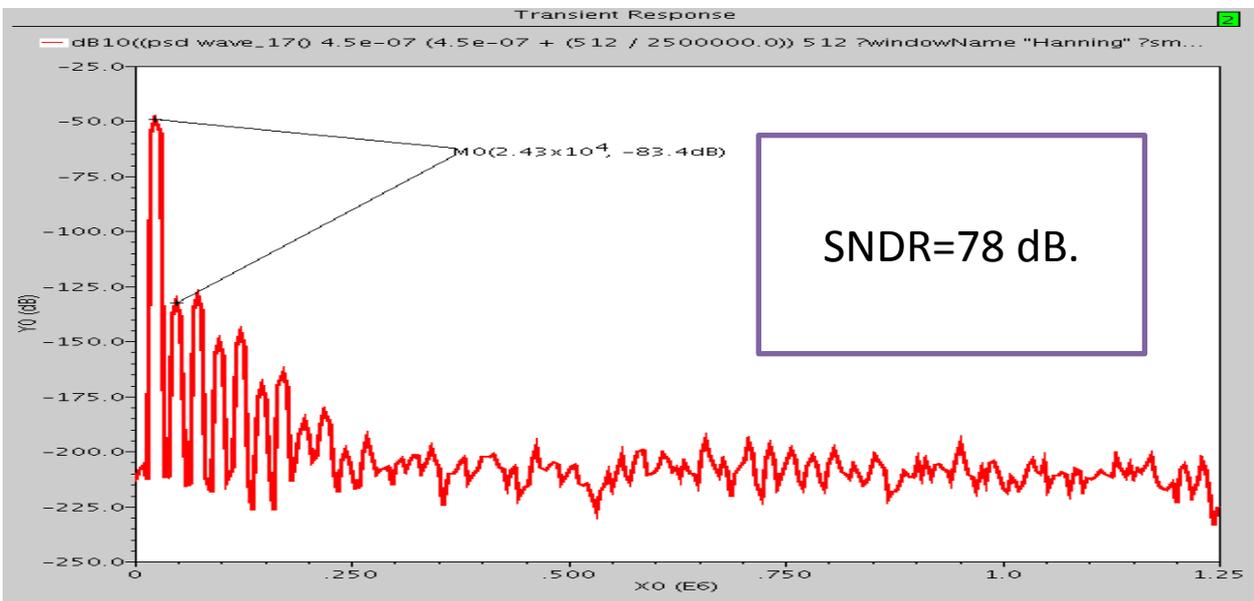
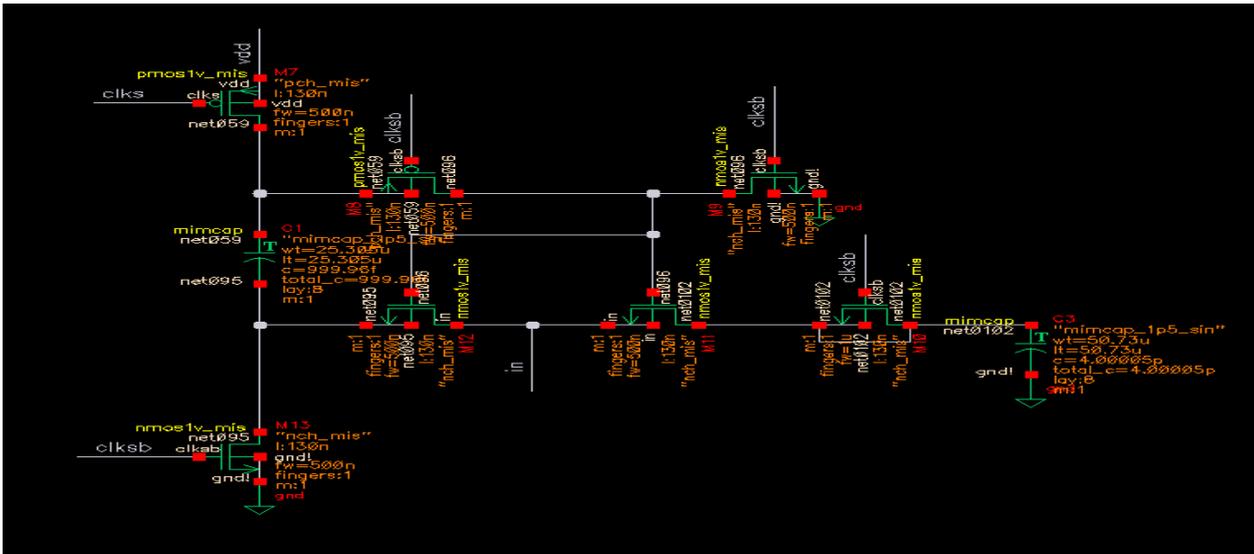


Figure 6.6.42 : sadollahy switch

6.11.1 Comparison:

specification	Dessouky	Sadollahy	Proposed
SFDR (dB)	90	83	100
SNDR (dB)	83	78	116
ENOB(bits)	13.5	12.7	19
Settling time	Less than half clk	Less than half clk	Less than half clk

6.12 Comparator:

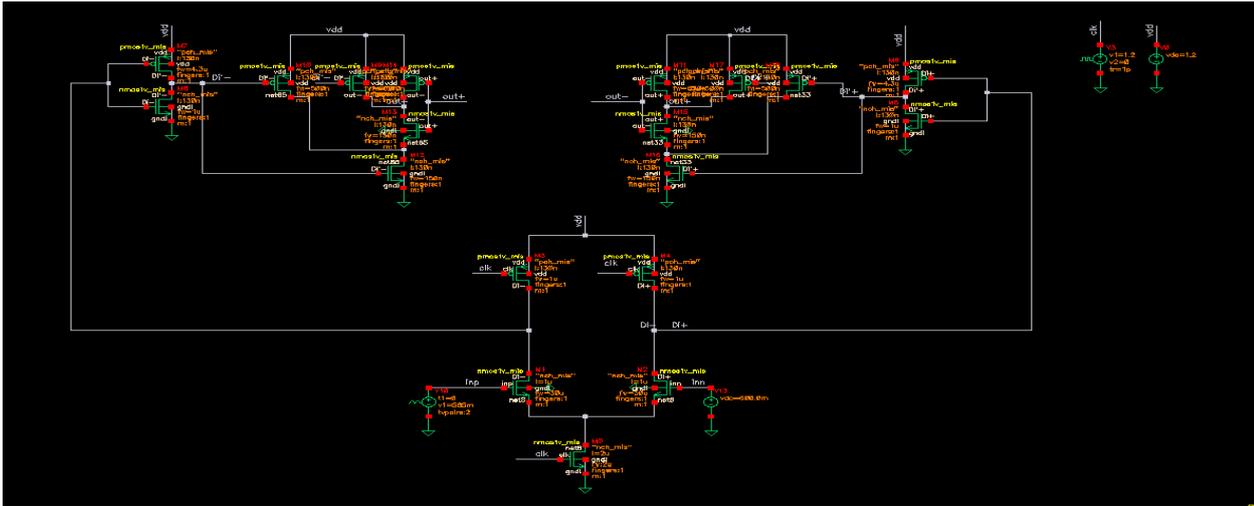


Figure 6.6.44 : comparator

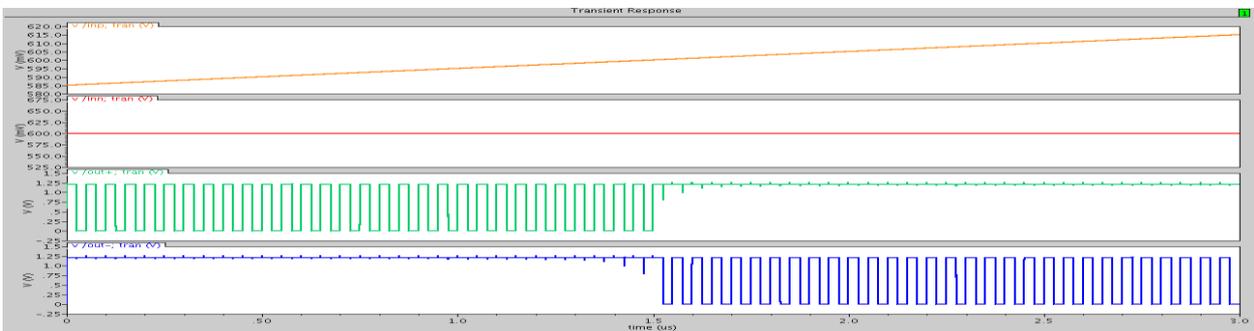


Figure 6.6.45 : simulation

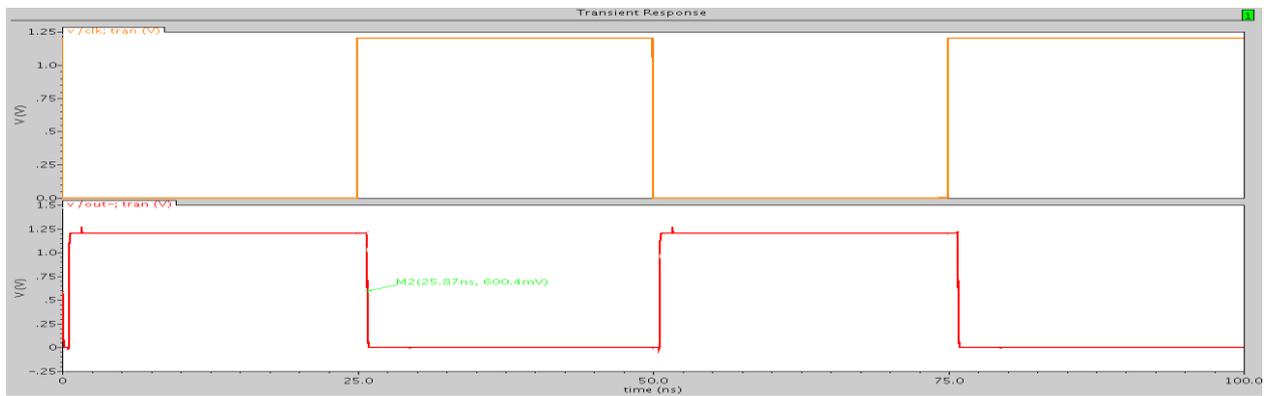


Figure 6.6.46 : settling time and delay

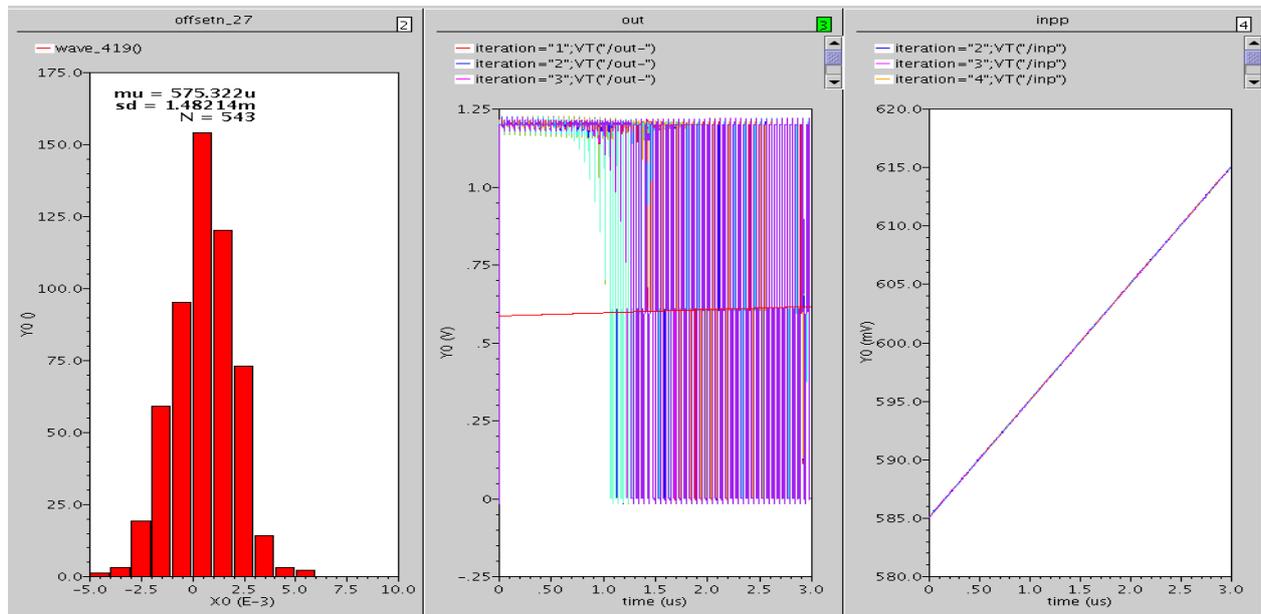
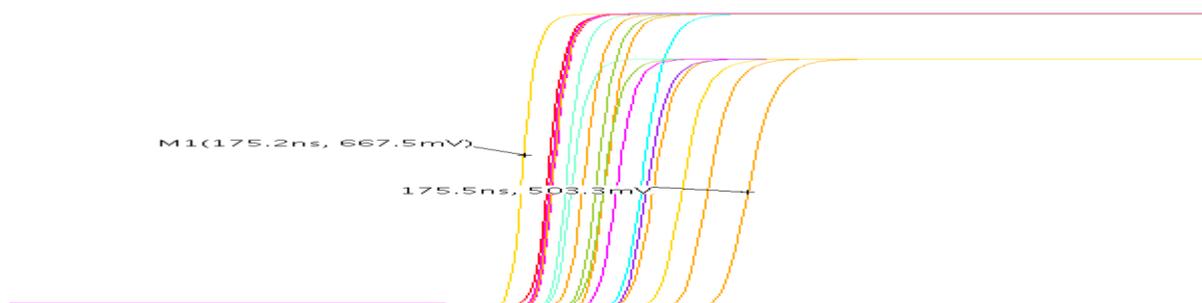


Figure 6.6.47 : Monte Carlo simulation and offset

6.12.1 Achieved specifications:

- Power=11 μ w.
- Delay =0.4n sec.
- Dc offset = 1.5m v.
- Maximum operating frequency =2G Hz.
- Pre-amp gain= 11 v/v.
- Input capacitance = 1.2 f.

6.12.2 Corners simulation:



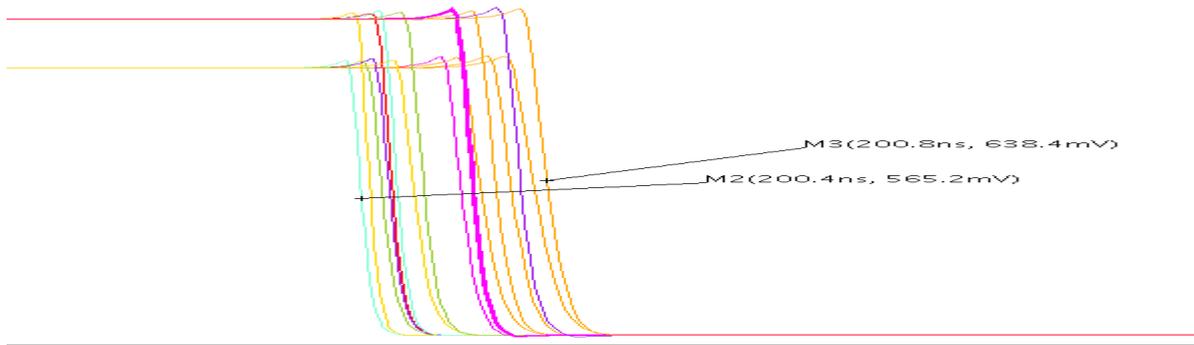


Figure 6.6.48 : corner simulation

6.13 Full system simulations:

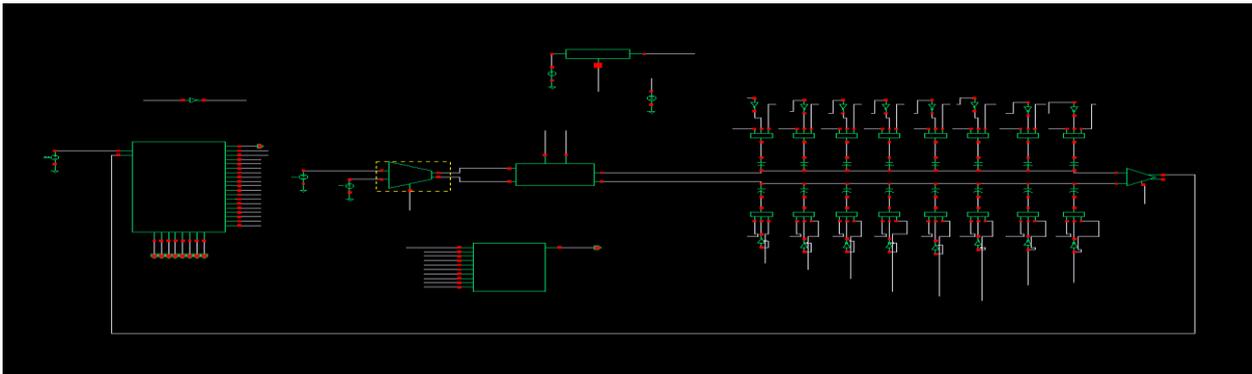
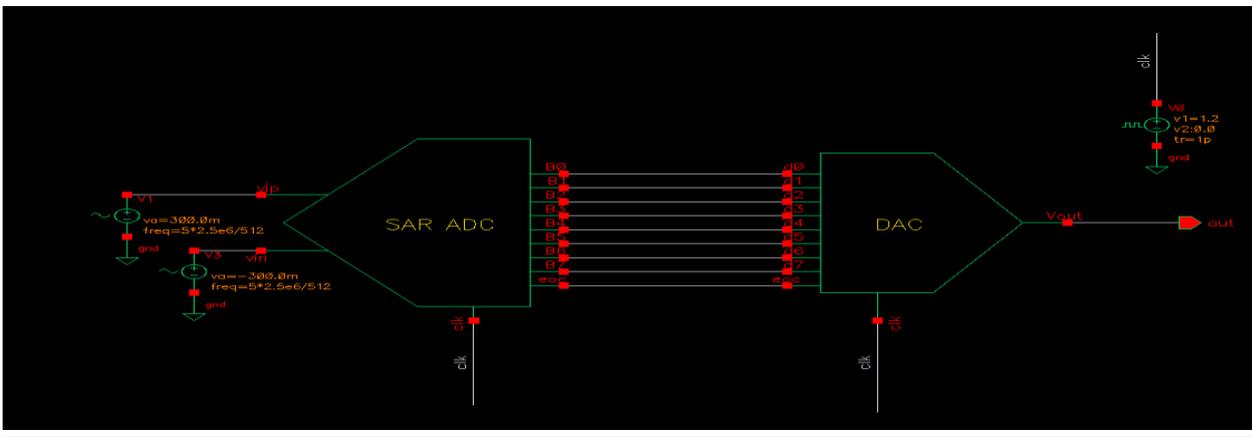


Figure 6.6.49 : SAR ADC



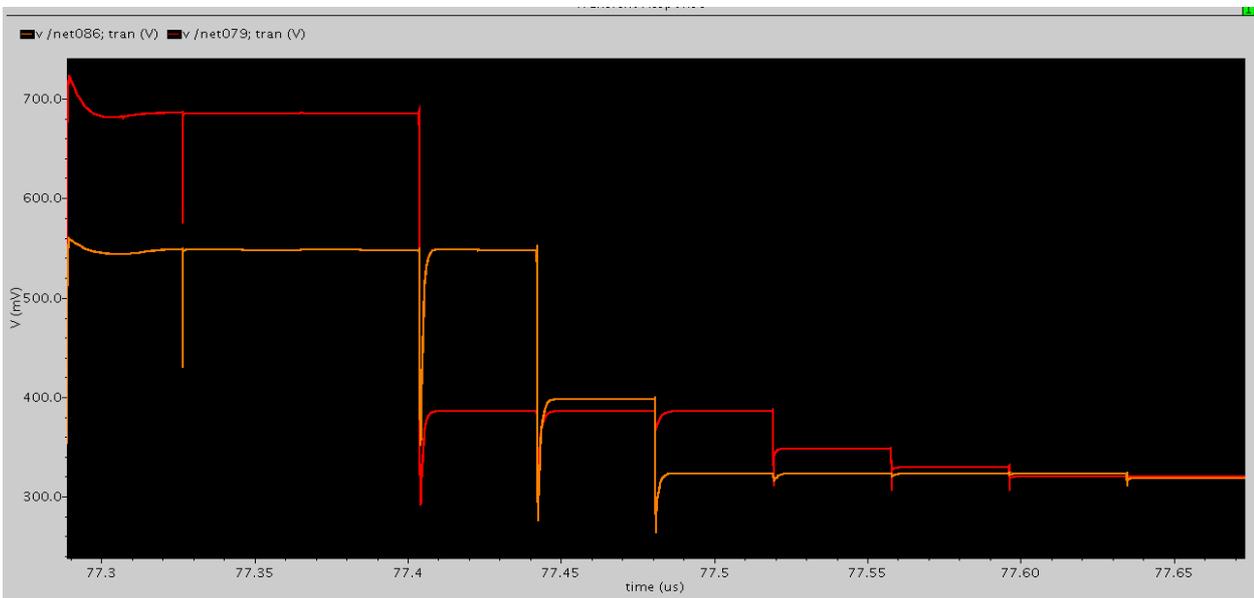
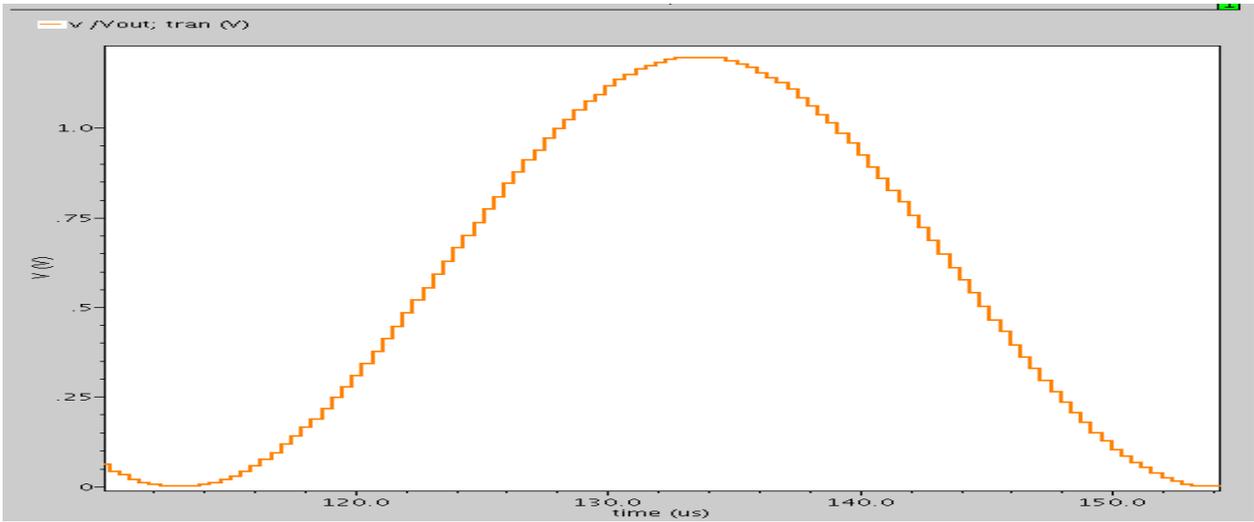


Figure 6.6.50 : algorithm

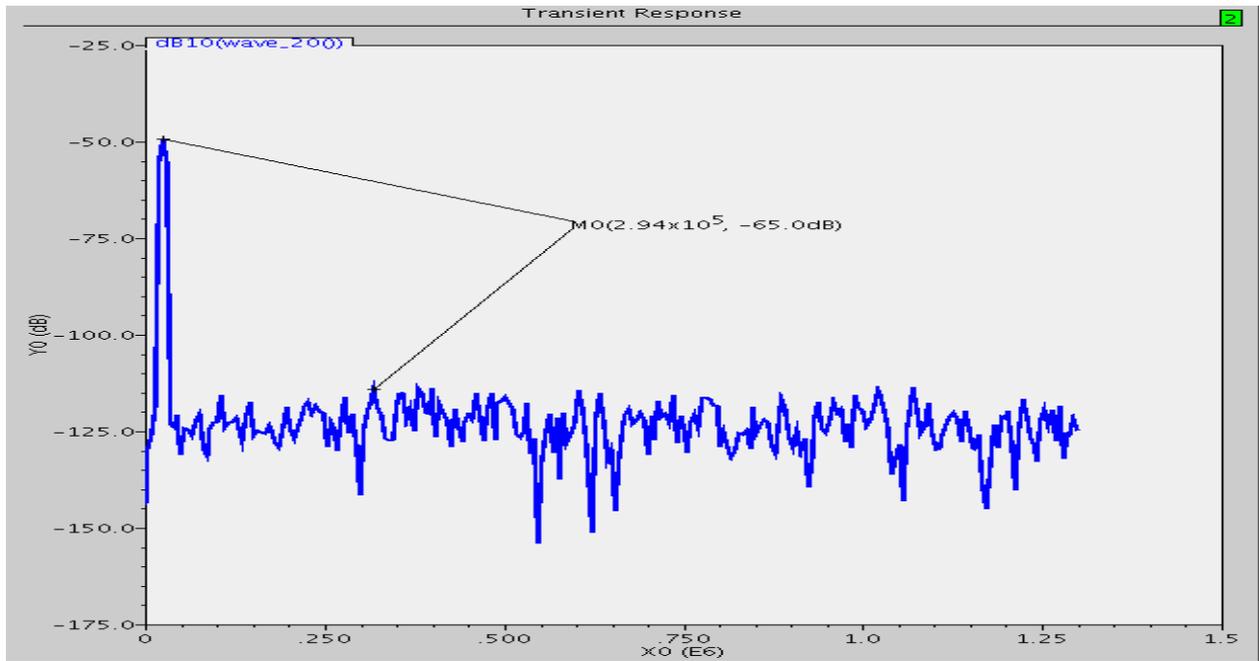


Figure 6.6.51 : PSD simulation

6.13.1 Corners summary:

Parameter	TT	SS,-40	SS,125	FF,-40	FF,125
SFDR(dB)	65	50.4	56.2	56.7	52.7
SNDR(dB)	49.9	49.8	49.9	94.9	49.9
ENOB(bits)	8	8	8	8	8
Power(us)	500	320	480	650	625

6.14 Mismatch analysis:

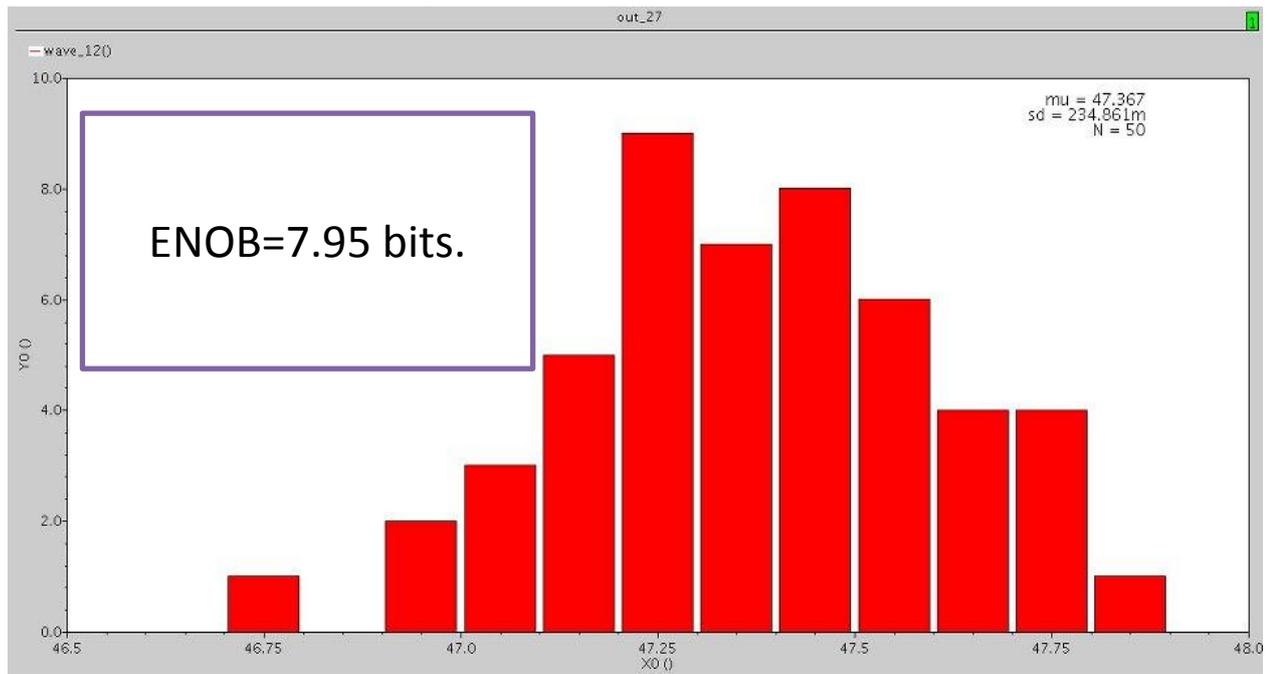


Figure 6.6.52 : monte-carlo simulation

6.14.1 Conclusion:

Specifications	Required by system	Achieved
Sampling frequency(MHz)	2	2.6
ENOB(bits)	5	7.95
Power(mw)	1	0.1

6.15 Final conclusion:

Specifications	This work
Architecture	SAR ADC
Technology	0.13 μm
Input range	Full scale
Supply voltage(V)	1.2
Sampling frequency(MHZ)	2.6
Resolution (bit)	8
Unit capacitance/sampling capacitance	30fF/4pF
SNDR(dB)	49.3
ENOB(bit)	7.95
POWER(us)	100
FOM(fJ/conv.-step) (all block with buffers)	15

6.16 Layout:

There are some rules that must be considered in layout:

- 1) communication with circuit designer.
- 2) Matching.
- 3) Be closing as possible for matching and temperature variation.
- 4) Using matching technique such as interdigitized and common centroid.
- 5) minimize parasitic and equalize routing.
- 6) use one unit and repeat it using multiplier for more matching
- 7) draw one half and mirror it.
- 8) Use wide metals and high metals for low R and so less IR drop (critical at low supply).
- 9) Use n-well shielding to minimize C.
- 10) Dummification.

6.16.1 Switches layout:

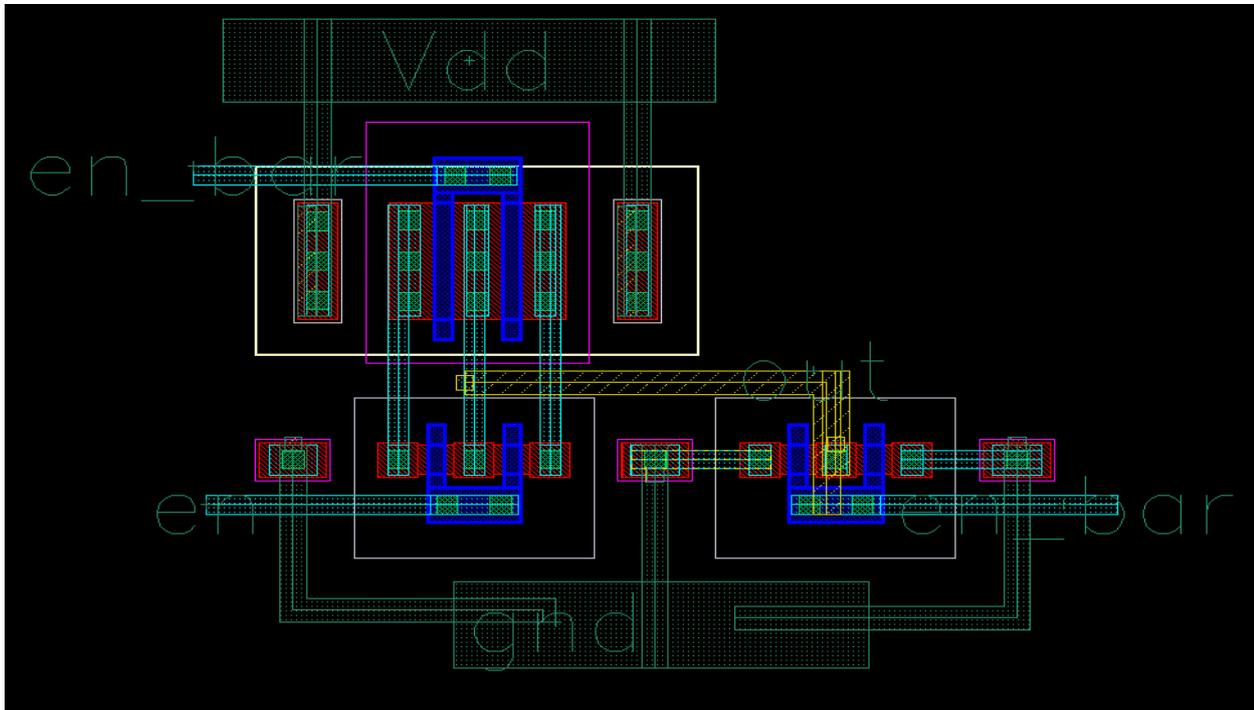


Figure 6.6.53 : switch layout

Reference switch:

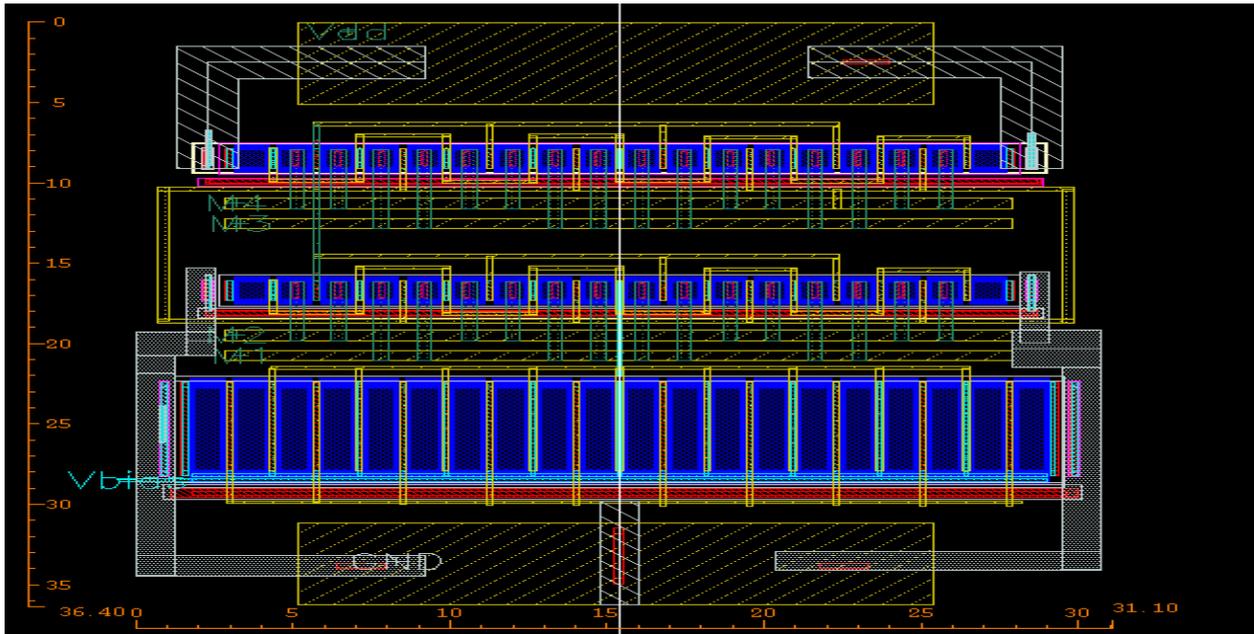


Figure 6.6.54 : reference buffer layout

1.3.5.3 Comparator:

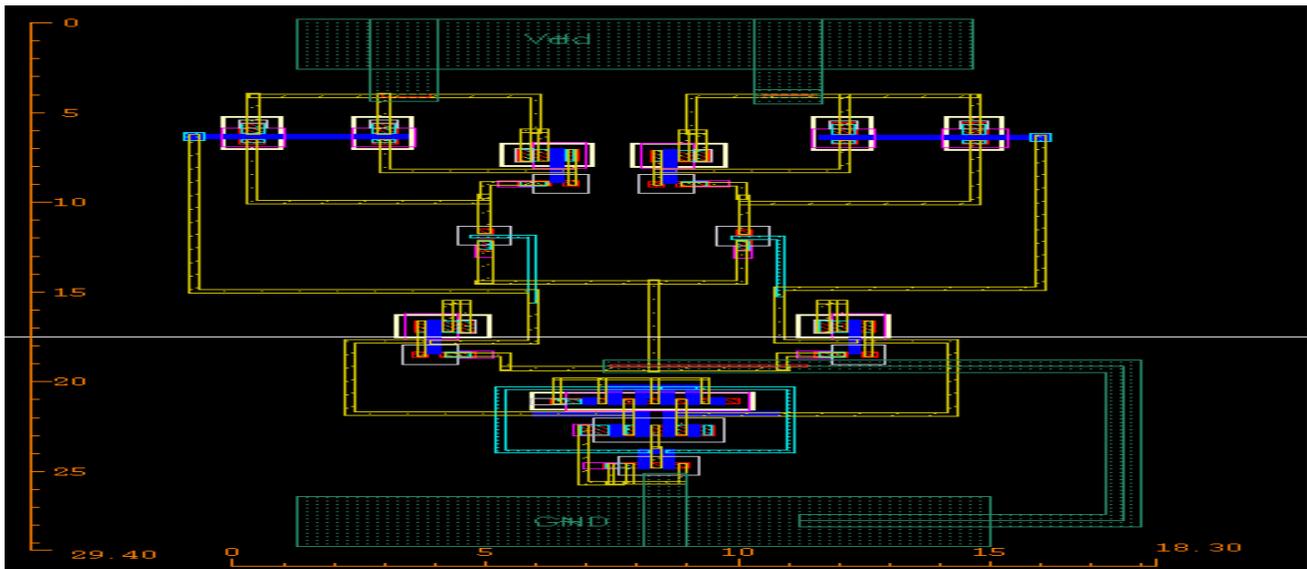


Figure 6.6.55 : comparator layout

Capacitor Array:



Figure 6.6.56 : Capacitor array

6.17 Summary:

SAR ADC in 0.13 um technology has been designed with vhdl synthesis, achieved 7.95 ENOB at 100 uw power. Perform layout of basics blocks with enhancing matching between them.

6.18 Future work:

Connect all blocks with layout together and perform DRC and LVS. Check PEX and simulate all blocks together and extract final specification after layout.

6.19References:

- Offset analysis of dynamic latched comparator , jeon.
- A 8-bit 500-KS/s low power SAR ADC for biomedical application, hang.
- High speed and low power dynamic latch comparator.
- Design of Cmos sampling switch for ultra low power ADCs in biomedical application.
- A method for reducing the variation in 'on' resistance of a mos sampling switch.
- A novel technique for improving the linearity of MOS sampling switch.
- A Cmos Bootstrapped switch with novel clock feed-through compensation.
- A new low voltage Cmos unity –gain Buffer.
- Low voltage wide swing fully differential Cmos voltage buffer.
- An 8-bit 1MHz Successive Approximation Register (SAR) A/D with 7.98 ENOB , Yafei Ye
- Accurate Testing of ADC's Spectral Performance Using Imprecise Sinusoidal Excitations
Zhongjun Yu, Degang Chen, Randy Geiger .
- Dynamic Testing of A/D Converters Using the Coherence Function Troy
Richards,Member, IEEE.
- Histogram Measurement of ADC Nonlinearities Using Sine Waves Jerome Blair,
Member, IEEE.
- An 8-bit Single-Ended Ultra-Low-Power SAR ADC with a Novel DAC Switching Method Weibo Hu,
Donald Y.C. Lie, and Yen-Ting Liu .
- A Novel Low-voltage Low-power SAR ADC for biomedical applications , Chao Yuan and Yvonne Y. H.
Lam School of Electrical and Electronics Engineering.
- Integrated circuit eda design of 10-bit sar adc , dan bu , nanjian.
- A CMOS LOW-POWER LOW-OFFSET AND HIGH-SPEED FULLY DYNAMIC LATCHED
COMPARATOR HeungJun Jeon, Yong-Bin Kim .
- Low-power and low-offset comparator using latch load Y. Jung, S. Lee, J. Chae and G.C. Temes.
- An Ultra-Low-Power SAR ADC with an Area-Efficient DAC Architecture Pouya Kamalinejad, Shahriar
Mirabbasi, and Victor C.M. Leung.
- A Low Power Consumption 10-bit Rail-to-Rail SAR ADC Using a C-2C Capacitor Array OBHoonki
Kim, YoungJae Min, Yonghwan Kim, and Soowon Kim.

-A 8-bit 500-KS/s Low Power SAR ADC for Bio-Medical Applications You-Kuang Chang, Chao-Shiun Wang and Chorng-Kuang Wang .

-Chri stian Jesus B. FAYOMI 1, Gordon W. ROBERTS 2 and Mohamad SAWAN .

-Wen-Yi Pang, Chao-Shiun Wang, You-Kuang Chang, Nai-Kuan Chou*, and Chorng-Kuang Wang .

-A 10-bit Low-Power SAR ADC With a Tunable Series Attenuation Capacitor Lado Filipovic.

-An Ultra-Low-Power 10-Bit 100-kS/s Successive-Approximation Analog-to-Digital Converter Reza Lotfi1,2, Rabe'eh Majidi2, Mohammad Maymandi-Nejad2, and Wouter. A. Serdijn.

-LOW-POWER TECHNIQUES FOR SUCCESSIVE APPROXIMATION REGISTER (SAR) ANALOG-TO-DIGITAL CONVERTERS ,Ramgopal Sekar.

-Low-power high-speed low-offset fully dynamic CMOS latched comparator HeungJun Jeon Northeastern University.

-Master's thesis performed in Electronic Devices by Raheleh Hedayati .

7 Receiver System verification

7.1 The LNA-Mixer interface :

The first interface verified was the LNA-Mixer one , it was simulated in the time domain , and the IIP3 and the NF wsa simulated also , and the following results were obtained . .

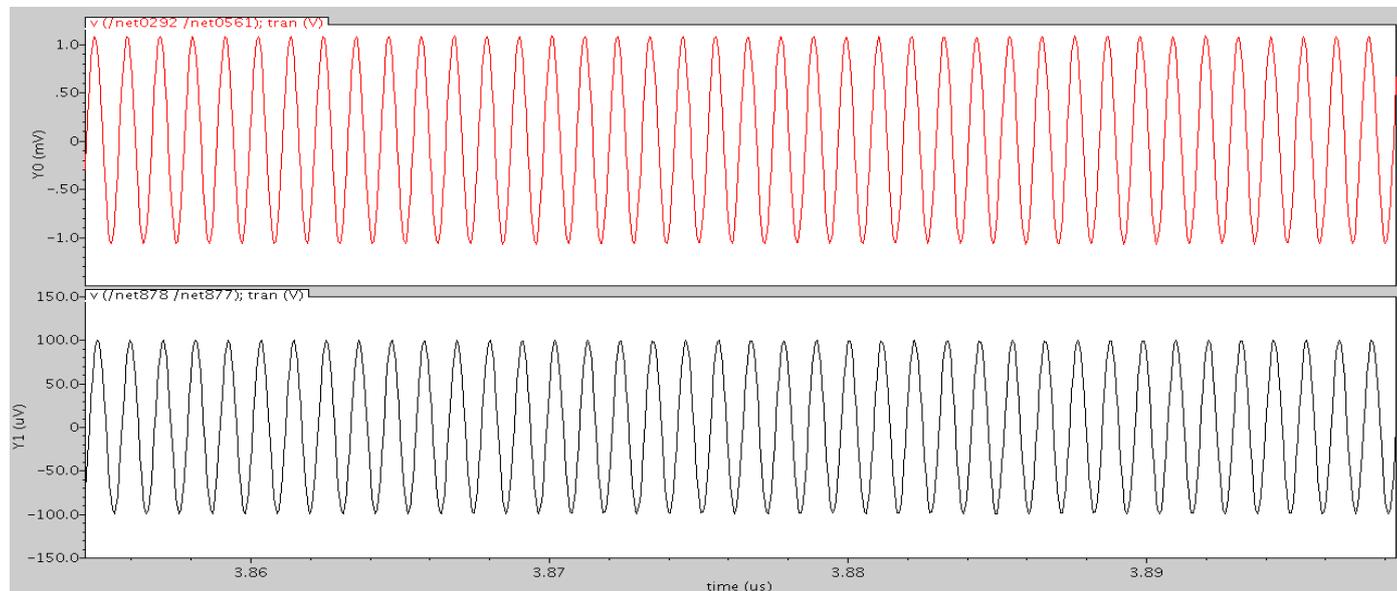


Figure 7.7.1 : the input and the output of LNA

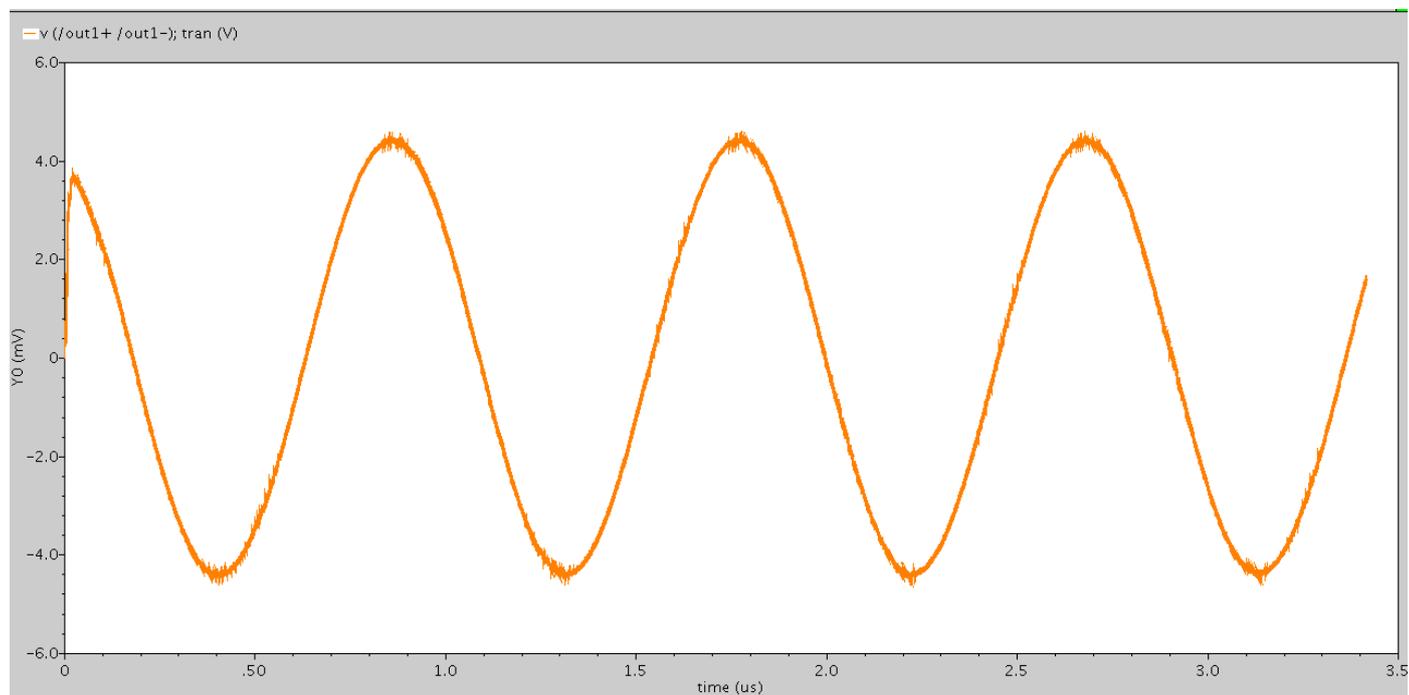


Figure 7.7.2 : the Mixer output

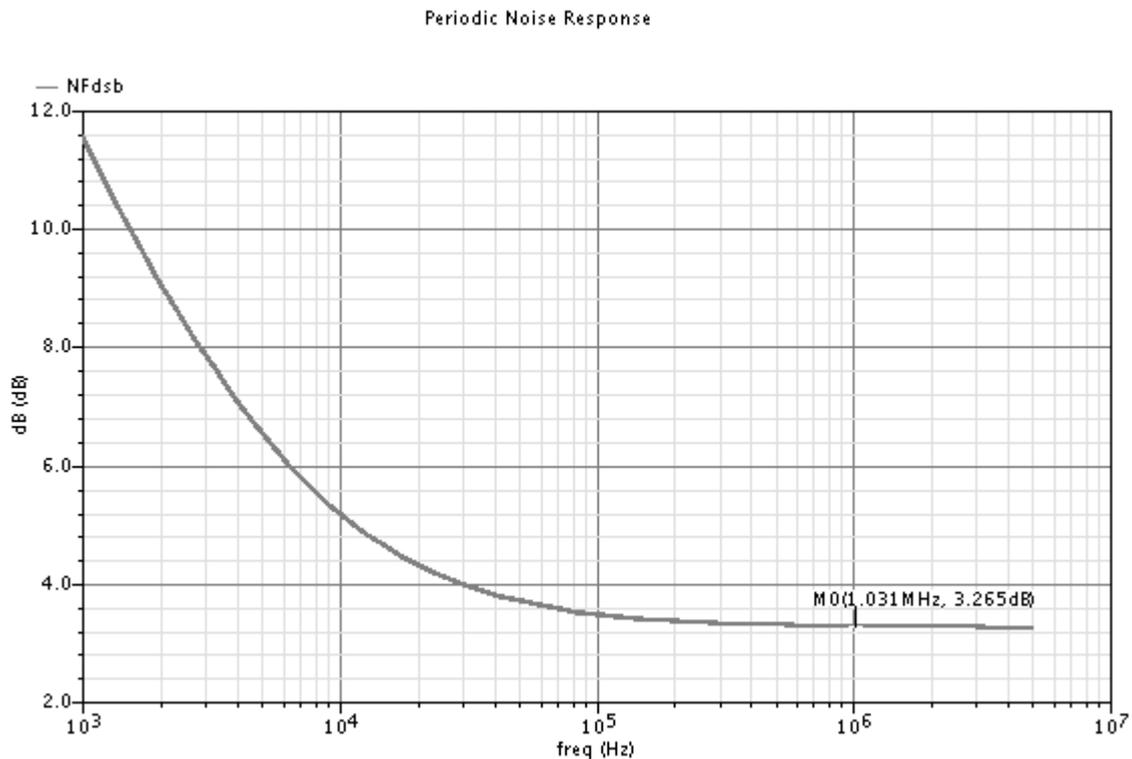


Figure 7.7.3 : the NF of both LNA and mixer

7.2 The LNA-Mixer-Filter interface :

The IF-filter is added to the Mixer and LNA , and a transient time domain analysis is run for all of them , and the following results were obtained the outputs of the LNA and the Mixer remains the same as the ones introduces in above . . .

The output of the filter in the case when the RF signal is in the channel side (the positive frequency side) is shown here clearly below

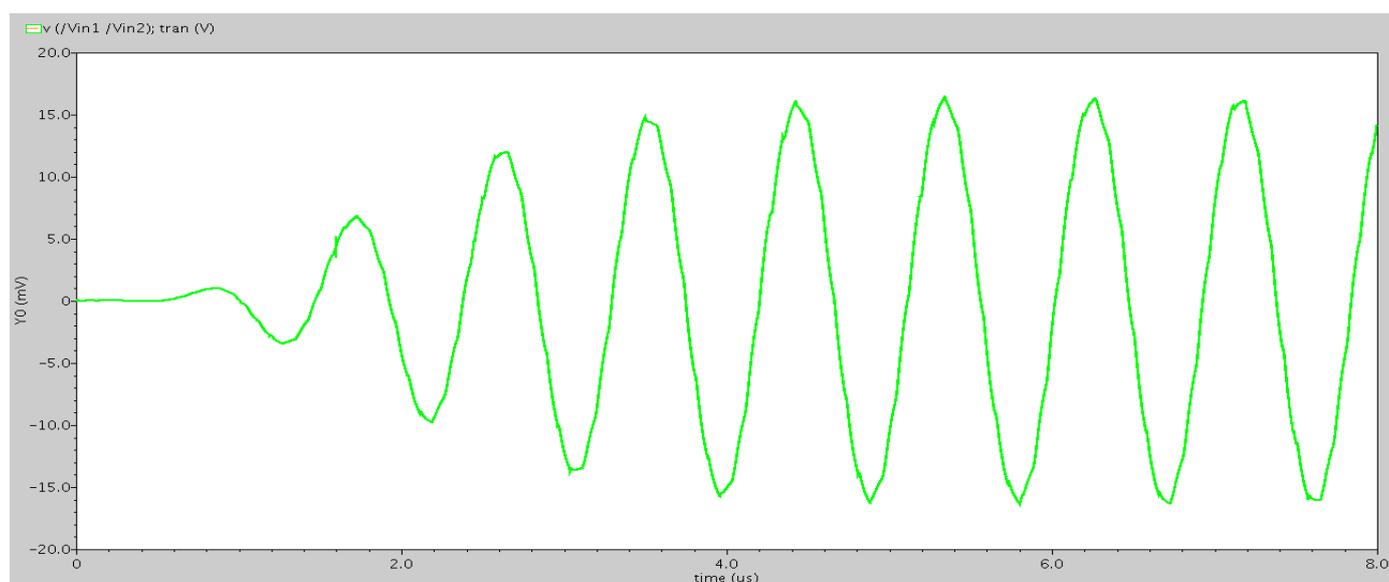


Figure 7.7.4 : the filter output when the RF-input signal is in the channel side

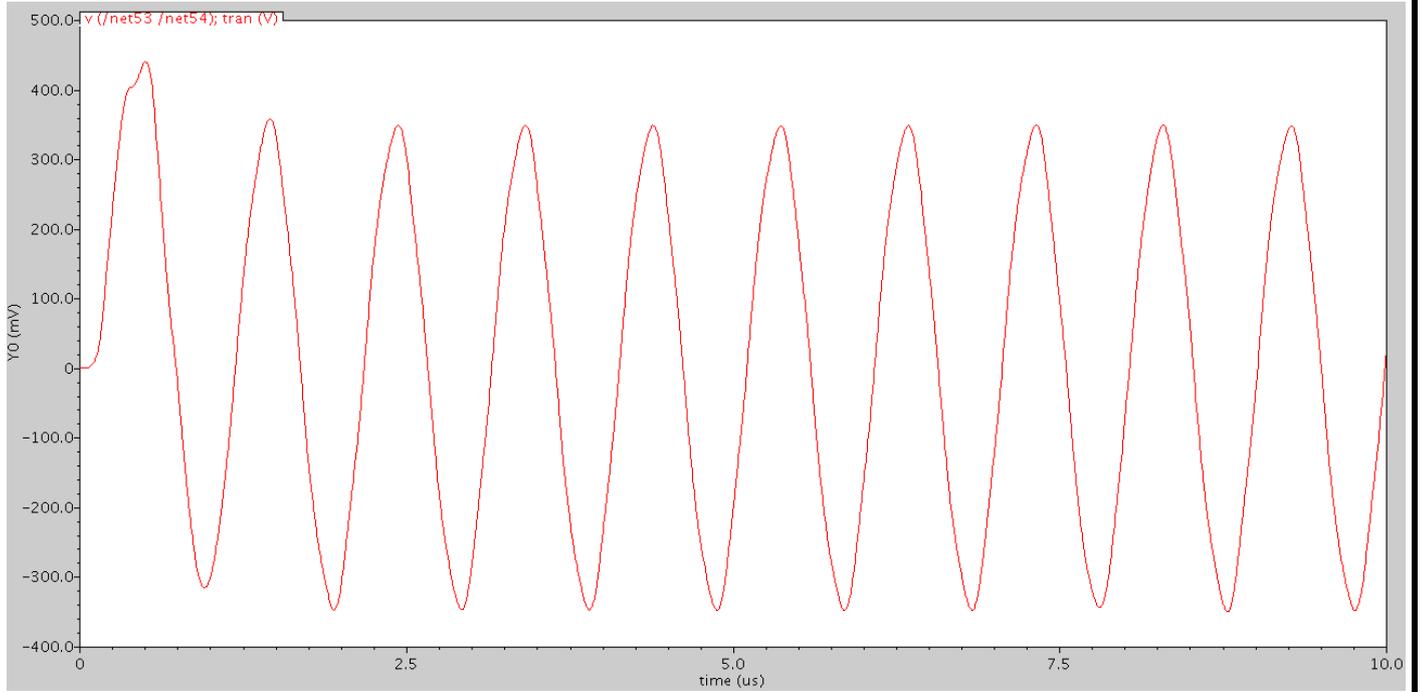


Figure 7.7.7 : the AGC output while interfacing

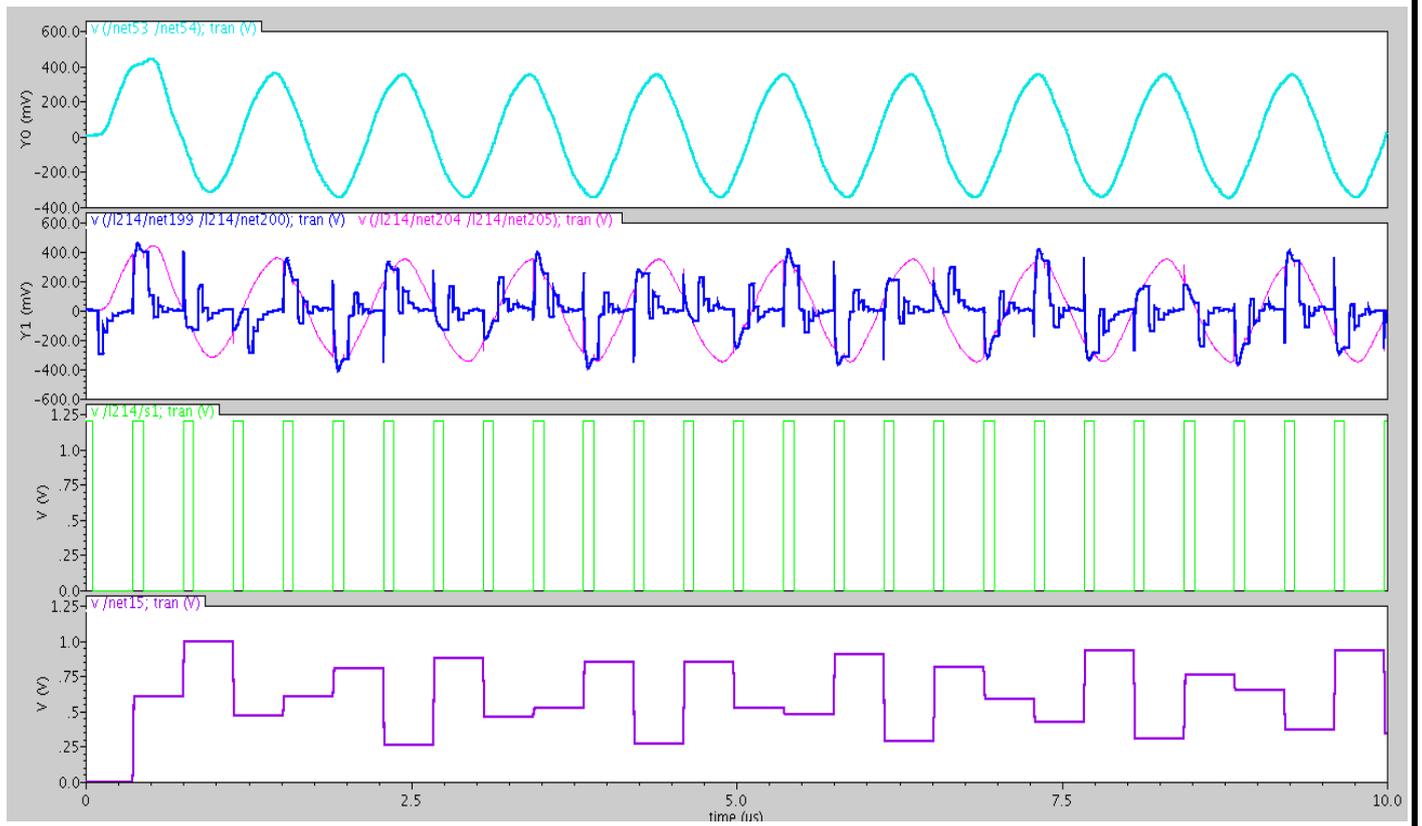


Figure 7.7.8 : the signals throughout the ADC

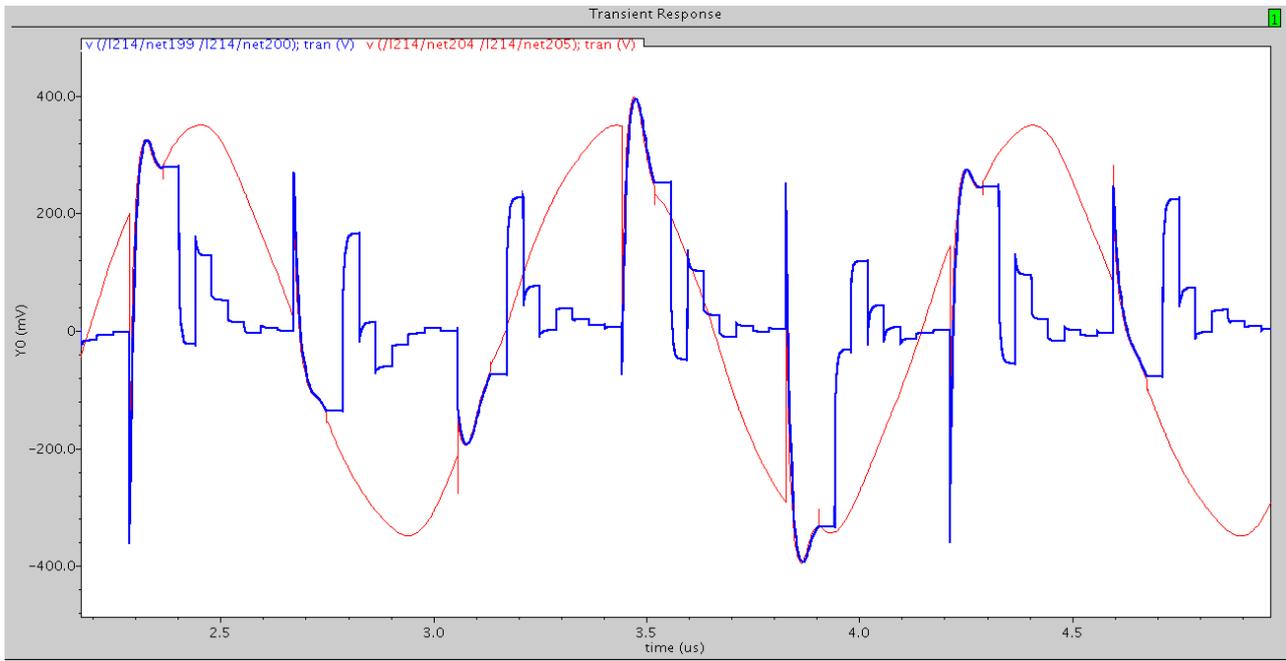


Figure 7.7.9: ADC signal tracking

7.4 The ALL RX chain interfacing results .

These is the final picture that summarizes - in an awesome - way the operation of the Rx chain in one picture , these are the signals after each block in the Rx chain while interfacing between them is held the schematic and the results are shown here

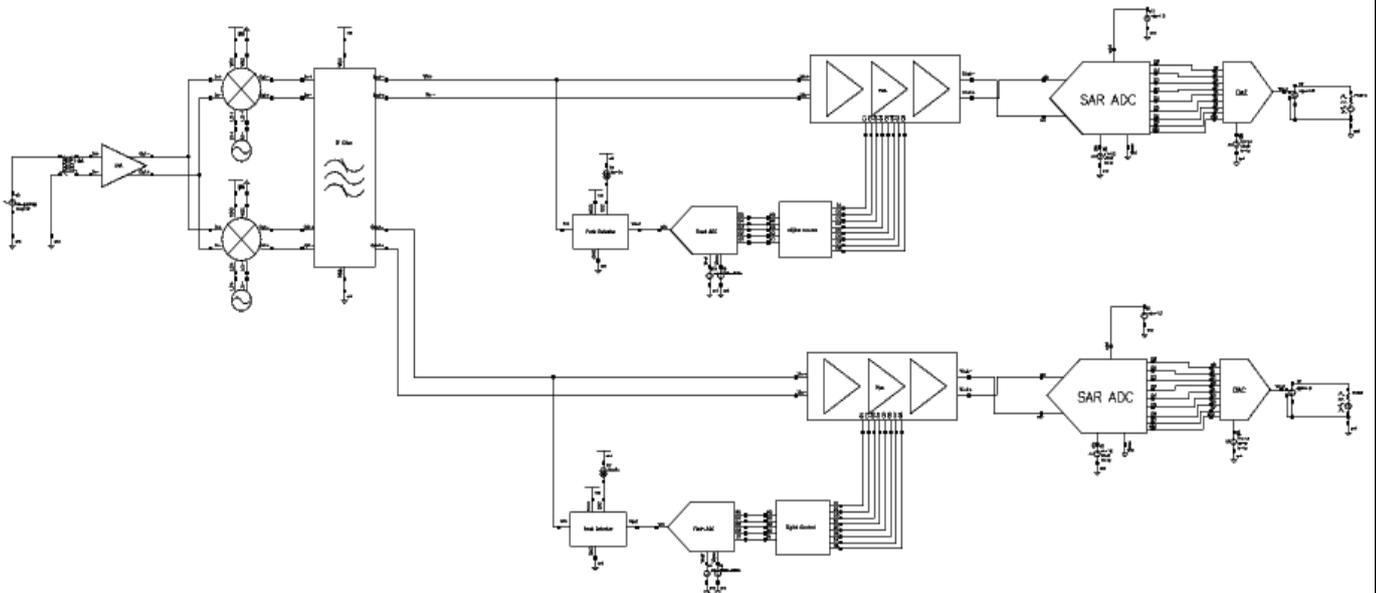


Figure 7.7.10 : the schematic of the all Rx

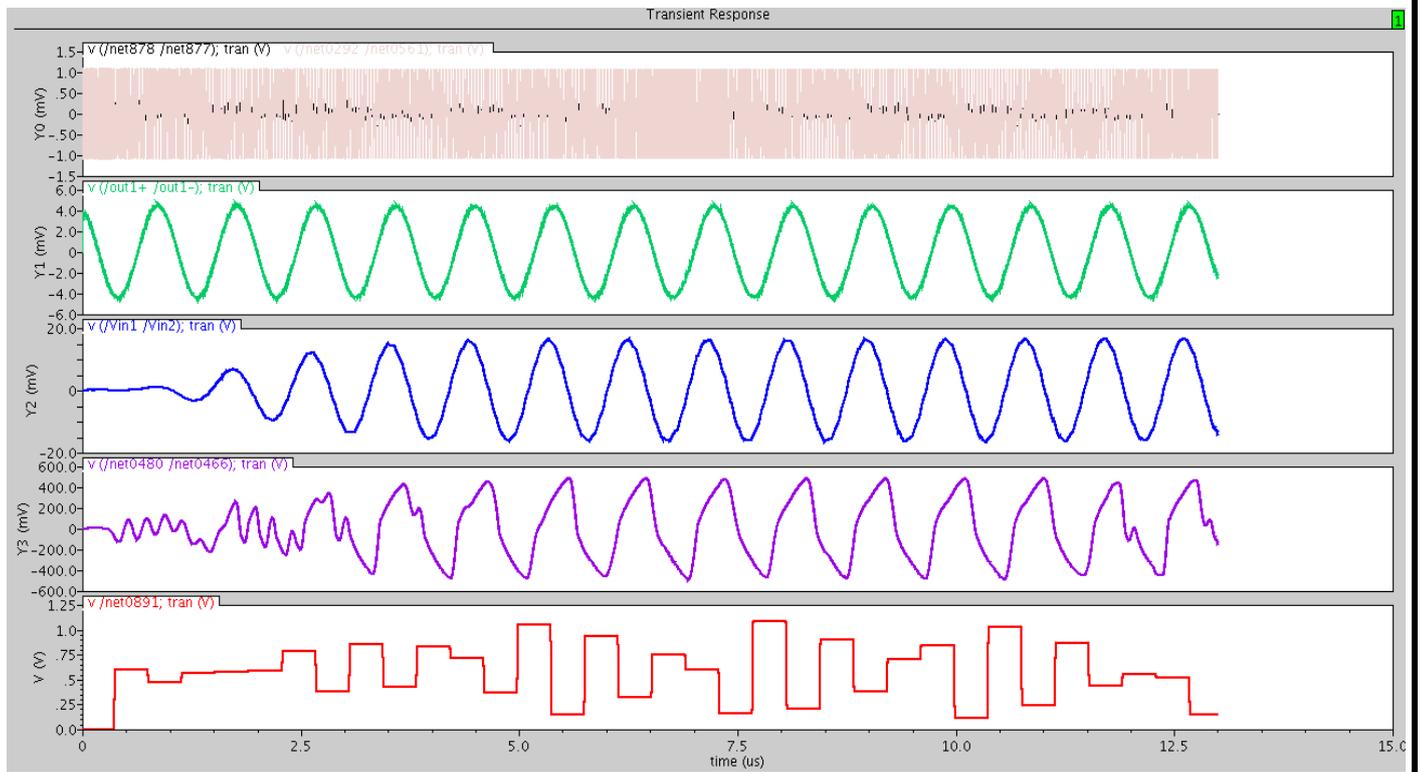


Figure 7.7.11 : the outputs of the each block of the Rx chain

8 Transmitter & PLL System Overview

8.1 Introduction

One of the key components in the integrated RF transceiver is the frequency synthesizer, which is a device capable of generating a set of signals of given output frequencies with very high accuracy and precision from a single reference frequency. The signal generated at the output of the frequency synthesizer is commonly known as local oscillator (LO) signal, used to down convert the message to the baseband or to up convert it to RF frequency to be transmitted.

The most important issues in the frequency synthesizer design are the speed and the spectral purity which differs depending on the wireless standard, we go through design phases for the frequency synthesizer by getting the required specifications from the datasheet, then distributing the specifications among different PLL blocks then show the design steps of each block and their integration verification at the end.

8.2 Specification Extraction

There are some specifications which are used to define any frequency synthesizer

Frequency Range: The synthesizer must be able to cover the entire frequency range of the system; it must be able to generate the channel frequencies required by the system with the required frequency accuracy.

Frequency Resolution: Minimum frequency step guaranteed by the synthesizer, it affects mainly choosing the suitable architecture.

Frequency Accuracy: is the acceptable error for the output frequency from the synthesizer, sets the limit to when the loop can be considered locked.

Locking Time: is the time required for the synthesizer to be locked for maximum frequency step within the required accuracy.

Phase noise: Ideally the output spectrum of the frequency synthesizer is an impulse however in practical it's a phase modulated or amplitude modulated signal with noise sources (Thermal noise, flicker noise, shot noise) measured in units of dBc/Hz, in time domain it's equivalent to timing jitter.

Spurs Level: It's a discrete tones appears at the synthesizer output due to non-idealities in the synthesizer loop as (current mismatch, Timing mismatch, using Fractional synthesizer, ...) & measured in units of dBc.

8.2.1 Frequency Range

We are targeting the ISM band (902-928 MHz) while we are using this frequency synthesizer for transceiver system so we need to use I & Q signals so we need double the frequency with using a divide by 2 divider out of the loop so our loop frequency range will be (1.804GHz - 1.856GHz)

8.2.2 Frequency Resolution

The channel resolution if the transceiver is our frequency resolution equals 750 KHz

8.2.3 Locking Time

From the data sheet [1] the locking time is 75µs.

SPEC	VALUE
Frequency range	902-928Mhz
Frequency accuracy	40 ppm
Frequency resolution	285Khz
Locking time	4µsec
Phase noise @100Khz offset	-92 dBc/hz
Phase noise @1Mhz offset	-107dBc/hz
Phase noise @5Mhz offset	-119dBc/hz
Reference Spurs	Less than -54dB

8.3 Brief Introduction for the phase locked loop (PLL)

8.3.1 Brief Review for PLL system

PLL simply is a negative feedback system where the output tracks the input when we have infinite open loop gain. The main blocks of PLL as shown in the following figure are the reference signal f_{ref} , that is derived from a crystal oscillator with high spectral purity (high SNR), a phase detector (PD) which is used to compare between the phases of two input signals and generates an output voltage directly proportional to the phase difference between them, a low pass filter (LPF) which is used to get the average of the PD output to be applied to the VCO so the loop takes an action by increasing or decreasing the output frequency in order to track the reference signal, a frequency divider N which is placed in the feedback path used to get multiples from the input reference signal frequency. The PLL is considered to be locked when $f_{out} = N \cdot f_{ref}$.

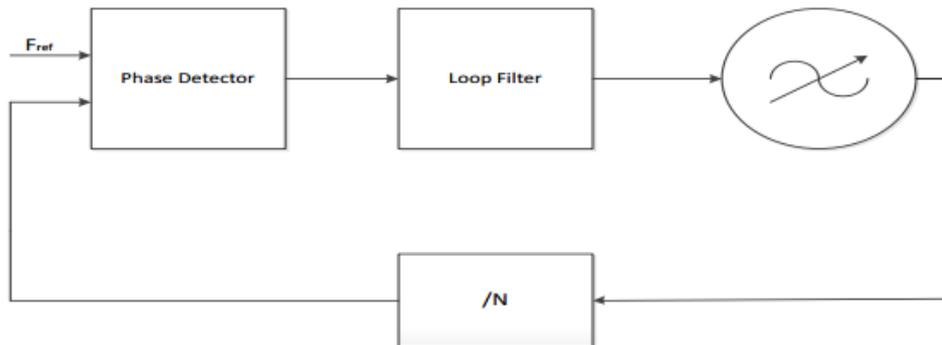


Figure 8.1 Integer N-frequency synthesizer

8.3.2 Building the linearized PLL model

Although the phase detector operation isn't linear it can be approximated by a linear function when the loop is near the locking. Different phase detector types are described in the following chapters. Here we highlight the type we use which is the PFD/CP. Its operation can be described that the PFD (phase frequency detector) can detect which of its 2 inputs have higher frequency pushing a current into or out of the filter changing the value of the control voltage in order to track the input reference signal. The linear model of the PFD/CP is shown in the following figure where, the VCO also can be described by a linear model as shown in the following figure.

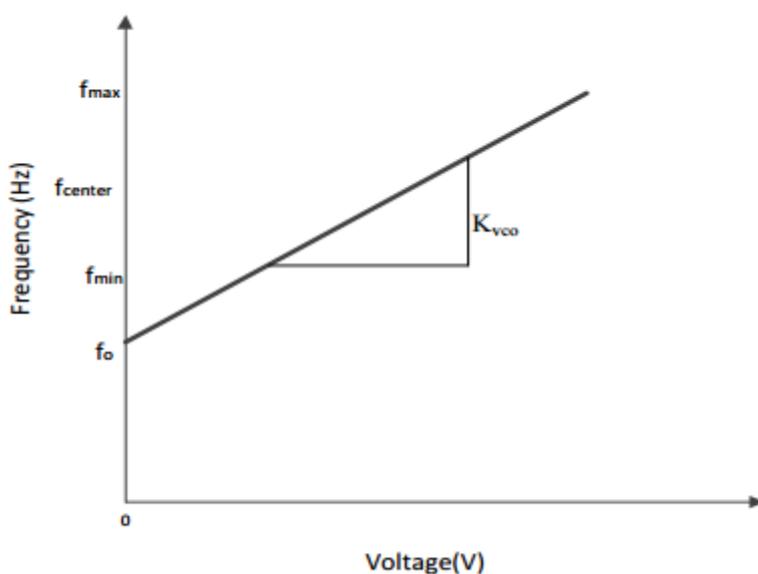


Figure 8.2 VCO linear tuning

Where $f = f_0 + K_{vco}V$

As the phase is the integration of the frequency we can describe the linear relation with.

$$\frac{\Delta\phi}{\Delta V} = \frac{K_{vco}}{s}$$

The divider divides its input frequency (phase) with ratio N so its linear model can be written With $\frac{\phi_{out}}{\phi_{in}} = \frac{1}{N}$.

So open loop gain can be described as $\frac{\phi_0}{\phi_{in}} = \frac{I_{cp} * Z(s) * K_{vco}}{s * N}$ where Z(s) is the loop filter impedance.

Let $K = K_{vco} * \frac{I_{cp}}{2 * \pi}$ then the closed loop gain $H(s) = \frac{N * Z(s) * K}{1 + Z(s) * K}$, if the open loop gain is large enough at the steady state then the closed loop gain approximated to N.

we use the $\frac{\phi_0}{\phi_{in}} = 1 - H(s) = \frac{s}{K * f(s)}$

Z(s) originally consists from only capacitor (one pole) at zero but as this affects the stability of the whole system as we have two poles at zero one from the capacitor and the other from the VCO to solve this we add a zero through adding a series resistance R1 to the filter capacitance then C2 is added to decrease the ripples effect on the control voltage due to switching effect of charge pump.

Then for first order filter $Z(s) = R_1 + \frac{1}{sC_1} = \frac{1 + s * R_1 * C_1}{sC_1}$

To get the transient response of the loop we enter a frequency step to the input $\frac{\Delta w}{s}$ and get the output $\Delta w_{out}(s)$ then taking it's Laplace transform we get the following relation The step response decays with a time constant of $(1/\zeta\omega_n)$ so increasing ζ leads to reaching the final value faster.

Where is the frequency step applied locking time is defined when the frequency reached a value of Δf (frequency accuracy) then by substituting in this equation by max frequency step and the required Δw we can get the locking time.

8.4 Frequency Synthesizer Architectures

The two main architectures are the integer N synthesizer and the fractional synthesizer the advantages and drawbacks of each one are discussed below.

8.4.1.1 Integer N synthesizer

The division ratio is limited to an integer value, this puts a constrain on the used reference frequency that it must be equal to the minimum channel spacing this is usually a small value, and as there's a constrain on the loop bandwidth that it must be smaller than so the continuous time approximation is valid, small loop bandwidth leads to longer locking time, also small f_{ref} means that we will need a large value of division ratios which makes the divider design more complex also this large division ratios leads to large amplification for the in-band phase noise.

8.4.1.2 Fractional N synthesizer

The constrain on the division ratio to be integer is removed, now we have the flexibility to choose large f_{ref} , so we can choose the loop bandwidth which can satisfy our locking time and phase noise specifications, as the divider design must be for integer values we add a simple Accumulator as shown in the fig(9.5) which can change the division ratio between two or more integer values to get the required average. The main problem of using fractional synthesizers the presence of quantization noise which to be filtered in the loop bandwidth it needs a higher order modulators than the simple accumulator but this also put a constrain on the loop filter that the PLL must have at least the same or higher order than the sigma delta modulator, also as the input for this modulators is constant then the output has a repeated sequences which leads to periodic ripples on the control voltage modulating the VCO generating a large discrete

spurs in the output spectrum this problem can be solved by using randomization techniques to break the periodicity of the output pattern then the fractional synthesizer has large advantages in trade of the complexity .

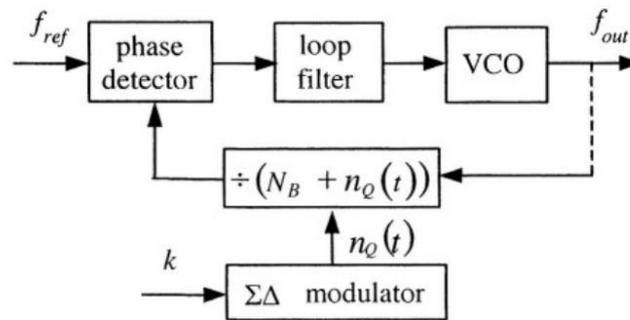


Figure 8.8.3 : fractional N synthesizer

8.5 Calculating Loop Parameters

8.5.1 Calculation for VCO gain and Tuning range

The VCO gain depends on the tuning range and the compliance range so choosing a compliance range of 0.5 volt from 0.4 to 0.9 for a tuning range from 1.802 Ghz to 1.856 Ghz

$$K_{vco} = \frac{1.856 - 1.802}{0.5} = 104 \text{ Mhz/V}$$

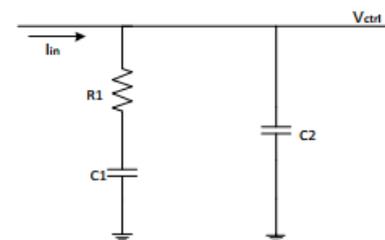
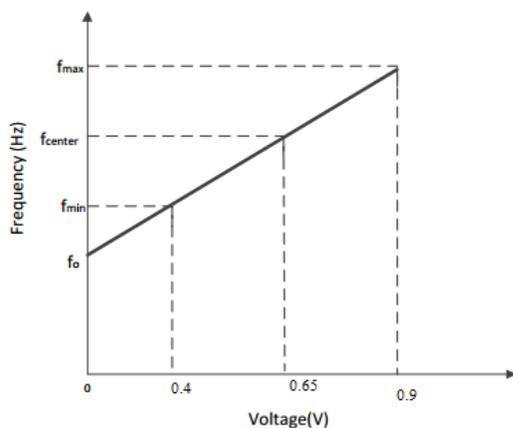


Figure 8.8.4: second order loop filter

8.5.2 Calculating the loop bandwidth

The loop bandwidth is one of the most critical parameters in the loop as changing it changes the locking time of the loop and it affects directly the output noise from the loop

The loop bandwidth has minimum & maximum values it can range between them so as to maintain the continuous time approximation

First we suppose to design the loop bandwidth targeting a phase margin of 60 degree which is on 2 poles but it degrades after adding the third pole due to the pole degradation in phase margin which is well known

$$\text{settling time} < \omega_{loop} < 0.1\omega_{ref}$$

8.5.3 Calculating Loop filter component elements

We begin our design by a second order loop filter as shown in the following Figure.

R1 is added in the loop filter to make the system stable as the loop contains two poles at origin, then C2 is added to decrease the ripples effect on the control voltage due to switching effect of charge pump.

Our goal is to find values of the R & C that satisfy the phase margin 60° and the loop bandwidth by using the following equation which defines open loop gain

$$H(s) = \frac{\frac{I_{cp}}{2\pi} * Z(s) * K_{vco}}{s * N}$$

$$Z(S) = \frac{1 + sT_1}{s(c_1 + c_2)(1 + sT_2)}$$

$$c_s = \frac{c_1 c_2}{c_1 + c_2}$$

$$T_2 = R_2 c_s \quad T_1 = R_1 c_1$$

$$H(s) = \frac{\frac{I_{cp}}{2\pi} * (1 + sT_1) * K_{vco}}{w^2 * N * (1 + wT_1)(1 + wT_2)}$$

So we can calculate the phase margin with the following equation

$$\phi(s) = 180 + \tan^{-1}(wT_1) - \tan^{-1}(wT_2)$$

So after calculating the maximum phase margin by differentiating the previous equation and equating it with zero we can find that the loop band width which maximized the phase margin is

$$w_c = \frac{1}{\sqrt{T_1 T_2}}$$

$$\phi(s) = \tan^{-1}(x) - \tan^{-1}(x)$$

$$w_p = x w_c$$

$$w_z = \frac{w_c}{x}$$

$$R_1 = \frac{I_{cp} K_{vco}}{N} \left(1 - \frac{1}{x^2}\right)$$

$$c_1 = \frac{1}{R_1 w_z}$$

$$c_2 = \frac{c_1}{x^2 - 1}$$

For choosing the I_{cp} I began with 20 μ A then increased it to 50 μ A to be more suppressing for noise

Loop parameter	value
R1	27.21k
C1	100pf
C2	6.61pf

The third pole is chosen to suppress the reference spurs & the noise to meet the needed specs

So the problem here was in the sigma delta quantization noise so I have designed the 3rd pole RC section for that purpose which added a pole to attenuate the sigma delta quantization noise which appeared in lower frequencies than the expected

Loop parameter	value
R3	50k
C3	1pf

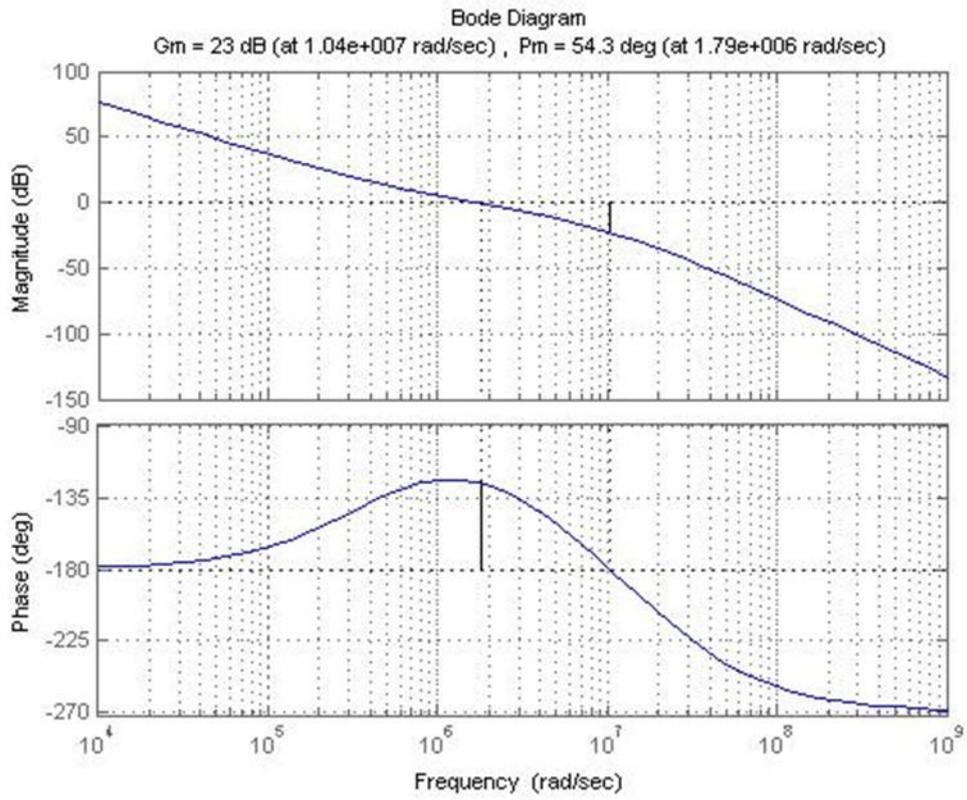


Figure 8.5: Loop stability

8.6 Phase Noise analysis

In this part we need to put specs on the noise of each block to satisfy the phase noise specifications from the required noise specs from the data sheet.

The contribution of each block in the output phase noise profile depends on the noise generated by this block and the transfer function seen by this block to the

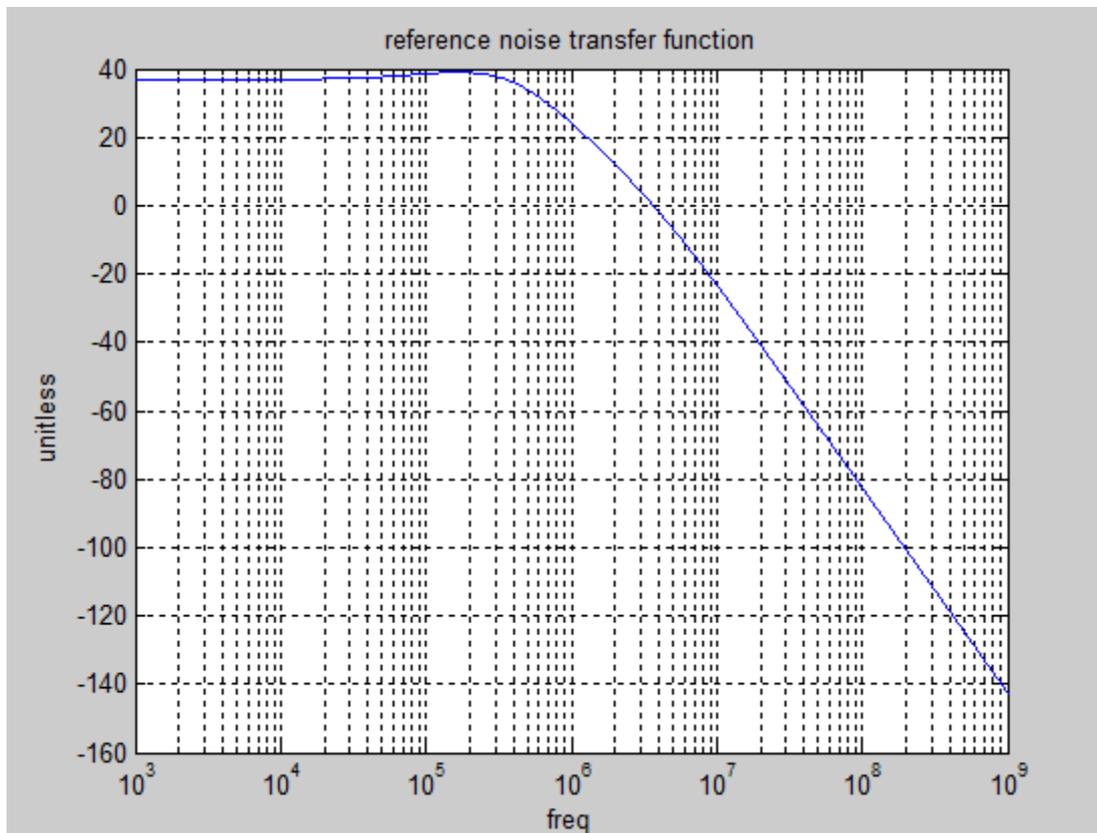
function seen by output of the PLL.

noise source	Phase transfer function		
input noise	$\theta_{out}(s) / \theta_{in}(s)$	$N \cdot \frac{H_{ol}(s)}{1 + H_{ol}(s)}$	low-pass
PFD/CP noise	$\theta_{out}(s) / i_{cp}(s)$	$\frac{N}{K_{pd}} \cdot \frac{H_{ol}(s)}{1 + H_{ol}(s)}$	low-pass
LF noise	$\theta_{out}(s) / v_{lf}(s)$	$\frac{K_{vco}}{s} \cdot \frac{1}{1 + H_{ol}(s)}$	band-pass
VCO noise	$\theta_{out}(s) / \theta_{vco}(s)$	$\frac{1}{1 + H_{ol}(s)}$	high-pass
divider noise	$\theta_{out}(s) / \theta_{div}(s)$	$-N \cdot \frac{H_{ol}(s)}{1 + H_{ol}(s)}$	low-pass

Figure8.8.6: phase noise transfer function of all blocks

8.6.1 Reference oscillator phase noise

It sees a low pass transfer function in the form of



And as of being an oscillator it generates a noise profile is described in lesson formula which consists of 3 regions directly proportional with $(1/f^3)$, $(1/f^2)$ & noise floor

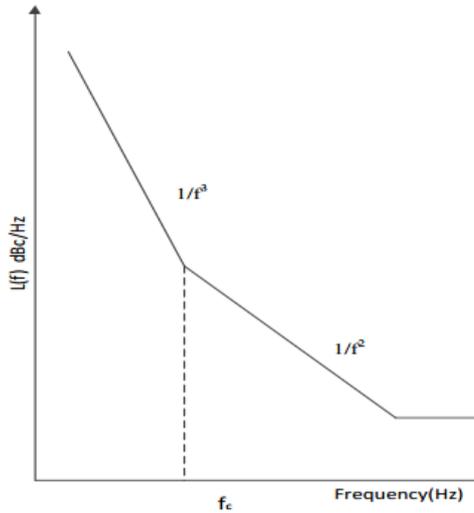


Figure 8.8.8: lesson equation

So the noise of the reference is like the following figure

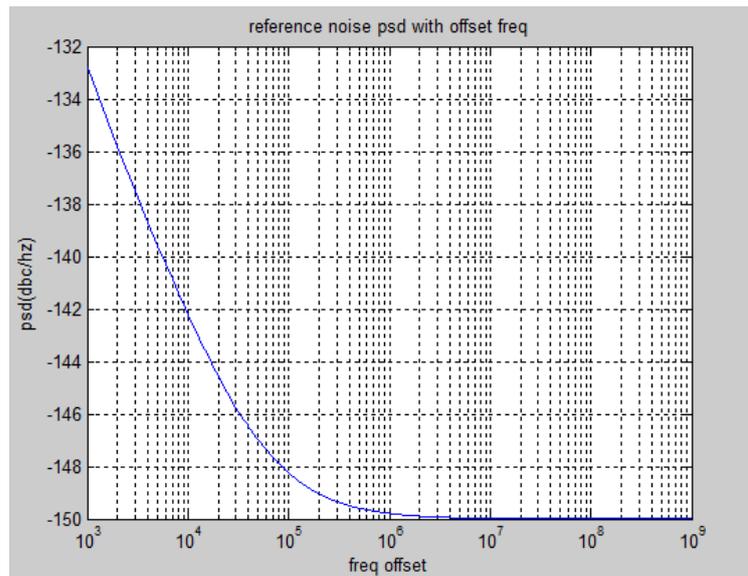


Figure 8.8.7 : reference oscillator phase noise

So its noise contribution in the output after multiplying in the transfer function is

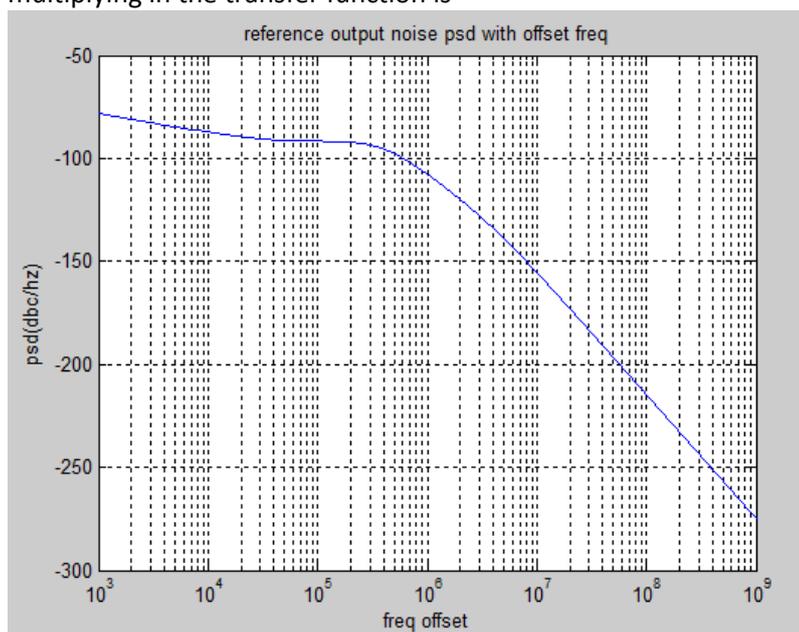


Figure 8.8.9 : Reference oscillator in the output phase noise

parameter	value
Q	83000
Phase noise @ 100Khz offset	-148dBc/hz
Phase noise @ 1Mhz offset	-149dBc/hz

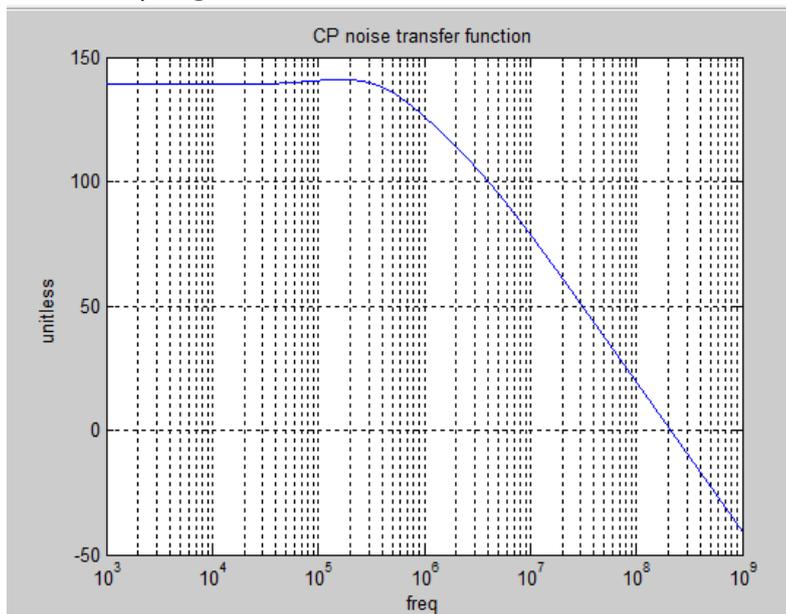
8.6.2 PFD & charge pump noise

Charge pump noise sees a low pass transfer function as said before so its noise appear in the in band region of the loop bandwidth when looking at the transfer function which the charge pump noise faces we can find that it is inversely proportional with the charge pump current so increasing the charge pump current decreases the output noise .

Choosing the noise floor which in terms responded on the noise of the charge pump based on leaving a noise margin between the vco phase noise and the charge pump phase noise by about 5 or 6 dB

The charge pump phase noise depends on the noise floor, flicker corner & alpha which is the fraction of time when the charge pump is ON and it is translated in the reset time as a fraction of the reference period

So our VCO noise is -107 dBc/hz @ 1MHZ offset frequency from the carrier so the charge pump noise is about to be -112dBc/hz @ 1Mhz offset



For this equation we can choose the charge pump output noise floor to be $4 \text{ PA}/\sqrt{\text{hz}}$

And alpha to be 0.01

And the flicker corner to be 100khz

Figure 8.8.10 : Charge pump noise transfer function

The charge pump consists of transistors mainly so it has flicker noise and thermal noise so its rms output noise

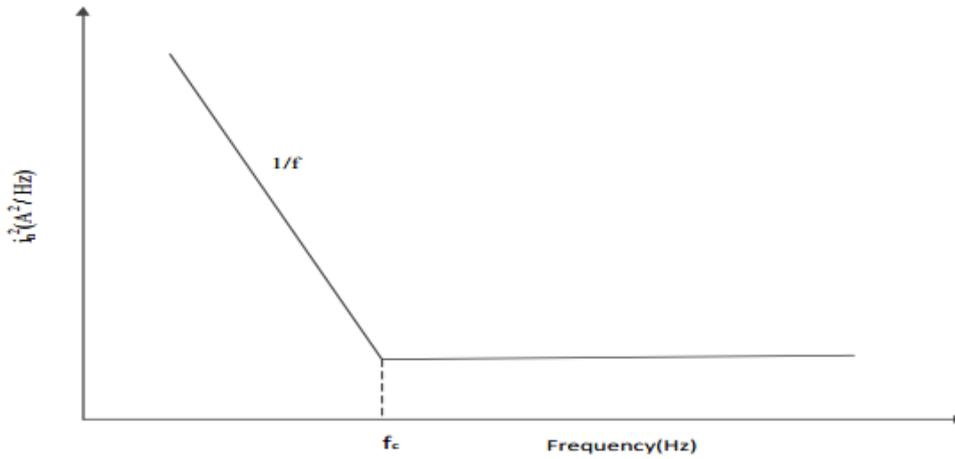


Figure 8.8.11 : Charge pump noise form

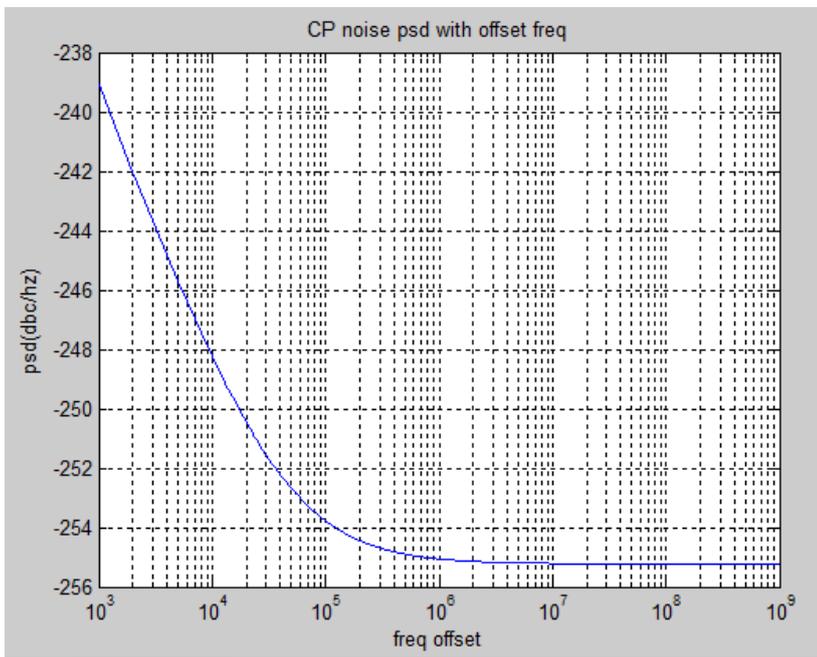


Figure 8.8.12 : Charge pump phase noise

So after the multiplication of the noise generated from the charge pump with the transfer function The contribution of the charge pump noise to the total output noise will be like the following figure and we can see that it is dominant in the in band region

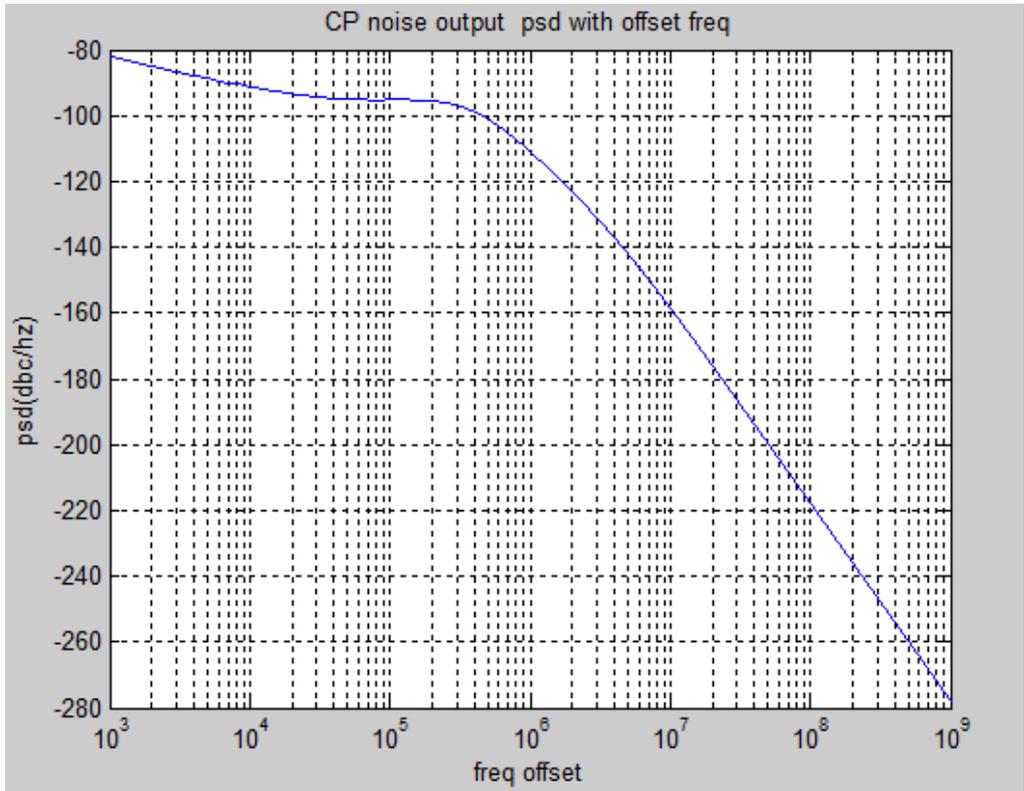


Figure 8.8.13 : Charge pump noise contribution in the output

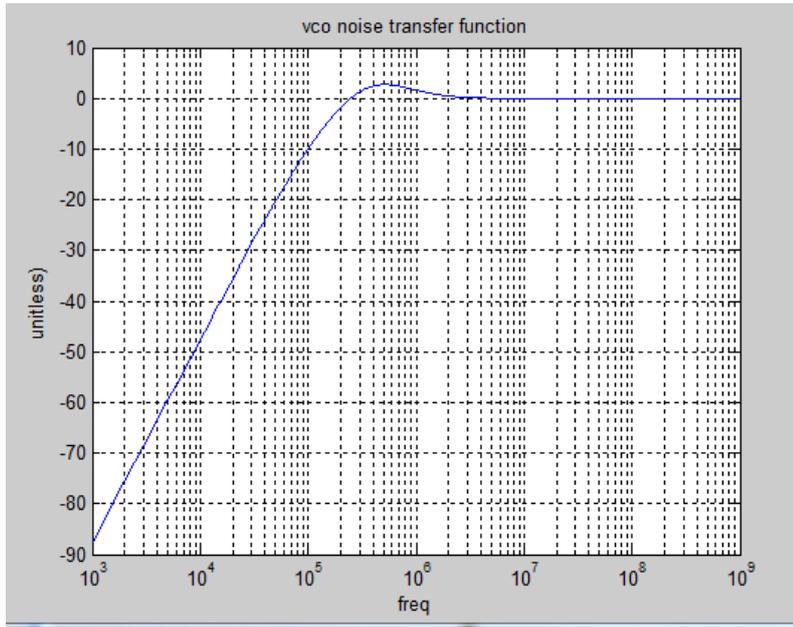
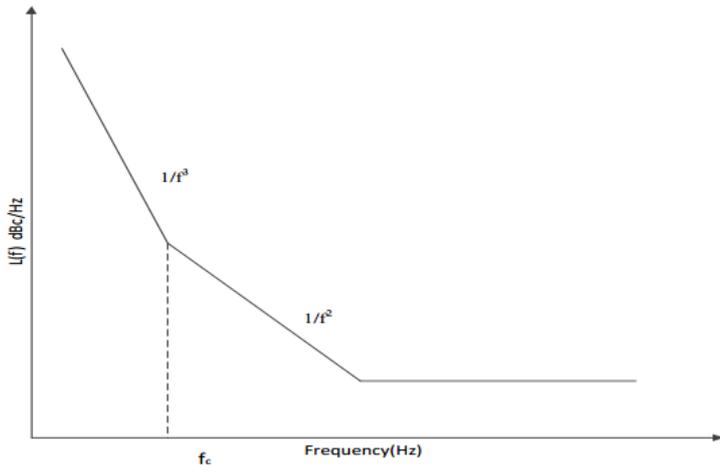


Figure 8.8.14 : VCO noise transfer function

The VCO noise obeys leeson equation as said before in the reference oscillator
 And as of being an oscillator it generates a noise profile is described in lesson formula which consists of 3 regions directly proportional with $(1/f^3)$, $(1/f^2)$ & noise floor

8.6.3 VCO phase noise

VCO noise is the most important factor in the noise of the phase locked loop as the Voltage controlled oscillator is a very noisy block its noise faces a high pass transfer function
 The following figure shows the VCO noise transfer function



So the phase noise generated from the VCO is like the following figure

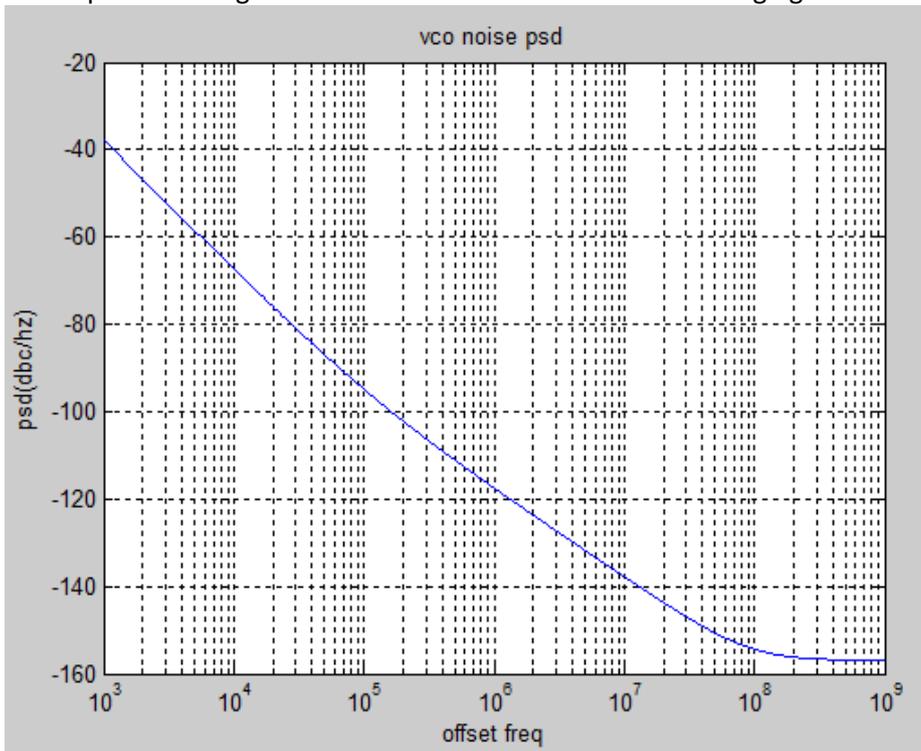


Figure 8.8.15 : VCO phase noise

So after the multiplication of that noise in the transfer function the VCO the noise contribution of the VCO in the output phase noise of the PLL

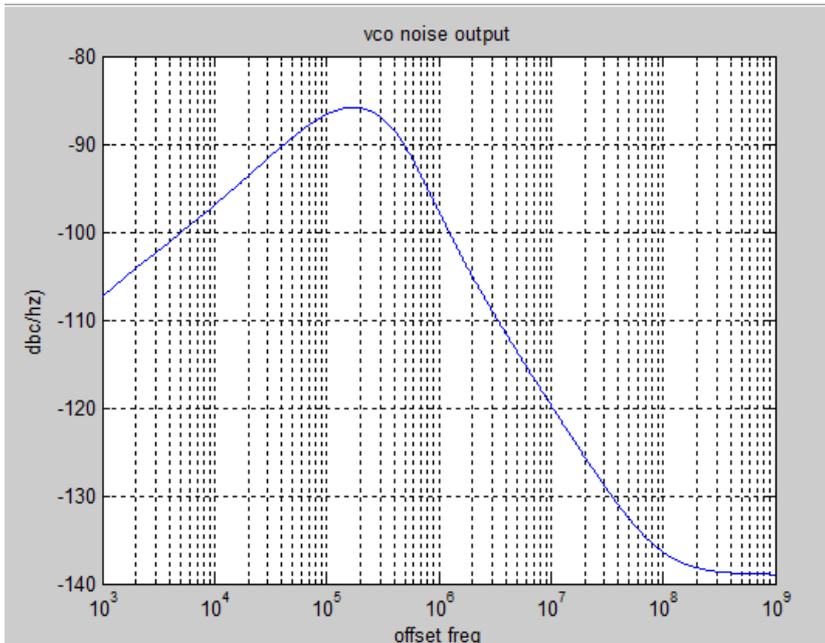


Figure 8.8.16 : VCO phase noise contribution in the output

8.6.4 Sigma delta phase noise

Sigma delta is a block which makes noise shaping but it has quantization noise comes from changing the sampling theory which uses the reference frequency as the sampling frequency .

the sigma delta noise faces a low noise transfer function but it's noise be more dominant at frequencies higher than that where the VCO noise is dominant .

we begin analysis with a second order sigma delta but due to its high in-band phase noise which doesn't satisfy our phase noise specs we switch for a third order modulator but another problem appears which is its out-band noise which required from us adding a third pole for the filter .

The sigma delta noise transfer function is as the following figure.

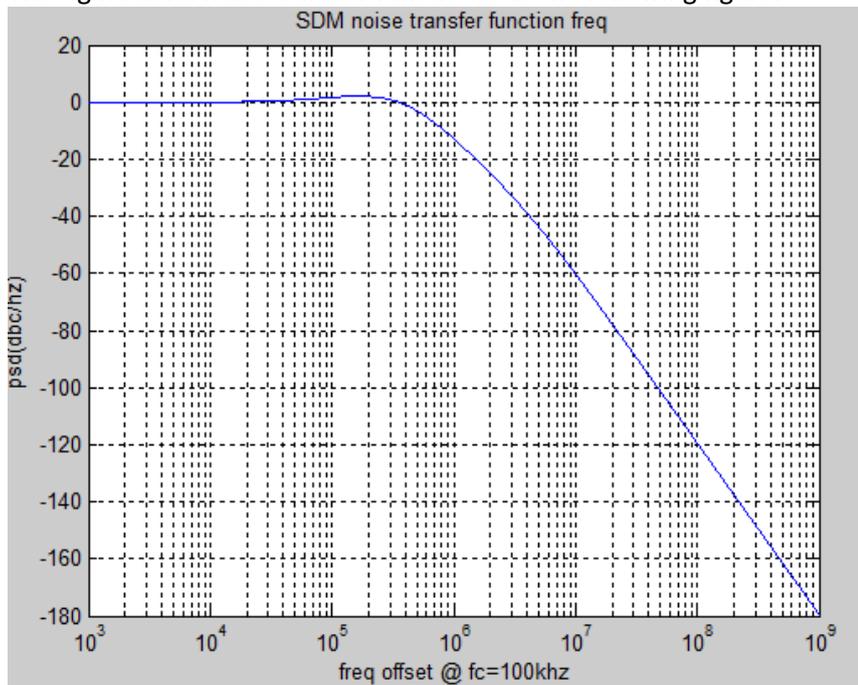


Figure 8.8.17 : SDM noise transfer function

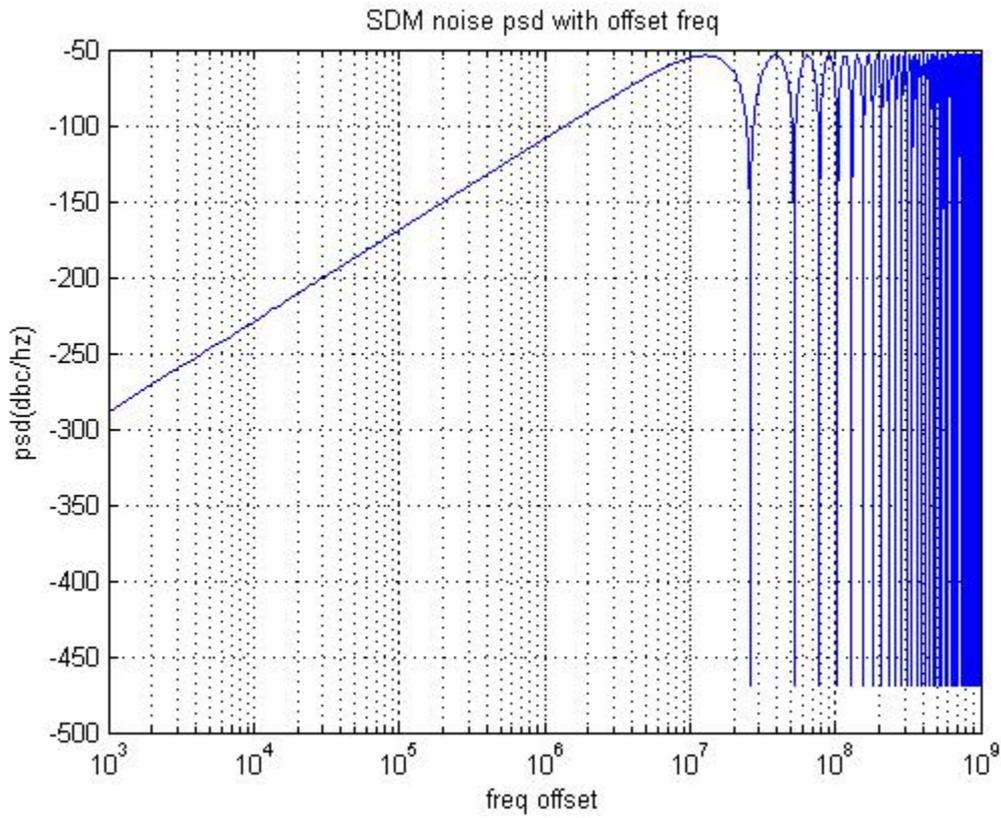


Figure 8.18: SDM noise psd

Parameter	Value	
Reference noise @ 100k	-148	dBc/hz
Reference noise @ 500k	-149.5	
Reference noise @ 1M	-149.78	
Reference noise @ 5M	-149.95	
Charge pump noise @ 100k	-253	
Charge pump noise @ 500k	254	
Charge pump noise @ 1M	-255	
Charge pump noise @ 5M	-255.19	
VCO noise @ 100k	-97	
VCO noise @ 500k	-113	
VCO noise @ 1M	-120	
VCO noise @ 5M	-134	
divider noise @ 100k	-136.39	
divider noise @ 500k	-139.2	
divider noise @ 1M	-139.5	
divider noise @ 5M	-139.91	

8.7 Reference spurs

In the PLL, the reference frequency modulates the VCO generating sidebands around the carrier.

Ideally, the up and down currents are equal, that is Therefore in the locked-state, the VCO control voltage variation is only due to noises in the PLL.

In practice, the non-idealities of the charge-pump cause periodic ripples on the control voltage (V_c), then we can get the output of the VCO (V_{out}) by assuming that the fundamental component of this ripples $V \cos(w_{ref}t)$

$$V_{out} = V_o \cos(w_0 t + \int V \cos(w_{ref} t))$$

$$V_{out} = V_o \cos(w_0 t + \frac{K_0 V}{f_{ref}} \sin(w_{ref} t))$$

By Assuming $\cos\left(\frac{K_0 V}{f_{ref}} \sin(w_{ref} t)\right) \approx 1$

$$\sin\left(\frac{K_0 V}{f_{ref}} \sin(w_{ref} t)\right) \approx \frac{K_0 V}{f_{ref}} \sin(w_{ref} t)$$

$$V_{out} = V_o \cos(w_0 t) - \frac{K_0 V}{f_{ref}} \sin(w_0 t) \sin(w_{ref} t)$$

$$V_{out} = V_o \cos(w_0 t) - \frac{K_0 V}{2f_{ref}} \cos((w_0 + w_{ref})t) - \frac{K_0 V}{2f_{ref}} \cos((w_0 - w_{ref})t)$$

Therefore

$$\text{Reference spurs} = \text{mag}\left(\frac{K_0 V}{2f_{ref}}\right) \text{ dBc}$$

All the previous analysis is an approximation that we assumed that the loop is opened during locking

Reference spurs can result due to 3 actions

- (a) current leakage
- (b) current mismatch
- (c) timing mismatch

Current Leakage: The leakage current from up and down signals affects the control voltage all the time which requires that the loop will make a small phase offset to compensate this leakage this periodic action resulted in periodic wave form on the control voltage which transferred to the output as discrete tones

Current Mismatch: As the up and down currents of the charge pump may have some mismatch and not pump the same current then during locking since the charge pump is on for the time (t) on due to the reset path in the PFD, the current pumped to the filter during this period isn't equal to zero then also the loop makes a small phase shift to compensate for this effect.

Timing Mismatch: The spurs resulted due to the mismatch between the up and down signals which needed to be compensated at the end of the on time.

Mainly the most effective Mismatch for the reference spurs is the current mismatch always the Timing Mismatch and Leakage current effect is much smaller than the effect of the current mismatch

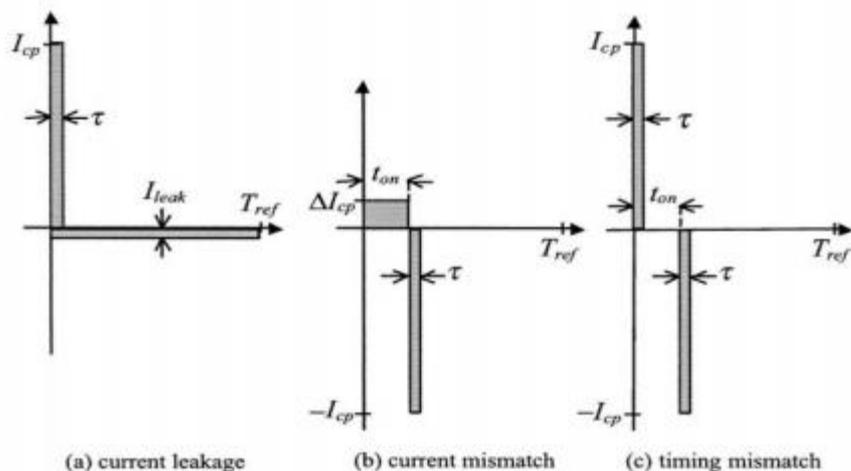


Figure 8.8.19 : Reasons for reference spurs

8.8 CPPSIM system design

Dynamic Parameters		Noise Parameters	
fo: 250e3 Hz	paris.pole: _____ Hz <input type="checkbox"/> On	ref.freq: 26e6 Hz	
order: <input type="radio"/> 1 <input type="radio"/> 2 <input checked="" type="radio"/> 3	paris.Q: _____ <input type="checkbox"/> On	out.freq: 1830e6 Hz	
shape: <input checked="" type="radio"/> Butter <input type="radio"/> Bessel	paris.pole: _____ Hz <input type="checkbox"/> On	Detector: -113 dBc/Hz <input type="checkbox"/> On	
<input type="radio"/> Cheby1 <input type="radio"/> Cheby2 <input type="radio"/> Elliptical	paris.Q: _____ <input type="checkbox"/> On	VCO: -107 dBc/Hz <input type="checkbox"/> On	
ripple: _____ dB	paris.pole: _____ Hz <input type="checkbox"/> On	freq.offset: 1e6 Hz	
type: <input type="radio"/> 1 <input checked="" type="radio"/> 2	paris.pole: _____ Hz <input type="checkbox"/> On	S-D: <input type="radio"/> 1 <input type="radio"/> 2 <input type="checkbox"/> On	<input type="checkbox"/> On
fz/fc: 1/10	paris.pole: _____ Hz <input type="checkbox"/> On	<input checked="" type="radio"/> 3 <input type="radio"/> 4 <input type="radio"/> 5	<input type="checkbox"/> On
	paris.zero: _____ Hz <input type="checkbox"/> On		
	paris.zero: _____ Hz <input type="checkbox"/> On		
Resulting Open Loop Parameters		Resulting Plots and Jitter	
K: 1.371e+011	alter: _____ <input type="checkbox"/> On	<input type="radio"/> Pole/Zero Diagram	<input type="radio"/> Transfer Function
fp: 3.750e+005 Hz	alter: _____ <input type="checkbox"/> On	<input type="radio"/> Step Response	<input checked="" type="radio"/> Noise Plot
fz: 2.500e+004 Hz	alter: _____ <input type="checkbox"/> On	Xmin? Xmax? Ymin? Ymax?	
Qp: 7.059e-001	alter: _____ <input type="checkbox"/> On	rms jitter: 2.548 ps	
PLL Design Assistant		Written by Michael Perrott (http://www.cppsim.com)	

Figure 8.8.20 : CPPSIM parameters

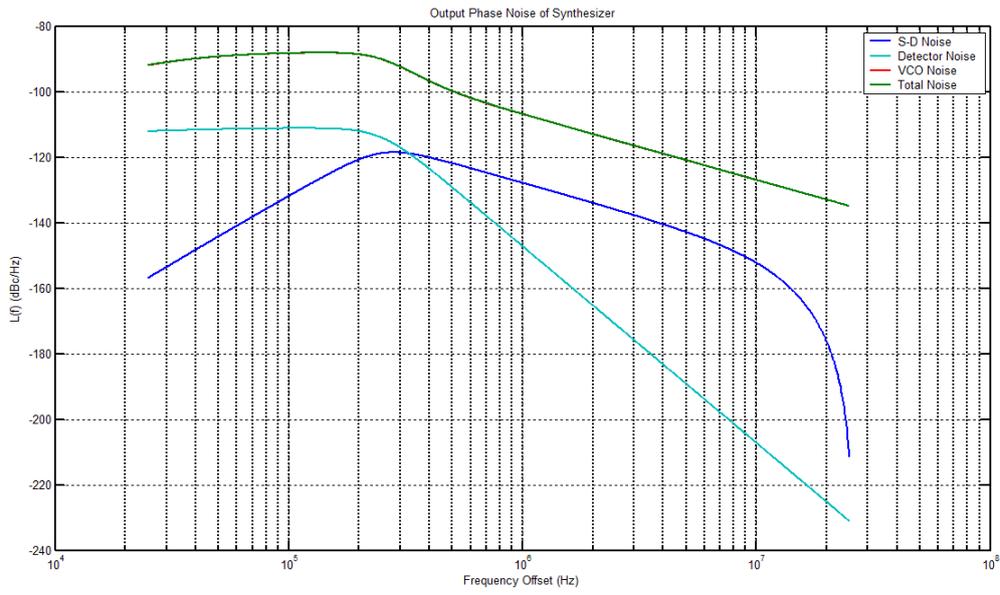


Figure 8.8.21 : CPPSIM noise results

9 Crystal oscillator

9.1 Introduction

9.1.1 Crystal oscillator in PLL

- Crystal oscillator can be abbreviated as XO where X is crystal and O is oscillator.
- Crystal oscillators use an external crystal to generate clock signals for designs that require high stability at a precise frequency.
- Phase Locked Loops (PLL) are used to multiply the crystal oscillator output frequency to higher values.

Contribution of crystal in PLL phase noise:

Contribution of XO in the phase noise of the PLL depends on the noise generated by the XO and the transfer function seen by the crystal oscillator until it reaches the output of the loop.

The transfer function seen by XO is:

$$\frac{\theta(S)_{out}}{\theta(S)_{in}} = N \frac{H_{ol}(S)}{1 + H_{ol}(S)}$$

Where N is the division ratio of the divider and $H_{ol}(S)$ is the transfer function of the open loop.

Specs achieved:

Frequency = 26.01 Mhz

Startup time = 143 μ sec

Current consumption in the core = 500 μ A

PN at 1 MHz = -147 dBc/Hz



Figure 9.1: crystal oscillator

9.1.2 Why crystal oscillator?

Circuits need a stable in frequency high quality factor, low phase noise and low power consumption as a reference oscillator

9.1.3 Crystal oscillator applications

<u>Military & Aerospace</u>	<u>Industrial</u>	<u>Consumer</u>
Communications Navigation IFF Radar Sensors Guidance systems Fuzes Electronic warfare Sonobouys	Communications Telecommunications Mobile/cellular/portable radio, telephone & pager Aviation Marine Navigation Instrumentation Computers Digital systems CRT displays Disk drives Modems Tagging/identification Utilities Sensors	Watches & clocks Cellular & cordless phones, pagers Radio & hi-fi equipment Color TV Cable TV systems Home computers VCR & video camera CB & amateur radio Toys & games Pacemakers Other medical devices
<u>Research & Metrology</u> Atomic clocks Instruments Astronomy & geodesy Space tracking Celestial navigation		<u>Automotive</u> Engine control, stereo, clock Trip computer, GPS

Figure 9. 2: Crystal uses

9.1.4 What is crystal oscillator?

Is an electronic oscillator circuit that uses the mechanical resonance of a vibrating crystal of piezoelectric material to create an electrical signal with a very precise frequency, This frequency is commonly used to keep track of time (as in quartz wristwatches), to provide a stable clock signal for digitalintegrated circuits, and to stabilize frequencies for radio transmitters and receivers. The most common type of piezoelectric resonator used is the quartz crystal, so oscillator circuits incorporating them became known as crystal oscillators, but other piezoelectric materials including polycrystalline ceramics are used in similar circuits, quartz crystals are manufactured for frequencies from a few tens of kilohertz to tens of megahertz. More than two billion crystals are manufactured annually. Most are used for consumer devices such as wristwatches, clocks, radios, computers, and cellphones. Quartz crystals are also found inside test and measurement equipment, such as counters, signal generators, and oscilloscopes.

9.1.5 Piezoelectricity

- Means electricity resulting from pressure it is derived from the Greek word “piezo” which means press and “electric”.
- It was discovered by Jacques and Pierre Currie.
- It is the electric charge that accumulates in certain solid materials like crystals, ceramics.
- It can be understood as the linear electromechanical interaction between the mechanical and the electrical state in crystalline materials , piezoelectric effect is also reversible which means you can generate mechanical strain from an applied electrical filed.

9.1.6 Some historical notes

- 1880 piezoelectric effect discovered by Jacques and Pierre Curie
- 1905 First hydrothermal growth of quartz in a laboratory by G. Speza
- 1917 First application of piezoelectric effect, in sonar
- 1918 First use of piezoelectric crystal in an oscillator
- 1926 First quartz crystal controlled broadcast station
- 1927 First quartz clock built
- 1949 Contoured, high-Q, high stability AT-cuts developed
- 1956 First commercially grown cultured quartz available

9.2 Crystal oscillator specifications

Phase noise:

It measures the signal quality of the XO output. It is simply the Noise-to-Signal ratio of the XO signal spectrum at a certain frequency offset normalized to 1Hz bandwidth. It is measured in dBc/Hz. This specification is one of the most important specification and sometimes the most difficult to meet.

Oscillation frequency:

It is the frequency of operation of the crystal oscillator.

Startup time:

Time at which the output wave reaches a stable amplitude.

Load capacitance:

Capacitance across the crystal terminals, also it decides if the crystal is series resonance or parallel resonance.

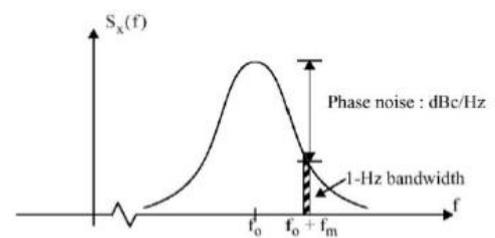
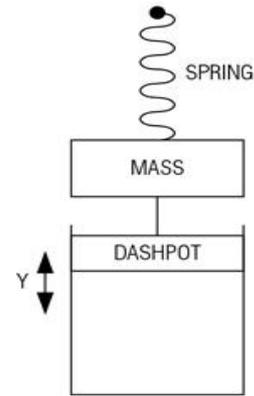


Figure 9.3: Phase Noise

9.3 Crystal modeling

Mechanical model

- $\sum F = 0$
- $M \frac{d^2Y}{dt^2} + D \frac{dY}{dt} + KY = 0$
- $\frac{d^2Y}{dt^2} + \frac{D}{M} \frac{dY}{dt} + \frac{K}{M} Y = 0$



Electrical model

- $\sum V = 0$
- $\frac{1}{c} M \int Idt + L \frac{dI}{dt} + IR = 0$
- $L \frac{d^2Q}{dt^2} + R \frac{dQ}{dt} + \frac{Q}{c} = 0$
- $\frac{d^2Q}{dt^2} + \frac{R}{L} \frac{dQ}{dt} + \frac{Q}{LC} = 0$

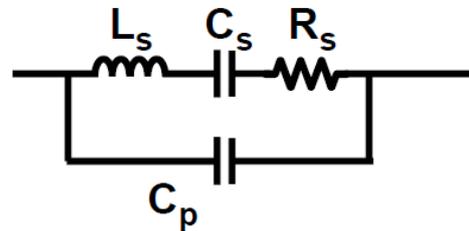


Figure 9.4: Electrical model and mechanical model

This mechanical and electrical models above are equivalent so the mass, the spring and the dashpot equivalent to the inductor, the capacitor and the resistor respectively

R_s , L_s , C_s are called the motion arm elements which are equivalent to the mechanical vibrations of the crystal and C_0 is called the shunt capacitance which is the capacitance between the plates of the crystal plus the stray capacitance due to crystal enclosure

Mechanical resonance frequency = $\sqrt{\frac{K}{M}}$ while electrical resonance = $\frac{1}{\sqrt{LC}}$

9.3.1 Series or Parallel Resonant Crystals

Actually there is no difference in construction between series and parallel resonant circuit both are manufactured exactly alike but when we decided using either? The answer is if there is load capacitance or not, if there is a load capacitance so use parallel resonance.

Series mode is at F_s while anti-resonance is at F_a between them there is a parallel mode which depends on the load capacitance.

$$F_s = \frac{1}{2\pi\sqrt{L_s C_s}}$$

$$F_a = F_s \left(1 + \frac{C_s}{2C_p}\right)$$

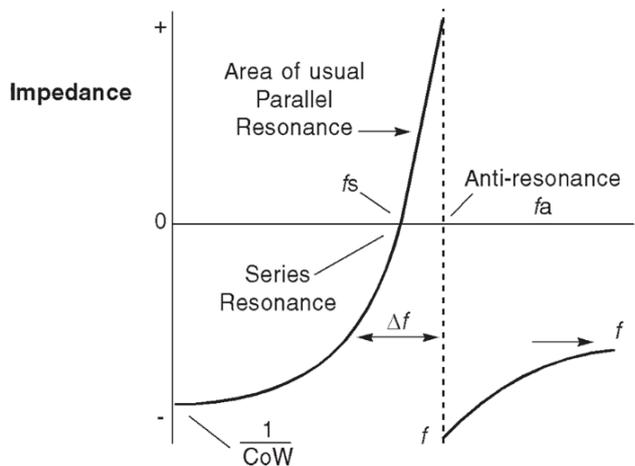


Figure 9.5: resonance modes

9.4 Theory of oscillation

To have oscillation on a certain frequency first there must be a feedback loop then at this loop and at the required frequency loop gain and loop phase must follow Barkhausen criterion which says that loop gain must be greater than 0 dB and loop phase must be zero degrees

$$|\beta A| \geq 1$$

$$\angle \beta A = 2\pi n \text{ Where } n = 0, 1, 2, 3, 4 \dots$$

- When oscillation initially energized, the only signal in the circuit is noise of the fundamental frequency and its harmonics then the gain of the loop determines which frequency that can propagate in the feedback loop.
- This signal propagates in the loop and amplified with increasing in amplitude but this rate of increase depends on the small signal, loop gain and BW of the crystal used, at steady state the closed loop gain =1.
- Amplitude of the signal is reduced either by nonlinearities of the active element or an automatic level control.
- Quartz crystal always want to oscillate with its fundamental frequency and must be forced to oscillate at a harmonic.

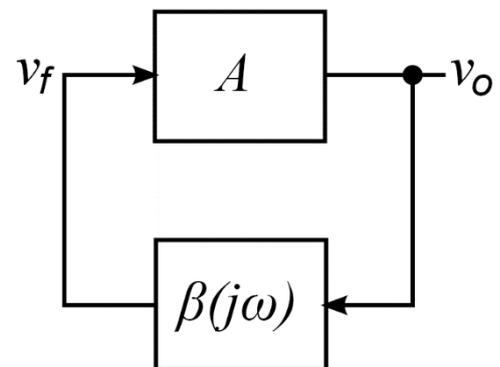


Figure 9.6: Feedback circuit

9.4.1 Negative resistance

For oscillation $Z_c + Z_m = 0$, so $\text{real}(Z_c) = -\text{real}(Z_m)$, therefore $\text{real}(Z_c)$ must be equals to $-R$

But we can't get a negative resistance from a passive circuit so an active circuit must be used to get this negative resistance

9.5 Single transistors devices

Excellent oscillators can be implemented using only single transistors with two impedances between its drain and source, gate and source respectively while between drain and gate there is the crystal as a feedback

As shown the impedance of the active circuit Z_c is given by

$$Z_c = \frac{Z_1 Z_3 + Z_2 Z_3 + g_m Z_1 Z_2 Z_3}{Z_1 + Z_2 + g_m Z_1 Z_2}$$

This relation is a bilinear function of g_m therefore the locus of $Z_c(g_m)$ is a circle where this circle always located in the lower half plane

9.5.1 Lossless linear circuit

Losses circuit is done by removing Z_1, Z_2, Z_3 and putting C_1, C_2, C_3 instead, lossless circuits are used to achieve minimum critical transconductance (thus minimum current needed for oscillation) and maximum frequency stability therefore Z_c will be :

$$\text{Real}(Z_c) = -\frac{g_m C_1 C_2}{(g_m C_3)^2 + \omega^2 (C_1 C_2 + C_2 C_3 + C_3 C_1)^2}$$

$$\text{Im}(Z_c) = -\frac{g_m^2 C_3 + \omega^2 (C_1 + C_2)(C_1 C_2 + C_2 C_3 + C_3 C_1)}{\omega [(g_m C_3)^2 + \omega^2 (C_1 C_2 + C_2 C_3 + C_3 C_1)^2]}$$

Maximum negative resistance is obtained for

$$g_m = g_{mopt} = \omega \left(C_1 + C_2 + \frac{C_1 C_2}{C_3} \right)$$

While the critical transconductance for oscillation is

$$g_{m \text{ crit}} = \frac{\omega}{QC} \frac{(C_1 C_2 + C_2 C_3 + C_3 C_1)^2}{C_1 C_2}$$

Also $C_3 = C_p + C_{gd}$ and $C_L = \frac{1}{\frac{1}{C_1} + \frac{1}{C_2}} + C_{par}$

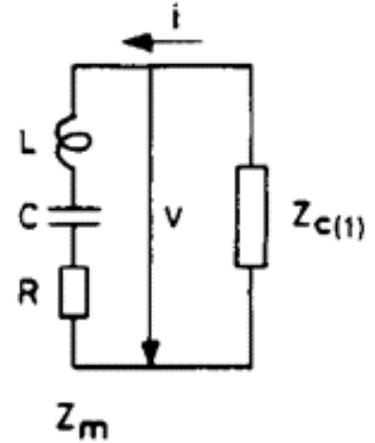


Figure 9.7: negative resistance

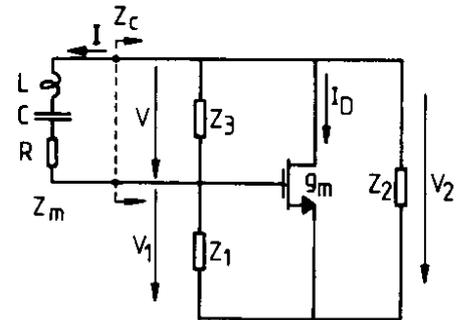


Figure 9.8: active transistor

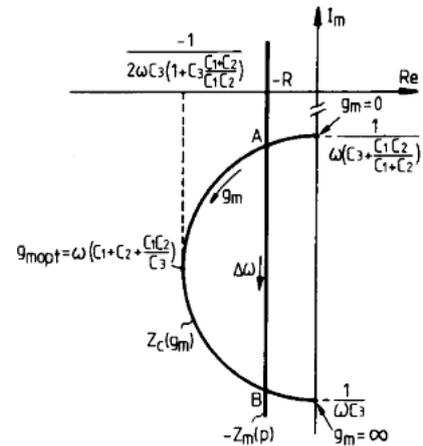


Figure 9.9: impedance in the complex plane

9.5.2 Derivations for critical and optimum transconductance

- Optimum transconductance is the g_m that gives you the maximum impedance

$$Real(Z_c) = -\frac{g_m C_1 C_2}{(g_m C_3)^2 + \omega^2 (C_1 C_2 + C_2 C_3 + C_3 C_1)^2}$$

$$\frac{dZ_c}{dg_m} = \frac{C_1 C_2 ((g_m C_3)^2 + \omega^2 (C_1 C_2 + C_2 C_3 + C_3 C_1)^2) - g_m C_1 C_2 (2g_m C_3^2)}{((g_m C_3)^2 + \omega^2 (C_1 C_2 + C_2 C_3 + C_3 C_1)^2)^2}$$

$$\frac{dZ_c}{dg_m} = 0, \quad g_m C_1 C_2 (2g_m C_3^2) = C_1 C_2 ((g_m C_3)^2 + \omega^2 (C_1 C_2 + C_2 C_3 + C_3 C_1)^2)$$

$$\therefore g_{mopt} = \frac{\omega (C_1 C_2 + C_2 C_3 + C_3 C_1)}{C_3} = \omega \left(C_1 + C_2 + \frac{C_1 C_2}{C_3} \right)$$

- Critical transconductance is which the real impedance of the active circuit equals to that of the crystal

$$Real(Z_c) = -\frac{g_m C_1 C_2}{(g_m C_3)^2 + \omega^2 (C_1 C_2 + C_2 C_3 + C_3 C_1)^2}$$

$$Real(Z_c) = -\frac{g_m C_1 C_2}{(g_m C_3)^2 + \omega^2 (C_1 C_2 + C_2 C_3 + C_3 C_1)^2}$$

$$-R = -\frac{g_m C_1 C_2}{(g_m C_3)^2 + \omega^2 (C_1 C_2 + C_2 C_3 + C_3 C_1)^2}$$

$$\frac{1}{R} = g_m \frac{C_3^2}{C_1 C_2} + \omega^2 \frac{(C_1 C_2 + C_2 C_3 + C_3 C_1)^2}{g_m C_1 C_2}$$

Neglecting the $g_m \frac{C_3^2}{C_1 C_2}$ term then:

$$\therefore g_{m crit} = \frac{\omega^2 (C_1 C_2 + C_2 C_3 + C_3 C_1)^2}{R C_1 C_2} = \frac{\omega}{QC} \frac{(C_1 C_2 + C_2 C_3 + C_3 C_1)^2}{C_1 C_2}$$

9.5.3 Active transistor topologies

For single transistor there are 3 topologies, these topologies are according to the position of the ground in the transistor, the transistor has 3 terminals so there are 3 topologies.

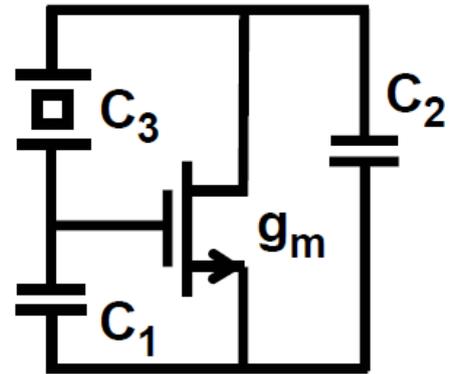


Figure 9. 10: single transistor

Grounded drain (Santos topology):

In this topology the drain of the active transistor is grounded, so there is an advantage in this topology that the crystal will connect to the active circuit by using only one pin and the other pin of the crystal will be connected to the ground, but there is a main disadvantage in this topology is that C_3 will increase by the shielded caps this will degrade the stability of the frequency.

Grounded gate (Colpitts topology):

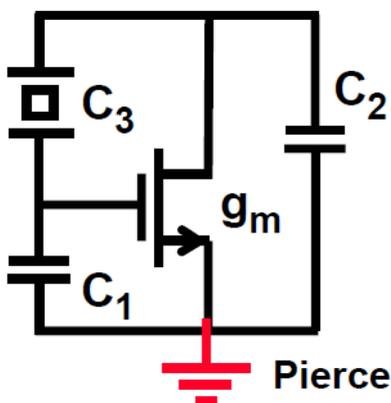
Same as Santos topology this topology has an advantage that it also needs one pin to connect with the transistor, the other pin is grounded too, but the main disadvantage is that the gate is grounded which means that we need equal bias current sources in the drain and the source which is not feasible

Grounded source (Pierce topology):

Its main advantages that it doesn't affect by the shielded caps or needs biasing that is not feasible so it has stable frequency but the disadvantage that it needs 2 pins to connect with the transistor.

Conclusion:

I will use Pierce topology as an active circuit as it is the most stable in frequency and has no biasing problems like Colpitts and no capacitance problems like Santos.



9.6 Amplitude control topologies

The roles of any amplitude control mechanism is:

1. Provide large gain (current) to the transistor at power up to guarantee oscillator startup.
2. Decrease the gain while amplitude is growing to reach with the amplitude in steady state.

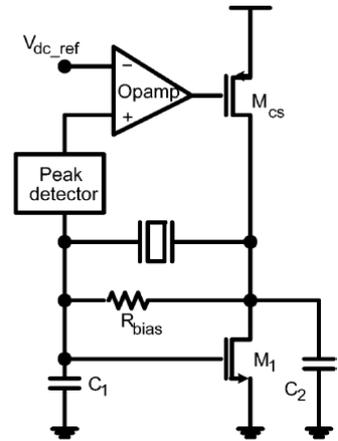


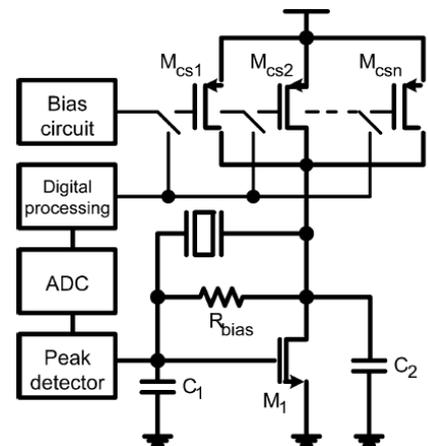
Figure 9. 11:Pierce,Colpitts and Santos topologies

9.6.1 Analog amplitude control crystal oscillator

It is a conventional control loop that sense the envelop of the growing

output signal then an error amplifier which compares this envelop to a reference, at start up envelop level is smaller than the reference level the current source transistor push maximum current in the active transistor then when the amplitude increase the amplifier output voltage increases so it decreases the current in the active transistor.

Figure 9. 12: analog amplitude control



But the main disadvantages for this method:

1. The amplifier output voltage is not a pure dc voltage, it has ripples so it will affect the phase noise.
2. This control loop is always on and hence consumes power.

9.6.2 Digital amplitude controlled crystal oscillator

Due to all the disadvantages of the analog control loops, digital control is devised, instead of the error amplifier ADC and digital processing are used, according to the envelop of the signal , the digital processing control the number of the current sources operating

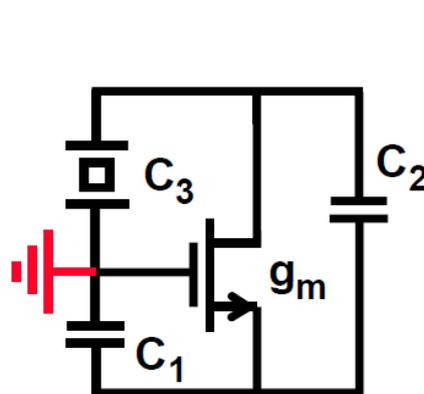
Figure 9. 13: Digital amplitude control

Its advantages are:

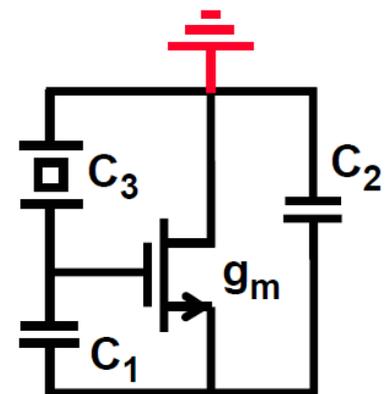
1. No DC ripples as in the error amplifier
2. Phase noise is not affected
3. Easier in stability
4. Turn off the circuit when not used so saves power

But the only disadvantage of it is the complexity of the design and the large area

- I will use the analog method for amplitude control as the digital method is too complex



Colpitts : 1-pin X=D



Santos : 1-pin X=G

and need a very large area.

9.7 Design methodology

9.7.1 Resonator

From data sheet:

$R_s=80\Omega$, $C_s=3.5\text{fF}$, $C_p=1\text{pF}$

$$\therefore L_s = \frac{1}{\omega^2 C_s} = \frac{1}{(2\pi \times 26 \times 10^6)^2 \times 3.5 \times 10^{-15}} = 10.706\text{mH}$$

$$F_s = \frac{1}{2\pi\sqrt{L_s C_s}} = 26\text{Mhz}$$

$$F_a = \frac{F_s}{\left(1 + \frac{C_s}{2C_p}\right)} = 26.05\text{Mhz}$$

$$Q = \frac{1}{2\pi F_s C_s R} = 21.86K$$

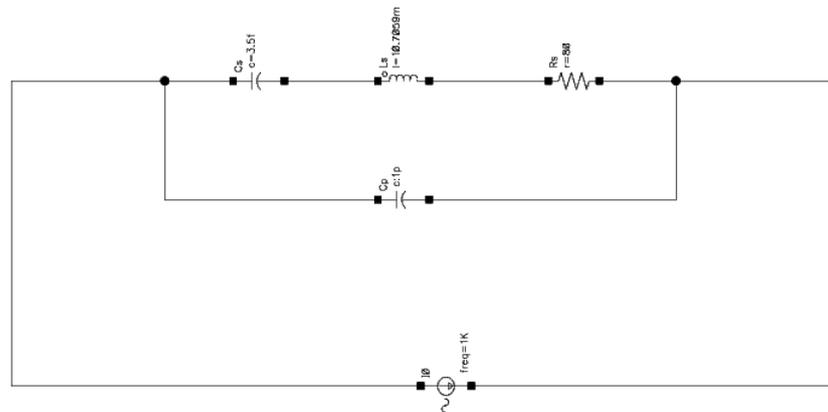


Figure 9. 14: crystal model

As shown in the figure a current source of unity amplitude is used to measure the total impedance of the crystal model which will be shown in the following figures

9.7.1.1 Simulation results

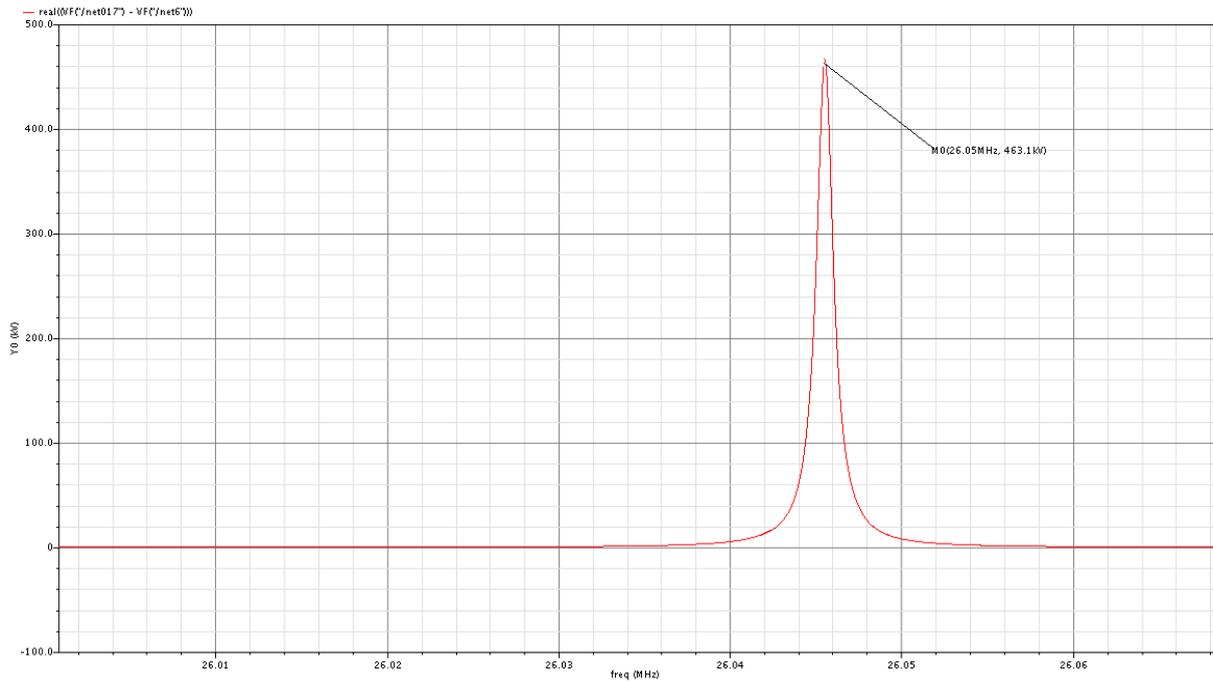
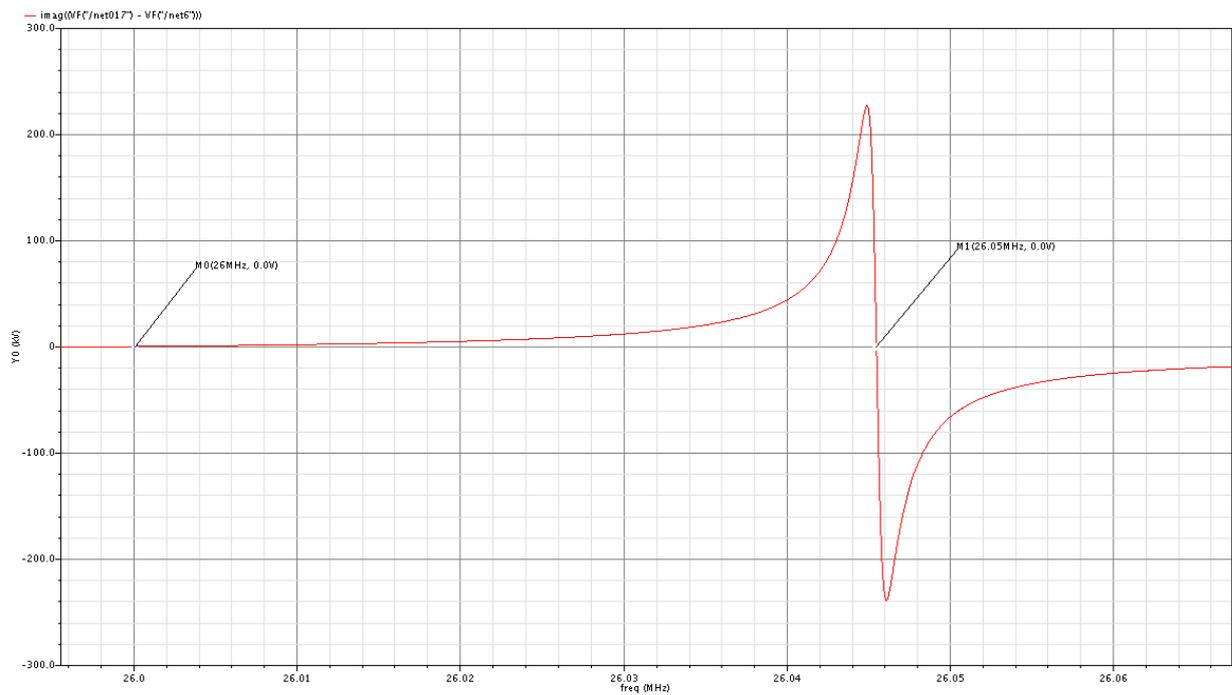


Figure 9. 15: model impedance-real part



As shown maximum real impedance at 26.05 MHz “Fa”

As shown in the figure that zero impedance is at 26 MHz “Fs”

Where infinity impedance is at 26.05 MHz “Fa”

9.8 Active circuit

9.8.1 Design

For the active circuit design we need to bias the active transistor with a current source and use a method to kick off the oscillations like:

1. Using initial condition in the inductor or the capacitor.
2. Using voltage pulses in series with the model or current pulses in parallel.
3. Using a step function instead of the usual DC voltage in the supply

Computing C_1 & C_2 :

$$C_L = \frac{1}{\frac{1}{C_1} + \frac{1}{C_2}} + C_{par}$$

Let $C_1=C_2$ and assume that $C_{par}=2.5\text{pF}$ & $C_L=20\text{pF}$ “from data sheet” Then $C_2=C_1=35\text{pF}$

$$g_{m\text{ crit}} = \frac{\omega}{QC} \frac{(C_1C_2 + C_2C_3 + C_3C_1)^2}{C_1C_2} = 3\text{mS}$$

$$g_m = g_{mopt} = \omega \left(C_1 + C_2 + \frac{C_1C_2}{C_3} \right) = 178\text{mS}$$

Figure 9. 95.: model impedance-imaginary part

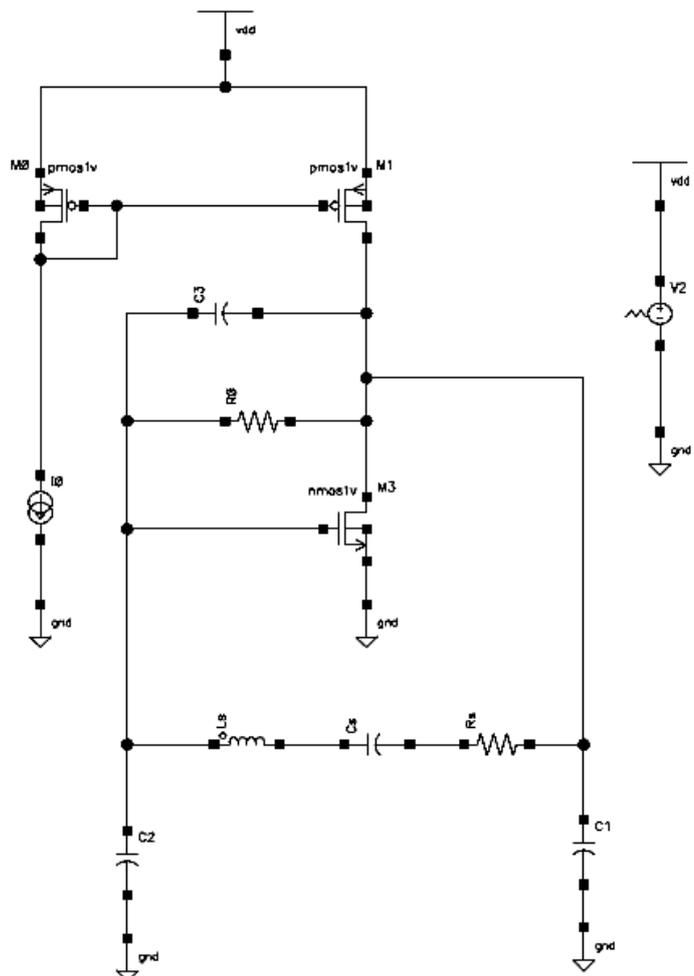


Figure 9.17: Schematic of the active circuit

9.8.1.1 Simulation results

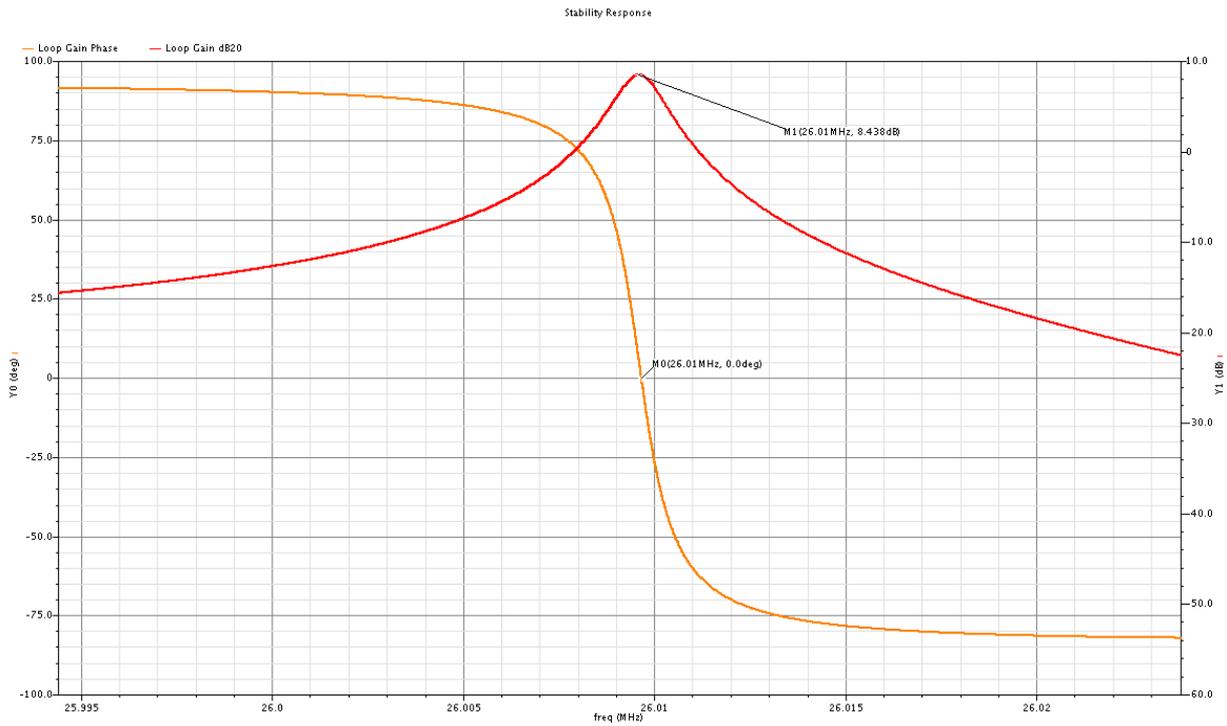


Figure 9.18: stability analysis

As shown in the figure Barkhausen criteria is verified, that at oscillation frequency gain is more than 0dB and phase is zero degrees

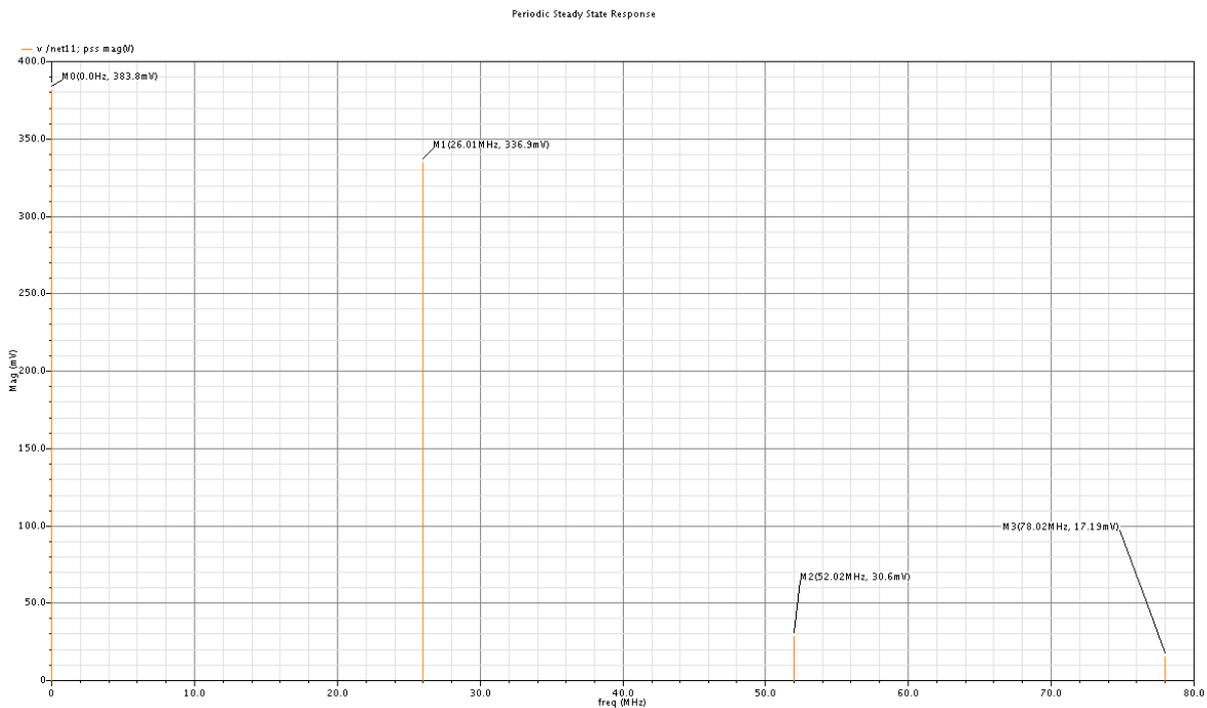


Figure 9.19: PSS analysis

As shown in the figure above the fundamental of this crystal is 26.01 MHz with 336mV amplitude

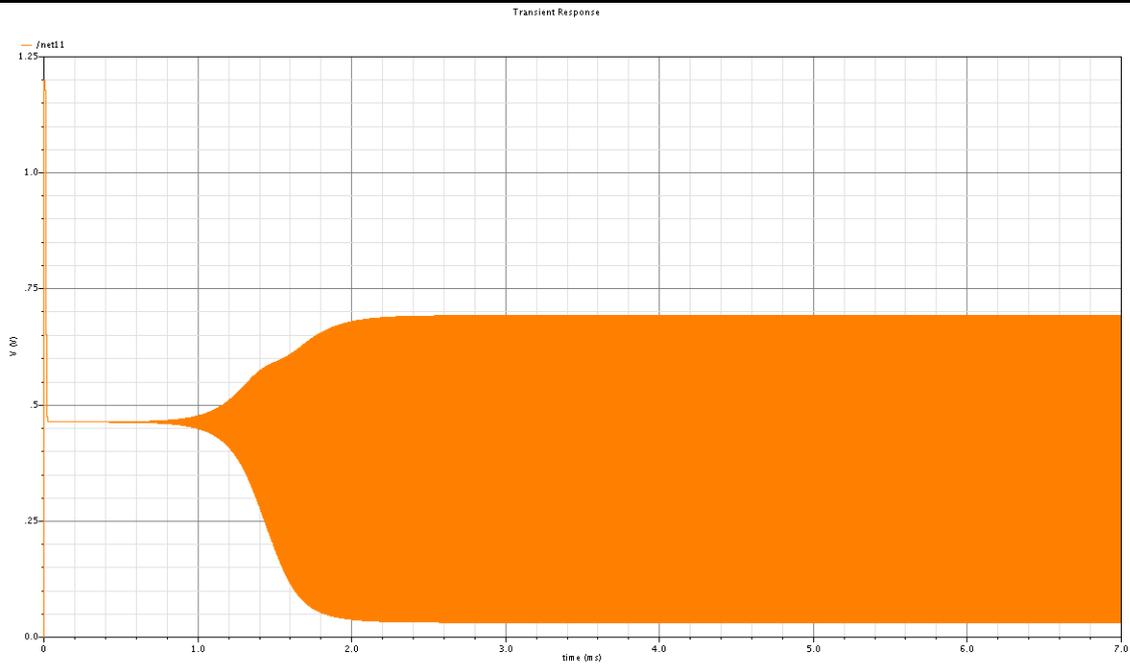


Figure 9.20:crystal output

As shown in the figure above the startup time = 2ms

As shown in the figure below sine wave from 30.72mV to 691.3mV which means 660.58mV peak to peak voltage.

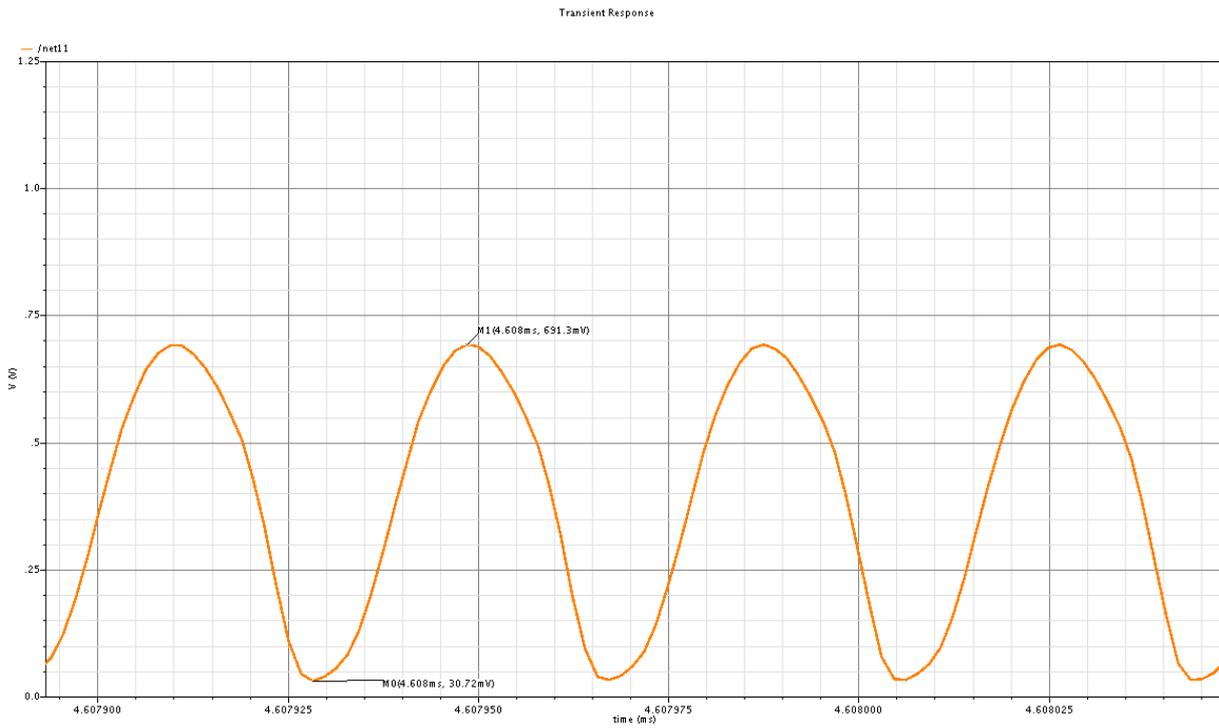


Figure 9.21: zoomed crystal output

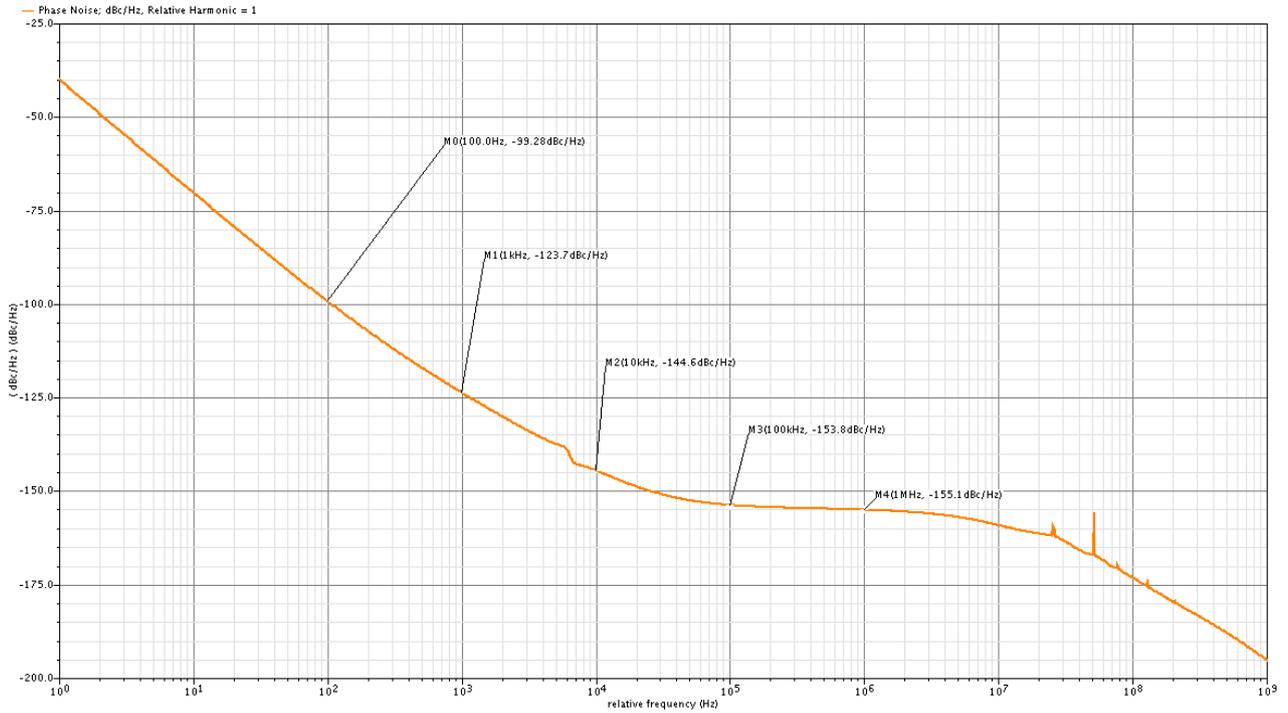


Figure 9.22: phase noise of the oscillator

Table 11: Active circuit Phase Noise

frequency	100Hz	1Khz	10Khz	100Khz	1Mhz
Phase noise	-99.28 dBc/Hz	-123.7 dBc/Hz	-144.6 dBc/Hz	-153.8 dBc/Hz	-155.1 dBc/Hz

9.9 Active circuit with current control loop

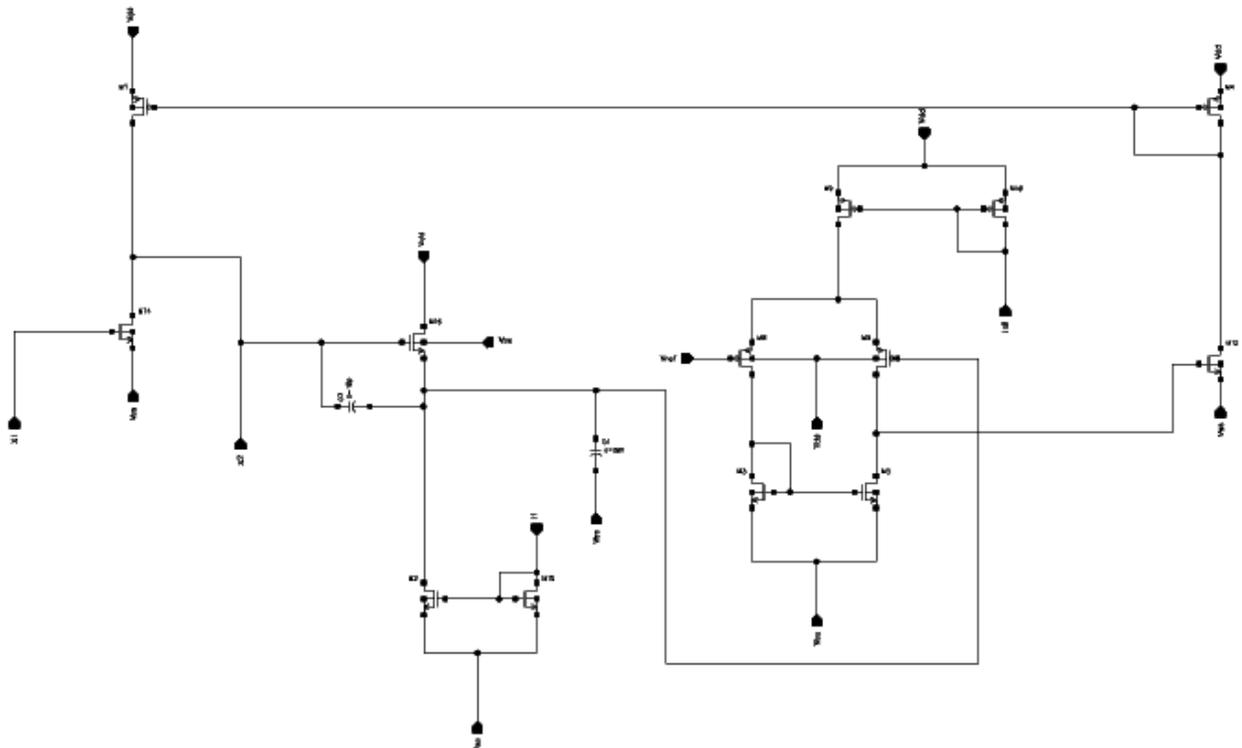
Properties of needed current control loop:

1. To start the oscillation with high current and small startup time
2. Then decrease this current to save power

So the requirements for a current control loop are (envelop detector, error amplifier, biasing)

- First of all the output of the OTA is high which pass a high current in the NMOS and this high current then mirrored in the active transistor to start the oscillation with a small startup time
- As shown in the figure below a common drain NMOS is acting as envelop detector which sense the raising sine wave from the oscillator.
- Then an error amplifier (single ended simple OTA) is used to compare between the output of the envelop detector and a reference voltage.
- When the output of the envelop detector increases above the reference voltage the output of the OTA decreases.
- Then the bias of the NMOS decreases which decrease the current passes in it.
- This decreased current then mirrored to the active transistor.

Advantages of this method that it is simple and decreased the current passes in the active transistor



so it decreases the consumed power but the main drawback of this loop is its stability

9.9.1 Simulation results

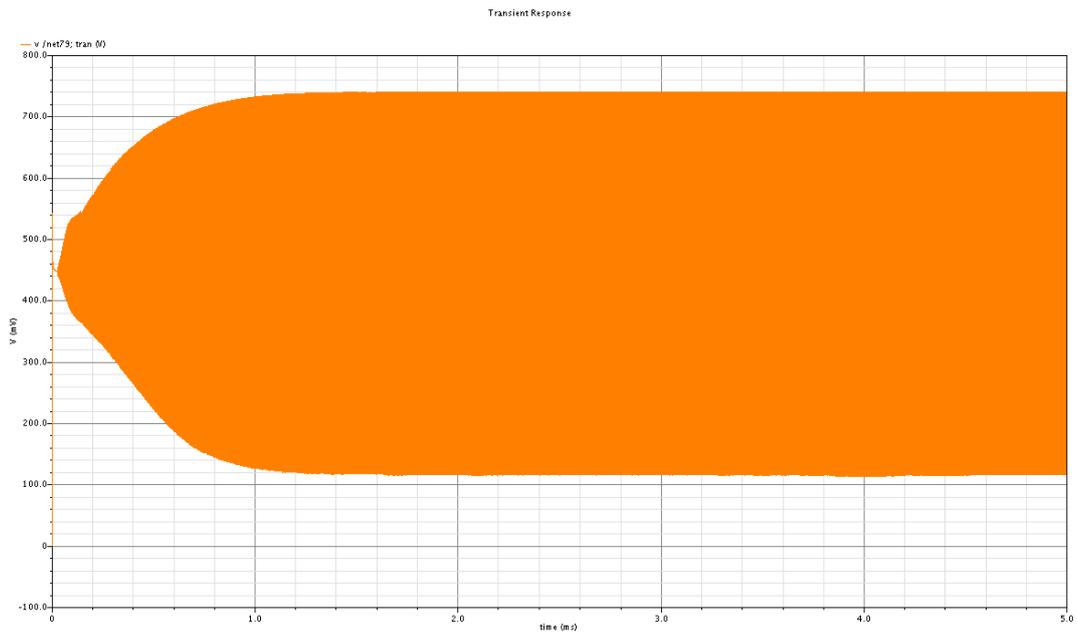


Figure 9.24: oscillation after putting control loop

As shown in the figure above that the startup time = 800us

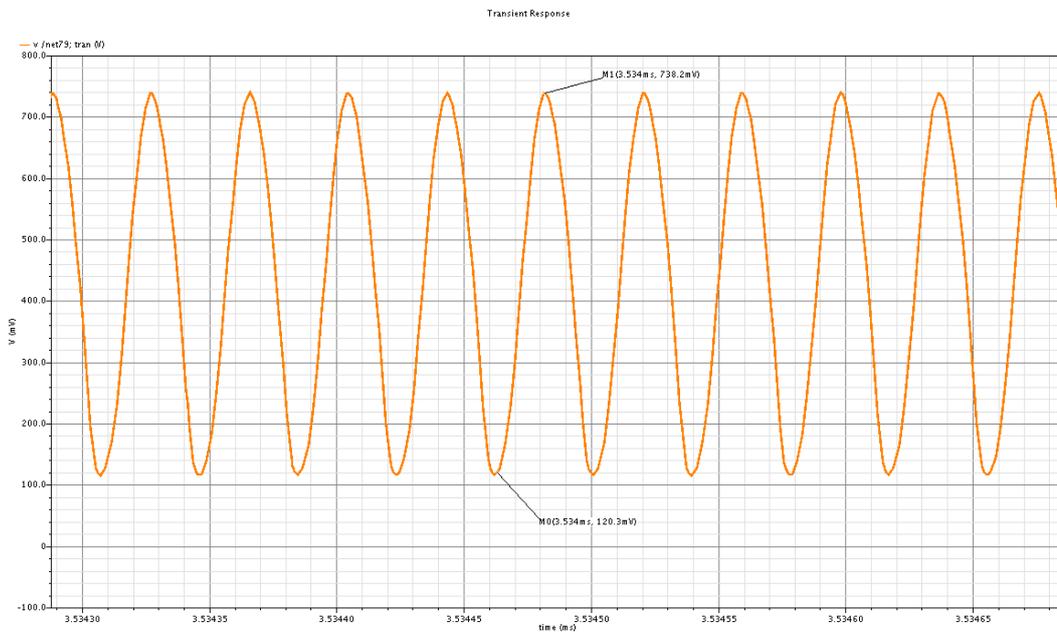


Figure 9.25: zoomed oscillations after control loop

This circuit consumes current 500 uA

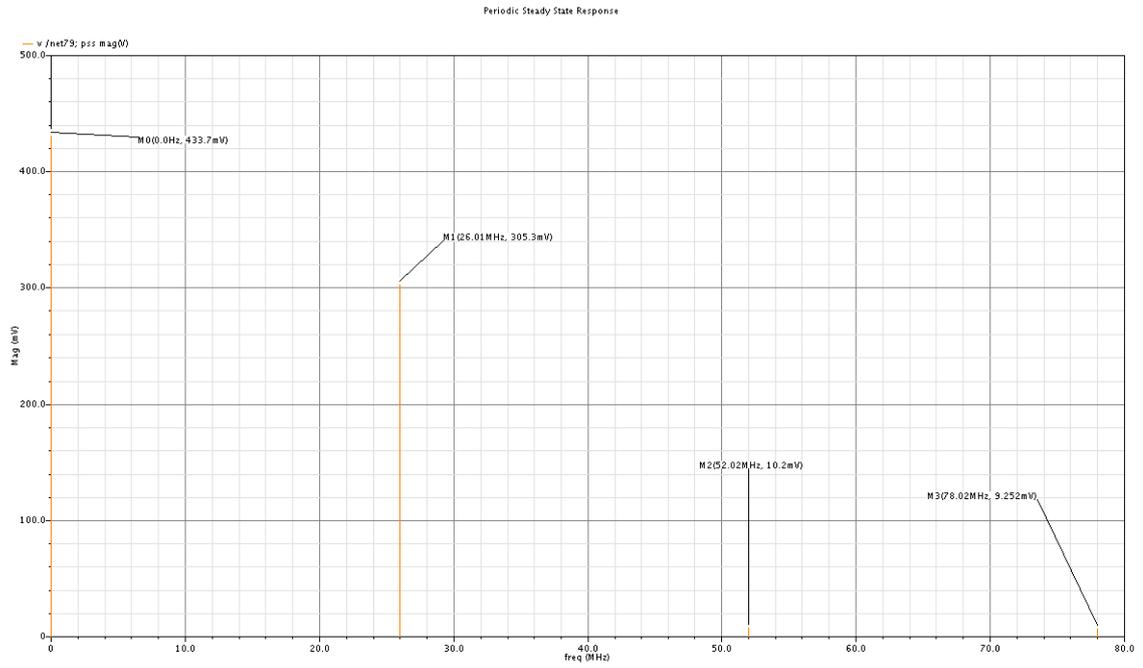


Figure 9.26: PSS after the control loop

Fundamental frequency = 26.01 MHz

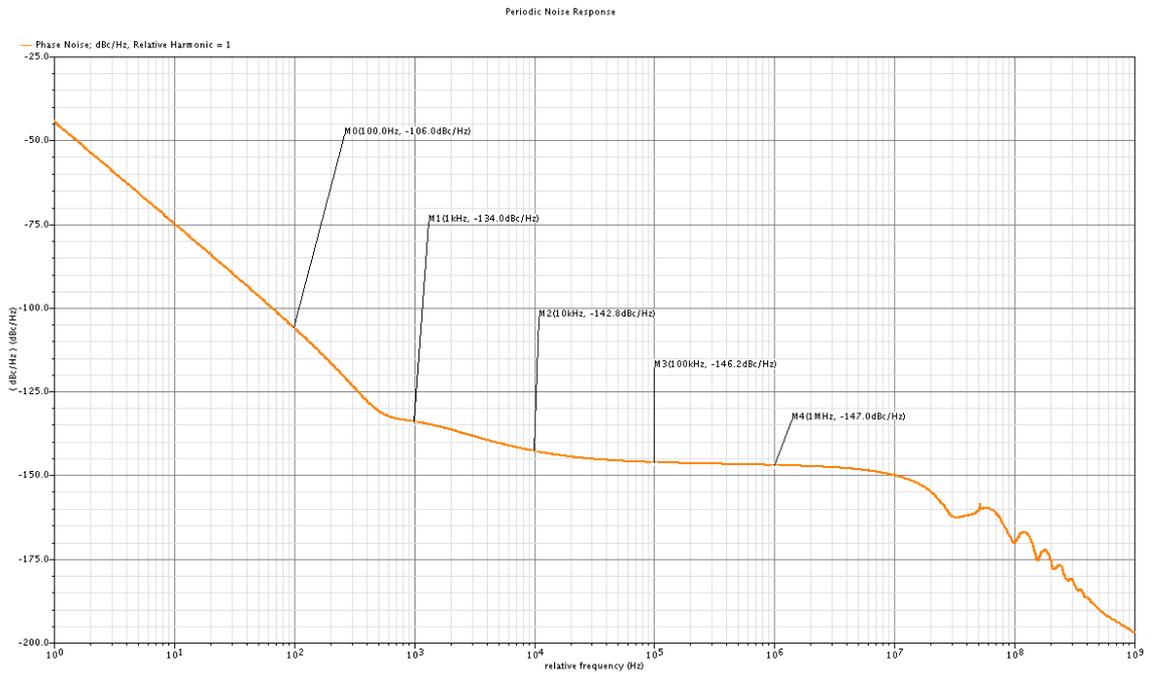


Figure 9.27: phase noise after control loop

Table 12: Phase Noise after Current Control Loop

frequency	100Hz	1Khz	10Khz	100Khz	1Mhz
Phase noise	-106 dBc/Hz	-134 dBc/Hz	-142.8 dBc/Hz	-146.2 dBc/Hz	-147 dBc/Hz

Comparison between the crystal oscillator before and after putting the current control loop:

Table 13: Comparison between with and without Current loop

	Without control loop	With control loop
Startup time	2msec	800μsec
Amplitude	336.9mV	305mV
Phase Noise	-99.28dBc/Hz at 100Hz -123.7dBc/Hz at 1KHz -144.6dBc/Hz at 10KHz -153.8dBc/Hz at 100KHz -155.1dBc/Hz at 1MHz	-106dBc/Hz at 100Hz -134dBc/Hz at 1KHz -142.8dBc/Hz at 10KHz -146.2dBc/Hz at 100KHz -147dBc/Hz at 1MHz

As shown in the table above that the current control loop decreases the startup time but in the other hand it decreases the amplitude and increases the Phase Noise but they are still meet the specs.

9.10 Buffer

- Crystal oscillator's output is sine wave but the PFD need a square wave.
- So a buffer is required to convert the sine wave to a square wave.
- The easiest way to convert from sine wave to square wave is inverters.
- The design of this buffer is a 2 stage inverters with common mode of 600mV.
- The input of the buffer don't necessarily have the 600mV as a common mode.
- So a filter is required to remove the DC of the input signal then adding the 600mV common mode.

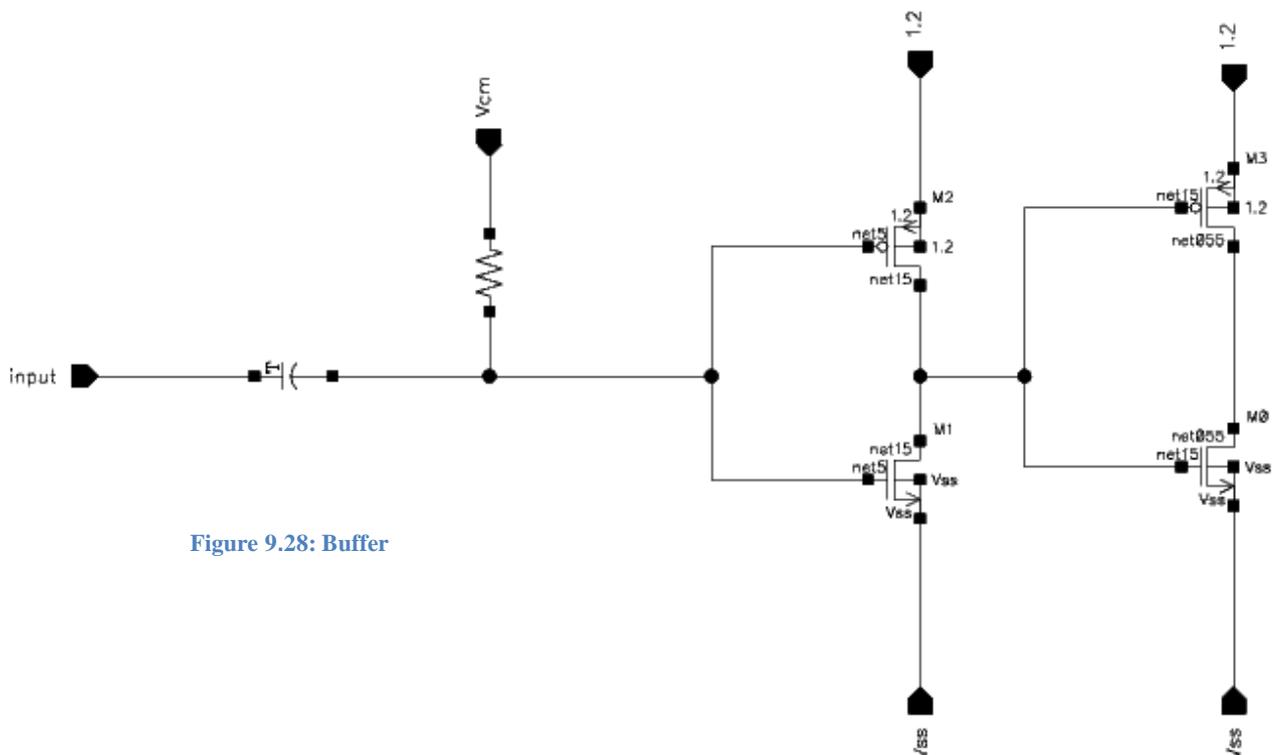


Figure 9.28: Buffer

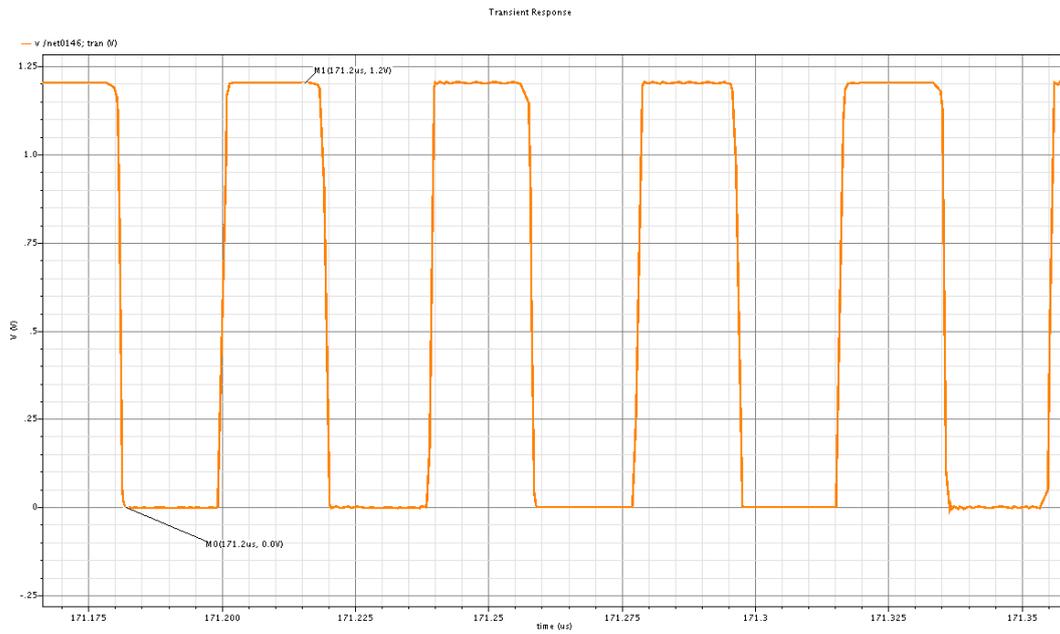


Figure 9.29: Buffer output

- The output of this buffer then is the input of the PFD.
- The startup time after putting the buffer is 143μsec.

9.11 Total circuit

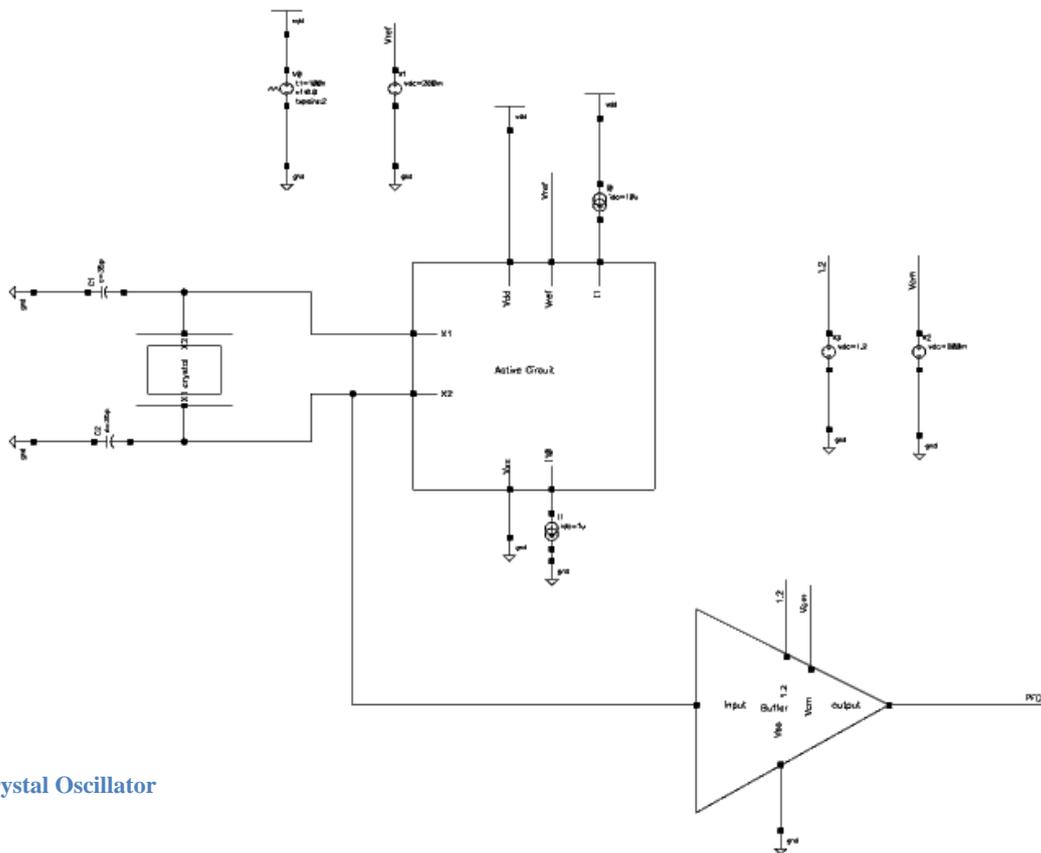


Figure 9.30: Crystal Oscillator

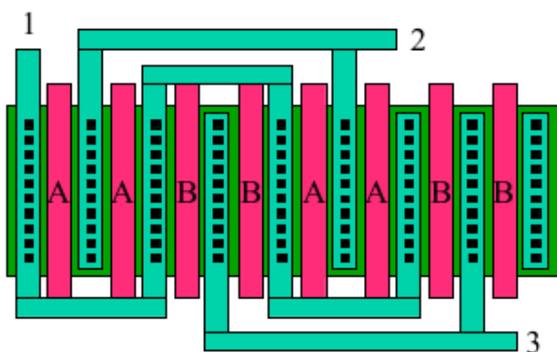
9.12 Summary

Table 14: Summary of results

	Required	Min	Typical	max	Unit
Freq.	26		26		MHz
Startup time	150	111.4 (SS,1.3V,0°C)	143	423.9 (FF,1.3V,85°C)	μsec
Current consumed	400	397.38 (FF,1.3V,85°C)	500	559.2 (SS,1.3V,0°C)	μA
Load Capacitor	13	10	13	20	pF
PN at 100Hz	-103	-106.9 (SS,1.1V,0°C)	-106	-101.5 (FF,1.3V,85°C)	dBc/Hz
PN at 1KHz	-132	-134.3 (FF,1.1V,0°C)	-134	-132.1 (SS,1.1V,85°C)	dBc/Hz
PN at 10KHz	-142	-142.8 (FF,1.1V,0°C)	-142.8	-139.8 (SS,1.1V,85°C)	dBc/Hz
PN at 100KHz	-148	-148.3 (SS,1.1V,0°C)	-146.2	-143.6 (FF,1.3V,85°C)	dBc/Hz
PN at 1MHz	-149.7	-149.3 (SS,1.1V,0°C)	-147	-144.5 (FF,1.3V,85°C)	dBc/Hz
temp	0 to 85	0	27	85	0°C

9.13 Layout

To make layout for integrated devices I used 2 matched transistors with one node in common as shown in figure below



So transistors are in an equal part of fingers so as in OTA there are 2 identical NMOS and 2 identical PMOS

Figure 9.31: 2 Integrated Devices Layout

9.13.1 OTA Layout

As shown in the figure the upper transistors are the 2 PMOS each one has 4 gates 2 drains and 2 sources

Also in the lower transistors are the 2 NMOS but here each of the has 2 gates and a drain and a source

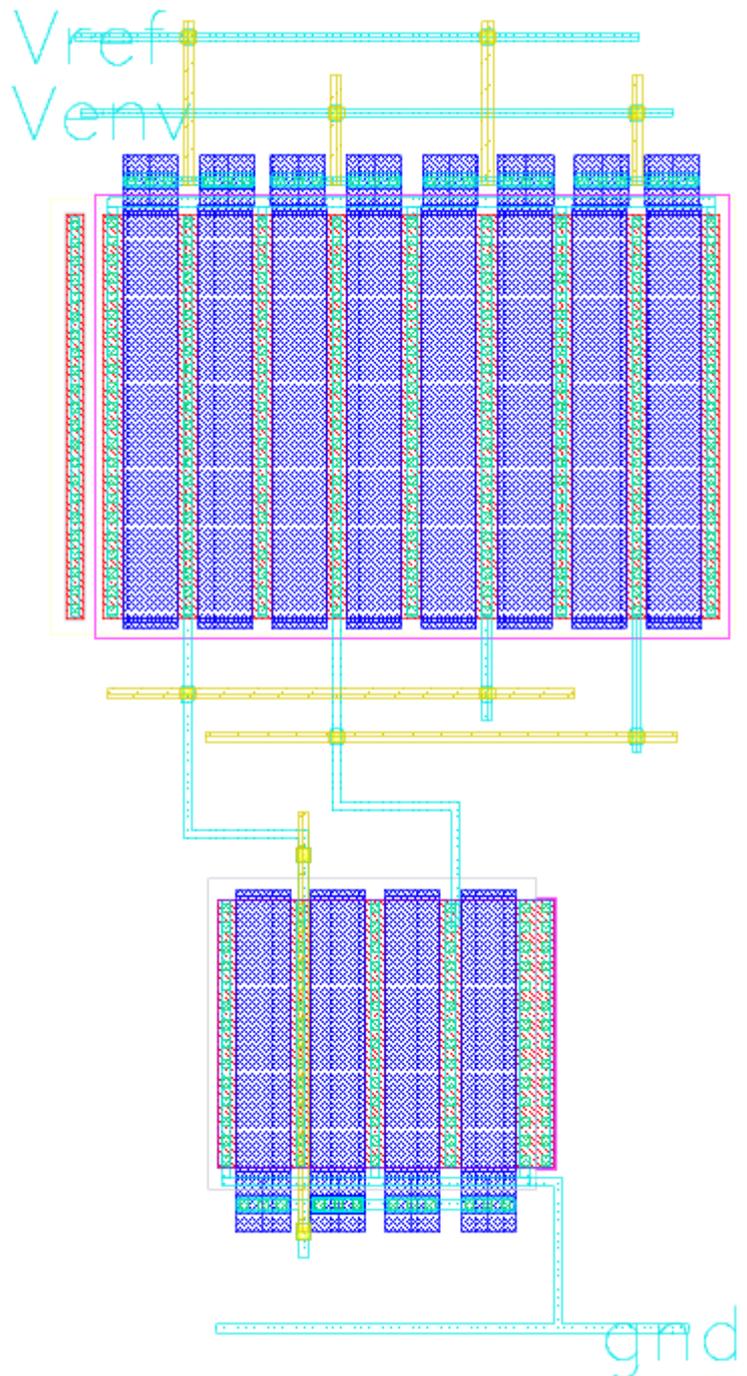


Figure 9.32: OTA Layout

9.13.2 Buffer Layout

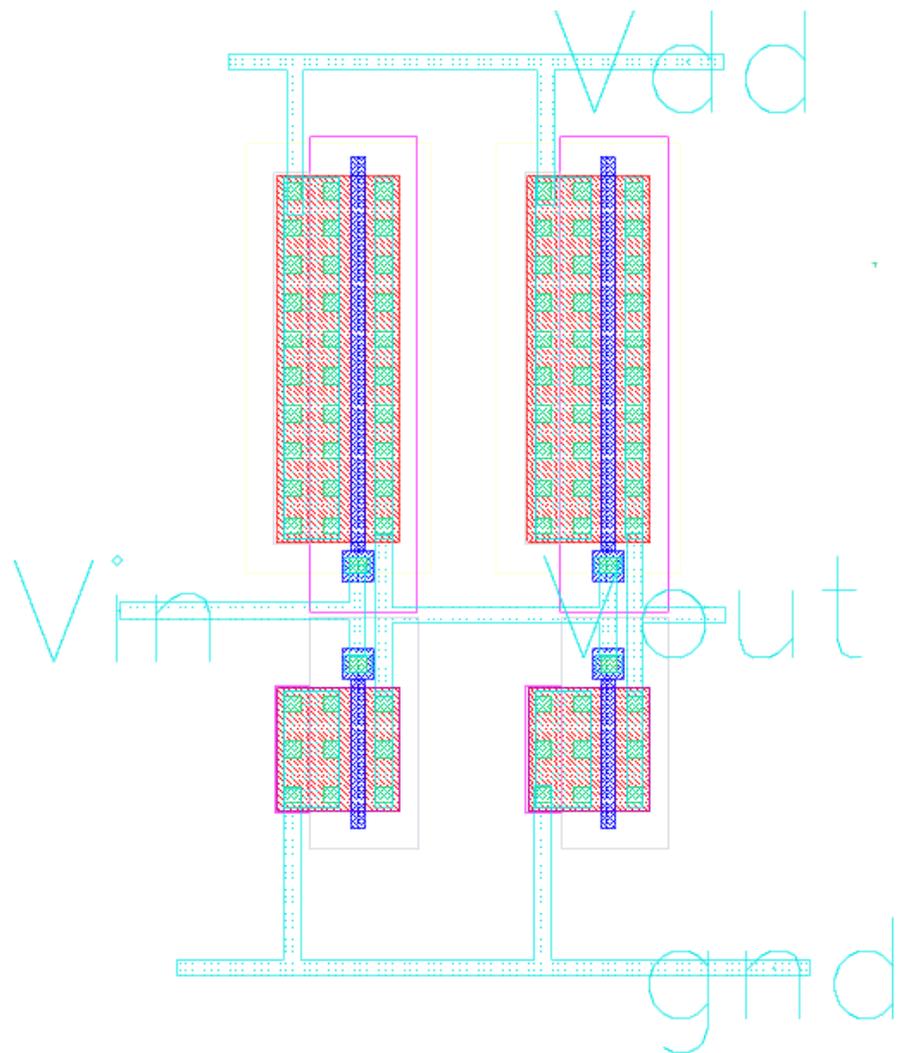


Figure 9.33: Buffer Layout

9.14 Conclusion

- Crystal oscillator is one of the fundamental blocks in the PLL as it gives an accurate clock cycle used to be compared with the divider output to ensure locking, so crystal oscillator output must be accurate.
- Startup time is inversely proportional with current in the core.
- Current control loop enhance current consumption but it degrades the phase noise.

9.15 References

- [1] E. A. Vittoz, M. G. R. Degrauwe, and S. Bitz, "High performance Crystal oscillator circuits: Theory and application," *IEEE J. Solid-State Circuits*, vol. 23, no. 3, pp. 774–783, 1988.
- [2] Low-Power Crystal and MEMs Oscillators by Vittoz
- [3] Razavi, Behzad, "RF Microelectronics", McGraw-Hill, New York, 1998
- [4] "An All-Analog Method to Enhance Amplitude Stability in Pierce Crystal Oscillators" by Karim M. Hussein and Emad Hegazi, Senior Member, IEEE
- [5] "A Study of Phase Noise in CMOS Oscillators" Behzad Razavi, Member, IEEE
- [6] "CX14SM AT CRYSTAL 12 MHz to 50 MHz Ultra-Miniature, Ultra-Low Profile Surface Mount AT Quartz Crystal" Data sheet
- [7] "Low-Power Sub-1 GHz RF Transceiver" Data sheet
- [8] "Design of crystal oscillators" Willy sansen lecture
- [9] Layout of Analog CMOS Integrated Circuit lecture by F. Maloberti
- [10] A Low Power Oscillator with Feedback Amplitude regulation by Rahul Rithe & Saurav Bandyopadhyay

10 PFD and Charge pump

10.1 Phase frequency detector

Phase frequency detector block compare phase and frequency of crystal reference signal and feedback signal from Divider, And give output proportional to difference of phase and frequency between the two signals.

10.1.1 Phase Detectors

Phase detector block detect phase difference between two input signals and its output proportional to the phase difference .The output of phase detector is $v_{out} = k_{PD} \varphi_e$, where k_{PD} is constant of proportionality. This linearized equation is valid only over limited range. There are many types of phase detectors we will mention some of them in the following section.

10.1.1.1 XOR Gate phase detector

XOR gate is the simplest kind of phase detectors which it can detect the difference between 2 input bits

But it's average voltage equals $v_{out} = \frac{v_0}{\pi} \varphi_e$ and its output waveform

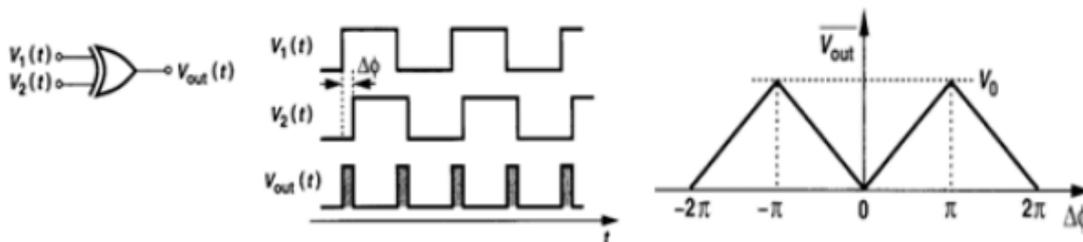


Figure 10.10.1 : XOR waveform

10.1.1.2 SR latch

Using SR latch during having a differentiator at the input of it and perform the phase comparison at the edge And the average output of SR phase detector

$$v_{out} = \frac{v_0}{2\pi} \varphi_e$$

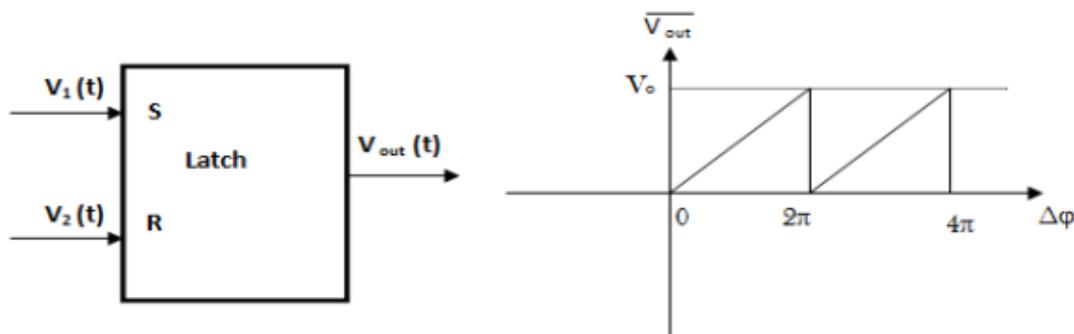


Figure 10.10.2 : SR latch phase detector

10.1.1.3 Mixer phase detector

Mixer used as phase detector if inputs are sinusoidal signals. If input signal assumed to be $A_1 \cos(w_1 t)$ & $A_2 \cos(w_2 t + \varphi_e)$, Mixer's output $v_{out} = \alpha A_1 \cos(w_1 t) A_2 \cos(w_2 t + \varphi_e)$, where α is conversion gain of Mixer and φ_e is phase error between to signals. So be using LPF after Mixer the output have frequency component at $((w_2 - w_1))$, If $w_1 = w_2$ output proportional with phase error between the two signals.

$$v_{out} = \frac{\alpha A_1 A_2}{2} \cos \varphi_e$$

10.1.2 Phase frequency detector

Replacing the phase detector with the phase frequency detector as it has many advantages than phase detectors

	Phase frequency detector	Phase detector
Detection	Detects phase & frequency	Detects phase only
False locking	No false locking	Suffer from false locking
Output voltage	Proportional to charge pump current	Proportional to phase difference

10.1.2.1 Tri state phase detector

It consists of 2 D flip flops with Asynchronous reset with an AND gate

This way of phase detection depends on creating 3 states the first state at the UP=1

Only and the second state at the DOWN=1 only while the third state at both the UP & DOWN signals=1

And at the third state the and gate generates the reset signal.

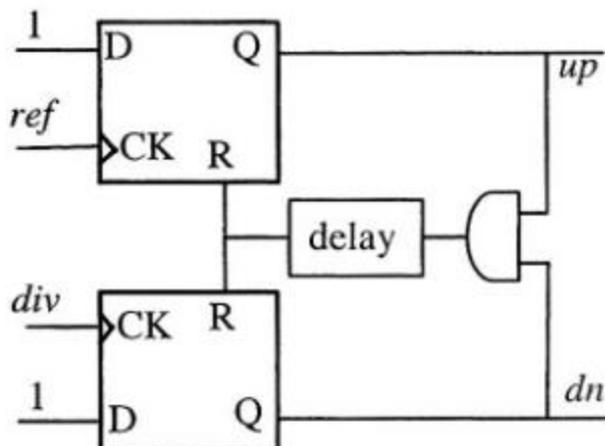


Figure 10.10.3 : Tri state PFD

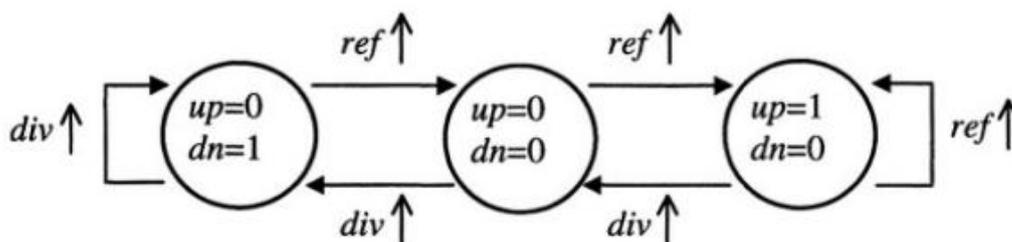


Figure 10.10.4 : Tri state PFD finite state machine

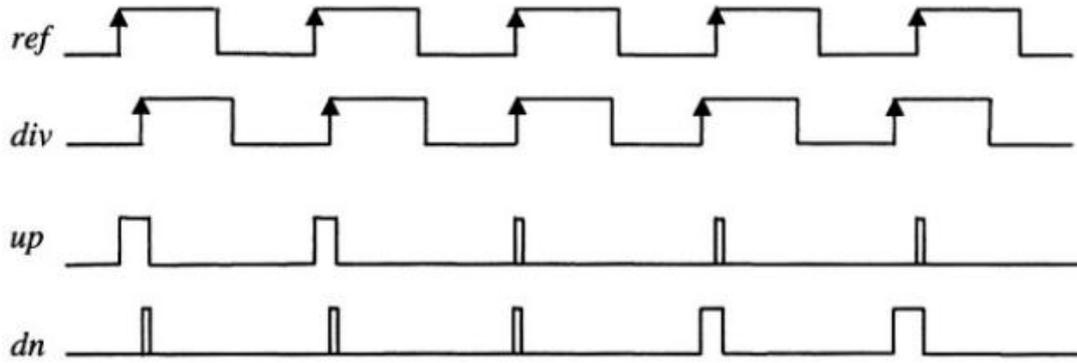


Figure 10.10.5 : Tri state PFD signals

This PFD consists of 2 latches triggered on the edge of the input signal so the input of the first latch is the input signal for the loop from the reference oscillator while the other latch input is the feedback signal

10.1.2.1.1 Flip flop implementation

The D flip flop is consists of 2 latches in a master slave topology so here we use the D flip flop as a Set reset flip flops with set =1 to use it as a D flip flop with D=1 so these latches are consisted of NOR gates

As in the following figure

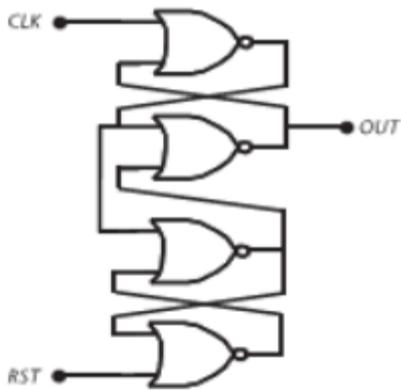


Figure 10.10.6 : D flip flop implementation

10.2 Design methodology

The basic design depends on designing the NOR gates which is the basic part of the flip flop which is chosen to be of minimum sized CMOS NOR gates to save power and area and also due to the low frequency of the reference input so no need to use more complex designs

Then the design of the NAND gate and the NOT gate is with the same way minimum sized

And after that we can choose to change the size of some cells according to the delay needed to eliminate the dead zone which was good in our case and there were no need for adding more delay

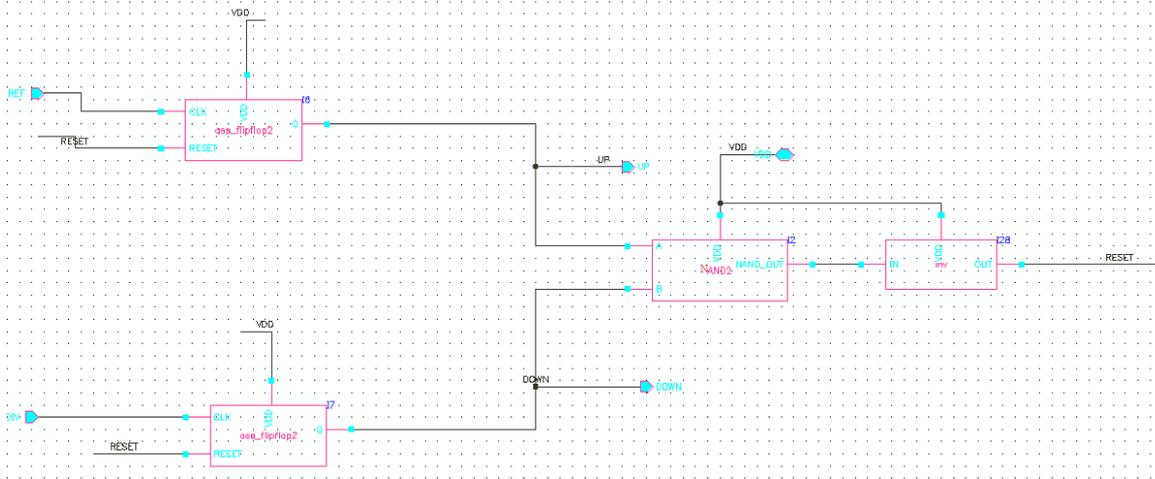


Figure 10.10.7 : tri- state PFD

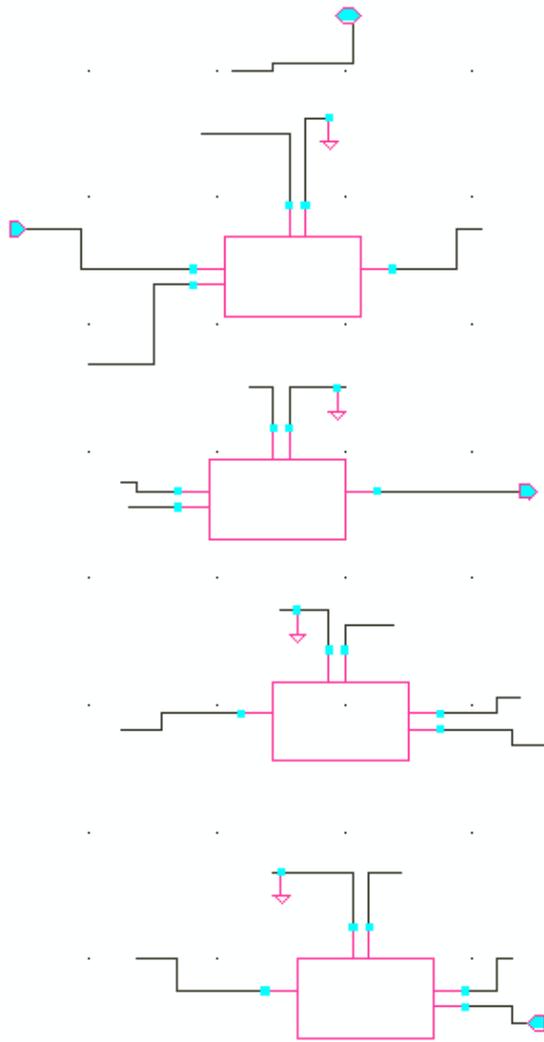


Figure 10.10.8 : D flip flop (connected with labels from inside)

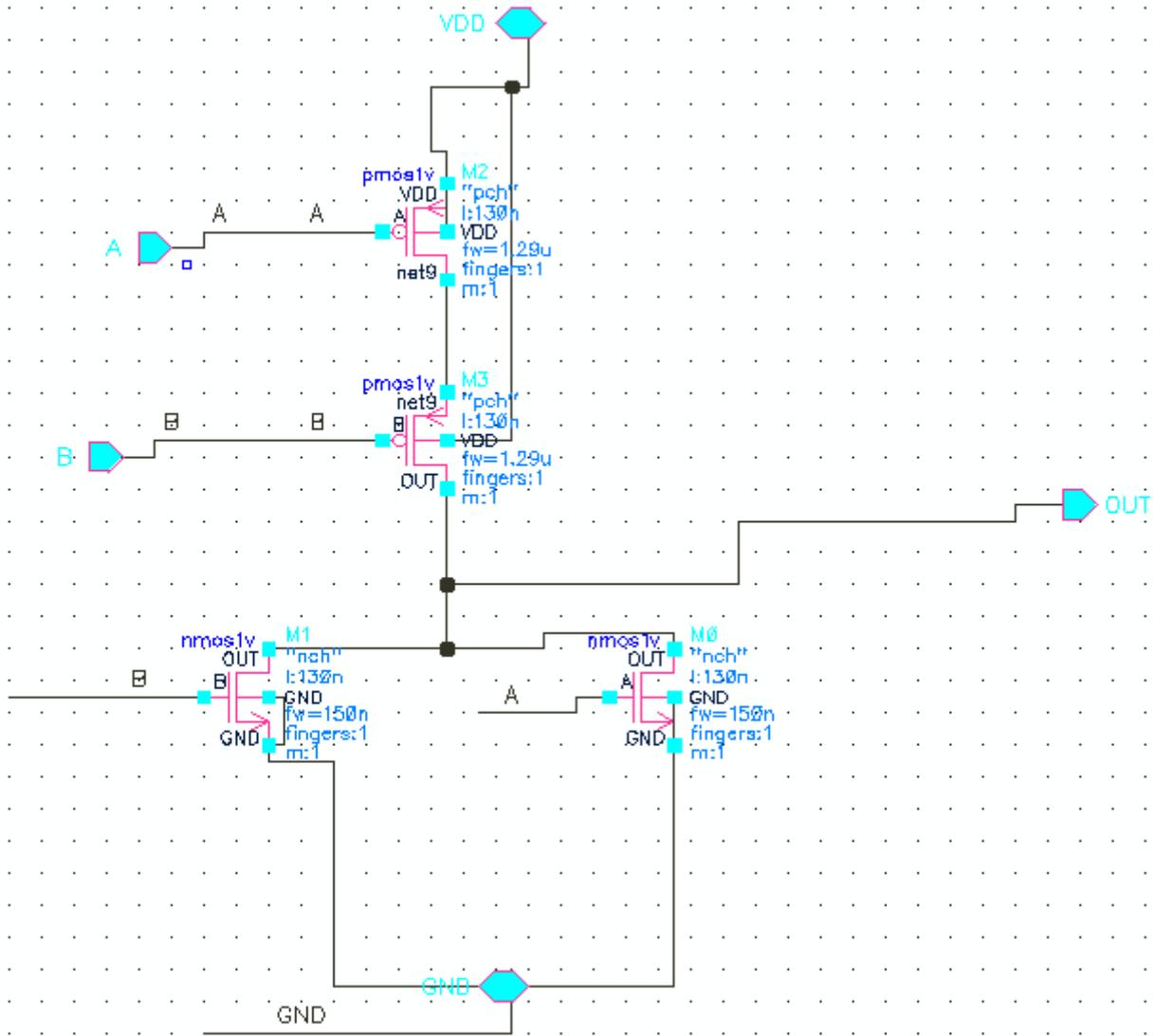


Figure 10.10.9 : CMOS NOR gate

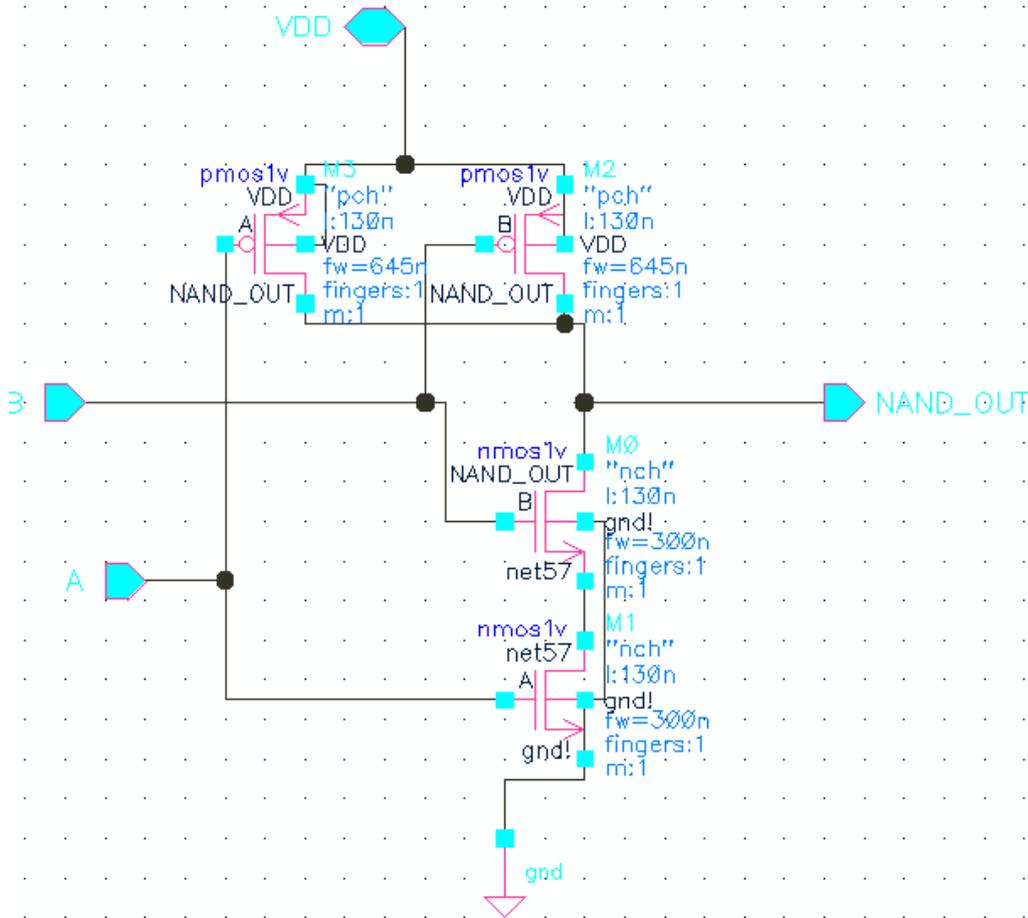


Figure 10.10.10 : CMOS NAND gate

10.3 PFD simulation results

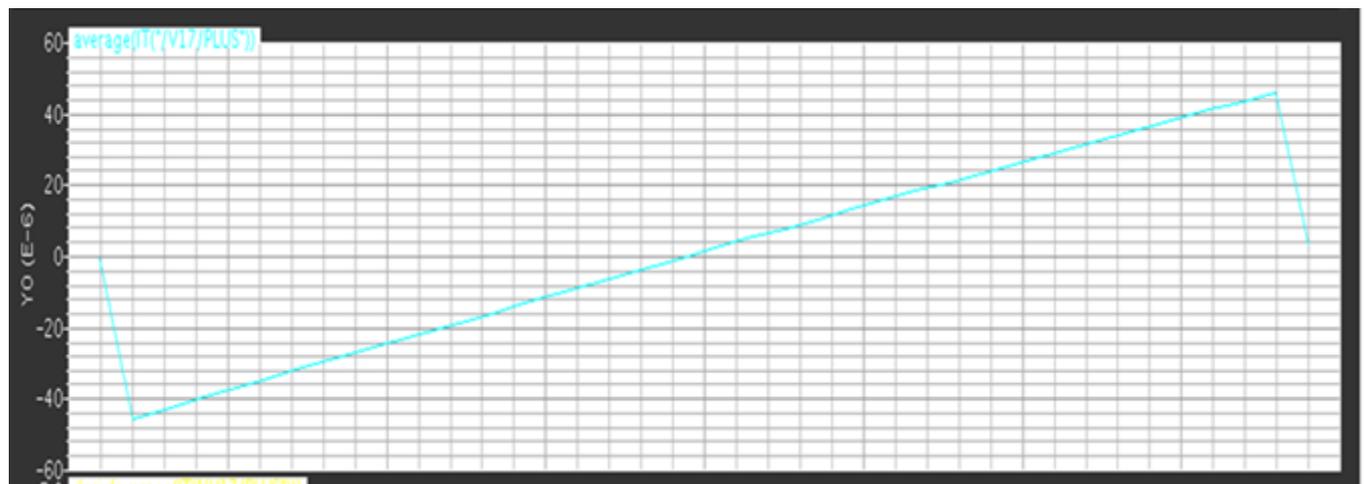
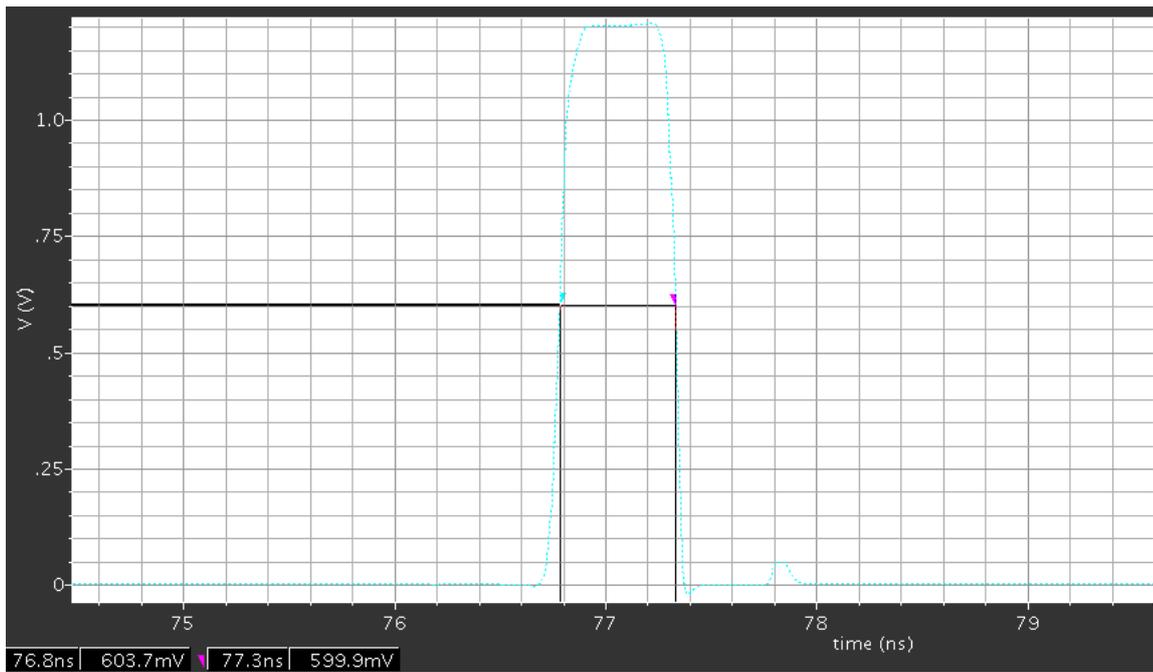


Figure 10.10.11 : PFD transfer function



10.4 Charge pump

10.4.1 Introduction

Charge pump is one of the important blocks in PLL which has great effect on loop dynamics. It consists of two current sources controlled by output signals of PFD. It converts digital signals of PFD into analog current to charge or discharge loop filter to change control voltage on VCO.

10.4.2 Charge pump architectures

Charge pump can be classified as 2 types single ended and differential ended outputs which reflects the need to a single loop filter or 2 loop filters

	Single ended charge pump	Differential charge pump
Advantages	<ul style="list-style-type: none"> -Need single loop filter -lower power consumption -require fewer components hence they occupy less area in a chip -Its problems are not those much difficult to handle 	<ul style="list-style-type: none"> -It doubles the range of output voltage compliance compared to single ended charge pump -Common mode and supply noise rejected -Differential stage is less sensitive to the leakage current since leakage current behaves as common mode offset with the dual output stages
Dis-advantages	<ul style="list-style-type: none"> -Switch mismatch, clock feed through, charge sharing problems -Limited output voltage compliance range 	<ul style="list-style-type: none"> -require two loop filters and common mode feedback circuitry -more number of transistors are required, with two or more current sources -occupy large silicon area and higher power consumption

10.4.2.1 Single ended charge pump

10.4.2.1.1 Switch in Drain

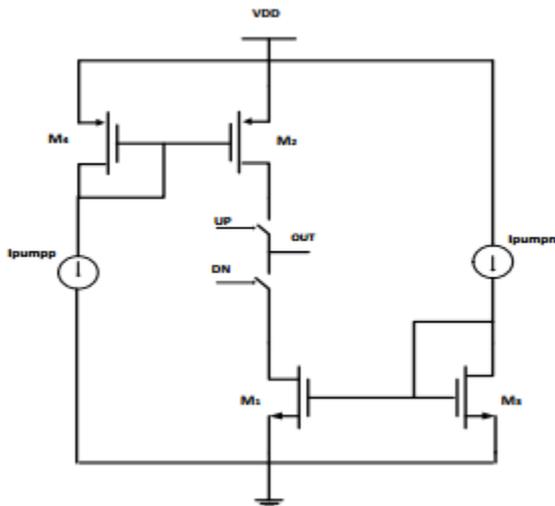


Figure 10.10.12 : Switch in drain single ended charge pump.

It is the most popular traditional charge pump, when DW switch is turned off the current pulls the drain of M1 to ground. After that when the DW switch is turned on the drain voltage of M1 increases from 0V to the control voltage on the loop filter, which means that transistor M1 operates in linear region for certain time until voltage on the drain of M1 reaches saturation voltage. In this period peak current is generated due to voltage difference of the M1 resistance series with the on resistance of the switch. The same action occurs for transistor M2 and the matching of this peak current is difficult which leads to reference spurs. This topology also suffers from charge sharing and clock feed through.

10.4.2.1.2 Switch in gate charge pump

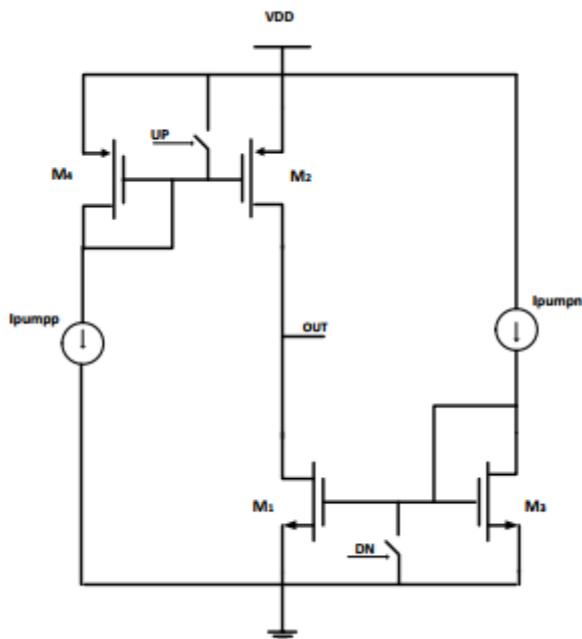


Figure 10.10.13 : Switch in gate charge pump

At this topology the two current mirrors (M1 and M2) are guaranteed to be in saturation region all the time. This topology increases the compliance range as there is no voltage drop on the two switches. But UP and DN signals face substantial capacitance from the current source and current mirror which makes it a low-speed topology, and also sizing of current sources affects the speed.

10.4.2.1.3 Switch in source charge pump

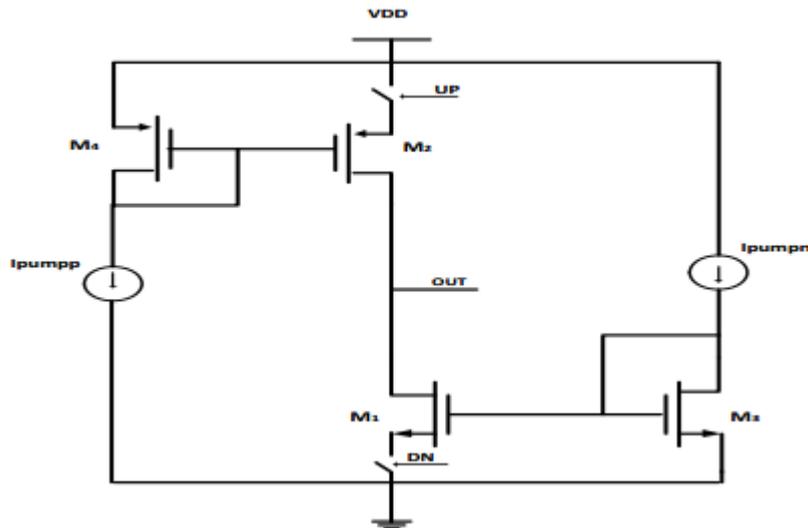


Figure 10.10.14 : Switch in source charge pump

In this topology M1 and M2 are in saturation all the time. This topology is faster than switch on the gate charge pump as the switch is connected to a single transistor with low parasitic capacitance. Sizing of current sources doesn't affect switching time, and this architecture solves the problem of charge sharing and clock feed through.

10.4.2.2 Differential charge pump

10.4.2.2.1 NMOS differential charge pump

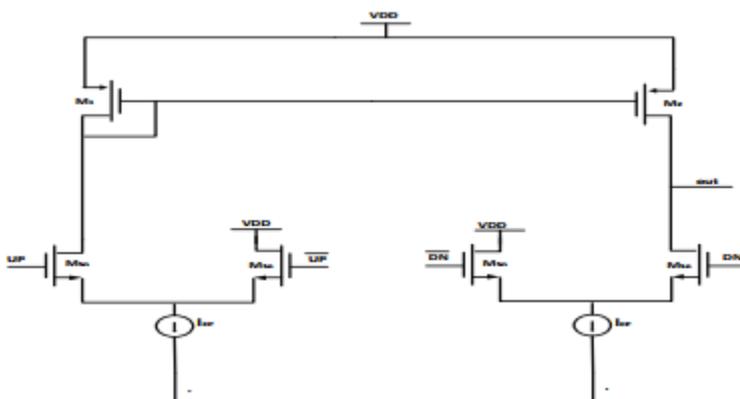


Figure 10.10.15 : NMOS differential charge pump

This topology avoids the inherent mismatch between NMOS and PMOS switches by using only NMOS switches. Since the current doesn't flow in the current mirror, M5 and M6, when the UP switch is turned off, so the current mirror still limits the performance.

10.4.2.2.2 Current steering charge pump

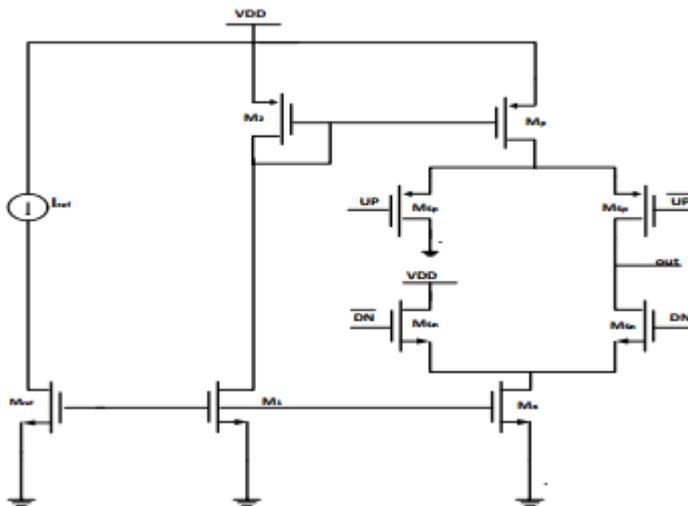


Figure 10.10.16 : Current steering charge pump

In this topology the two current sources are open all the time and the current steer between the two branches which greatly improve the switching speed. Increasing dimensions of current source and mirror transistors doesn't affect the switching speed. But its performance is similar to switch on drain topology which mean it suffer from charge sharing and clock feed through. To improve charge sharing problem in this architecture we use unity gain buffer to allow V_{ref} node to track the output node. So the drain of current sources virtually sees no difference when switching which solve charge sharing problem.

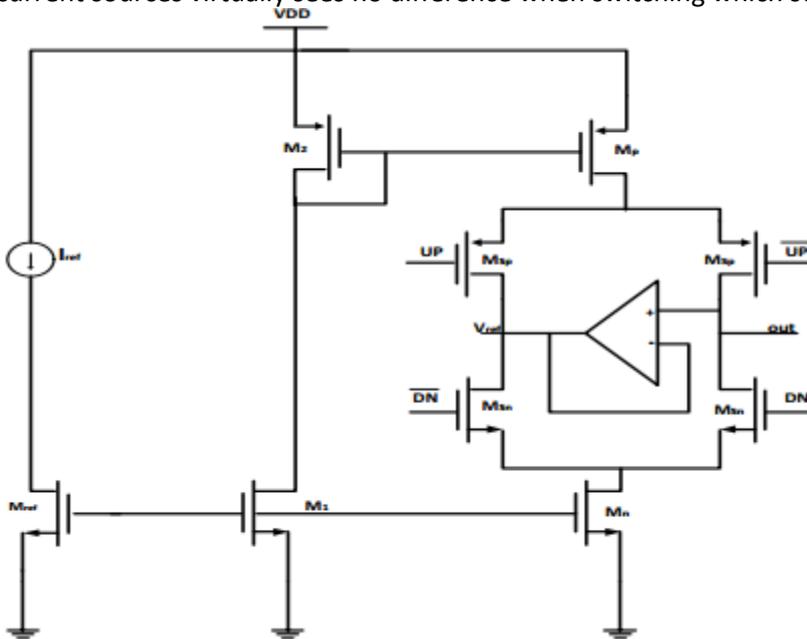


Figure 10.10.17 : Current steering charge pump with unity gain buffer

10.4.3 PFD & CHARGE PUMP non-idealities

10.4.3.1 DEAD zone problem

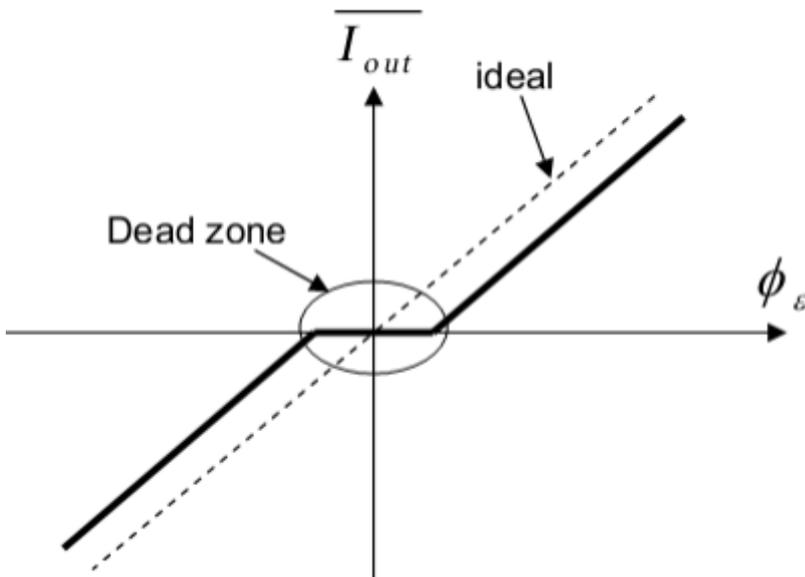
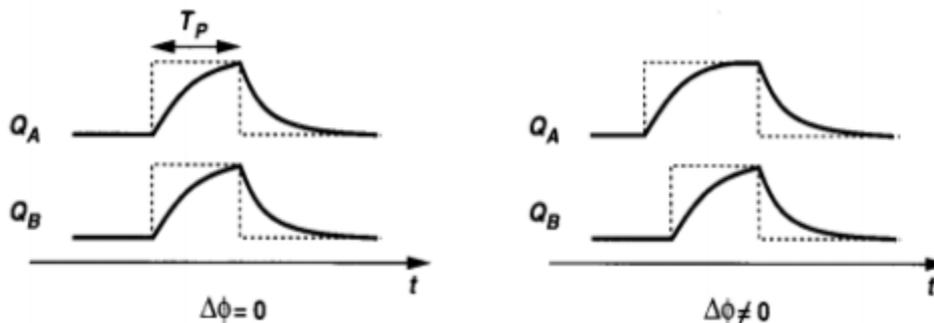


Figure 10.10.18 : PFD & CHARGE PUMP dead zone

Dead zone problem occur at small phase error between the two inputs of PFD. At small phase error PFD generates very narrow pulses on its output. Due to finite rise and fall times which resulting from input capacitance of charge pump switches, the pulses can't turn on the switches of charge pump, so no pumped current at output node. Loop gain become zero and the loop considered open which cause jitter. Solution of this problem by adding delay in the reset path of PFD until the output pulses is sufficiently wide to turn on charge pump current.



This figure shows two cases for PFD output signals, one at zero delay and the other at positive delay between the input signals of PFD. If the reset delay (T_P) is long enough to allow PFD outputs (Q_A , Q_B) to reach a valid logical level it will turn on the charge pump and eliminate dead zone.

Output spectrum of VCO should to be one Delta-Dirac function at certain frequency but Charge pump non-idealities causes reference spurious tones at output spectrum which lead to reciprocal mixing for undesired signals.

10.4.3.2 Current mismatch

Mismatch between Up and Down currents of charge pump lead to non-zero net current charge or discharge loop filter which result in changing the control voltage and change VCO output frequency and led to

reference spurs. PLL to prevent locking the average of control voltage must remain constant so PLL create phase error between input and output so that average net current injected to loop filter equal zero.

$$\Delta\phi = \frac{2\pi}{I_{cp}} \cdot \frac{1}{T_{ref}} \cdot \int_{t_0}^{t_0+T_{ref}} I_{out}(t) dt$$

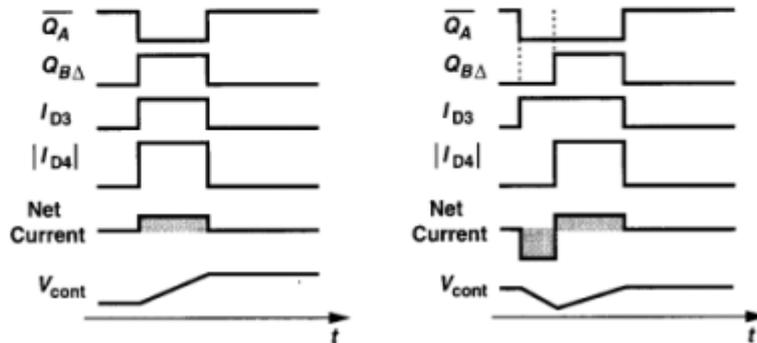


Figure 10.10.19 : Current mismatch effect

10.4.3.3 Leakage current

In locking state even if no current mismatch between charge pump current sources, there are leakage current $I_{leakage}$ flowing into loop filter and change control voltage and lead to reference spurs. For PLL hold locking condition it generates phase error between input and output so CP generate current pulse its width equivalent to phase error such that average pulse current equal to leakage current and net injected current to the loop equal to zero.

Leakage current results from non-ideal switching of current sources or VCO varactor leakage current or lossy capacitors in loop filter.

$$\Delta\phi = 2\pi \cdot \frac{I_{Leak}}{I_{cp}}$$

10.4.3.4 Timing mismatch

This issue arises from presence of delay between output signals of PFD which lead to disturb the control voltage which results to reference spurs even though there are no current mismatch and leakage current. Solution of this issue is to try to equalize delay between PFD signals as much as possible.

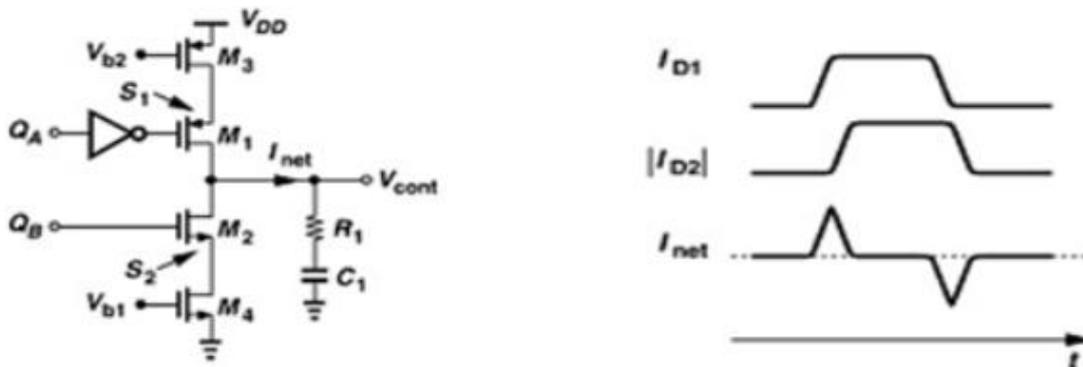


Figure 10.10.20 : Timing mismatch

10.4.3.5 Charge sharing

This phenomenon occurs due to finite capacitance at the Drains of current sources. When the switching transistors are open, the voltages on Drains of the current sources transistors move towards V_{DD} and GND. When the switches close, instantaneously, some of the charge stored on these parasitic capacitors will be transferred to the loop filter. Even if $C_x=C_y$, which lead to control voltage jump. The solution of this problem is to use unity V gain buffer to hold the control voltages to the parasitic capacitance during switching.

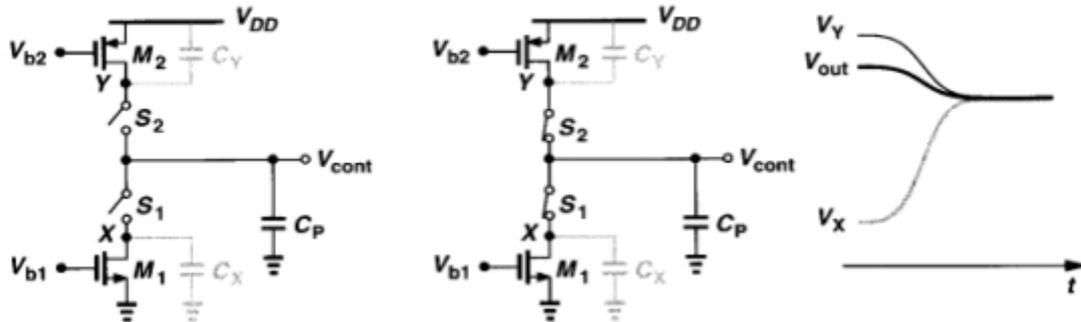


Figure 10.10.21 : Charge sharing

10.4.3.6 Clock feed through

Clock feed through occur due to finite gate to drain parasitic capacitance C_{gd} of the switches. The high frequency signal provided at the gate of switch transistor passes to the output node through C_{gd} . Clock feed through results in jumps in control voltage. Since the VCO sensitivity is high, even a small jump in control voltage results a large jump in output frequency. Solution of this issue is placing the switches near supply rails or by reducing sizing of the switches to reduce gate to drain capacitance.

10.4.4 Design methodology

I used the single ended charge pump to decrease the power consumption , area & to avoid using 2 loop filters .

The topology of switch in source is the used topology due to its fast speed and no parasitic capacitance, charge sharing& feed through.

The kind of the current mirror used in the topology is the low drop cascode current mirror which is used in the N& P networks and has a high impedance output node so the output current become stable across a large range of output voltage.

It was required to design a charge pump with a current of

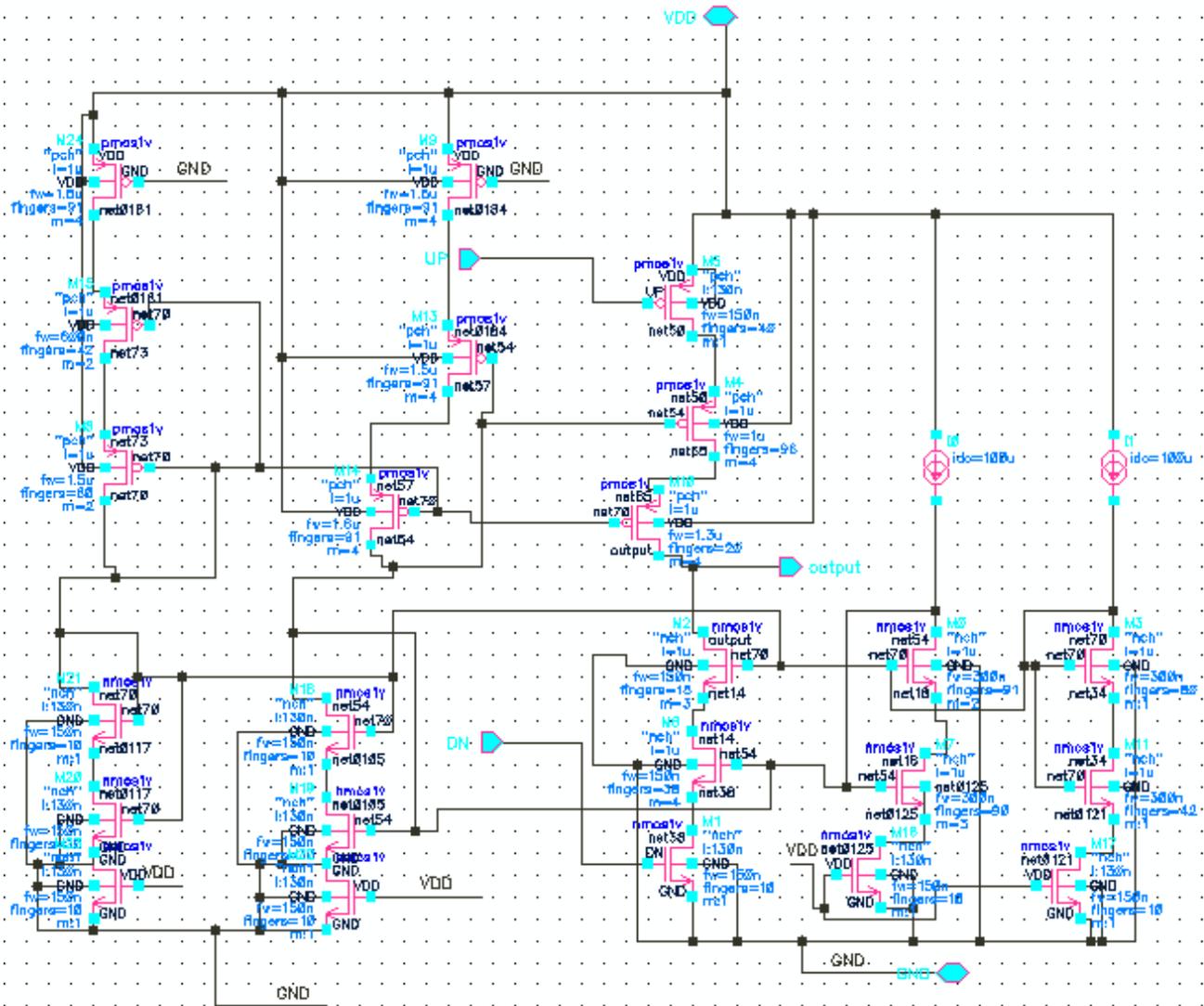


Figure 10.10.22 : SWITCH IN SOURCE CHARGE PUMP CIRCUIT

10.4.5 Simulation results

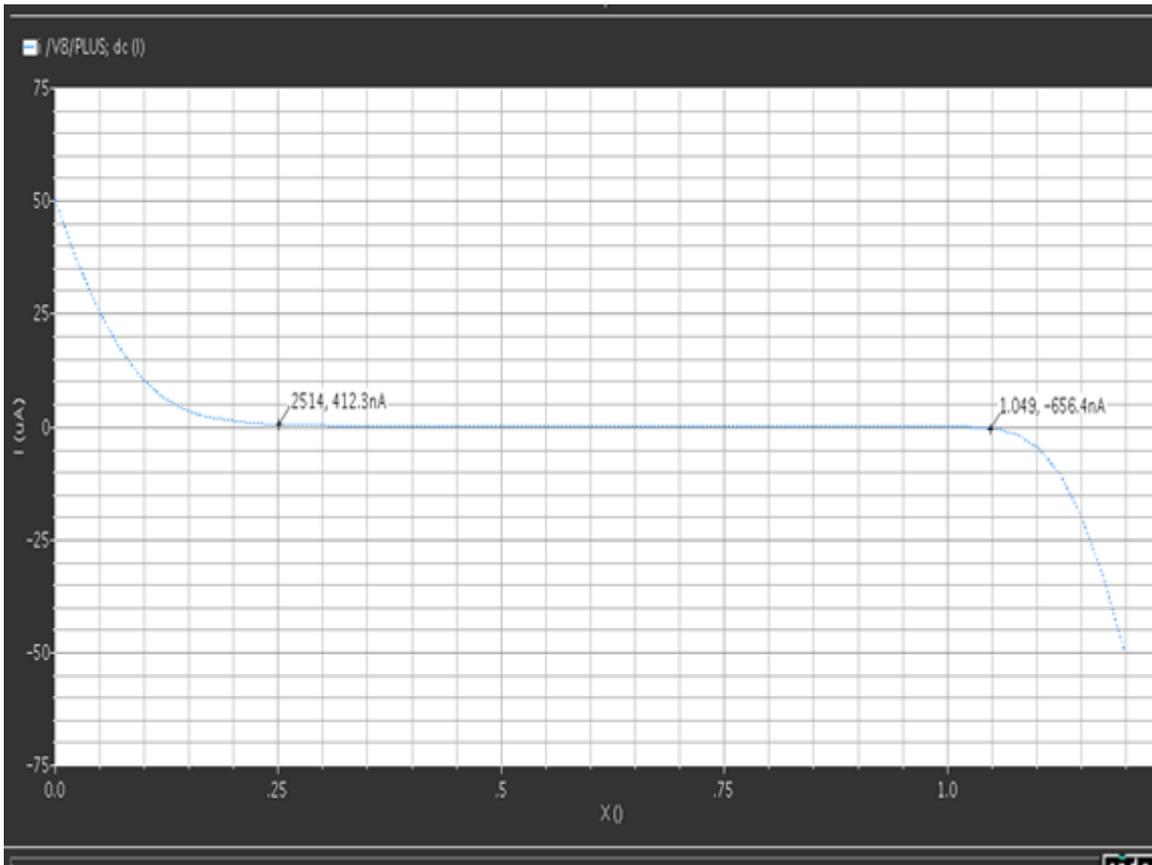


Figure 10.10.23 : DC mismatch

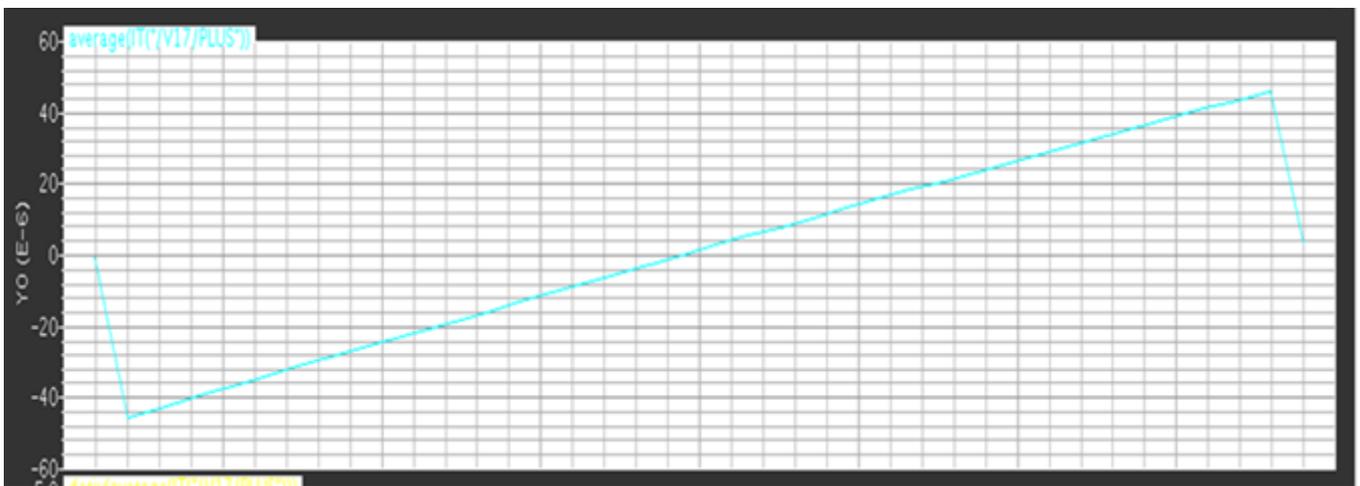


Figure 10.10.24 : PFD charge pump transfer function

Device	Param	Noise Contribution	% Of Total
/I45/M6	id	1.86516e-12	21.63
/I45/M4	id	1.83126e-12	20.85
/I45/M5	id	1.74305e-12	18.89

Spot Noise Summary (in A/sqrt(Hz)) at 1M Hz Sorted By Noise Contributors
 Total Output Noise = 4.01045e-12
 Total Input Referred Noise = 2.89483e-07
 The above noise summary info is for noise data

Figure 10.10.25 : CHARGE PUMP output RMS noise summary

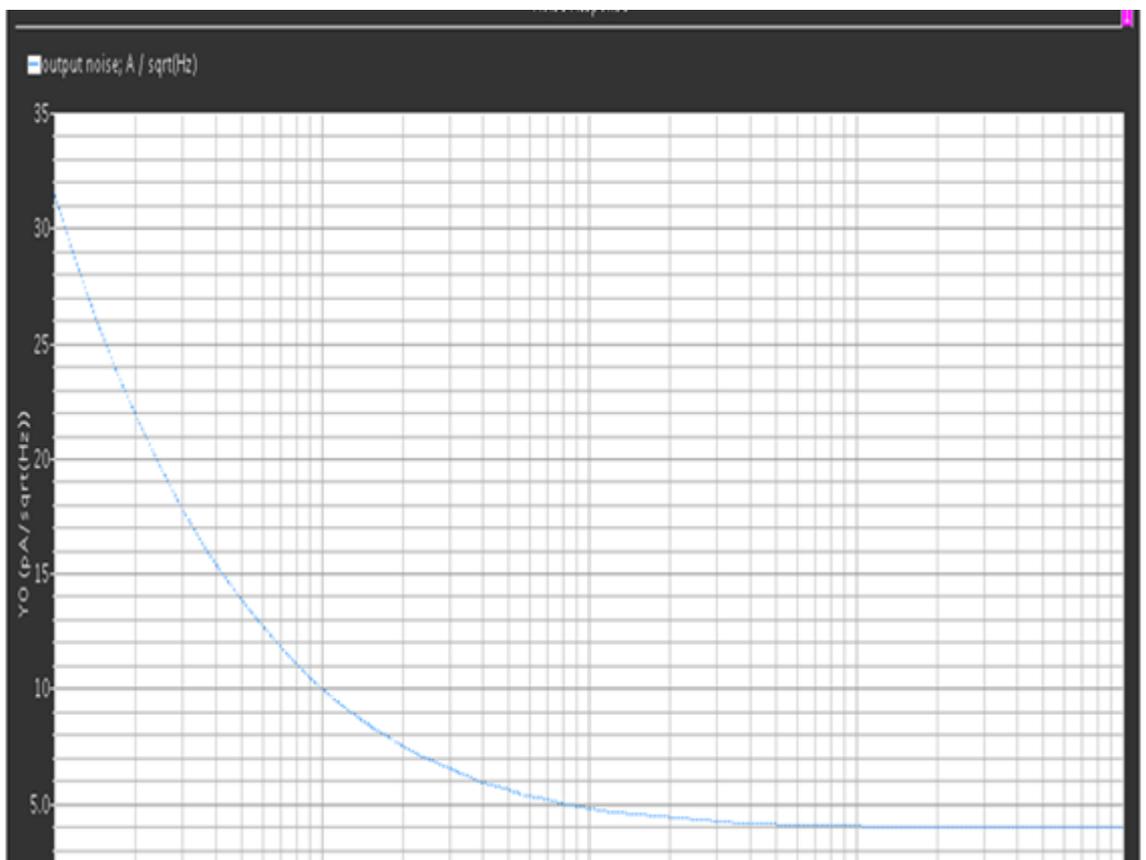


Figure 10.10.26 : Output noise of charge pump

11 Voltage controlled oscillator “VCO”

11.1 Introduction

A voltage controlled oscillator (VCO) is an oscillator whose output is a periodic signal (mostly sinusoidal or square wave) whose frequency is controlled by varying its control voltage.

Voltage controlled oscillator is one of the most important blocks in modern communication systems as it provide a precise reference frequency that can be used in modulation and demodulation of RF signals, in receivers the oscillator wave form is used as a reference signal to down convert the radio frequency signal (RF) to intermediate frequency signal (IF) and it also can be used in digital circuit as a clock.

VCO is usually used in Phase Locked Loops (PLL) in order to provide different LO frequencies which is used for channel selection.

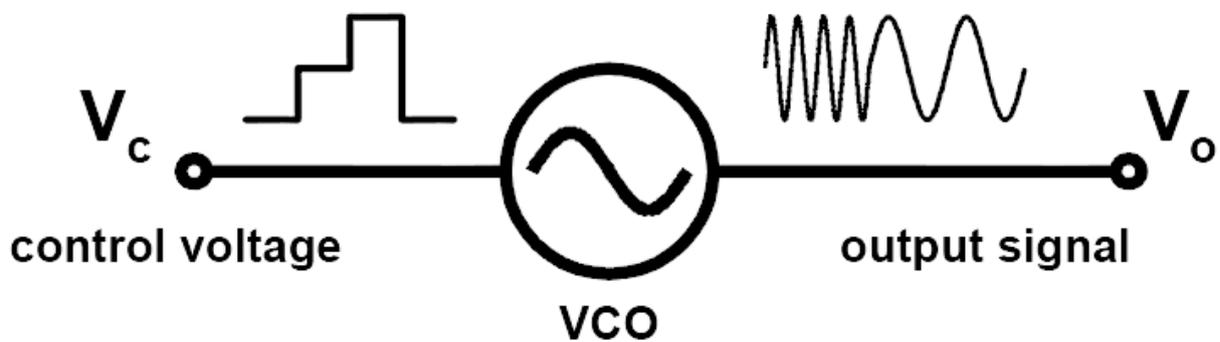


Figure 11.11.1 VCO

11.2 VCO specification

11.2.1 Phase noise

The spectrum of the oscillator output is not a sharp impulse but rather it is broadened by the noise of its constituent devices which is called “phase noise“.

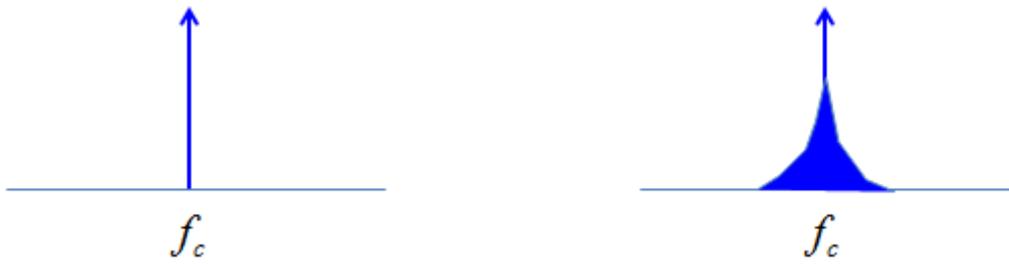


Figure 11.11.2 spectrum without and with phase noise

The effect of phase noise result from the noise spurs that hit the zero-crossings of the signal in time domain which result in phase change (jitter).

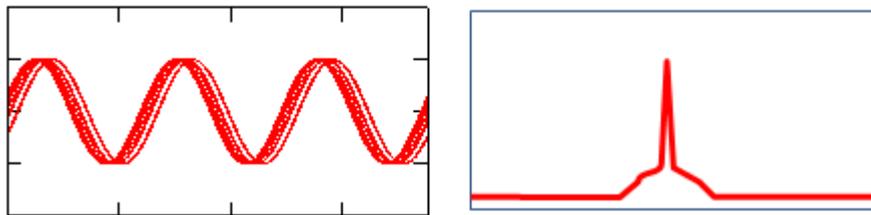


Figure 11.11.3 phase noise effect in time domain

Phase noise is measured with respect to the carrier frequency at offset Δf with units of dbc/Hz

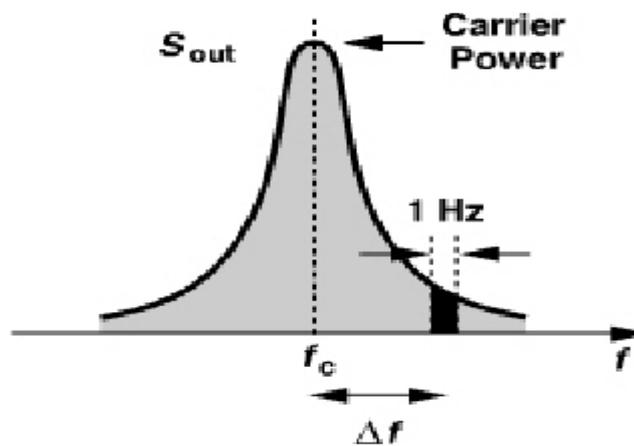


Figure 11.11.4 Phase noise specification

11.2.1.1 Phase noise effect

It can cause interference with nearby signals making it hard to be probably decoded.

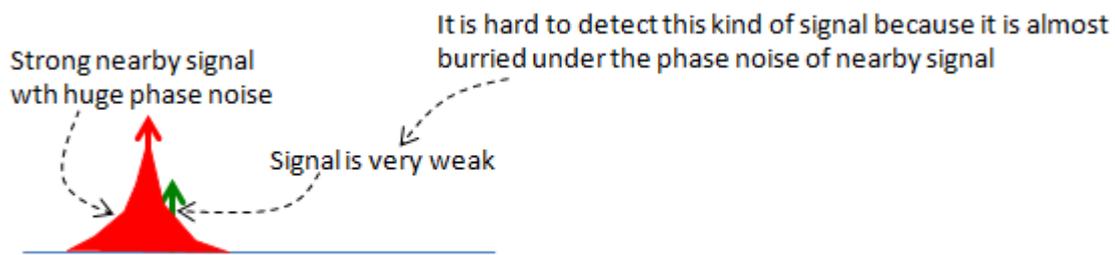


Figure 11.11.5 phase noise effect at the RX

11.2.2 VCO gain (K_{VCO})

It is defined as the slope of the frequency variation with respect to the control voltage

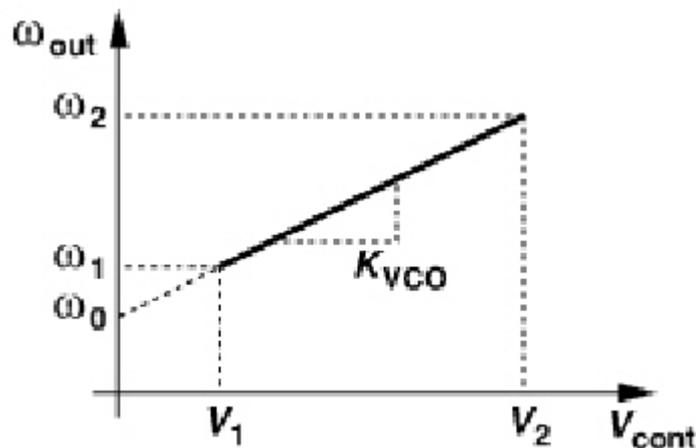


Figure 11.11.6 output frequency vs. vctrl

K_{VCO} value has a direct influence on the overall PLL noise performance as any noise on the loop filter control voltage will be amplified by the K_{VCO} value so it is desirable for this value to be smaller and it also has an influence on the loop bandwidth.

11.2.3 Tuning range

It is defined as the frequency range that VCO can cover and it is divided into 2 component:

1. The frequency range that is required by the system specification.
2. Additional margin to cover process and temperature variation

11.2.4 Output swing

As the VCO following stage (in RF systems) is usually a divider or mixer so the VCO must produce sufficient output swing to insure complete switching of the transistors of the next stage.

11.2.5 Supply sensitivity

It is defined as the change in the VCO output frequency with any variation in the supply voltage which leads to the translation of supply noise to frequency noise.

11.2.6 Frequency pulling

It is defined as the change in VCO output frequency with load impedance variation.

11.2.7 Drive capability

It is defined as the load capacitance which the VCO can drive and it is a main parameter to determine the power consumption in the output buffers.

11.2.8 Power dissipation

The power drained by the VCO and its buffer considered as the main power consumers in the whole PLL and always there is trade-off between the required phase noise and the power consumption.

11.3 Basic concepts

11.3.1 Feedback model

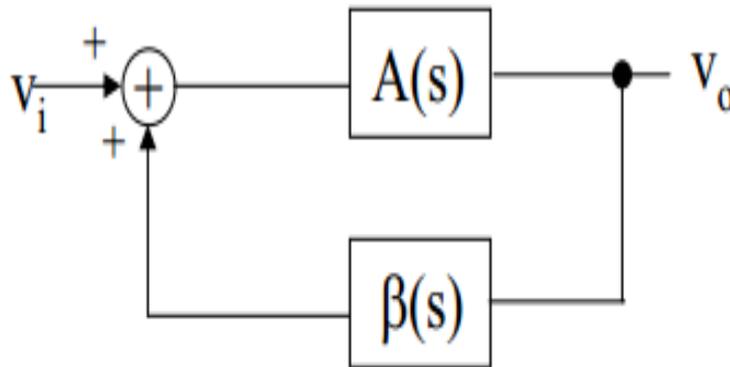


Figure 11.11.7 feedback model

The transfer function of this system is given by:

$$H(s) = \frac{A}{1 - AB}$$

From Barkhausen criteria, for a system to oscillate it must satisfy 2 conditions:

1. Loop gain $|AB|=1$
2. Phase shift $\angle AB=2n\pi, n=0,1,2,\dots$

The need for a feedback system is a must for any oscillator in order to compensate for the losses and to insure a sustained oscillation the loop gain must be greater than or equal to unity otherwise the oscillation will start and end rapidly due to the domination of losses (damped oscillation).

11.3.2 One port model

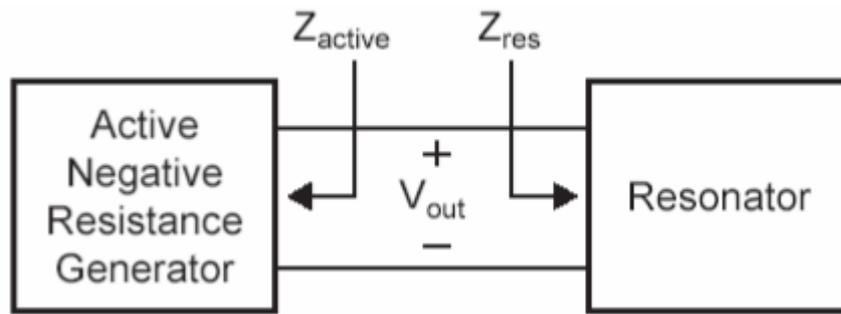


Figure 11.11.8 one port model

A design concept that treat the resonator as a one port network with losses equal to R_{res} and in order to compensate this losses active circuit is required with $-ve$ resistance = $-R_{res}$

11.4 Topologies survey

11.4.1 Ring oscillator

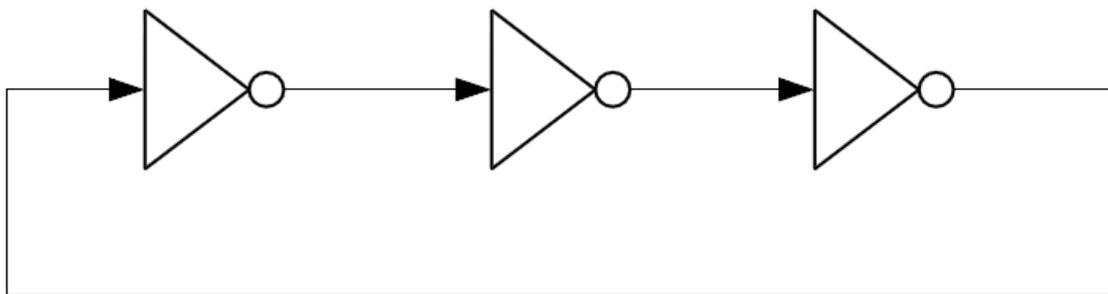


Figure 11.11.9 3 stage ring oscillator

A ring oscillator is a type of untuned oscillator which consist of multiple gain stages in a loop, each stage can be simple as inverter.

In order to form an oscillation, ring oscillator must obey Barkhausen criteria at which the total gain around the loop must be greater than or equal to unity and the total phase shift to be equal 180° , thus for a 3-stage ring oscillator the phase shift required by each stage is equal to 60° .

Thus, frequency of oscillation is given by:

$$\omega_{osc} = \sqrt{3} \omega_0$$

Where ω_0 is the 3db bandwidth of each inverter.

As the number of stages increase the required gain per stage decrease but also the oscillation frequency decrease.

Due to the absence of a highly tuned circuit (tank circuit) in the ring oscillator its noise performance is very poor and it is not used when low phase noise is required while on the other side that makes it use a smaller area on the chip.

The ring oscillator got a wide tuning range for its large VCO gain but that makes it more sensitive to noise and disturbance.

11.4.2 LC oscillator

11.4.2.1 Tank circuit

The LC oscillator is a type of the highly tuned oscillator in which the frequency of oscillation is determined by the tank circuit.

$$\omega_{osc} = \frac{1}{\sqrt{LC}}$$

The presence of the tank circuit play an important role in oscillation buildup as initially the charges on the capacitor flow into the inductor (as the capacitor discharge) creating a magnetic field on the inductor which leads to the flow of current in the opposite direction which recharge the capacitor again (assuming no losses) and

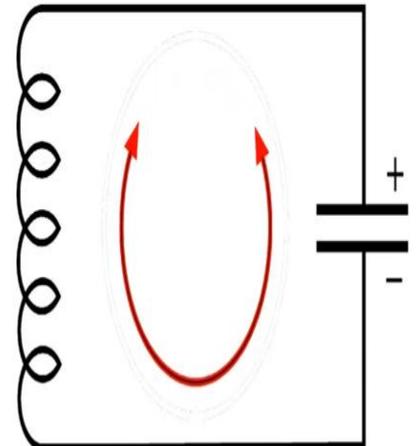


Figure 11.11.10 ideal LC tank circuit

so on which leads to oscillation build up.

Also the presence of the tank circuit act as a main parameter in lowering the oscillator phase noise, as from lesson's equation the phase noise is given by

$$L(f_m) = \frac{FKT}{2P_{av}} \left[1 + \frac{f_c}{f_m} + \left(\frac{f_o}{2f_m Q_L} \right)^2 \left(1 + \frac{f_c}{f_m} \right) \right]$$

Where;

Q_L : is the loaded quality factor of the tank.

P_{av} : is the average power.

f_o, f_c, f_m : is the flicker corner frequency & the carrier center frequency & the offset frequency

F : noise factor of active devices.

K : Boltzmann constant.

Lesson's equation show the inversely proportional relation between the phase noise and the tank quality factor so as the quality factor of the tank increase \rightarrow Phase noise decrease and that's why the LC oscillator can reach a very good noise performance at RF frequencies.

In the following section we will discuss the famous LC oscillator topologies (NMOS LC oscillator, PMOS LC oscillator & the complementary) their configuration, advantages & disadvantages and we will end up with a comparison between them and choosing the best topology for our specifications.

11.4.2.2 Cross coupled LC oscillator

For a single stage CS amplifier with a tank circuit connected at its drain and at the tank oscillation frequency, the phase shift is 180° (due to CS configuration) and the phase shift for the tank at the oscillation frequency equal to 0° .

Thus, from Barkhausen criteria to create

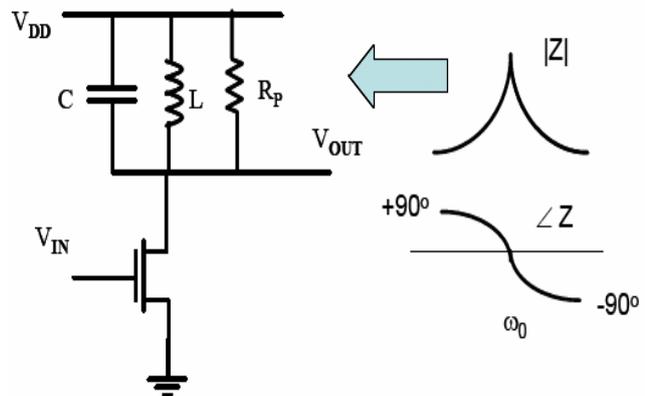


Figure 11.11.11 CS amplifier with a tank circuit

oscillation we need to:

1. Gain ≥ 1 .
2. Phase shift of $2n\pi$ (0, 360,...) \rightarrow add two stages in cascade & create feedback.

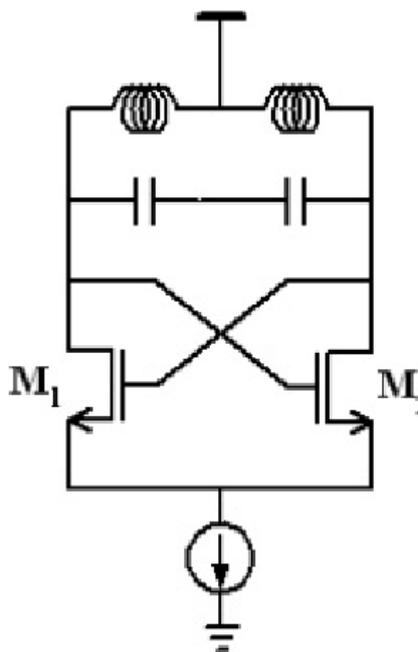


Figure 11.11.14 NMOS cross-coupled LC oscillator

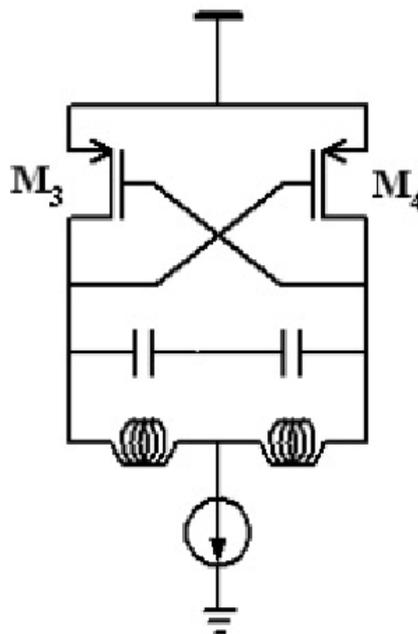


Figure 11.11.13 PMOS cross-coupled LC oscillator

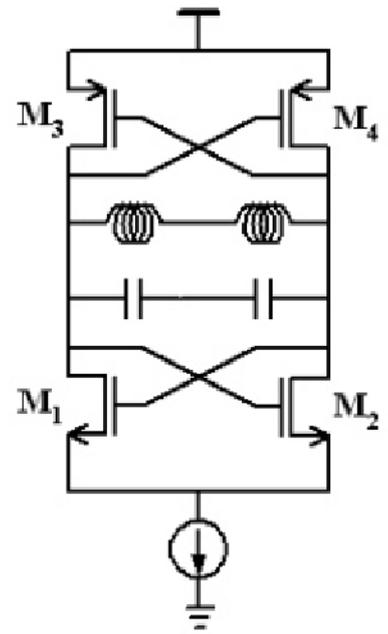


Figure 11.11.12 CMOS cross-coupled LC oscillator

And from this concept we have 3 main cross coupled LC oscillator topologies:

And the choice between these configurations depends upon the intended specifications about phase noise, output swing & supply headroom.

But first we had to discuss the need of current biasing mechanism which result from the following facts:

1. Without current biasing the only factor that control the trans-conductance of the devices will be there sizes thereby lacking a flexible approach in establishing control on the trans-conductance.
2. Current biasing sets a limit for the total power dissipation.
3. It aids the designer to achieve a compromise between the Phase noise and the power dissipation.

11.4.2.2.1 NMOS only vs. PMOS only vs. CMOS

NMOS has a higher mobility than PMOS at the same size which lead that PMOS has a smaller flicker noise than NMOS but due to the fact that the PMOS got a lower mobility thus, PMOS size has to be 3 to 4 times as the size of the NMOS.

For CMOS the voltage swing is essentially limited by the supply voltage and the bias current while for NMOS only or PMOS only it is limited with the bias current thus, for a very low supply voltage the NMOS or PMOS only configuration is better as the CMOS require higher supply value so as to operate with the same performance.

For NMOS or PMOS only the voltage swing can exceed VDD but for CMOS this voltage swing is limited by PMOS transistors driven into cutoff.

Also the use of more than 2 active devices other than NMOS or PMOS only increase the number of noise sources and the parasitic which lead to a significant effect in the overall noise performance.

Thus for the same size and the same bias current and the same supply voltage the choice will be for the NMOS only configuration.

Thus, the comparison between NMOS only, PMOS only and CMOS configuration end up with choosing the NMOS only configuration thus, the only remaining comparison is between LC oscillator and the Ring oscillator taking into consideration the required low Phase noise profile.

Table 11.15 LC oscillator vs. Ring oscilaltor

	LC OSCILLATOR	RING OSCILLATOR
PHASE NOISE	Lower	Higher (very poor)
TUNING RANGE	Lower	wider
AREA	larger	smaller
INTEGRATION WITH DIGITAL CHIPS	Poor integration	Easier integration
MULTI-PHASE OUTPUT	Harder	easier

11.4.2.2.1.1 Conclusion

From the previous comparisons we choose NMOS LC cross coupled oscillator for its low noise profile which is the main aspect required from VCOs for high frequency applications.

11.5 Design methodology

In this section we will discuss the design sequence starting from the tank circuit and ending up with the capacitor bank and the buffers.

First, I would like to mention some of the most important guiding equations that guide for understanding the design main trade-offs.

Lesson's equation:

$$L(fm) = 10\log\left[\frac{2FKT}{P_{sig}}\left(1 + \left(\frac{f_o}{2fmQl}\right)^2\right)\left(1 + \frac{fc}{fm}\right)\right]$$

As mentioned above, lesson's equation gives a direct insight about different parameters and their relation with the phase noise, so from lesson's equation we can deduce that in order to achieve a low phase noise profile:

1. Increase the tank quality factor (Q).
2. Decrease the overall Noise Figure (F).

The overall circuit Noise Figure is given by

$$F(\Delta f) = 1 + \frac{2YI_{bias}R_p}{\pi A} + \frac{YI_{bias}R_p}{2V_{od, bias}}$$

The first term is the noise from the tank resistance,
2nd term is the noise from the active transistors,
3rd term is the noise from the current source.

For the lowest noise profile we had to minimize the Noise Figure as much as possible.

And to guarantee the design optimization, the Noise contribution of all the circuit component

Must be arranged as given by the noise figure so, in order to achieve the minimum noise figure (1) we need to minimize the noise contributed by the current source and the active transistors.

But first we need to maximize the tank quality factor.

11.5.1 Tank circuit

First, we compare different types of the available inductors in the technology library.

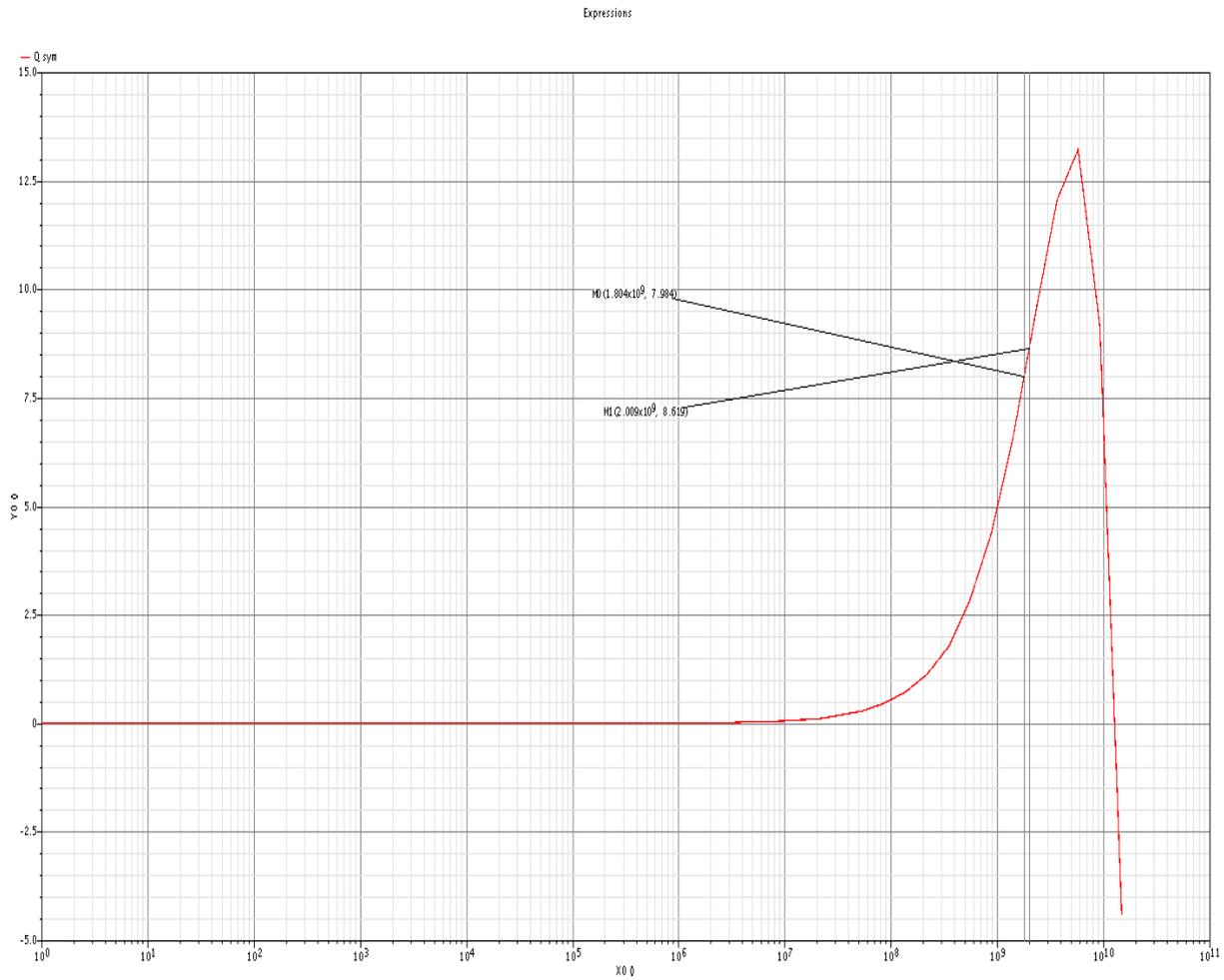
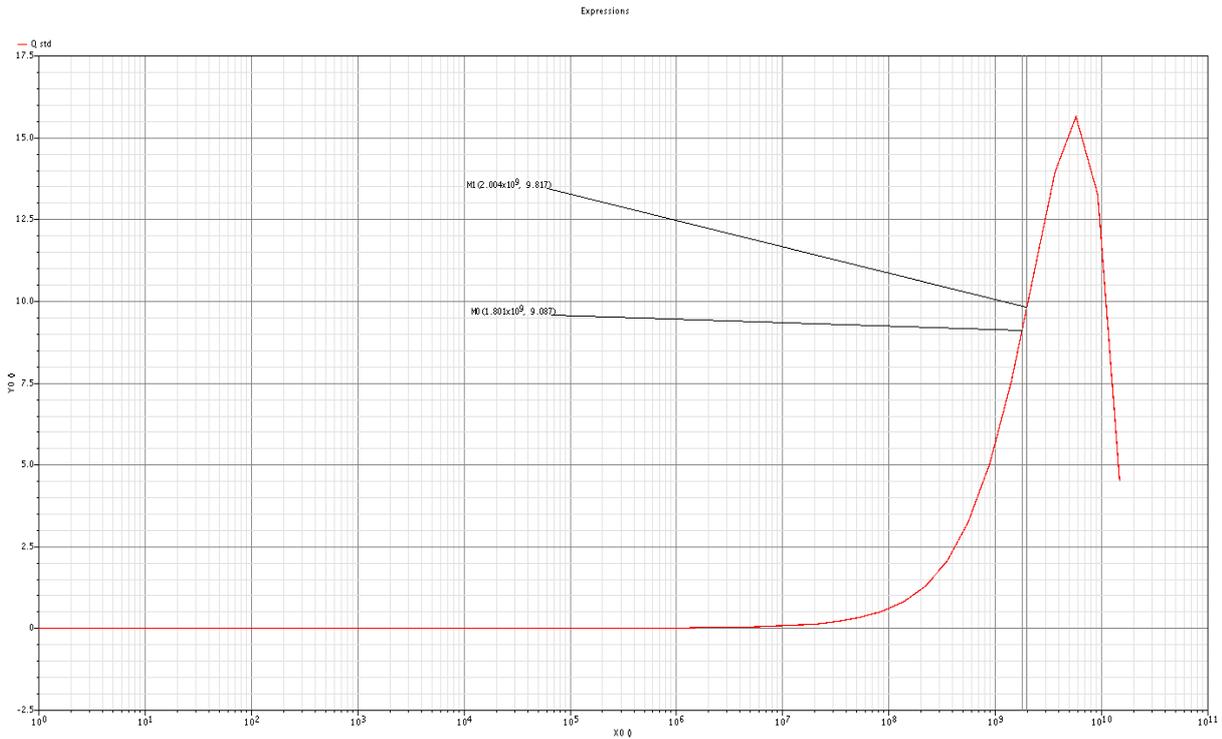
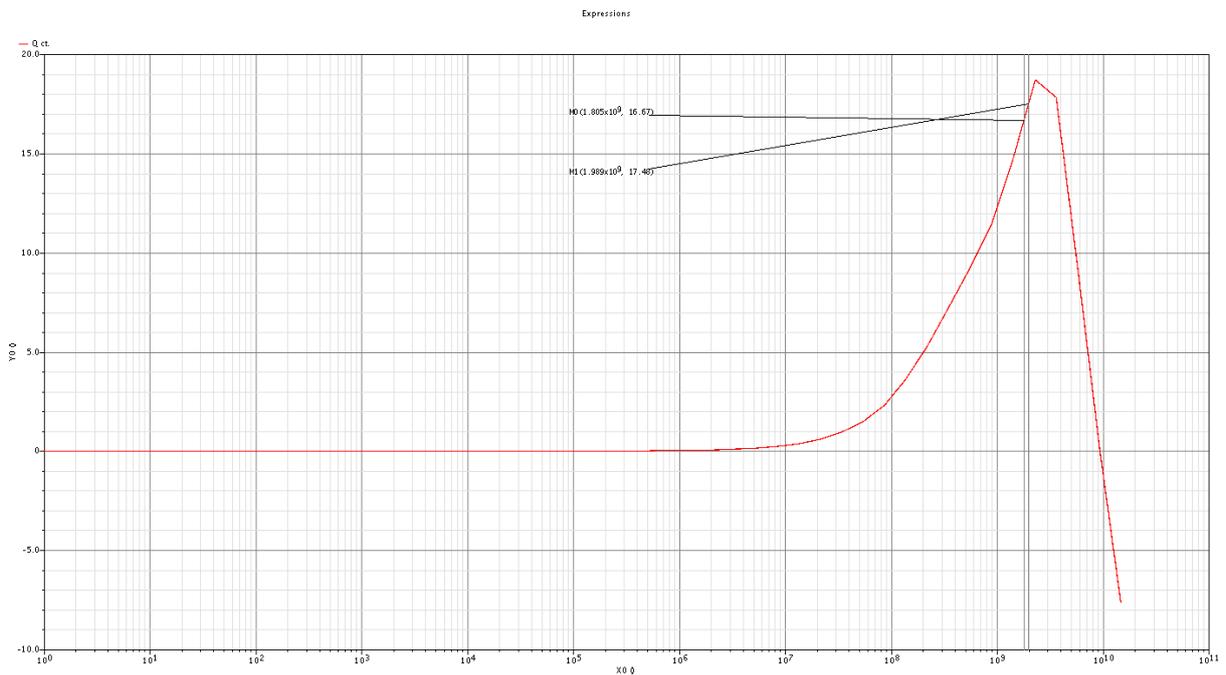


Figure 11.11.15 Q of symmetrical inductor

Q=7.9 @ 1.8G & Q=8.6 @ 2G



**Figure 11.11.16 Q of the standard (STD) inductor
Q=9 @ 1.8G & Q=9.8 @ 2G**



**Figure 11.11.17 Q of the Centre tap (CT) inductor
Q=16.67 @ 1.8G & Q=17.48 @ 2G**

Accordingly, **we choose the center-tap** as it got the highest quality factor and also because the center-tap (differential inductor) is area efficient than other inductors.

2nd, we calculate the inductor loss.

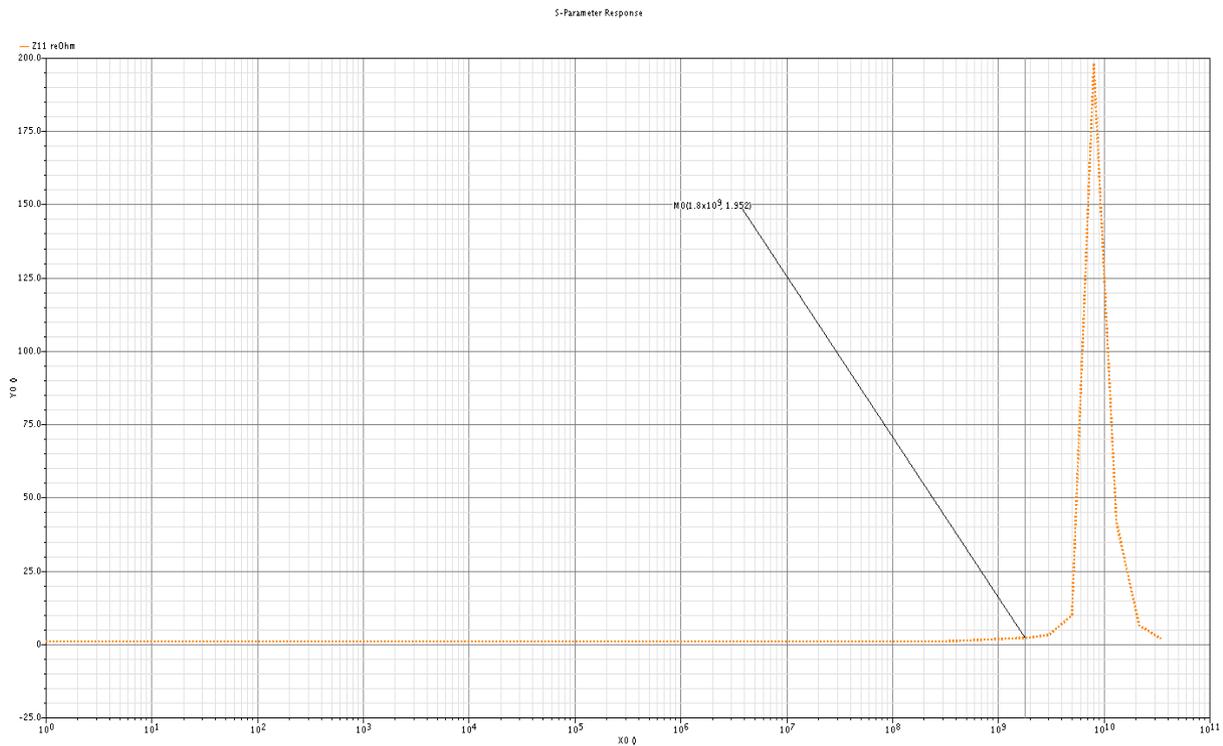


Figure 11.11.18 inductor series resistance (ohm)
 $R_s = 1.952 \text{ ohm}$

Thus, from this simulation we can calculate the inductor parallel resistance and we find it to be $R_p = 531.432 \text{ ohm}$.

11.5.1.1 Varactor design

The equivalent circuit of the tank, Varactor, capacitor bank that can be used for the quality factor calculation is given by [1] as follow

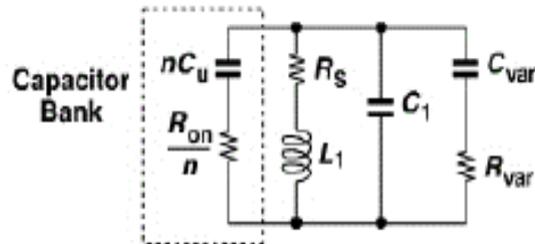


Figure 11.11.19 equivalent circuit for Q calculation

And the overall quality factor can be found as

$$\frac{1}{Q_{tot}} = \frac{1}{Q_L} + \frac{C_{var}}{C_1 + C_{var} + nC_u} \frac{1}{Q_{var}} + \frac{nC_u}{C_1 + C_{var} + nC_u} \frac{1}{Q_{bank}}$$

Q_L : inductor quality factor

C_{var} : Varactor capacitance

Q_{var} : Varactor quality factor

C_u : unit capacitance of the capacitor bank

Q_{bank} : capacitor bank quality factor.

And this equation can be used as a guiding equation in choosing the Varactor and the capacitor bank.

Thus, in order to keep the overall quality factor **we choose the Varactor with the highest quality factor**.

But here **we encounter a problem** which is the large variation in the VCO gain (K_{vco}) within the compliance range and that can be shown by the following figure

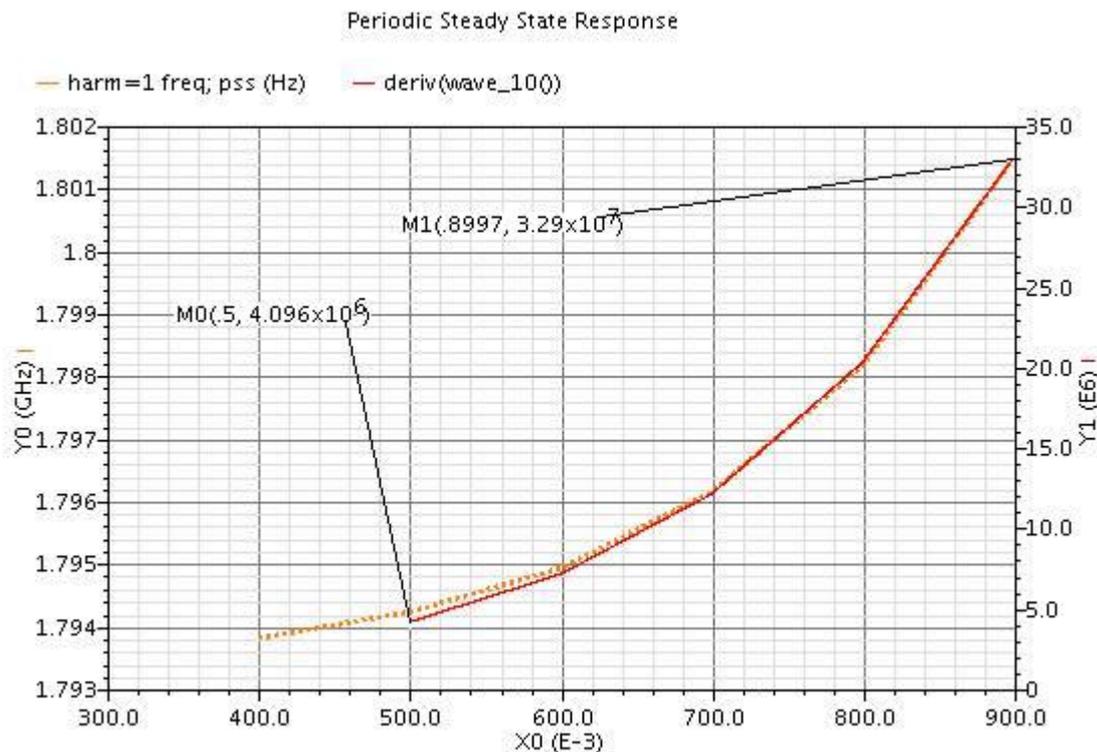


Figure 11.11.20 KVCO variation (solid) across the compliance range
KVCO varies from 5 MHz/v to 40 MHz/v

Which is a large variation across the compliance range so we had to control the voltage on the Varactor thus we use the following technique.

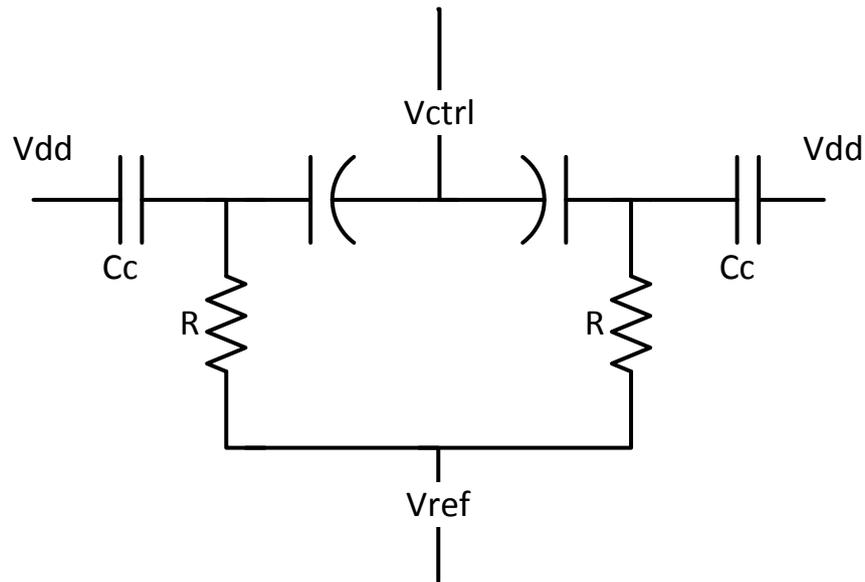


Figure 11.11.21 Capacitor coupled Varactor

C_c is a coupling capacitor which is added in series between the Varactor and the oscillating node thus, the varactor node can be forced to any DC voltage (V_{ref}) so as to operate at the most linear region of the varactor characteristics.

The value of C_c is chosen much larger than the varactor capacitance value so as to minimum affect the tuning range.

V_{ctrl} is the control voltage from the loop filter and the value of R is chosen large enough so as not to degrade the tank quality factor consequently the Phase noise and the following figure show the effect of R value on the Phase noise

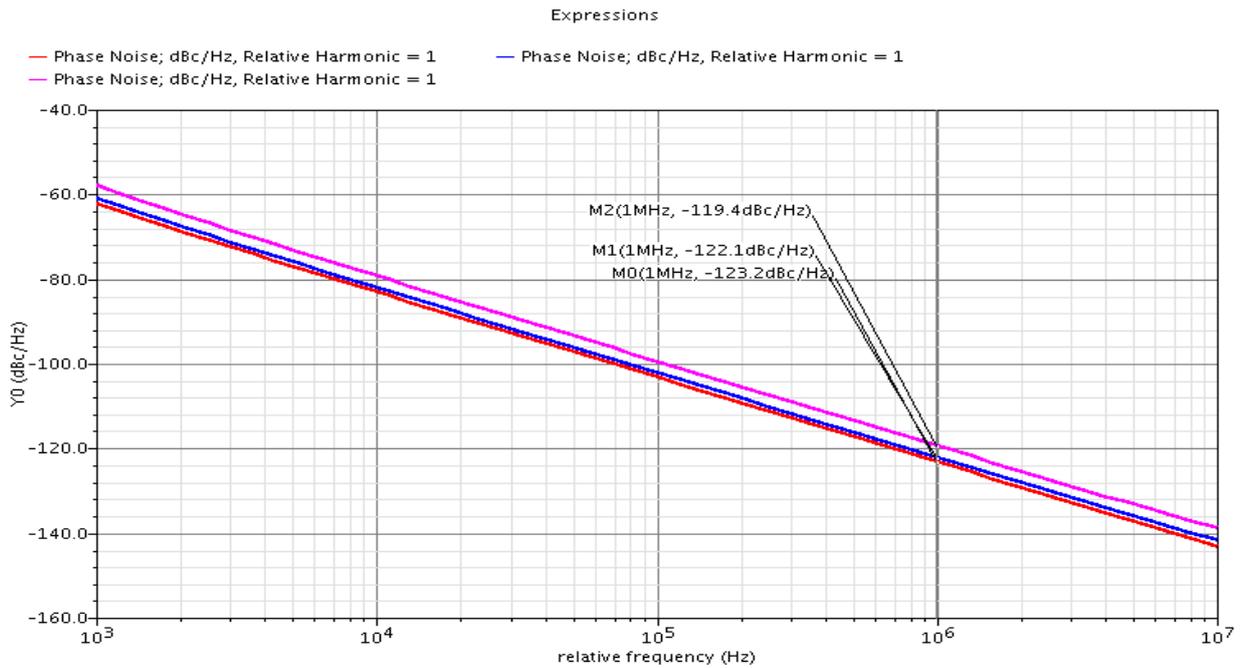


Figure 11.11.22 show the effect of R on the Phase noise
for R =500 ohm, P.N @1 MHz = -119.4 dbc/Hz
for R =1K, P.N @1 MHz = -122.1 dbc/Hz
for R =2K, P.N @1 MHz = -121.1 dbc/Hz
for R =3K, P.N @1 MHz = -123.2 dbc/Hz

Accordingly, **we choose R = 3K** and by this technique we minimize the Kvco variation across the compliance range and that can be shown by the following figure

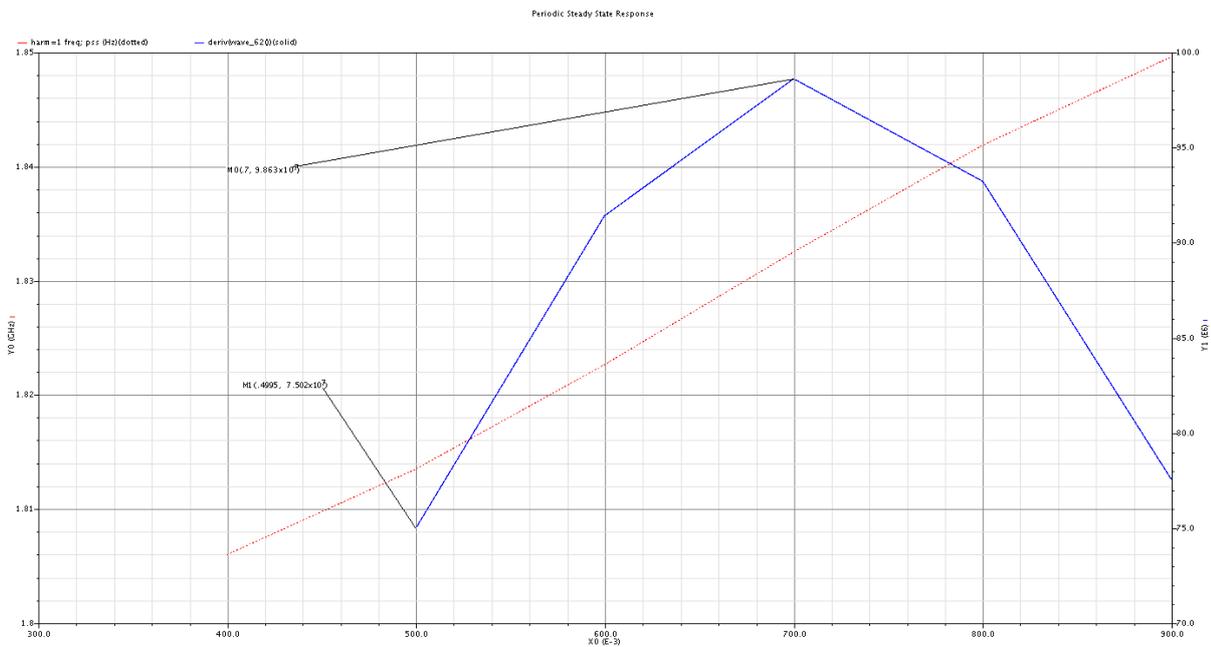


Figure 11.11.23 KVCO variation
KVCO varies from 75 MHz/v to 95 MHz/v
average = 90 MHz/v

So, summarizing the tank circuit in the following table

Table 11.16 tank circuit values at 1.8Ghz

L	2.95nH
Q	16.5
R_{EQ}	531.4 ohm
C_F	2.7PF
C_C	6pF
C_{VAR}	607.5fF
VARACTOR	10/5
C_{UNIT}	220fF

11.5.2 Active devices design

As mentioned above the main role of the active devices is to provide the negative resistance required to compensate for the tank losses and to maintain a loop gain ≥ 1 to satisfy Barkhausen criteria.

Thus,

$$(gmRp)^2 \geq 1$$

$$gm \geq \frac{1}{Rp}$$

In most practical cases a safety factor (α) is usually taken to insure the oscillation condition and its value from 2~3 thus,

$$gm = \alpha \frac{1}{Rp}$$

$$Rp = 531.4 \text{ ohm}$$

And by taking $\alpha = 3$, we got

$$gm = 5.6\text{mS}$$

Thus, we choose gm to be equal 7.4mS to account for additional losses that may be encountered from the capacitor bank.

We consume 1.445mA in the core in order to achieve the required output voltage swing and that can be calculated from the following equation

$$V_{\text{tank}} = \frac{2}{\pi} I_{\text{bias}} Rp$$

11.5.3 Tail current

Any noise on the tail current source is frequency translated by the switching pair as in a single balanced mixer and injected to the tank and due to the varactor this noise is translated directly into phase noise thus, for the tail current we had to minimize its flicker noise in order to decrease noise up-conversion thus, we size it for the minimum g_m that keep it in saturation.

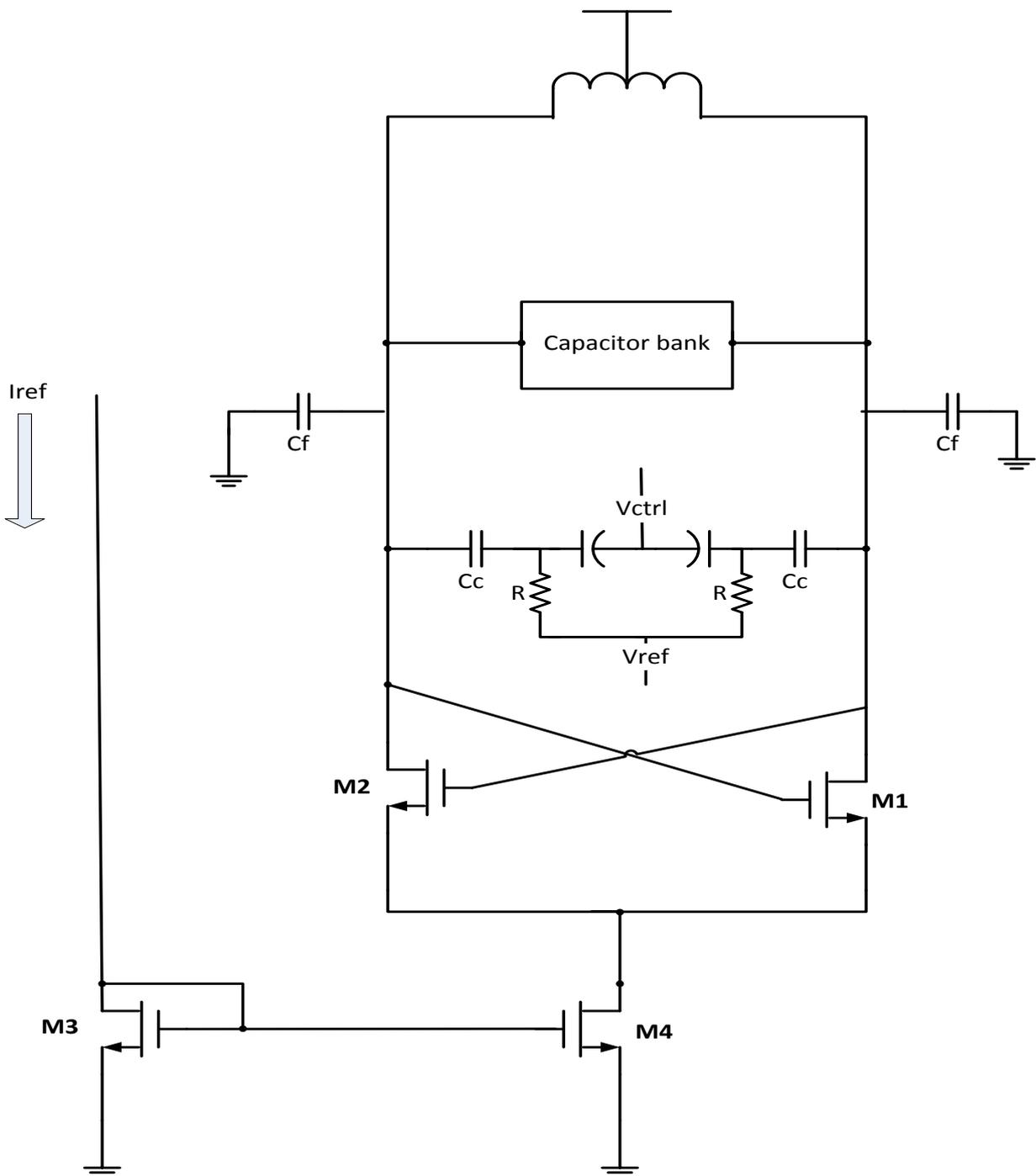


Figure 11.11.24 VCO core schematic

11.5.4 Buffer design

Our main goal in choosing the buffer was to achieve the largest output bandwidth-with a reasonable value of current-so that we minimize the attenuation on the output signal, Accordingly we choose the

source follower while the inverter can't be used because the common mode of the output signal from the NMOS-only VCO is always equal to the supply voltage so it will not switch.

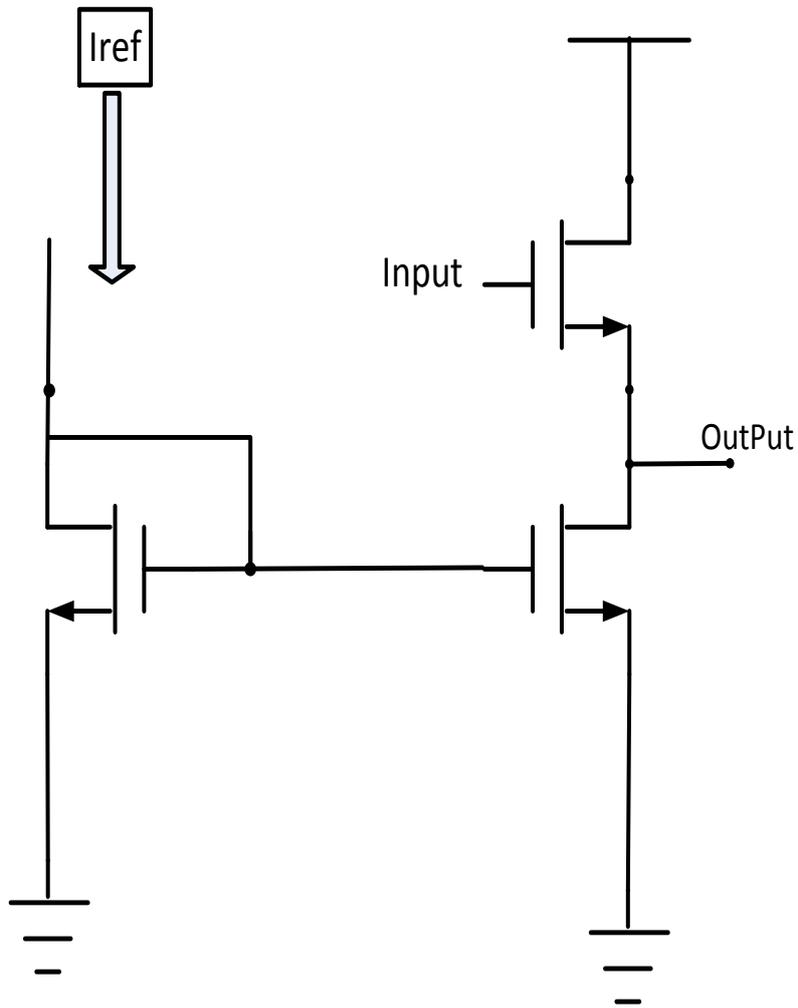


Figure 11.11.25 source follower topology

In designing the source follower we had to choose carefully the transistor transconductance (g_m) and the bias current so that we can get a minimum attenuation for the input signal.

The output pole of the source follower is given by

$$output\ pole = \frac{g_m}{C}$$

Where C is the capacitance seen from the output node, and by this equation we deduce that increasing the transistor g_m by increasing its width will increase the output bandwidth but will also increase the transistor parasitic capacitance so it will decrease the bandwidth rather than increasing it so we had to choose the optimum g_m for the source follower and that will be deduced from the next figure

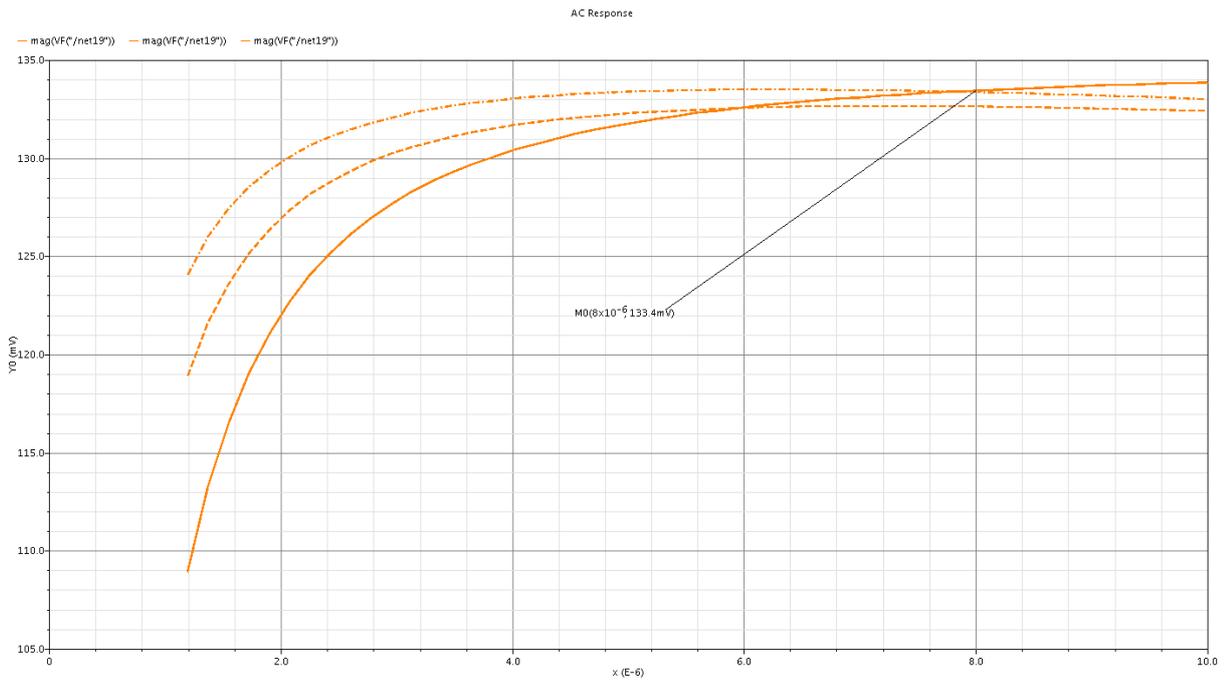


Figure 11.11.26 source follower output vs. transistor width

Accordingly we choose the source follower gm for a certain value of the bias current.

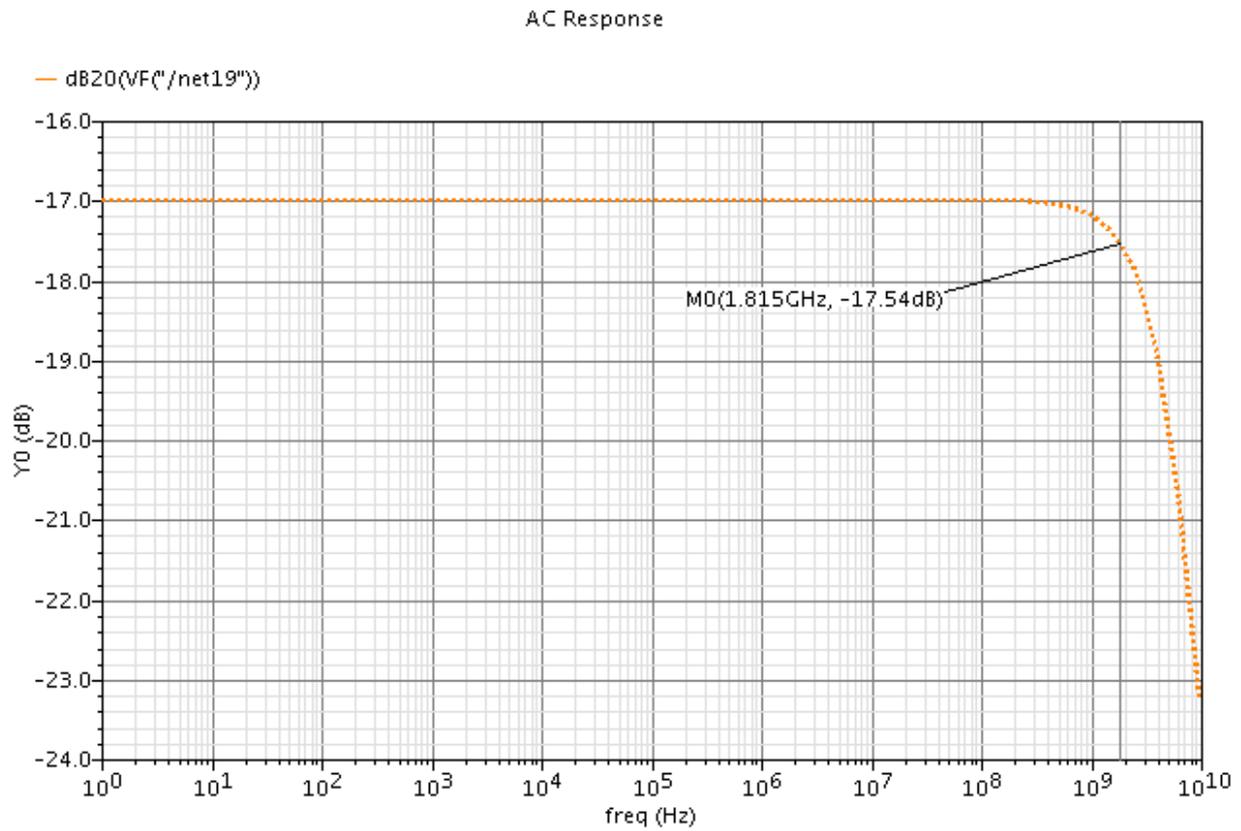


Figure 11.11.27 source follower output bandwidth

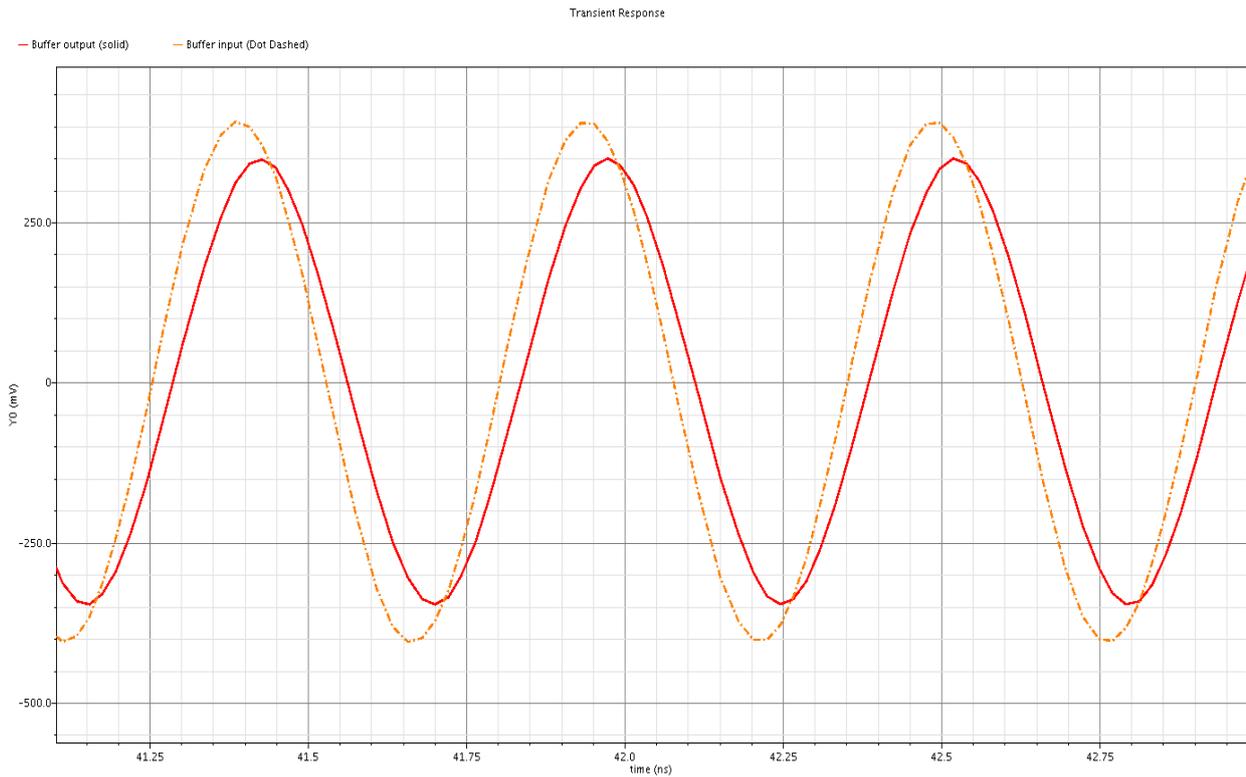


Figure 11.11.28 input (dotted) & output (solid) of the buffer

11.5.5 Calibrator design

Due to the fact that the VCO output carrier frequency varies with process variation and in order to provide a sharp carrier frequency so a frequency calibrator is a must.

The counter based frequency calibrator depends on taking two input signals which is the divider output and the crystal oscillator output and according to the difference between them it switch on and off the capacitor bank switches till the divider output frequency become nearly equal to the crystal oscillator frequency

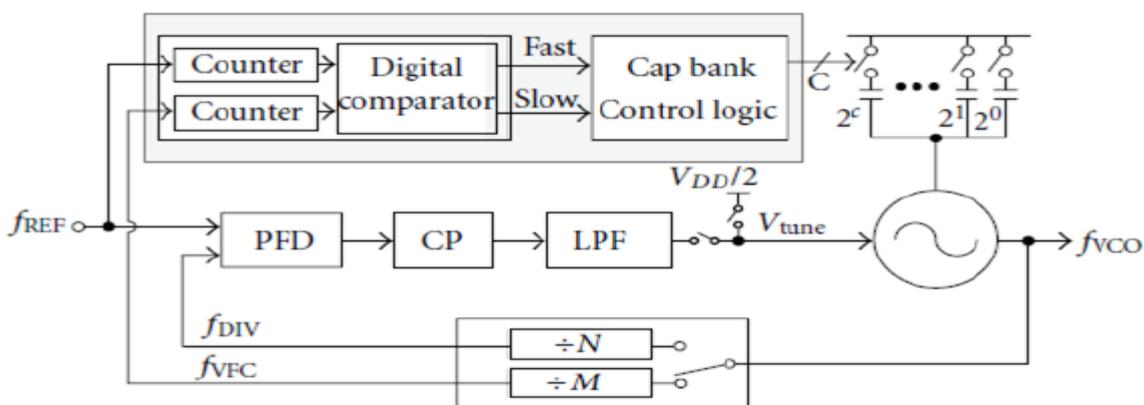


Figure 11.11.29 counter based frequency calibrator

11.5.5.1 Calibrator mechanism

As shown above the frequency calibrator count the input signal crosses with a certain threshold and that done by a function called “Cross” which return the number of crosses the input did with a certain threshold so the input frequency with higher frequency can be easily determined and accordingly it switch on & off the capacitor bank switches till the two inputs become nearly equal and the following figure show the output frequency calibration after the calibrator is added.

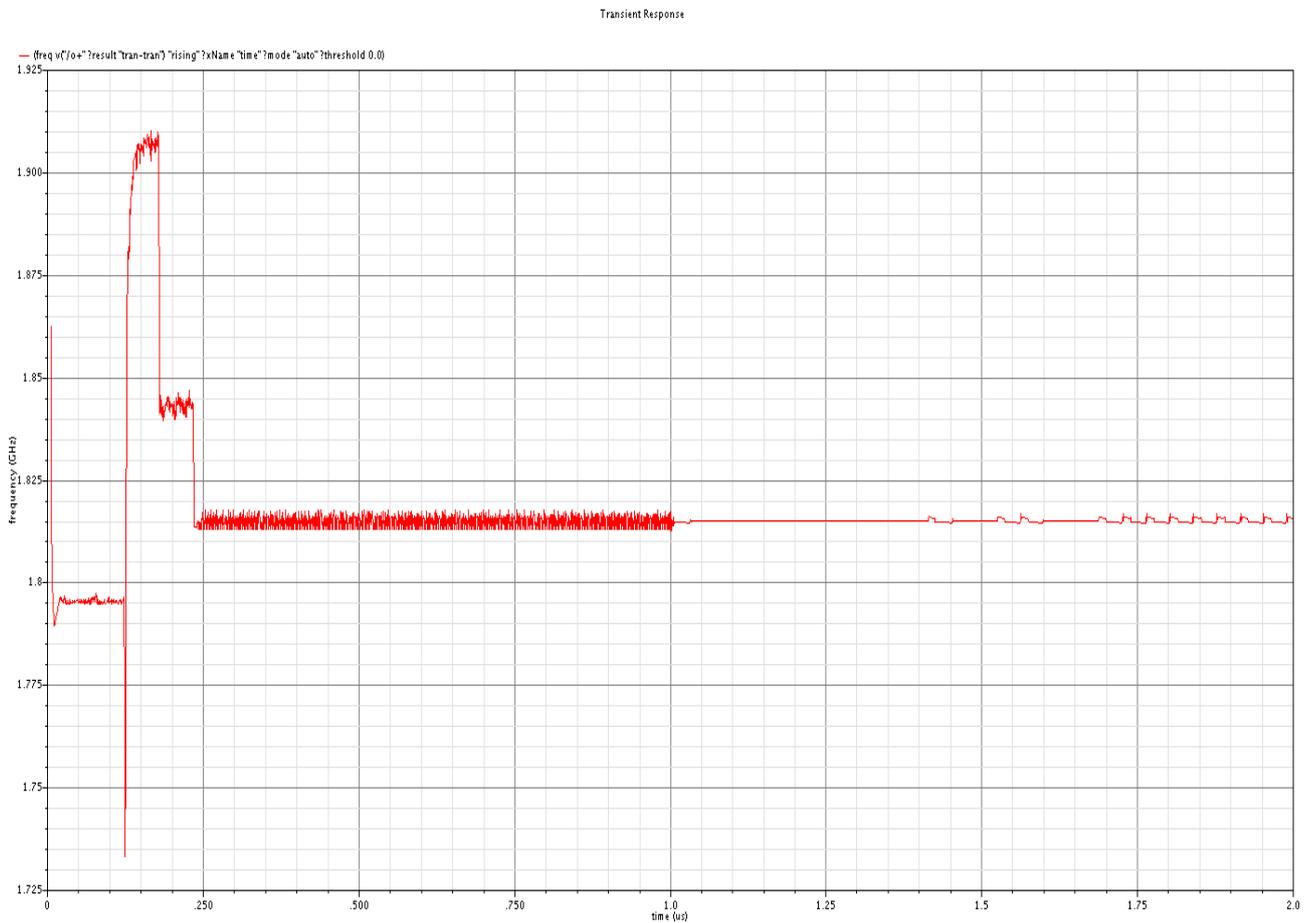


Figure 11.11.30 output frequency calibration

11.6 Important simulations

11.6.1 Transient output

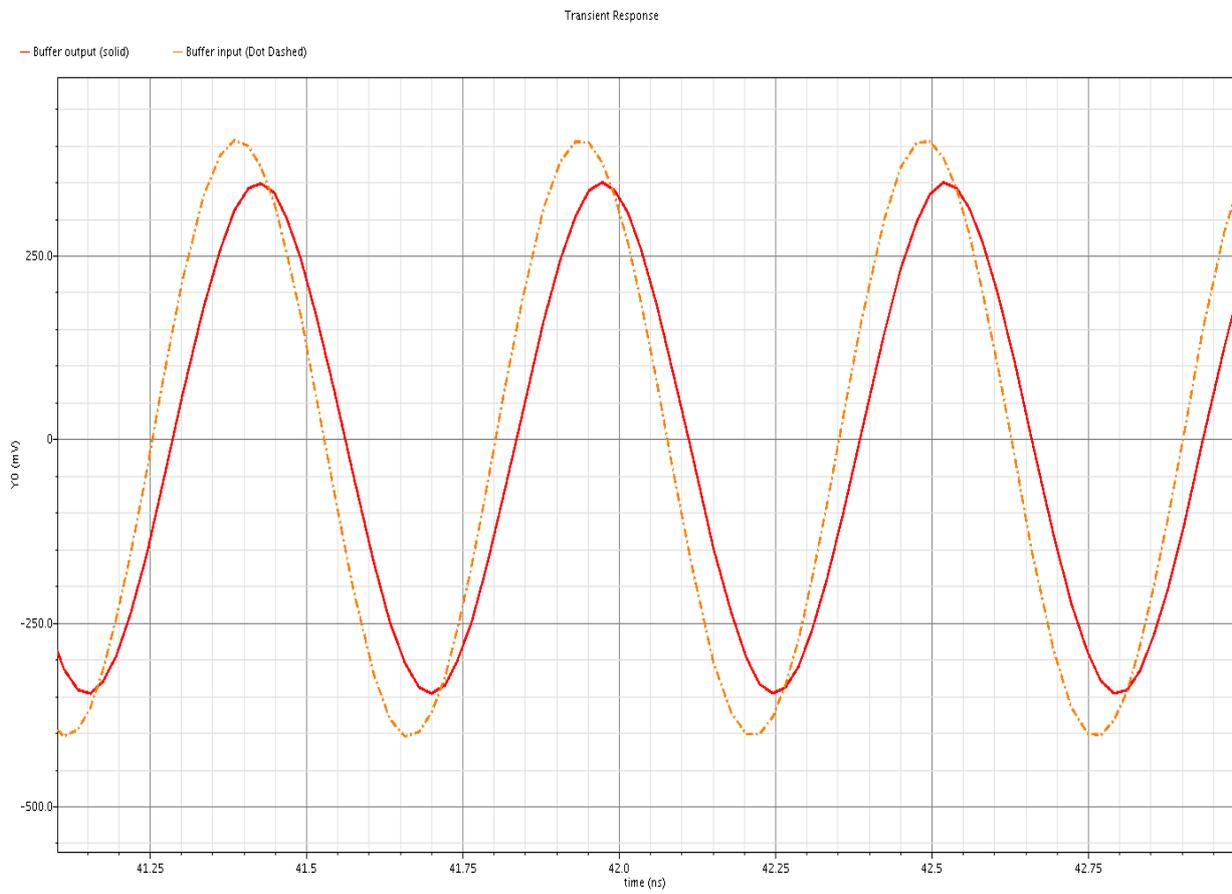


Figure 11.11.31 Buffer input (dotted) & buffer output (solid)

11.6.2 Tuning range

The following figure show the covered range and its variation with corners (SS, FF) and it can be seen obvious that the covered range shift down ward in SS-corner and that due to the fact that the capacitance value increase with SS-corner while in FF-corner it shift up-ward and that due the fact that the capacitor value decrease in FF-corner

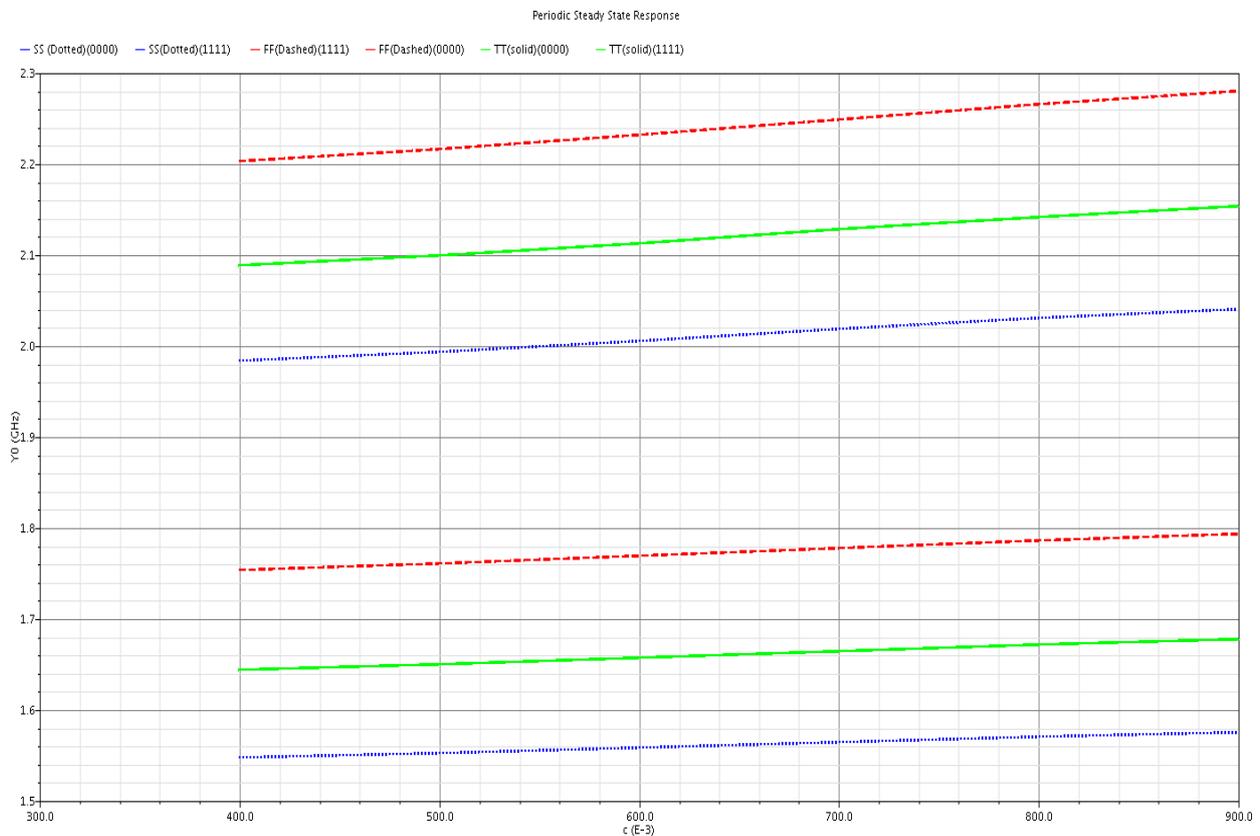


Figure 11.11.32 covered frequency range in FF, TT, and SS

And it can be seen also that the required frequency range (1815 MHz \rightarrow 1865 MHz) is covered in all of these corners.

11.6.3 VCO gain (Kvco)

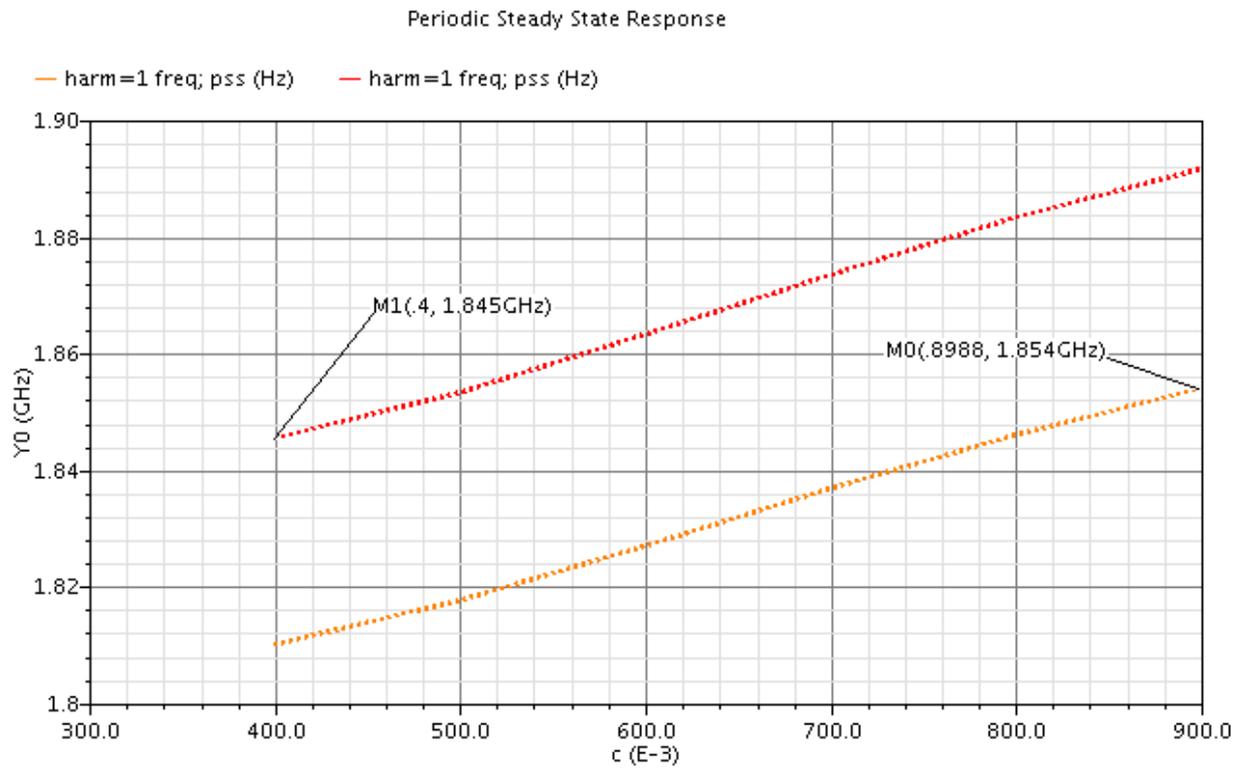


Figure 11.11.33 tuning curve sub-band for the required range in TT

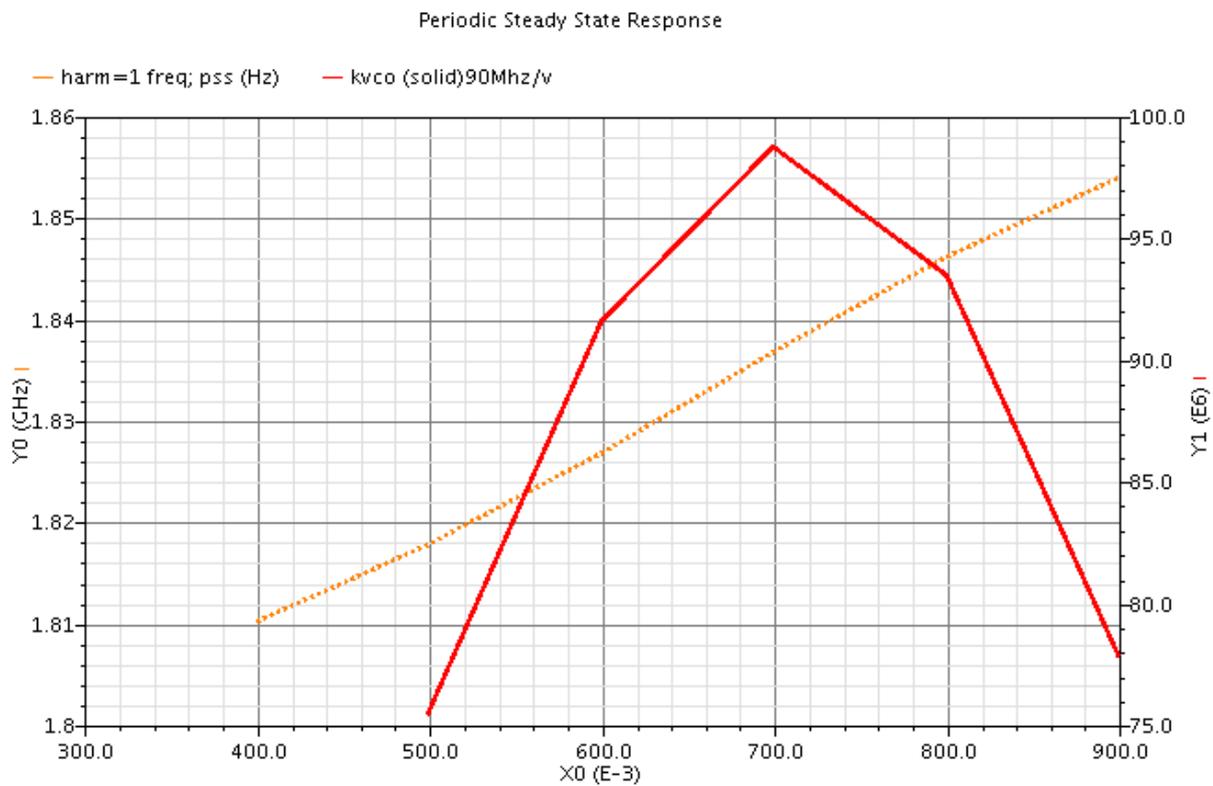


Figure 11.11.34 tuning curve (dotted) & VCO gain for sub-band 1 (TT)
average Kvco = 90Mhz/v

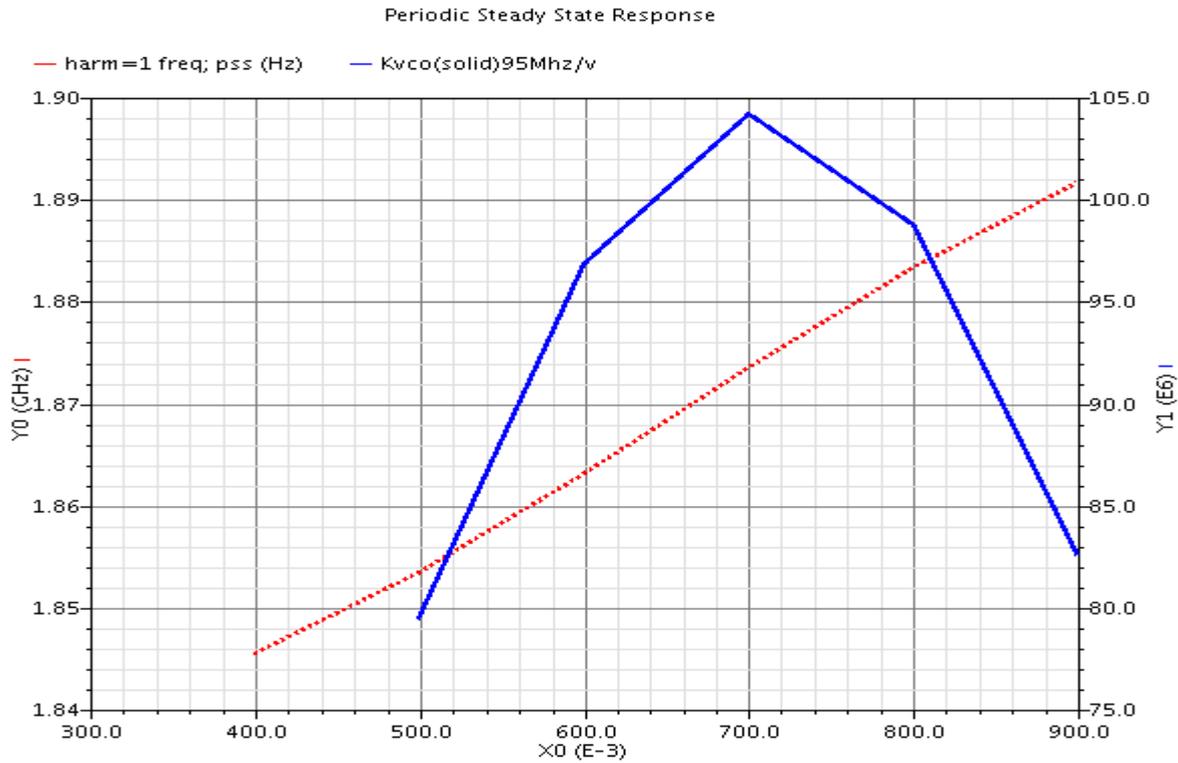


Figure 11.11.35 Tuning curve (dotted) vs. VCO gain (solid) for sub-band 2 average KVCO = 95 MHz/v

Thus the kvco varies across the tuning range from 90MHz/v \rightarrow 95MHz/v.

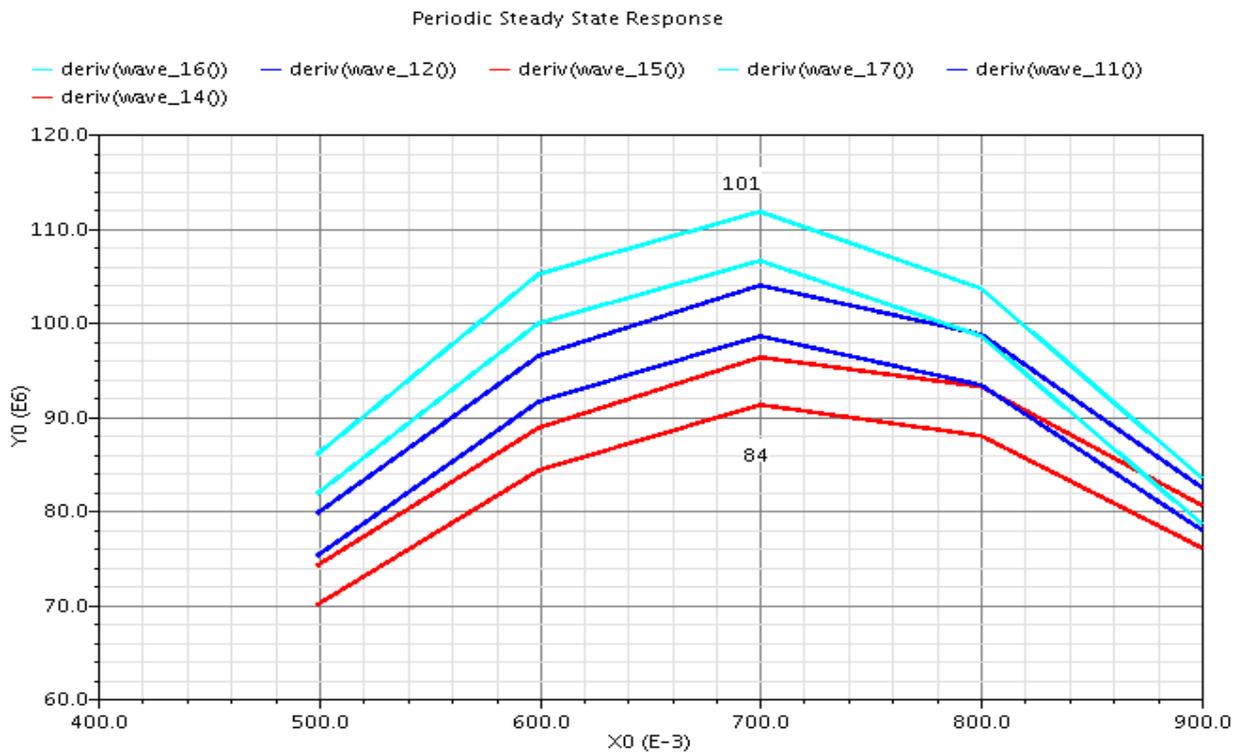


Figure 11.36 KVO variation with tempretur

11.6.4 Phase noise

The following figures show the phase noise and its value across corners (SS, FF) with temperature variation (-40^o,27^o,85^o) while its result across the rest of corners is tabulated below.

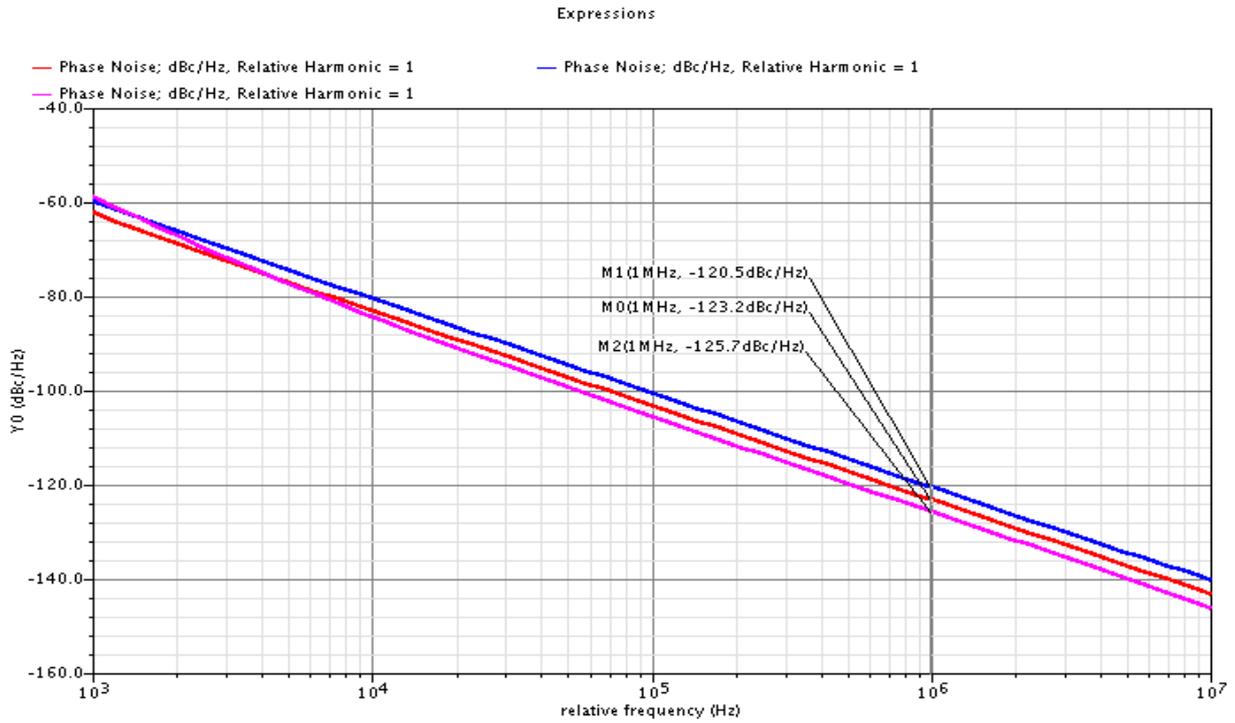


Figure 11.11.37 P.N.(TT)

-123.2 dbc/hz @ 27^o
 -120.5 dbc/hz @ 85
 -125.7 dbc/hz @ -40

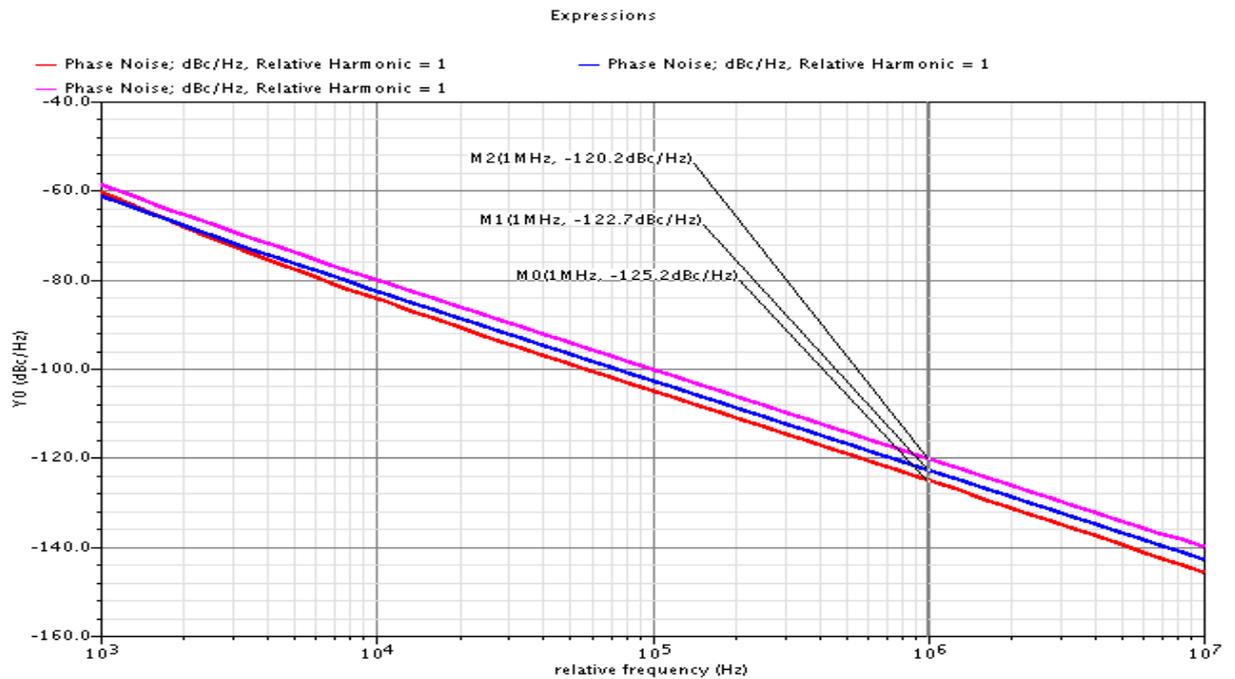


Figure 11.38 PN (SS) -120.2dbc/hz(85^o),-122.7dbc/hz(27^o),-125.2dbc/hz (-40^o)

Expressions

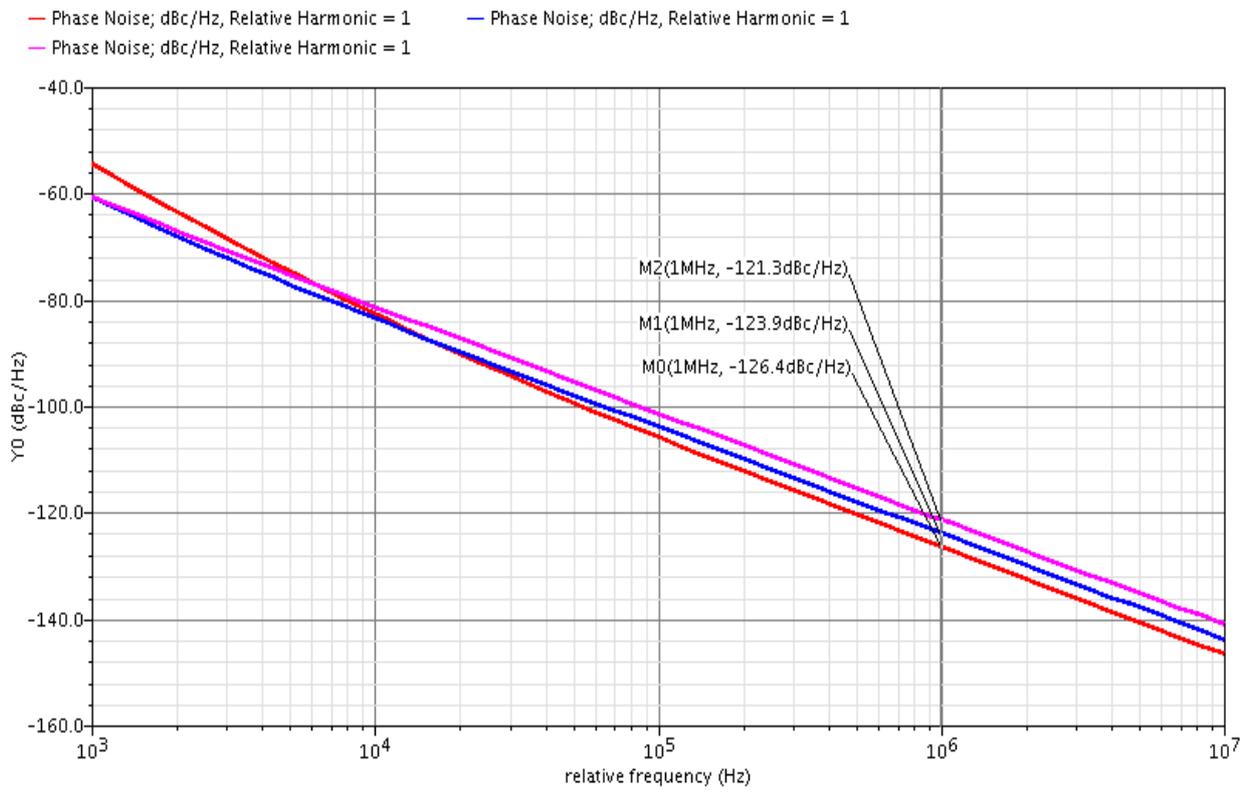


Figure 11.11.39 PN(FF)
-121.5 dbc/hz (85)
-124.5 dbc/hz (27)
-126.3 dbc/hz (-40)

11.6.5 16 covered sub-bands

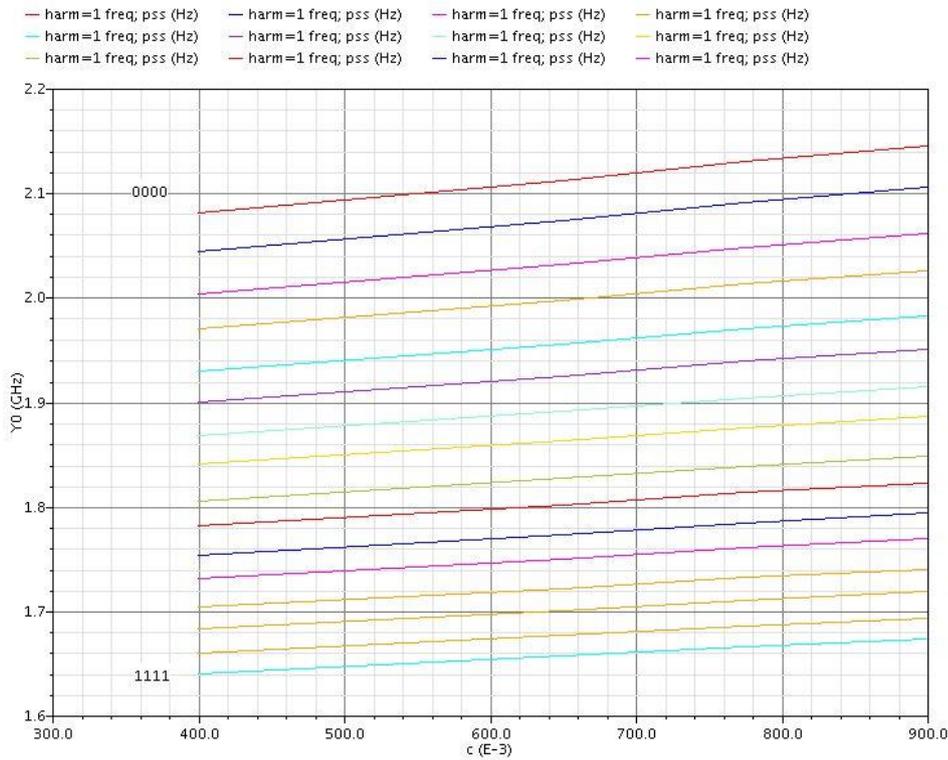


Figure 11.40: covered sub-bands

11.7 Summary of the result

spec	Required	Min.	Typical	Max.	units
Centre frequency	1.834Ghz	1.834 (with calibrator)	1.834	1.834 (with calibrator)	GHz
Tuning range	52		52		MHz
VCO gain (Kvco)	100	84 TT,85 ⁰	95 TT,27 ⁰	101 TT,-40 ⁰	MHz/v
Phase Noise (1M)	-107	-120.2 SS,85 ⁰	-123.3	-126.3 FF,-40 ⁰	dbc/hz
Buffer output	600	580 @ 85 ⁰	750	850 @ -40 ⁰	mv
Maximum load	100f		100f		
Operating temperature	-40 ⁰ to 85 ⁰	-40 ⁰	27 ⁰	85 ⁰	Degree celsuis

Table 11.17 summary of the result

11.8 Design schematics and test bench

11.8.1 VCO core, Buffer, capacitor bank

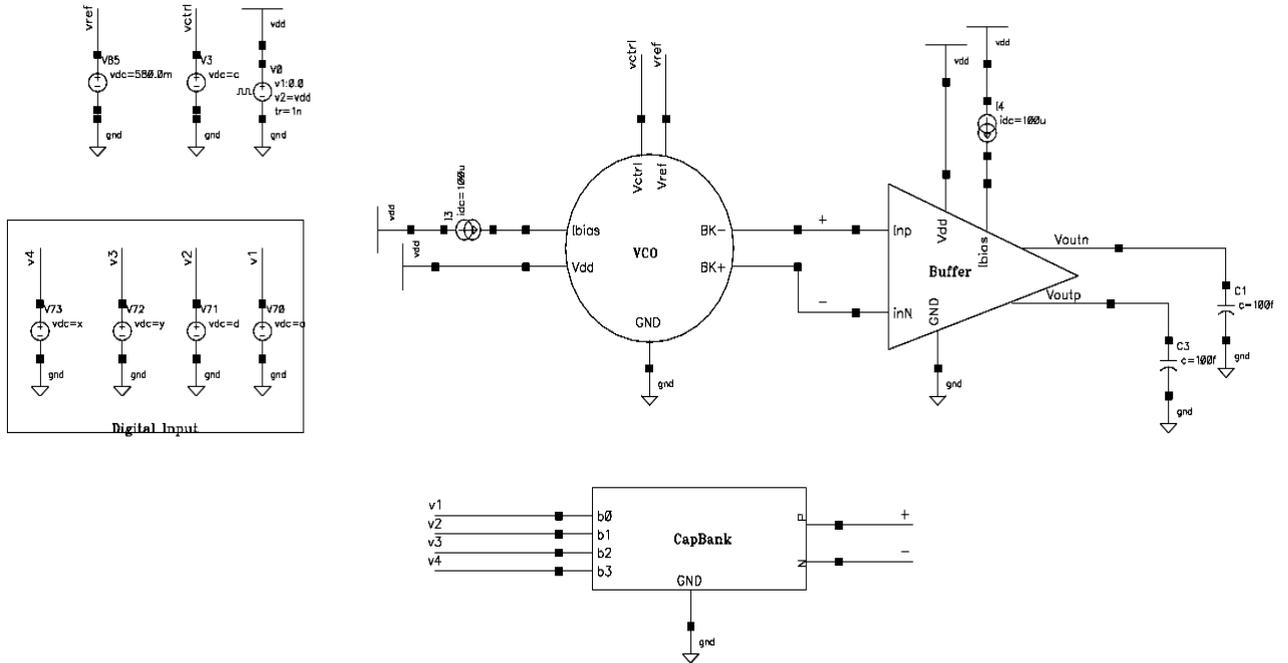


Figure 11.11.41 VCO core & Buffer & cap bank

11.8.2 VCO core schematic

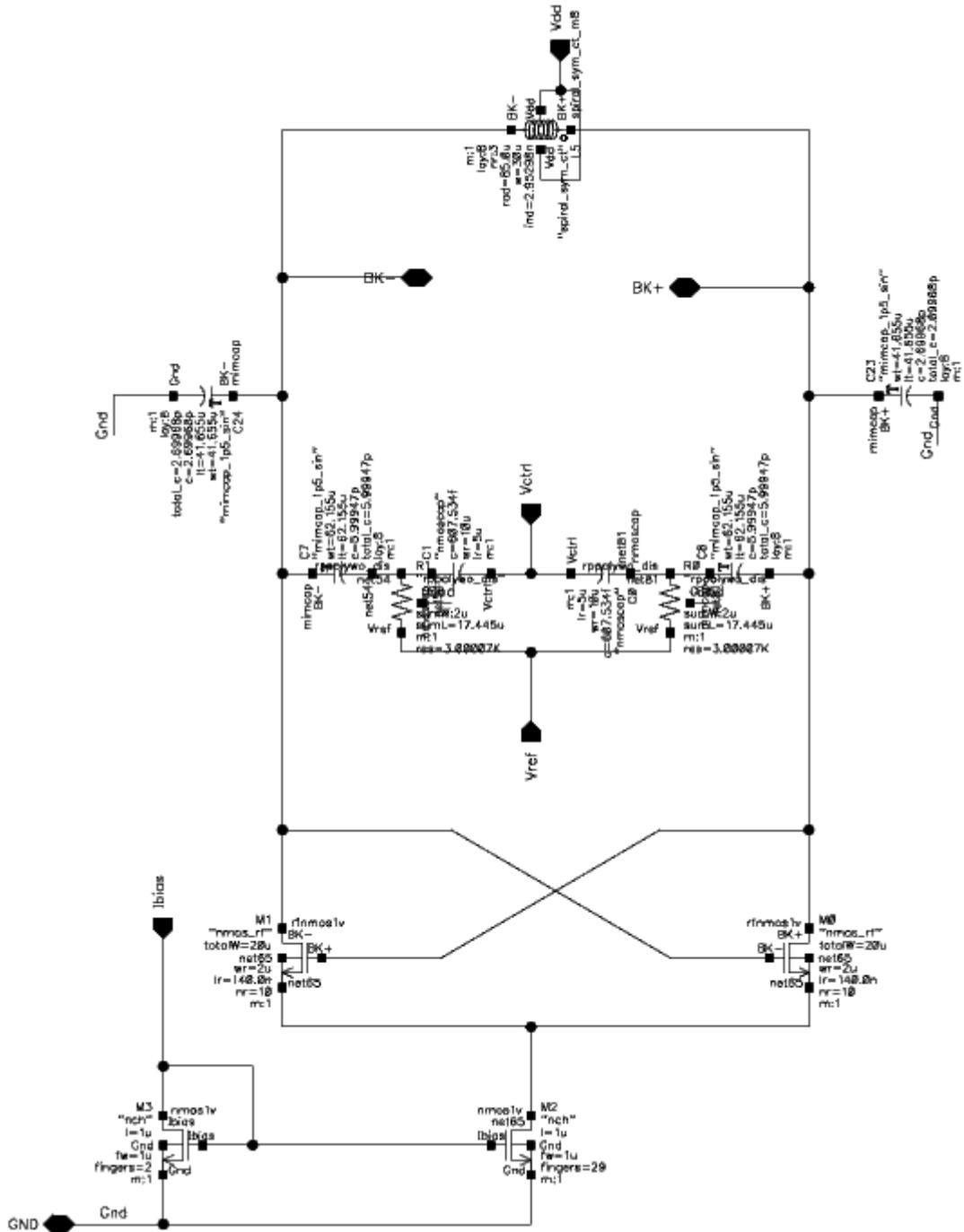


Figure 11.11.42 VCO Core schematic

11.8.3 Capacitor bank schematic

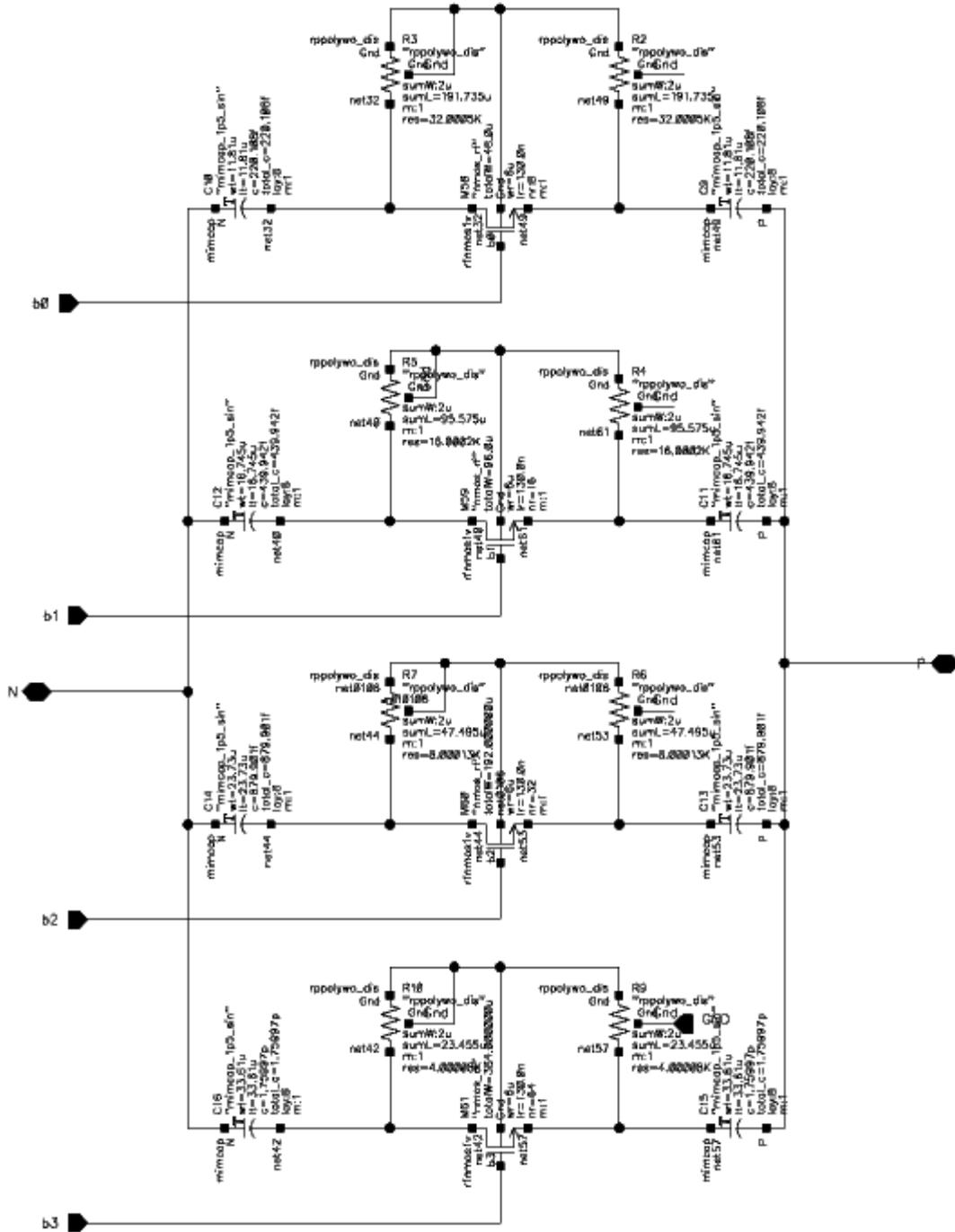


Figure 11.11.43 Capacitor bank Schematic

11.8.4 Buffer schematic

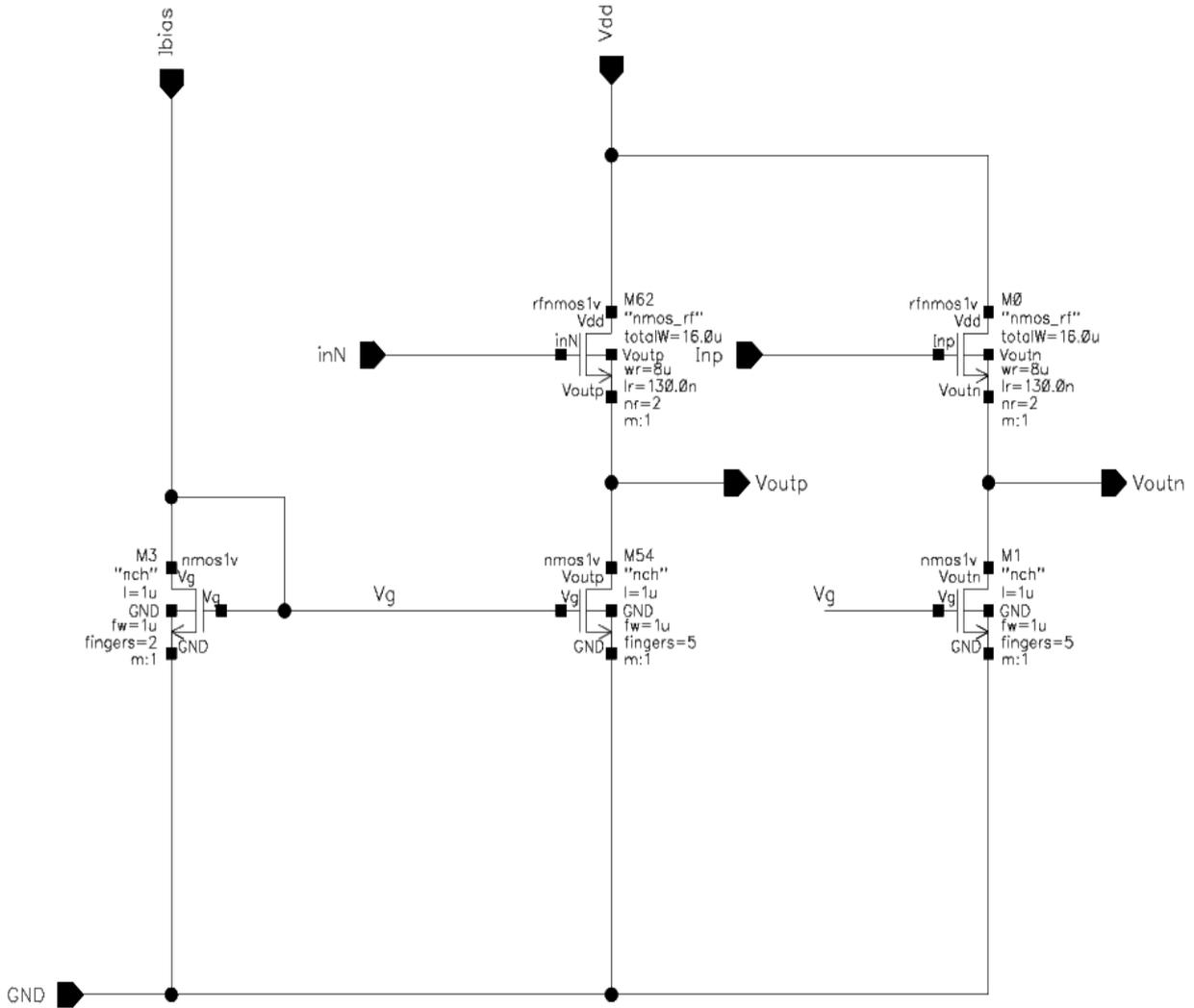


Figure 11.11.44 Buffer Circuit

11.9 References

- [1]. Razavi, Behzad, "RF Microelectronics". McGraw-Hill, New York, 1998.
- [2]. E. Hegazi, J. Rael and A. Abidi, The Designer's Guide to High-Purity.
- [3]. Oscillator <http://ocw.mit.edu/OcwWeb/Electrical-Engineering-and-Computer-Science/6-776Spring-2005/CourseHome/index.htm>.
- [4]. Design of CMOS LC voltage controlled oscillators-thesis-Chetan Shambhulinga Salimath University, India, 2002 December 2006.
- [5]. Maria del Mar Hershenson¹, Ali Hajimiri¹, Sunderarajan S. Mohan¹, Stephen P. Boyd, Thomas H. Lee, "Design and optimization of LC oscillators", Nov. 1999.
- [6]. Pietro Andreani and Sven Mattisson, "On the Use of MOS Varactors in RF VCO's", IEEE JOURNAL OF SOLID-STATE CIRCUITS, VOL. 35, NO. 6, JUNE 2000
- [7]. Bin Wan, "A design and analysis of high performance voltage controlled oscillators", thesis, August 2006.
- [8]. Nobuyuki Itoh, "A Study of Capacitor-Coupled Varactor VCO to Investigate Flicker Noise Up-Conversion Mechanism", IEEE International Workshop on Radio-Frequency Integration Technology, Nov 30-Dec 02, 2005, Singapore.
- [9]. Francesco Svelto, Stefano Deantoni, Rinaldo Castello, "A 1.3 GHz Low-Phase Noise Fully Tunable CMOS LC VCO", IEEE JOURNAL ON SOLID STATE CIRCUITS, VOL. 35, NO. 3, MARCH 2000.
- [10]. Liheng Lou^{1,2}, Lingling Sun^{2,*}, Haijun Gao² and Jincai Wen, "A 0.68-1.65GHz CMOS LC Voltage-Controlled Oscillator with Small VCO-Gain and Step Variation" 2011.
- [11]. Julien Mira¹, Thierry Divel¹, Serge Ramet¹, Jean-Baptiste Begueret¹, Yann Deval, "Distributed MOS Varactor Biasing for VCO Gain Equalization in 0.13µm CMOS Technology" 2004 IEEE Radio Frequency Integrated Circuits Symposium.
- [12]. Li Wenyuan, Li Xian, and Wang Zhigong, "LC voltage controlled oscillator in 0.18-µm RF CMOS", Vol. 32, No. 11 Journal of Semiconductors November 2011.
- [13]. Sung Tae Moon, "Introduction to RF VCO Design", Nov. 2004.
- [14]. Samuel Jia-Chi Chieh, "VCO", March 2006
- [15]. Jongsik Kim, Jaewook Shin, Seungsoo Kim, and Hyunchol Shin, "A Wide-Band CMOS LC VCO With Linearized Coarse Tuning Characteristics", IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—II: EXPRESS BRIEFS, VOL. 55, NO. 5, MAY 2008

12 Frequency Divider

12.1 Introduction:

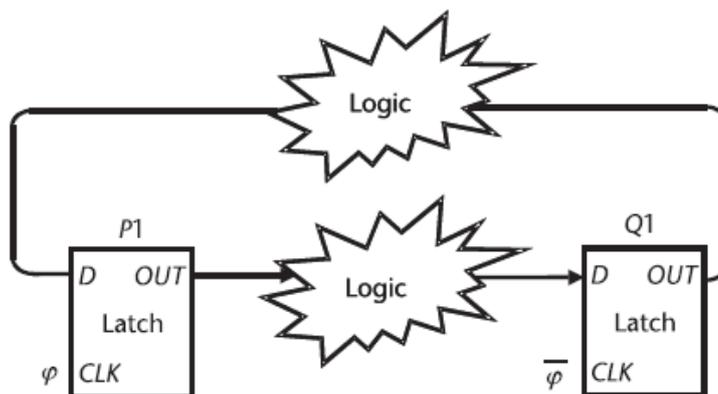
Frequency Divider is a device which performs frequency division of the signal in the PLL- based frequency synthesizer (the Programmable or Feedback divider) to reach the locking mode as it divides output from VCO and Feed it back to the PFD to compare it with the reference signal from crystal oscillator and also the output divider(fixed divide by 2) used to support quadrature signals to serve A mixer or PA (needs any channel just divide by 2) as For the ISM band PLL, Transmitter and Low IF Receiver VCO frequency needed = $2f_{Lo}$ frequency used at mixer .so it is designed at the highest frequency of the PLL .

This work presents A programmable Feedback frequency Divider with a CML as 1st Divide by 2/3 circuit and other as CMOS Dividers to achieve a division ratio from 62 to 72 a CML to Cmos circuit is interfacing these two stages .we use only one CML as Max input frequency is 1.83 GHZ after 1 stage max output is about 915 MHz which is less than max operating frequency of CMOs at our 130 n technology (with reasonable dynamic power as less frequency) ,it consumes about 1.9 mA and noise floor -154 dBc/HZ as noise floor , it is well tested across corner analysis. The second implemented divider is fixed divider at output to generate IQ signals divide by two with 1.5 mA current consumption and 1.8 degree IQ mismatch and stable across corners .The sections are divided as follows: 12.2 general consideration discussing concepts that are used in frequency Dividers, 12.3 System Topology survey, 12.4 circuit level survey , and 12.5 Design methodology.

12.2 General Consideration:

A **frequency divider** also called a **clock divider** or **prescaler** in its basic implementation is sequential logic circuits (counters). They are made up of logic gates and delay units (Flip-Flops and Latches). It divides the clock signal. Generally, there will be two sets of latches in each circuit, the 1st set operates with CLK while 2nd operates with CLKNOT as shown in Fig.12.1

In Asynchronous Divider the input signal of the divider feeds the first flip-flop, which triggers the second and so on. To achieve a low power design, it is desirable to use an asynchronous divider structure to minimize the amount of circuitry operating at high frequencies. The dual-modulus approach achieves such a structure, and has been successfully used in many high speed, low power



designs

Figure 12. 1: Synchronous Divider

12.2.1 Fractional N PLL:

It uses sigma-Delta to produce fractional division Ratio. Fractional-N synthesizers work by periodically changing the division ratio from N to $N+1$ and back such that the average is $N + F/M$ where $0 \leq F < M$; N, F and M are integers. For example, if N is 5 for 99 cycles and 6 for one cycle, 5.01 is the average division ratio.

12.2.2 Divider Phase noise

Consider the PLL that multiplies the reference frequency by M . We want to simulate the contribution of part of the divider, for example the first divide-by- n , to the output phase noise. The VCO oscillates at frequency f_{vco} and the divider output is a periodic waveform of frequency. We consider that the next stage is triggered when the divider output crosses some threshold voltage V_{th} , which for CMOS gates is usually close to half of the supply voltage. Because of device noise and possibly from interference from other on chip signals the edge of interest is noisy, that is it triggers the next stage at some time instant different than the ideal. This time error is a discrete time random process, and we will denote its power spectral density (PSD) with $S_{t, f_{vco}/n}(f)$, we can get the results [12] $S_{t, f_{vco}/n}(f) = S_{If}(f) + S_w(f)$ so PSD is periodic in frequency with period $= \frac{f_{vco}}{n}$

They are also even functions of frequency

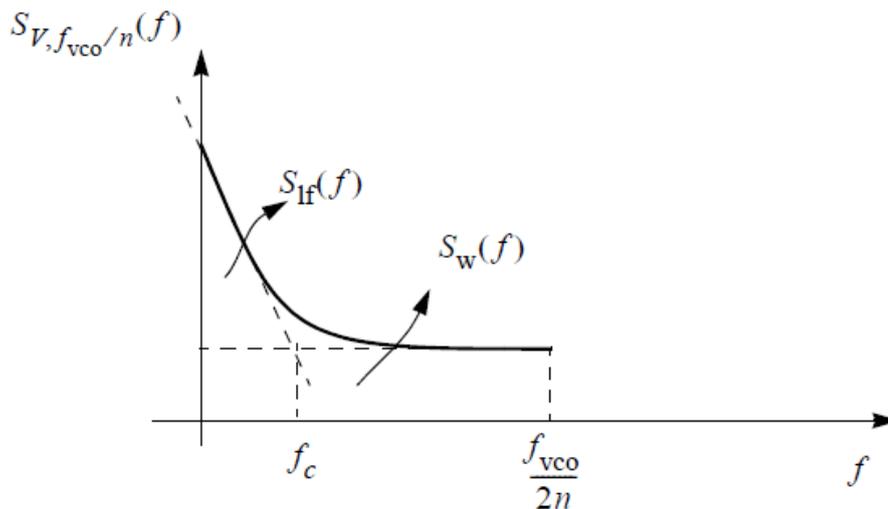


Figure12.2: A noise spectrum at the output of the divider

12.2.3A Modular Architecture

It is a Commercial option and power scalable architecture for low power programmable frequency dividers. (Key Circuit: 2/3 divider cell).

Short time to market demands architectures providing easy optimization of power dissipation, fast design time and simple layout work.

Other Advantages are: Miniaturization, low cost, high reliability, low power dissipation and good EMC (electromagnetic compatibility) and long battery life time where Divider complexity and high operation frequency leads to high power dissipation so it needs the simpler one. The Choice of the divider Architecture is essential for achieving high reusability of existing building blocks, low power dissipation and high design flexibility. A modular Architecture complies with these requirements.

12.3 Architecture Review

The frequency divider can be categorized as: Fixed and Programmable divider Fixed Dividers: has fixed division Ratio, it can be used in the Feedback dividers before the programmable one to relax there power requirements and maxing operating frequency Requirements, also it can be used in the output of the PLL to generate quadrature output that are needed in the mixer, figure 12.3 &12.4 shows an example of fixed division ratio synchronous divider.

The multi-modulus divider presented in this chapter is an extension of the popular Dual-modulus topology; the Swallow counter contained in the dual-modulus design is Replaced with cascaded divide-by-2/3 sections. This section provides further details of the multi-modulus divider architecture used in the prototype. We begin with a general description of the dual-modulus approach, and then explain the extension of this method to multi-modulus structures. Finally, architectural details are provided of the divide-by-2/3 sections used in the divider design

The structure in Fig.12.3 consists of two level-triggered latches that are cascaded Within a negative feedback loop. Maximum operating speed of this structure is determined by the propagation delay of the input of each latch to its respective output . When the latch is made transparent by the clock. Specifically, the input period, TIN, Must satisfy the relation:

$$\frac{TIN}{2} - Ts > delay1 = delay2$$

Where Ts is defined as the setup time of each latch.

The structure in Fig.12.4 is a divide by 3 cells.

Programmable Dividers can be:

- 1) Digital : MultiModulus Divider, Pulse swallow Divider(Dual Modulus)
- 2) Analog frequency dividers are less common and used only at very high frequencies like ILFD and miller dividers

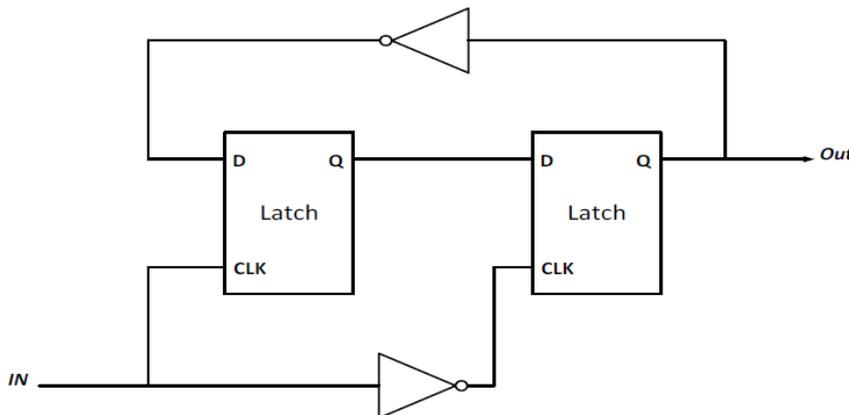


Figure12. 3: Divide by two

Basic building Blocks are Divide by 2/3 Cell it operates as follows If Div=1 it ÷3 and if Div=0 it ÷ 2, When Div=1 Will be explained Later.

-Rule of thumb: Dual modulus dividers are about factor of 2 slower than ÷2

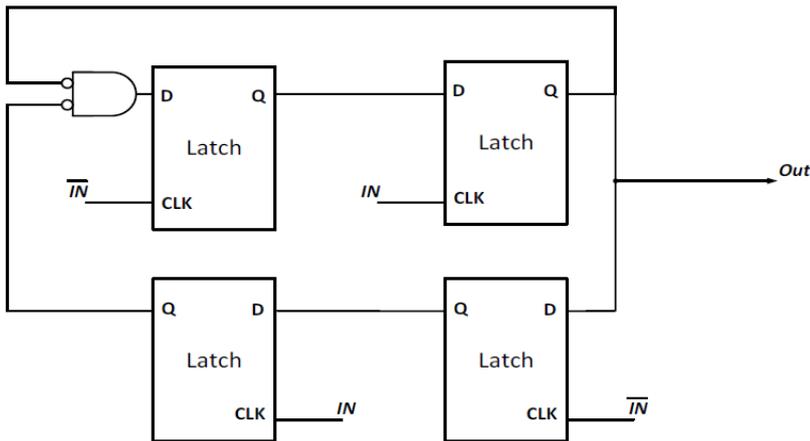


Figure 12.4: Divide by three circuits

12.3.1 PULSE SWALLOW DIVIDER

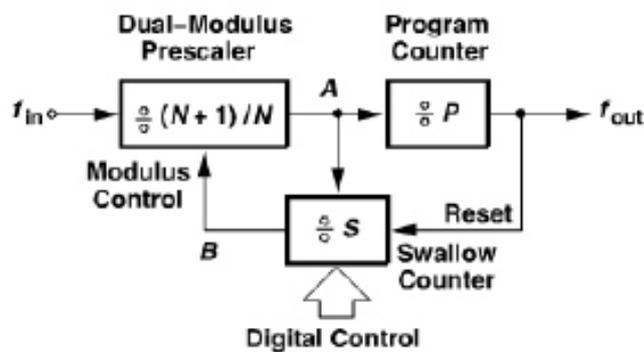


Figure 12.5: Fully Programmable Divider based on Dual Modulus Prescaler (PULSE SWALLOW)

A common Realization of Feedback divider that allows unity steps in modulus is called "pulse swallow divider" shown in figure 12.5[1], it consists of:

1) A "dual-modulus prescaler": counter provides division ratio of $N+1$ or N according to the logical state of its modulus control B input.

2) A "swallow counter": this circuit this circuit divides the input frequency by S , $S \geq 1$

in unity step by digital input this counter controls the modulus of the presale and also has arrest input.

3) A "program counter" divides by P , when the program counter fills up after it counts P pulses at input it resets the swallow counter.

-If all 3 counters start from reset Presale counts $N+1$ pulses giving 1 pulse at A , program counter counts output pulses of the prescaler .till the counter Fills up (get S pulses at its input) so main input receives $(N+1)*S$.the swallow counter then changes the modulus to N and begins from zero again and the program counter needs $p-s$ pulses to fill up, Now the prescaler divides by N producing $p-s$ another pulses to fill up the program counter. In this mode main input receives $N (p-s)$ pulses so

$$D_{tot} = NP + S, \quad p \geq S$$

$$f_{out} = (NP + S)f_{ref}$$

it has Lack of modularity than the MULTI-MODULUS Divider , The Programmable counters represents a substantial load at the output Of the Dual Modulus prescaler so power dissipation is

When R_1 is high then Mod_1 will come into play. The case of $R_1=1$ & $R_2=0$ is shown on Fig. 12.11. In this case, the second divider will always divide by two, but the Mod_2 signal will be high every second cycle of F_0 . Thus, for one complete cycle of the output, the first divider will swallow one extra pulse (divide by three once), causing the whole Circuit to divide by five. Likewise, it can be shown that with both inputs set to one, divide-by-seven is achieved as illustrated in Fig. 12.10. In this case, the second Divider always divides by three, and, once again, the Mod_1 signal causes the first Divider to swallow one extra pulse every output cycle of the second divider. This Adds one input period to the period of the output signal. Thus, in this circuit, to

Produce one output clock cycle, a minimum of four input clock cycles is required.

If R_2 is high, the second block causes the circuit to swallow two additional inputs Pulses, and if R_1 is high, the first block will swallow one additional input pulse. Thus, the total output period is given by:

$$T_{out} = T_{in} (R_1 + 2 * R_2 + 4)$$

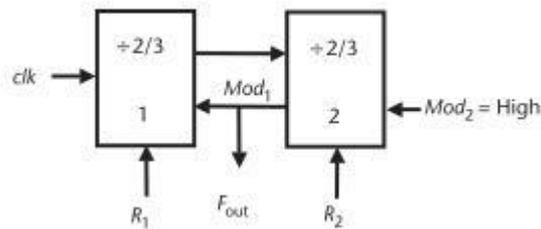


Figure 12.8: two cascaded divide by 2/3 stages

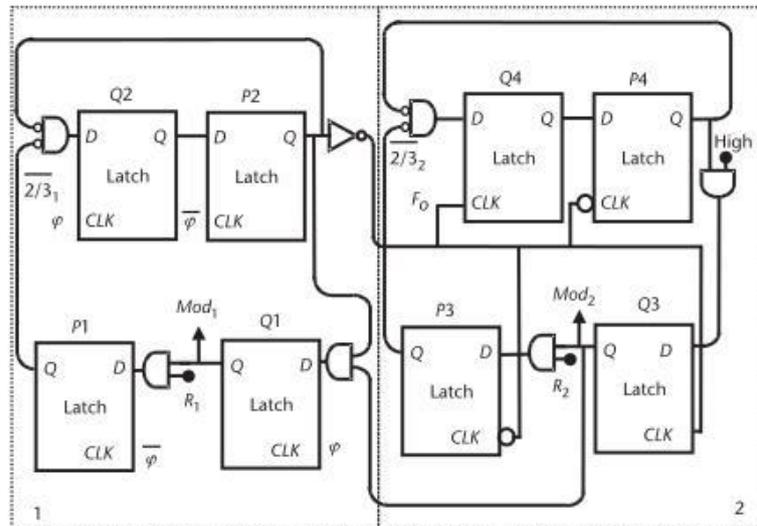


Figure 12.9: A divide by 4/7 circuit made from two 2/3 stages shown at latch Level

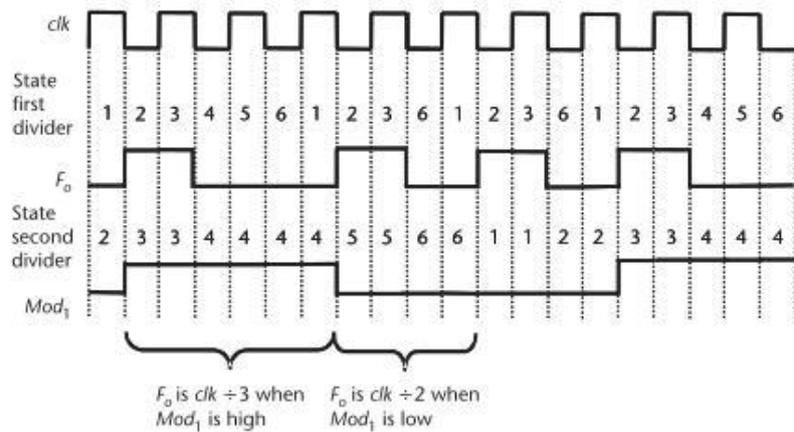


Figure 12.10: Waveforms in divide-by-7 mode (R1 = high and R2 = high)

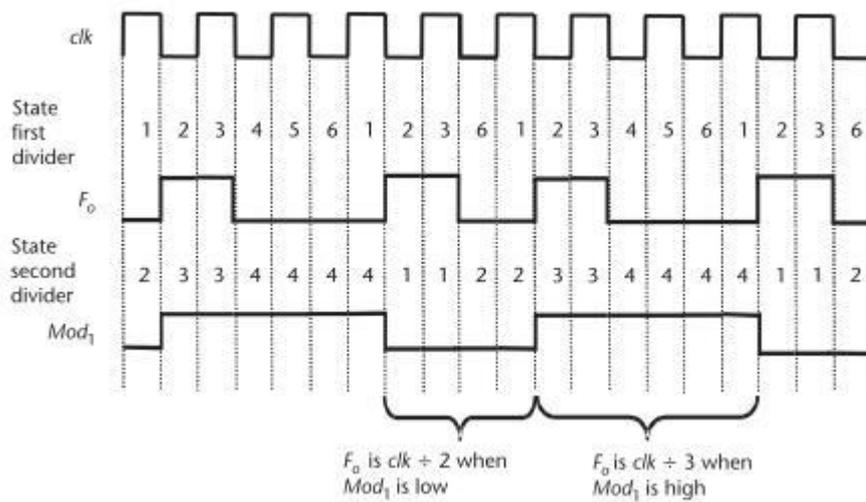


Figure 12.11: Waveforms in divide-by-5 mode (R1 = high; R2 = low).

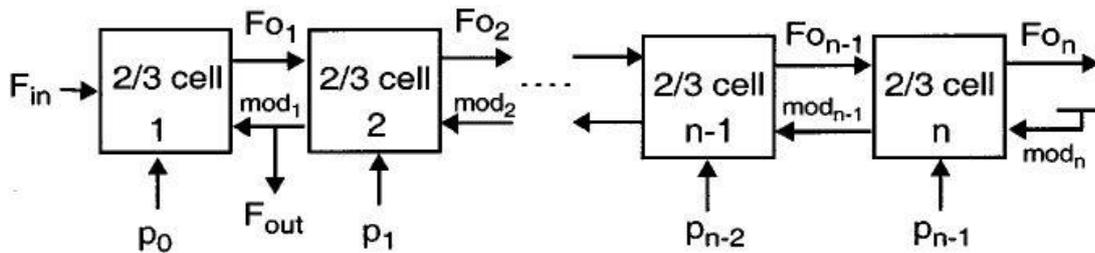


Figure 12.12: MMD Architecture

The maximum operating speed of the divide-by-2/3 structure is less than the Basic divide-by-2 topology since the gating logic adds to the propagation delay From the output of latch 2 to the input of latch 1.

The basic programmable architecture is depicted in Fig.12.12 [3] .The modular structure consists of a chain of 2/3 divider cells connected like ripple counter .this structure is characterized by the absence of long delay loops, as feedback lines are only present between adjacent cells. This “local feedback” enables simple optimization of power dissipation. Another advantage is that the topology of the different cells

In the prescaler is the same, therefore facilitating layout work.

The programmable prescaler operates as follows. Once in a division period, the last cell on the chain generates the signal Mod_n . This signal then propagates “up” the chain, being reclocked by each cell along the way. An active mod signal enables a cell to divide by 3 (once in a division cycle), provided that its programming input is set to 1. Division by 3 adds one extra period of each cell’s input signal to the period of the output

Signal. Hence, a chain of $2/3$ cells provides an output signal with a period of

$$T_{out} = (2^n + 2^{n-1}P_n - 1 + 2^{n-2}P_n - 2 + 2^{n-3}P_n - 3 + \dots + 2P_1 + P_0) T_{in}$$

P_0, P_1, \dots, P_{n-1} are the binary programming values of cells 1 to n . So all integer Division ratio range from: 2^n (all $P_n = 0$) to $2^{n+1} - 1$ (all $P_n = 0$).

Note each cell speed is $1/n$ Less than previous one.

Dual modulus dividers with other moduli can be designed using synchronous dividers as in [2]. Synchronous Dual modulus dividers have limited speed and increase the loading on their input (CLK load), therefore we can use synchronous divide by $2/3$, divide by $3/4$ or divide by $4/5$ as the core of the divider followed by divide by two stages in an asynchronous way leading to improvement in the speed of the whole divider and decrease the loading of the divider on its input. There is another technique to improve the

Speed of dual modulus dividers given in [5] based on asynchronous dividers and phase switching technique.

12.3.3 Generic MMD

The MMD architecture with all $2/3$ cells may not necessarily end up with minimum gate count and power consumption. This section discusses a generic MMD design algorithm, along with the implementation of each cell, which takes the minimum hardware and current [4]. The MMD architecture discussed in the previous section, based on [3], is a special case of the generic algorithm. When the MMD division range is not large compared to its minimum division ratio, the generic algorithm will result in an optimal MMD architecture that is different from the one given in [3]. For the case of a large MMD division range, the generic algorithm leads to the architecture with all divide-by- $2/3$ cells, as discussed above. The generic MMD architecture includes a number of divide-by- $2/3$, dual-modulus cells cascaded with a divide-by- $(P/P + 1)$ dual-modulus cell (P being an integer and $P \geq 2$) in a ripple fashion. Here only the last cell is selected to be a divide-by- $(P/P + 1)$ cell so that all of the division ratios in the required range can be programmed with a unit step increment. If any one of the preceding cells is not a $2/3$ cell, then the unit step increment is not guaranteed. If a step increment other than one is desired, the optimal architecture is to place a fixed-ratio, divide-by- S stage in front of the MMD cells, as shown, so that the MMD has a programmable step size of S . The architecture shown in Figure 12.13 provides the division ratio as:

$$N = (2^{n-1}P + 2^{n-1}R_n - 1 + 2^{n-2}R_n - 2 + \dots + 2R_1 + R_0) * S$$

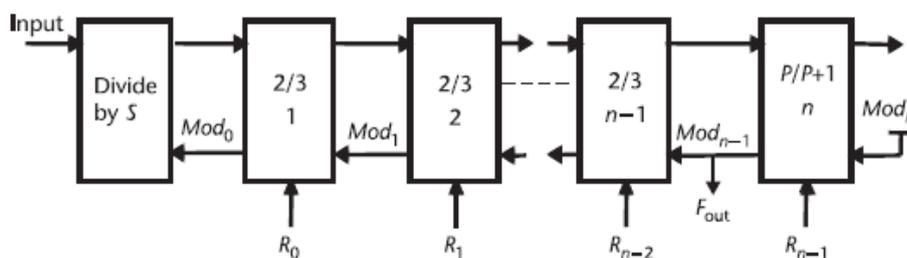


Figure 12.13: A generic MMD architecture

So P increase min division ratios, C_0, \dots, C_{n-1} are MMD control bits , P/P+1 cell increases the Minimum division Ratio & range.

This generic optimize Min current and have smallest area & min. number of control bits than the normal MMD .

Generic Architecture for MMD Design algorithm:

- 1) Required division ratio ($D_{min}:D_{max}$) , #divisor steps (range)= ($D_{max} - D_{min} + 1$)
- 2) If the required range is greater than D_{min} , the MMD is preferred
- 3) Implemented range can be larger than the required range .initially set $M = D_{min}$
- 4) No of required cells $n = \lceil \log_2(D_{max} - M + 1) \rceil$ (take nearest larger integer)
- 5) $P = \lfloor M/2^{n-1} \rfloor$ (take nearest smaller integer)
- 6) If $M/2^{n-1}$ not integer reset $M = P * 2^{n-1}$ and go back to step 4
- 7) If $M/2^{n-1}$ is an integer it is necessary to evaluate the best way achieving lower current consumption .using(a single p/p+1 or 2/3 cell and a $((p/2) / (P/2) + 1)$ cell
- 8) If only 2/3 cells are used Total number of required cells
 $N_{2/3} = \lceil \log_2(D_{max} + 1) \rceil - 1$

12.3.4 Comparison & Conclusion

Table 12.01: Comparison for Different System Level Architectures

Points of comparisons	Pulse Swallow Divider	MMD or Generic MMD
Modularity	Less	More
Programmability	Limited –more if a delta sigma shaper	High
Layout time	More	Less
Design time , complexity	More	Less
Power consumption	More	less
Power optimization	Hard	simple
Division Ratio	NP+S	2^n to $2^{n+1} - 1$

Conclusion: For the Feedback Divider the MMD or Generic MMD is preferred than the PSD as I need to select the least power consumption one and least Time.as shown in Fig 12.14, 12.15 &12.16 [10]

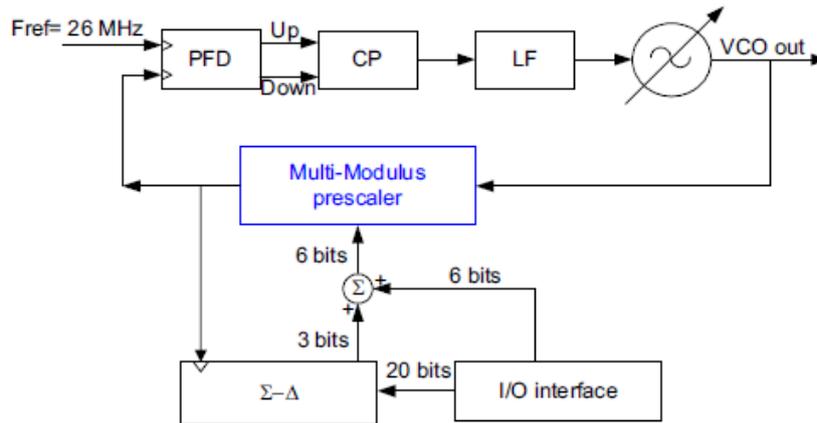


Figure 12.14 : MMD chosen as Feedback Divider with Sigma-Delta interface

Another Fixed Divide by two to be chosen to generate quadrature outputs

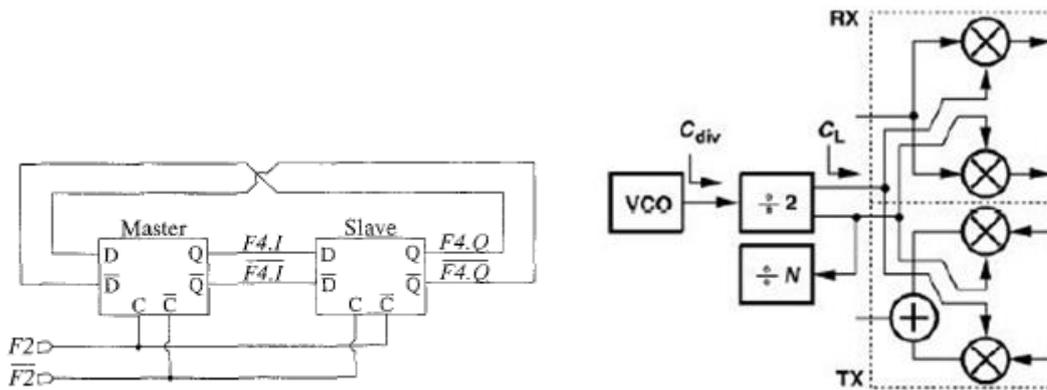


Figure 12.15: quadrature output generator

Figure 12.16: : Fixed Divide by 2 with Mixer and PA.

12.4 Circuit review

The frequency divider can be realized by many logic styles, The choice of Divider topology is governed by: Input swing (available from vco), the input capacitance (presented to the vco) The maximum speed, the output swing (as required by subsequent stages), the minimum speed (dynamic logic versus static logic), leakage current and the power dissipation

With the scaling down of CMOS transistors, many issues now have become a factor in design the different logic styles are: CML, Cmos rail to rail, ECL, SCL, CRL and TSPC.

12.4.1 CMOS Rail to Rail (static)

CMOS rail-to-rail logic swing is the voltage difference between supply rails. by a large amount and hence requires larger charge and discharge time rail-to-rail logic is often slower than the other types of logic like CML. it approx. it doesn't consume DC power

It Consumes Dynamic power = $C_L * F * VDD^2$

So as Freq inc. more power consumed than CML, and also as CL inc. it will consume more power .so it is used for lower-speed applications.

Cross-coupled NAND/NOR (positive feedback) or cross coupled inverters can be used to implement a latch function is shown in Fig.12.17. it has relatively small number of transistors & Robust

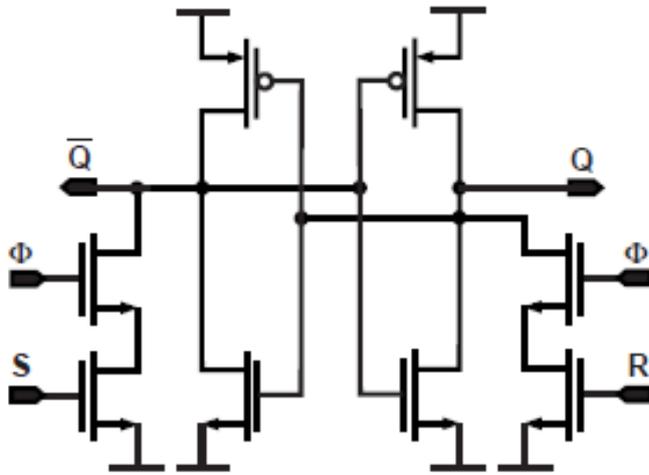


Figure 12.17: Static latches

12.4.2 Current Mode Logic (CML)

CML implements some analog components to compute logic. CML have reduced swing than CMOS rail to rail which is more preferable in high speed applications as the charging and discharging of output nodes takes less time. It is based on current steering between differential pair of transistors which give it high immunity against common mode noise as it couples to both lines simultaneously. The current is constant which leads to less switching noise, a cleaner power, ground rails and less noise injected into the substrate. It is important to switch the pair in digital applications, so a large enough differential Voltage must be applied to the input.

For high-speed applications and for low switching noise, synthesizers do not always use standard CMOS logic, but use CML instead, the differential pair can be used to make an inverter, AND, OR, and XOR gates, can be implemented

The CML or Source coupled logic (SCL) D-latch is given in Fig.12.18 and SCL AND gate

Figure.12.19, 12.20 the power consumption of CML circuit is basically static power consumption

$$P = I_{bias} * V_{DD}$$

In contrast with rail to rail CMOS circuits which consumes theoretically no static power but at high frequency the dynamic power consumed by them is larger than static power of CML. Therefore this is another reason to use CML at high frequencies while at low frequencies CMOS is less in area and power and preferred on CML.

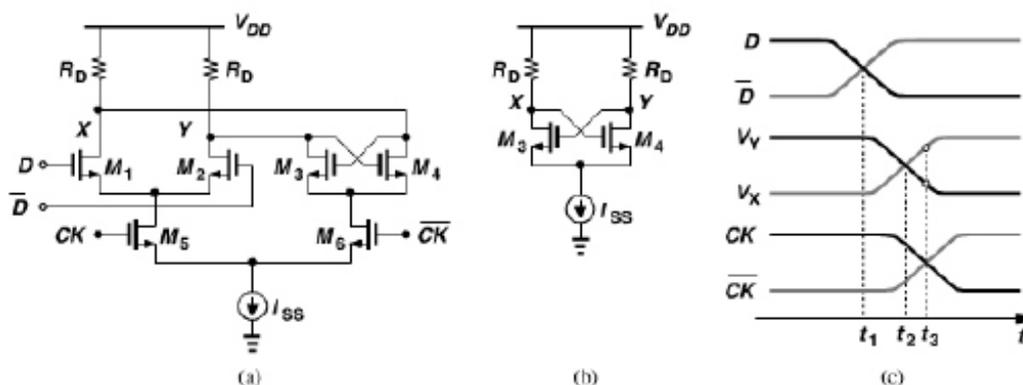


Figure 12.18: a) CML D latch – b) regeneration mode Equivalent circuit - c) waveform

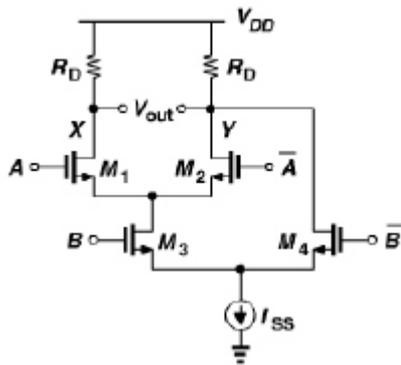


Figure 12.12.19: CML NAND/AND

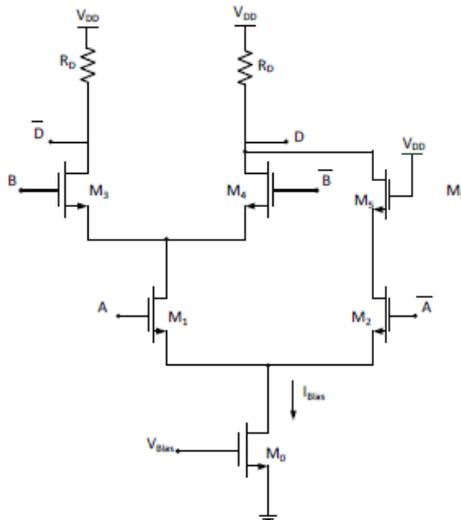


Figure 12.12.20: CML AND combined with Latch

12.4.3 TSPC (True single phase clocking)

It is a type of Dynamic Logic that uses only 1 clock phase instead of the normal one which reduces the problems of clock skews. As shown in Fig. 12.21 TSPC circuits can be used in higher speeds than static CMOS with lower power consumption due to less capacitive load. It has the advantages of higher speed, less clock delay & no static power, no need for differential clock, They take smaller area than CML and don't need biasing network therefore they are simpler than CML but usually they reach less speed due to PMOS stacking and rail to rail swing. While it increases the number of transistors, need full swing (rail to rail) input CLK & not provide quadrature outputs, divider fails at very low clock frequencies (Dynamic logic), they have less immunity to noise sources as they are not differential, also they suffer from high switching noise due to the spiky nature of their supply current. Due to their dynamic nature they suffer from some hazards like charge leakage, charge sharing and glitches. Some of these problems are solved in E-TSPC [8], [9], [11] there is architecture that can run as high as 1.8 GHz

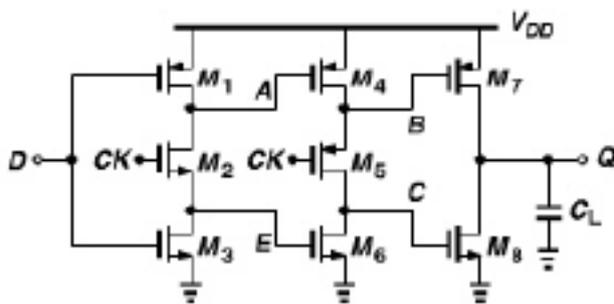


Figure 12.21: TSPC Flip FLOP

12.4.4 OTHER STYLES

Razavi, CRL, Ratioed Logic & inductive peaking dividers they are rarely used [1], [8]

12.4.5 Comparison & Conclusion

Comparison of Power vs. Frequency shown in Fig. 12.22

Table 12.02: Comparison Different Logic styles

Points of comparisons	CML	CMOS rail to rail	TSPC
Nature	Static	Static	Dynamic
Maximum Speed	Highest	Medium	High
Power Dissipation	High at low frequency ,low at high frequency	Low at low frequency ,High at high frequency	Lower than CML at high frequency
Swing	Lower swing	Rail to Rail swing	Rail to Rail swing
Area and Biasing Network	Large area and biasing network	Small area and no biasing network	Small area and no biasing network
Switching noise	Less switching noise(constant I dc)	Large switching noise (current spikes)	Large

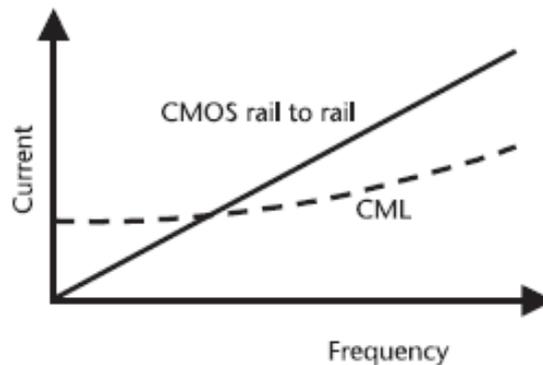


Figure 12.22: Power consumed VS. Frequency

Conclusion:

12.4.5 .A) the quadrature generator:

SPECS Required:

Differential swing $\geq 600\text{mVpp}$ peak to peak differential

Power $< 1.6\text{mA}$, freqs: 600 M to 2 GHZ, I & Q Mismatch < 2 deg, CL= 250 fF.

12.4.5. A.1) the Latches

-**CML:** very low swing, so you can to work in higher speeds, also the current steering is faster than changing the load capacitors

- **TSPC:** if quadrature signals are needed 2 divide by 2 single ended will be used & this will consume more power, Uncontrollable power consumption.

12.4.5. A.2) the Buffers

- At the output use buffer to Isolate Load variation & Mixer kick back noise from the quadrature generator to decrease the IQ mismatch

- At RF frequencies, it is better to use lower swing digital than the rail to rail one, supply noise will be very high.

And so we choose CML buffer rather than CMOs one as it has less swing.

Finally I choose CML Latch as it is the most power efficient one with not very high frequency and CML buffer as CML is preferred than Cmos in high speed-low voltage apps.

12.4.5 .B) The Feedback programmable divider

Required design Specs Given: choose a MMD to be implemented

Division Ratio: 69.38 - > 71.38

MMD power: 2mW, input swing differential 600mvPP

PN = -135 dBc/HZ noise floor >> get better than that

1st stage will be CML Prescaler (divide by 2/3) till frequency decrease to

Be 1GHZ or more as it will consume more power for CMOS also more noise

For TSPC, Simulating Intrinsic Delay of CMOS Inverter at our Technology gives Max operating frequency For CMOS is equal to 10 GHZ so we can you use CMOS/TSPC after CML From 1GHZ till 15MHZ consuming Less power as Frequency is Less, we choose CMOs as Less noise, and has no minimum operating frequency and less supply noise, We will use CML to CMOS to convert Swing to Rail to Rail swing

12.5 Design Methodology

In these section we will discuss the use methodology, design equation & optimization curves used for the actual final design that achieves the specs

12.5.1 Design method for output Divider quadrature generator

For The quadrature output generator as shown in Fig. 12.23. Simple CML Flip flops in divide by 2 modes with CML latch.

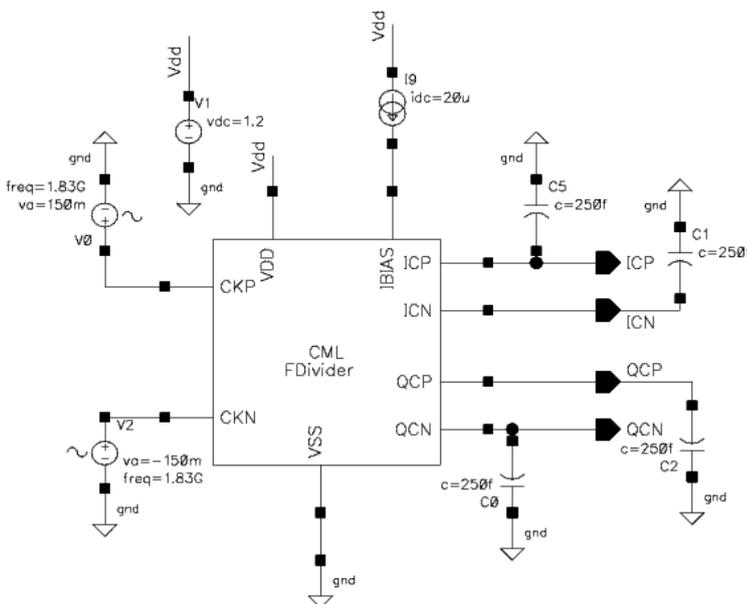


Figure 12.23: Quadrature outputs Generator

12.5.1.1 CML LATCH

The Divide By two consists of two latches in Feedback mode as shown in Fig.12.24, The CML latch is shown in Fig.12.25 is Designed and optimized For Min IQ mismatch and Min power as follows:

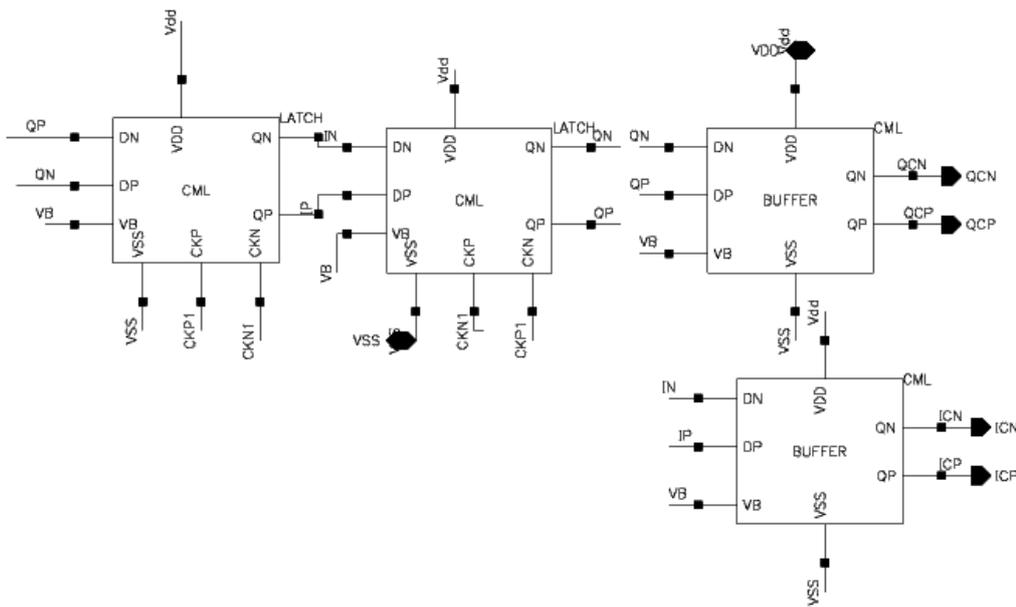


Figure 12.24: Quadrature Output generator from inside

Design steps of CML:

- **Length of transistors**

The length of all transistors will be chosen minimum to get maximum speed.

1) Voltage Levels

For 1 node Out High= VDD, out Low=VDD-IR, for the CML as whole out high IR and Low level = -IR, so differential output swing = 2*I*RD, this swing much so as to be able to switch the following stages between logic '1' and logic '0'. The bias current ISS is completely steered to one of the two output branches for a magnitude of the differential input voltage >

$$\sqrt{2 * ISS / \mu n * Cox * (Wn / Ln)} > 300mv \text{ (we set it) [1]}$$

Also $V_{GD} = V_G - V_{DD} + R_d * ISS < V_{TH}$ to be out of Triode region

2) Choosing I Bias and RD

We can see that the pole of this circuit is $RD * CL$, therefore the delay will be $0.69 * RD * CL$. Increasing the value of current (I Bias) at same swing leads to decreasing the value of the resistance (RD) which decreases the delay and increases the maximum operating frequency at which the SCL circuit operates. Frequency = $1/2 * \text{delay}$, Power latch = $V_{DD} * ISS = 1.2 * ISS$

3) RD implementation

RD can be implemented as PMOS nonlinear load operating in triode region which adds parasitics that decrease max. Frequency. We can use Diode connected PMOS transistor, it will consume large headroom = $(V_{od} + V_{th})$, solved by Source follower with Gm function of switching current slowing down the operation. or use Poly resistance but it takes larger area than active loads but it is linear and we can overcome its inaccuracy by using Poly current as a reference which is reference current

which changes with the variations of the Poly resistance keeping the swing constant across process and temperature corners. Therefore we choose to implement RD with Poly resistance as it is more linear than active loads.

for one latch power < 300mW so ISS=300 uA , Rd*Iss > 300 mV take Rd *ISS = 0.3 so Rd >1K ,out DC = 1.2-0.3/2=1.05 will be input Dc also 1.05-1.2+0.3=0.15 < vth , if vth=450 mV >> max I*R =0.6 V > Rmax = 2 Kohm .

4) Sizing of CLK transistor (M1 & M2)

The operation of this pair of transistors is to steer the current totally between the cross coupled pair (Hold transistors) and the sample differential pair. From characteristics of the differential pairs it is required for the input swing to be larger than Vod. Since we have > 300 mV input swing therefore we choose Vod to be within 100mV or less. For choosing their width, we will follow the relation given in

$$W_{min} = \frac{2 * I_{bias} * L}{Mn * Cox * Vod^2}$$

$\left(\frac{w}{L}\right)_{min} 1,2 = 1.44 \mu$ at Vod)max=V input swing magnitude

5) sampling transistor M3 & M4)

They act as amplifier with gain = gm*RD. They also act as sampling circuit for the input data when clock = '1'. Its gain is chosen to be greater than one at the frequency of operation for robust operation.

(W/L)3,4 >> gm * RL > 1 so gm > 0.625 mS, for vod = 0.174 so ((W/L)3,4)=24 so w3,4) min about = 1.2 u (0.7u actual but min W for RF =1.2u)

6) Latch transistors (M5 & M6)

They are responsible for holding data when clk = '0'. They are cross coupled pair and their gain must be greater than 1 for positive feedback. (W/L)5,6 as (w/L)3,4 or Less so (W)5,6=1.2 u , Actual vod will be chosen for moderate inversion as low as possible to get highest efficiency and good Ft

7) increase transistor widths

Transistor widths need to be exceeded thus by hand analysis cause

- M5,6,3,4 tail node voltage may be small drive M1,2 in triode
- M1,2 tail may be small leaving little headroom for ISS transistor
- At very high speed voltage swing at X & Y not reach Rd*ISS need wider transistor for current steering So tail current must be high to be wider range of linearity & larger gm
- larger sizes means larger Capac, also to decrease IQ mismatch.

12.5.1.2 CML BUFFER

CML buffer is designed with the rest of current 0.5 or less for each one As shown in Fig. 12.26 . it is approximately same methodology as CML buffer , but here we will consume up to 0.5 mA for each . $I \cdot R > 300\text{mV}$ so $R_{\text{min}} = 600 \text{ ohm}$ and $WCK)_{\text{min}} = 2.888 \text{ u}$ to steer current .

12.5.1.3 Current Sources and Biasing

we may choose $L=1 \text{ u}$ or more , W according to square law / GM/ID design for long channel Current Source .

we will choose cascode Bias source to have high impedance node to be stable across corners.[13] as shown in Fig.12.27 . We will choose Poly biasing to have stable swing across corners.as shown in Fig.12.27

12.5.1.3 Final Design Values after optimization :

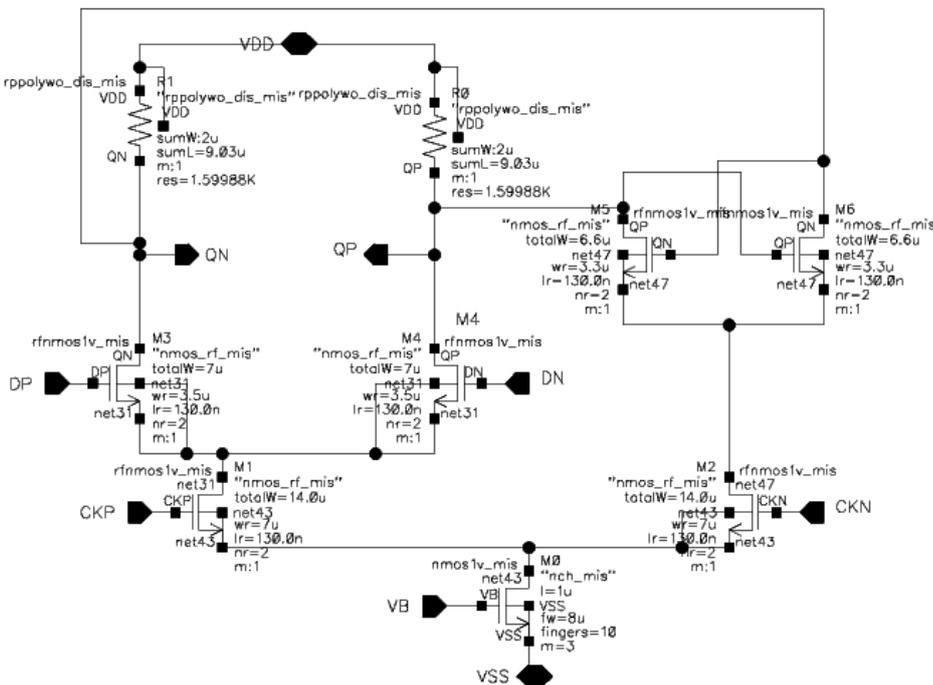


Figure 12.12.25: CML Latch used to generate IQ signals

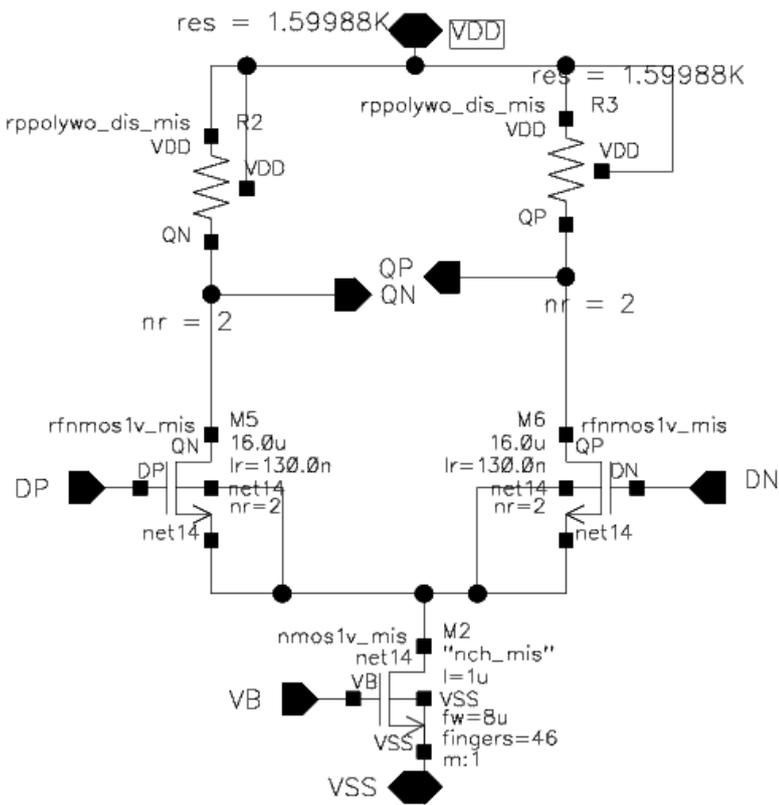


Figure 12.26: CML Buffer

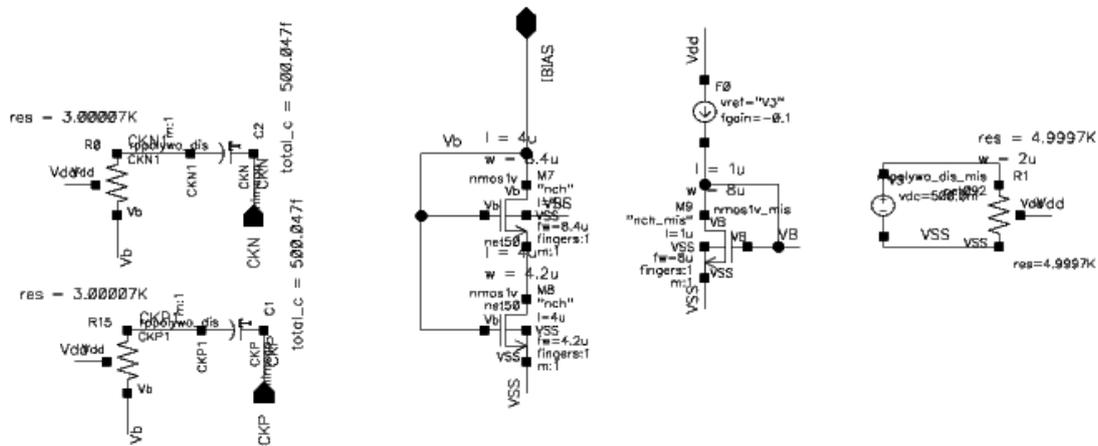


Figure 12.12.27 : Quadrature Bias and Poly current

Table 12.03: Optimized Sizes of transistors after many simulation to satisfy specs (Latch ISS=300 uA , Rd=1.6 K)

CML Latch	Transistor Width	Transistor Length
M0	30*8 u	1u
M1,2	2*7u	130n
M3,4	2*3.5u	130n
M4,6	2*3.3u	130n

Table 12.04: CML buffer optimized Sizes For MIN IQ mismatch (ISS=640 uA, Rd=1.6K)

CML Buffer	Transistor Width	Transistor Length
M2 "CS"	64*8 u	1u
M5,6	2*8u	130n

12.5.1.4 Typical Simulation results

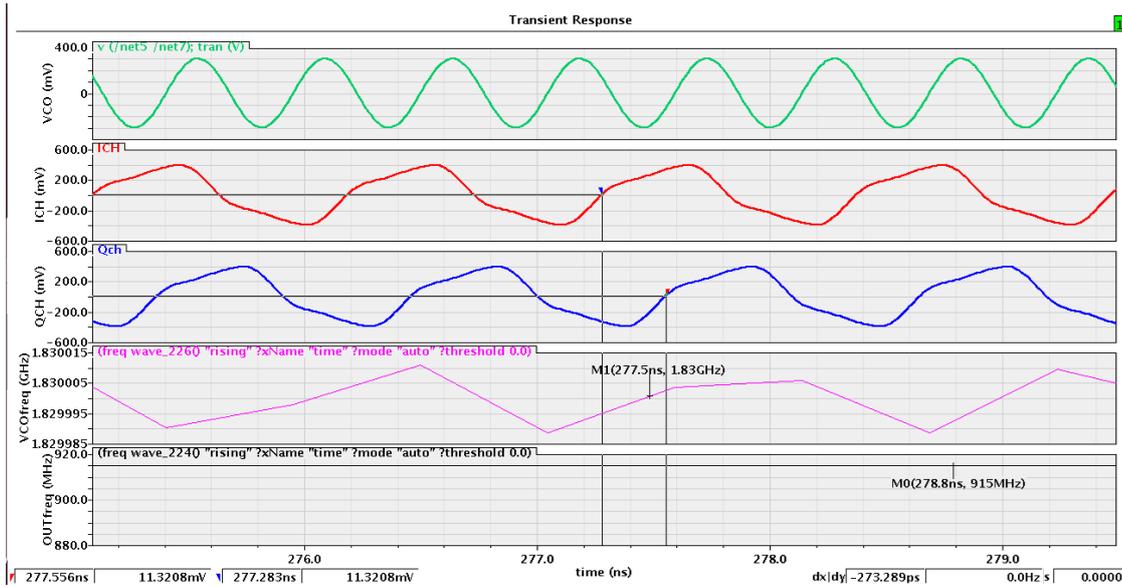


Figure 12.28: Transient Simulation of input, IQ outputs, Frequencies , Delay \equiv IQ Mismatch = $T/4 = 0$ Mismatch

For Monte Carlo analysis $N=100$ Run, $sd=1.88524p$, $\mu=273.51 p$ as $Freq=915MHz$ so $T/4=273.224 p$, So IQ Mismatch in phase = $\frac{90 \cdot 3 \cdot 1.88524}{273.51} = 1.86 \text{ deg} < 2 \text{ deg}$

Another worst case bias curve at 12.29 gives 1.9 deg

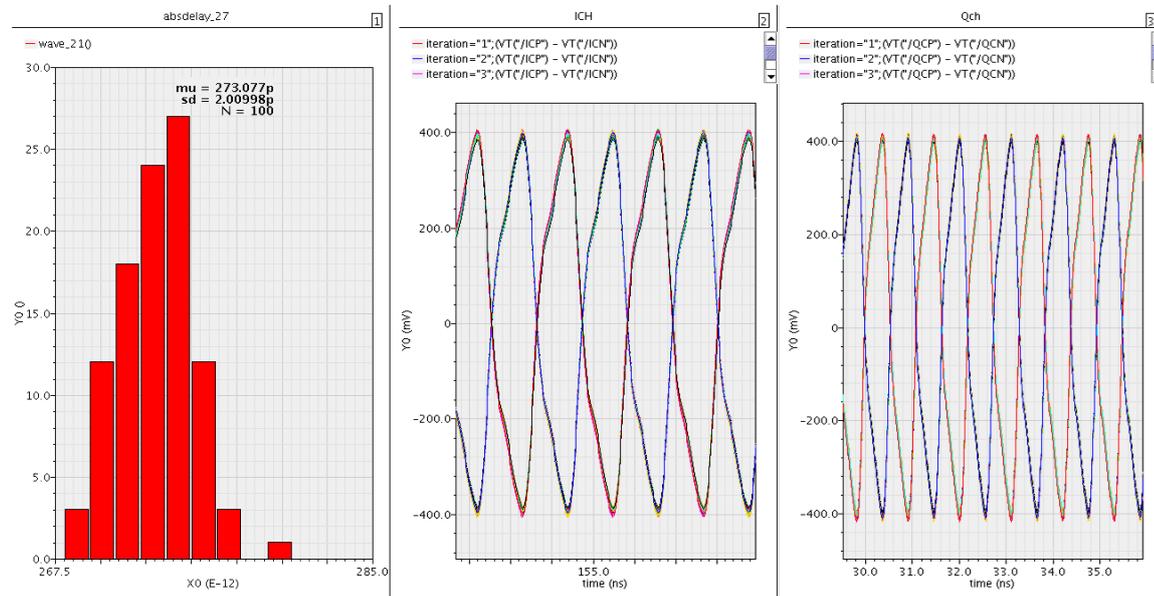


Figure 12.29 : Monte Carlo Simulation after Adding bias Cells

12.5.1.5 Corner Simulations

We run ocean script (Appendix 12), also you can see the Delay function across corners to ensure that the circuit pass across all corners and computed our delay function across corners which gives good results.

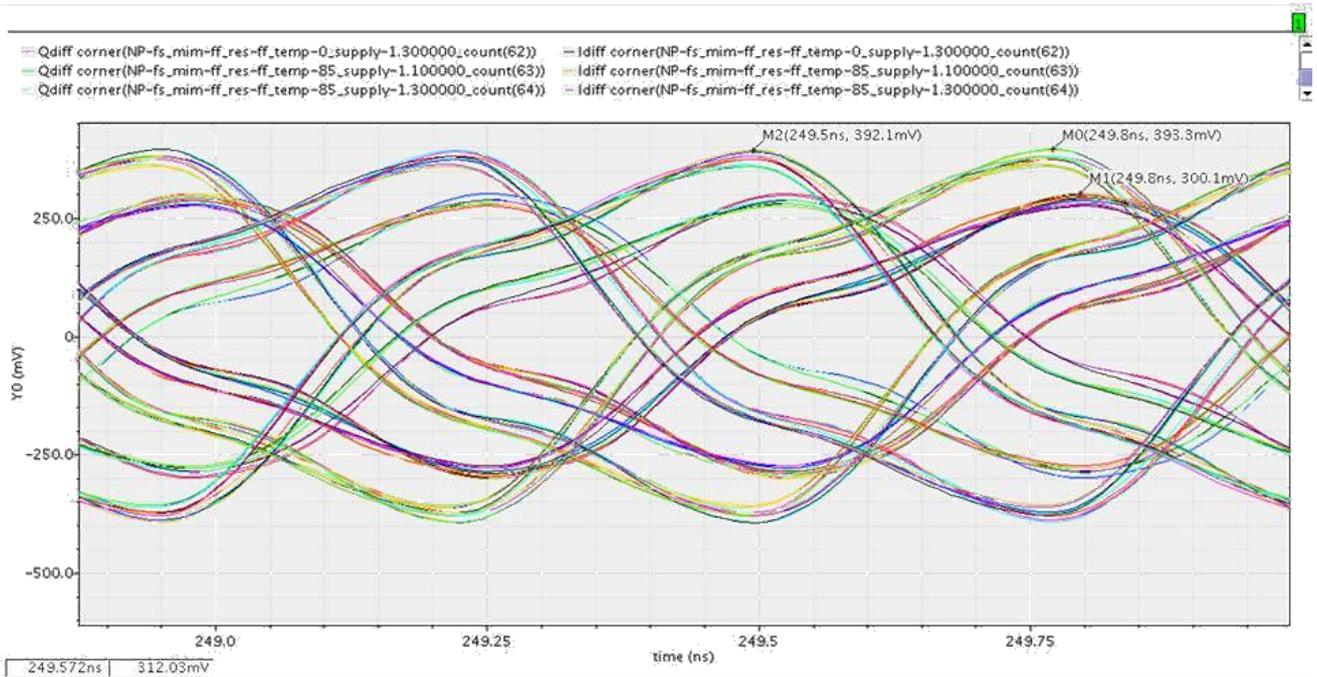


Figure 12.12.30: I and Q signals across corners , and max swing across corners is 393 mV from 350 mV

The divider works across all PVT corners at 2.3 GHz input frequency as the VCO reaches 2.3 GHz before calibration and the digital calibrator uses the output of the divider in calibration. We will present some of the critical corners such as

- Process → SS ,R&C → FF, Temperature → 85 ° and Voltage → 1.1V

This is the worst case corner from speed point of view (The slowest corner). As fast resistant Increase parasitic capacitance which decrease swing at these corner (not compensated by poly current)

- Process → FF, Temperature → 0° and Voltage → 1.3V & 1.1V

This is the fastest corner which gives the maximum possible operating frequency.

- Process → SS, Temperature → 0° and Voltage → 1.1V

This corner is critical for biasing conditions of CML stage

12.5.1.6 Achieved Specs

Table12.05: Achieved and Typical Specs

SPECS	Required	Achieved
Current	1.6 mA	1.52 mA
IQ phase mismatch	< 2 deg	1.8 deg
Works up to	2.3 GHZ	2.3 GHZ
CL	250 fF	250 fF
Swing	>600mV PP	TT:800 , SS: 660, ff:920 m

12.5.2 Design method for feedback divider

For A MMD in Fig.12. based on Design Algorithm we need Min division ratio = $2^n = 64 = 2^6$ near to 69.38 so we will take 6 bits . which will gives us max division ratio = $2^7 - 1 = 127 > 71.38$. it gives us using MMD not generic MMD.

The proposed architecture is given in Fig. We will present the design of each stage .

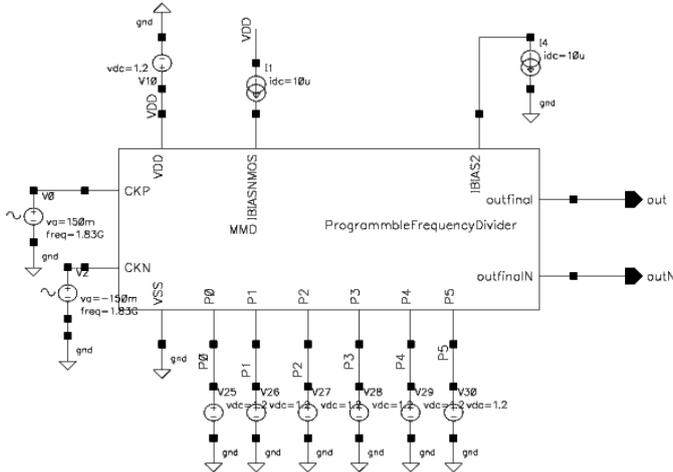


Figure 12.12.31: overall Feedback Divider

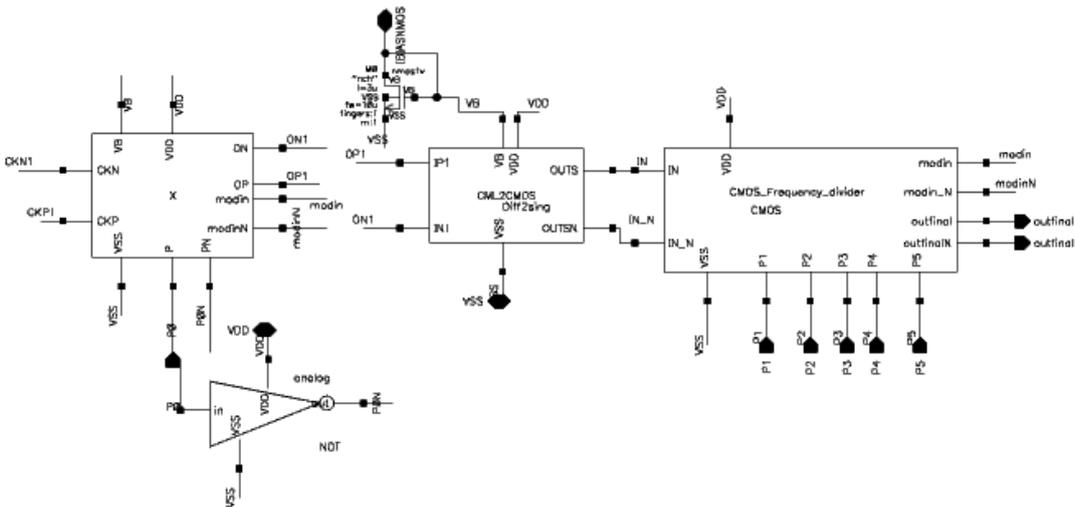


Figure 12.32: Feedback divider construction

12.5.2.1 Design method For CML 2/3 Divider

For a divide by 2/3 as shown in Fig. it consists of many Blocks as CML(Latch , NAND , Level shifter)

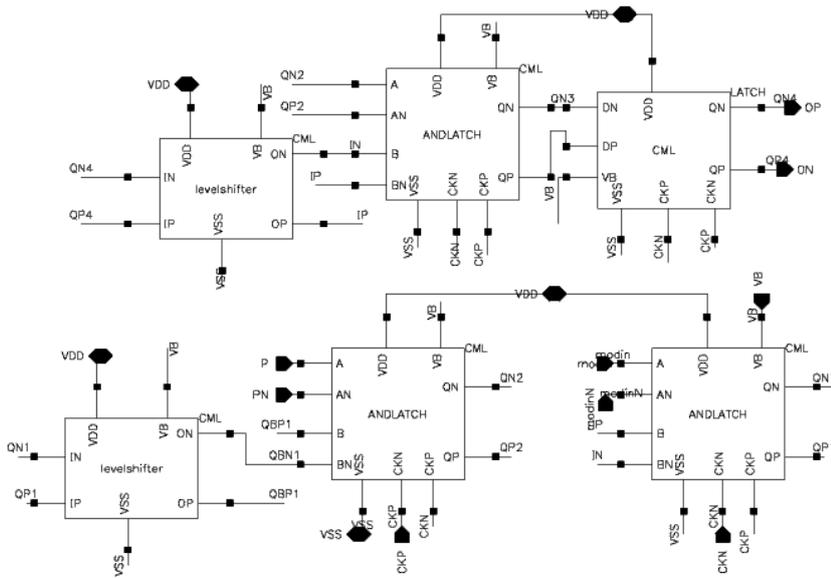


Figure 12.33: A CML MMD Construction

12.5.2.1.1 CML LATCH & CML NAND

Same Design Methodology as mentioned previously with different Current Consumption gives Different Widths. I will not use Latch combined with NAND as I will not pass all corners at this supply level and Rather I will use AML NAND alone after CML latch

12.5.2.1.3 CML Level shifter

Translates From output Level To CLK level at input of NAND gate, it must consumes $INAND*2$, $R=RNAND/2$, $VH=VDD-I*R$; $VL=VDD-I*R-I*R$, put IR =difference in CM of levels.

12.5.2.1.4 Final Design Values after optimization :

Table 12.06 : Final Design Values For CML MMD divide 2/3 Cells

CML LATCH	Transistor Width	Transistor Length
M1,M2	1.5 u	130 n
M3,4,5,6	1.2 u	130 n
M0 CS &(R=5.72K)	7*10 u	3 u
CML NAND	Transistor Width	Transistor Length
M1,2	1.5 u	130 n
M3,4	1.2 u	130 n
M0 : Cs & (R=2.5K)	12*10 u	3 u
CML Level Shifter	Transistor Width	Transistor Length
M3,4	1.2 u	130 n
M5 :CS & (R=1.25K)	24*10 u	3 u

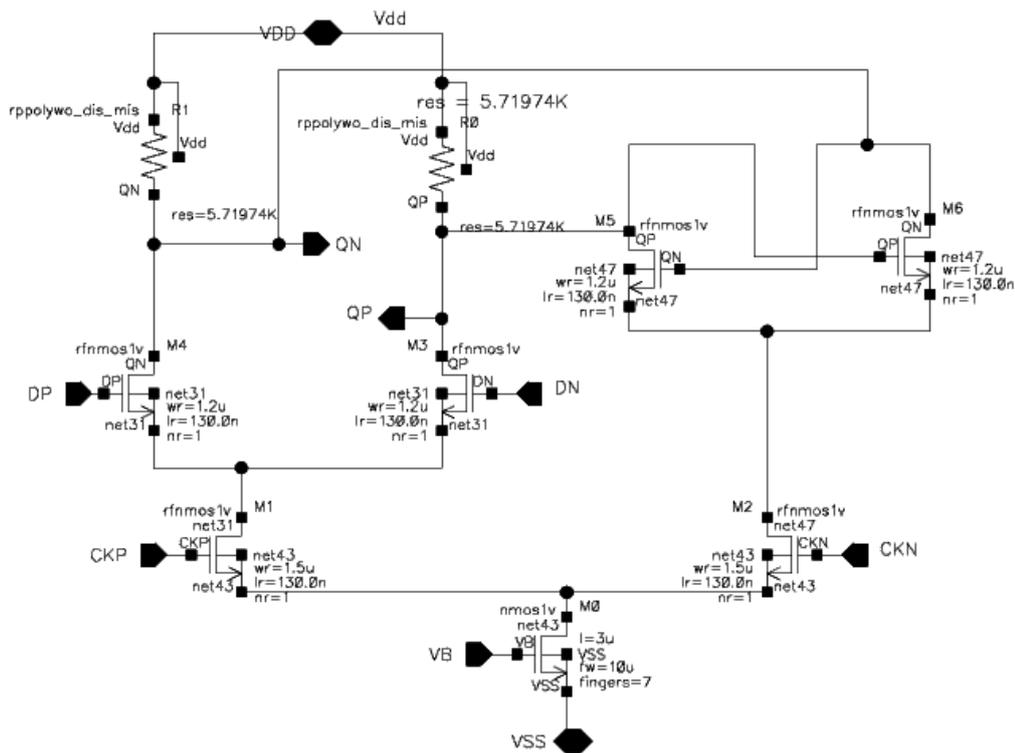


Figure 12.34 : CML latch

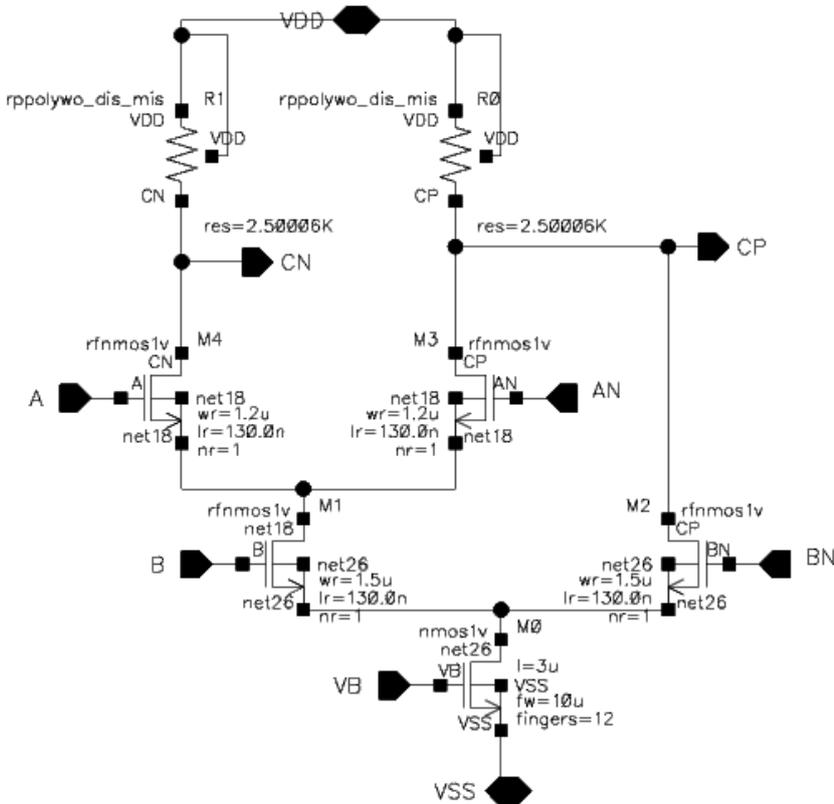


Figure 12.35: CML NAND

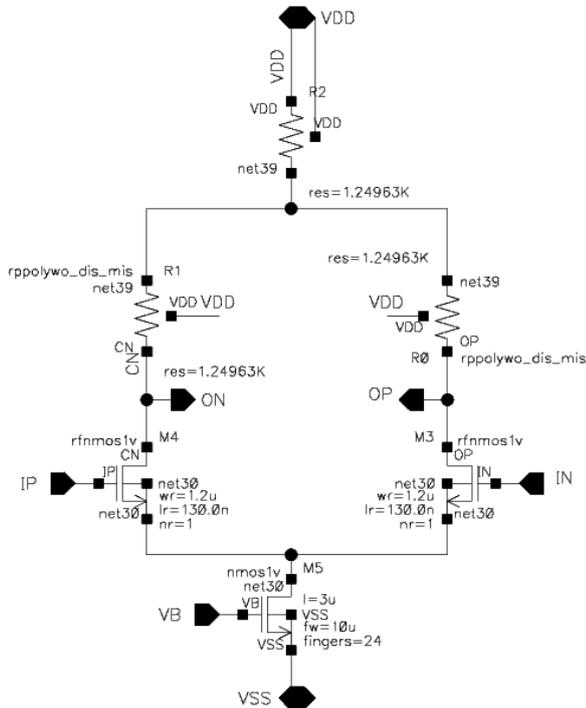


Figure 12.12.36: CML LEVEL SHIFTER

12.5.2.2 Design method For CML to CMOS converter

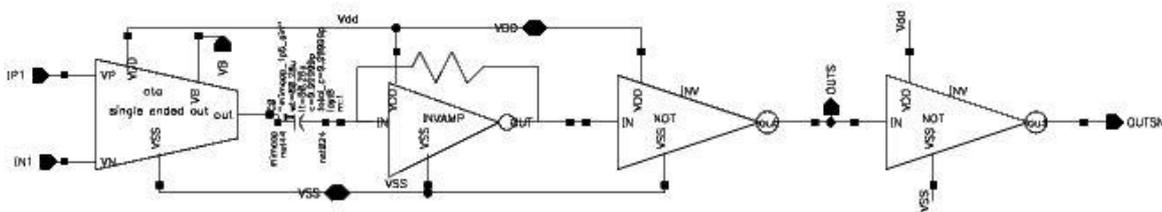


Figure 12.377: CML TO CMOS Converter

It is used to change the swing of the CML stage which is 400mV to rail to rail swing also changing the differential nature of the signal to be single ended to be compatible with static CMOS circuit. It consists of differential input single ended output amplifier, then inverter with feedback resistance which converts the inverter from voltage amplifier to transresistance amplifier so that the circuit become robust against DC offsets at the amplifier output then followed by chain of inverters to drive the load of the next stage.

- The NMOS input differential pair (M₁ & M₀) is sized to compromise between linearity and gain.
- The PMOS pair (M₃ & M₄) is sized to get suitable dc voltage at OUT₁ node.
- Current source is sized to get the suitable current to get the required **gm_{1,0}**
- Inverter with feedback resistance
- M₅ and M₆ are sized to get suitable gain (g_m).
- Resistance value is chosen to get the required gain and BW.
- The total gain of the two stages is given by

$$Av1 * Av2 = (gm_{1,0} \frac{RF}{1-Av2}) *(g_{mn}+g_{mp}) *Rf$$

To size these inverters to get min delay we will use the sizing methodology given in [10]. Using the optimum delay equations in [8], where we choose fanout of each inverter = 3.6 We will use 2 stages .

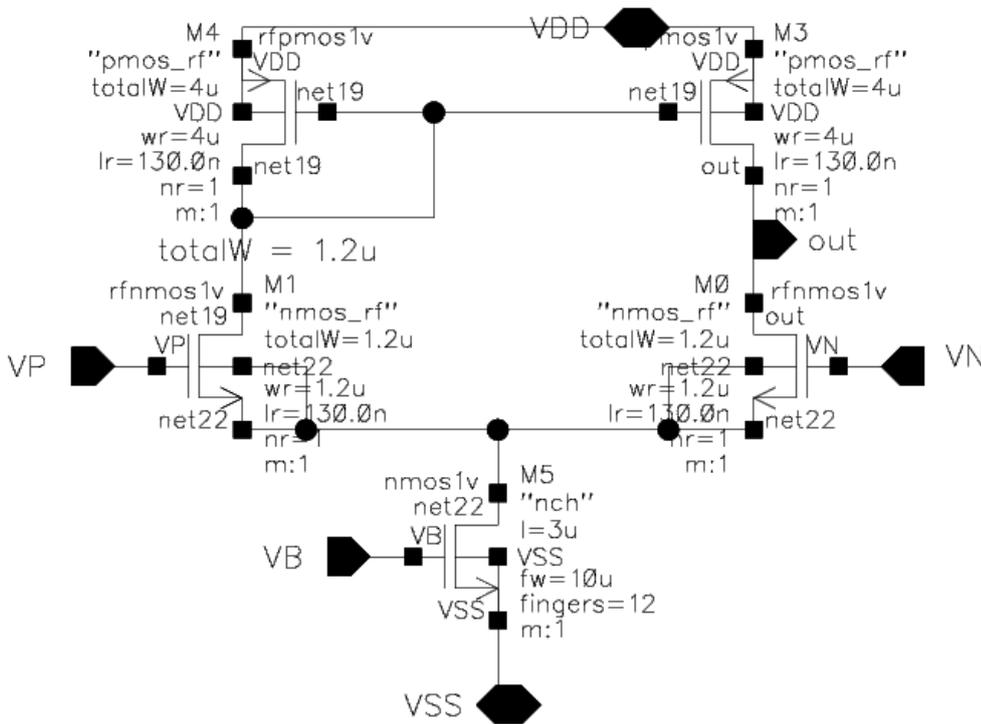


Figure 12.38: Differential to single ended amplifier

Table 12.07 : CML 2 CMOS Design Summary

Single ended output	Transistor Width	Transistor Length
M1,M0	1.2 u	130 n
M3,4	1.2 u	130 n
M5 : CS Consumes 120 uA	12*10 u	3 u
CMOS Inverter amplifier	Transistor Width	Transistor Length
M1	1* 3.7 u	130 n
M0	2*3 u	130 n
R = 10 K , it Consumes 201 uA		
Reference inverter	Transistor Width	Transistor Length
M0	1.2 u	130 n
M1	2 u	130 n

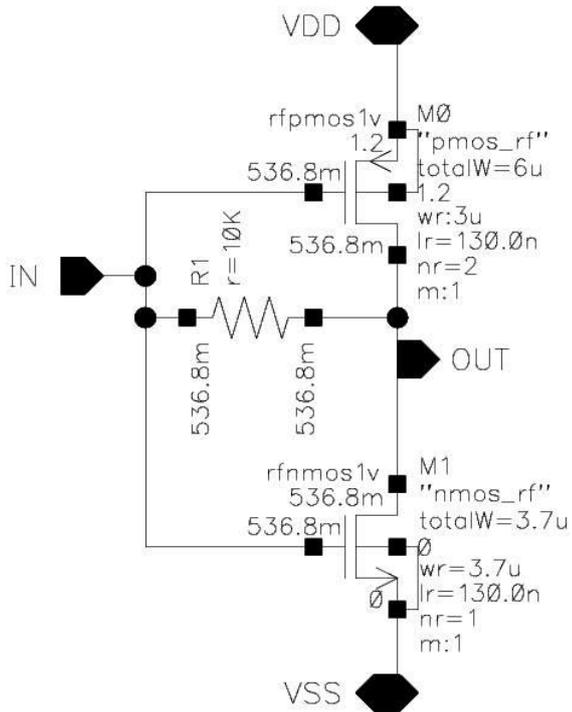


Figure 12.38: Inverter as amplifier

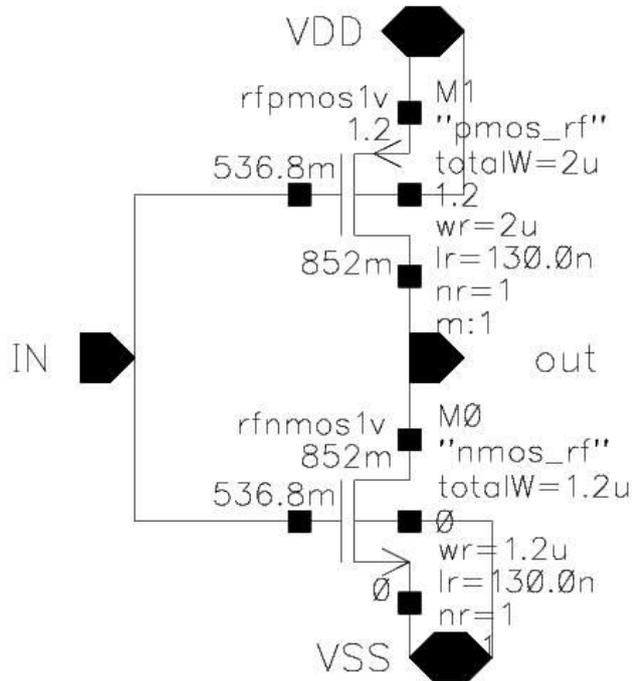


Figure 12.40: CMOS Reference inverter

12.5.2.3 Design methodology for CMOS divider

We will Design Cmos 5 MMD divide 2/3 using same inside structure Like CML MMD to divide the frequency from 1.15GHZ till 35.9375MHz so we will use for the first 2 stages only RFMOS to model accurately

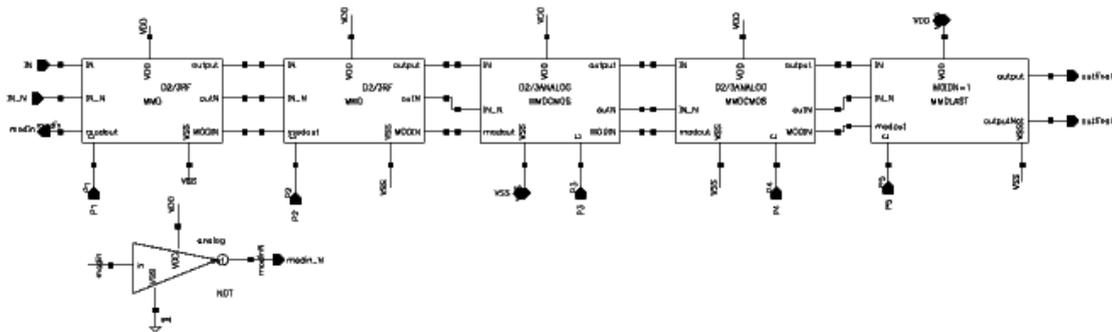


Figure 12.391: Overall CMOS MMD DIVIDER

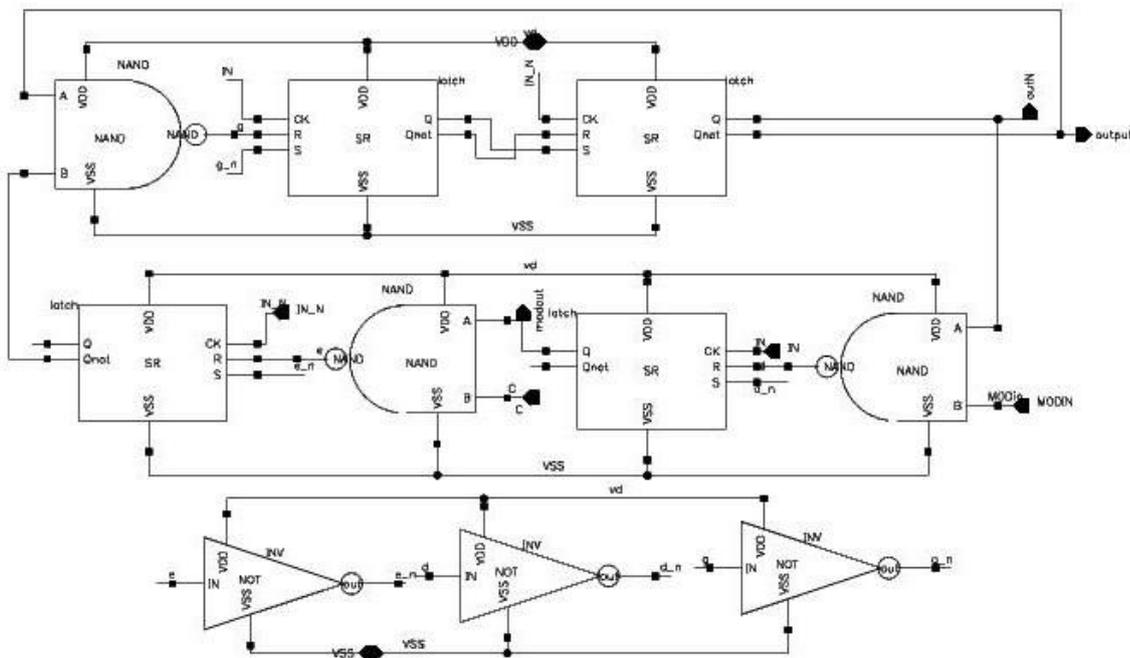


Figure 12.12.40: INSIDE ANY DIVIDER CELL

CML SR LATCH : we can use the normal latch. We could also use combined latch with AND gate to decrease the delay. The CMOS stage are given.

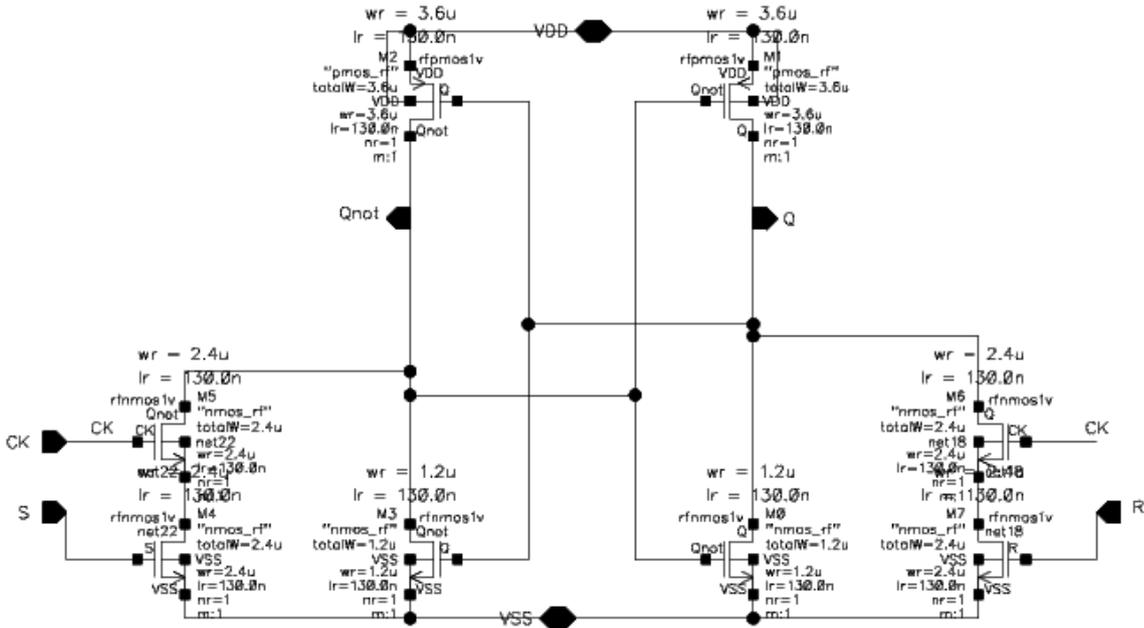


Figure 12.41: CMOS SR LATCH

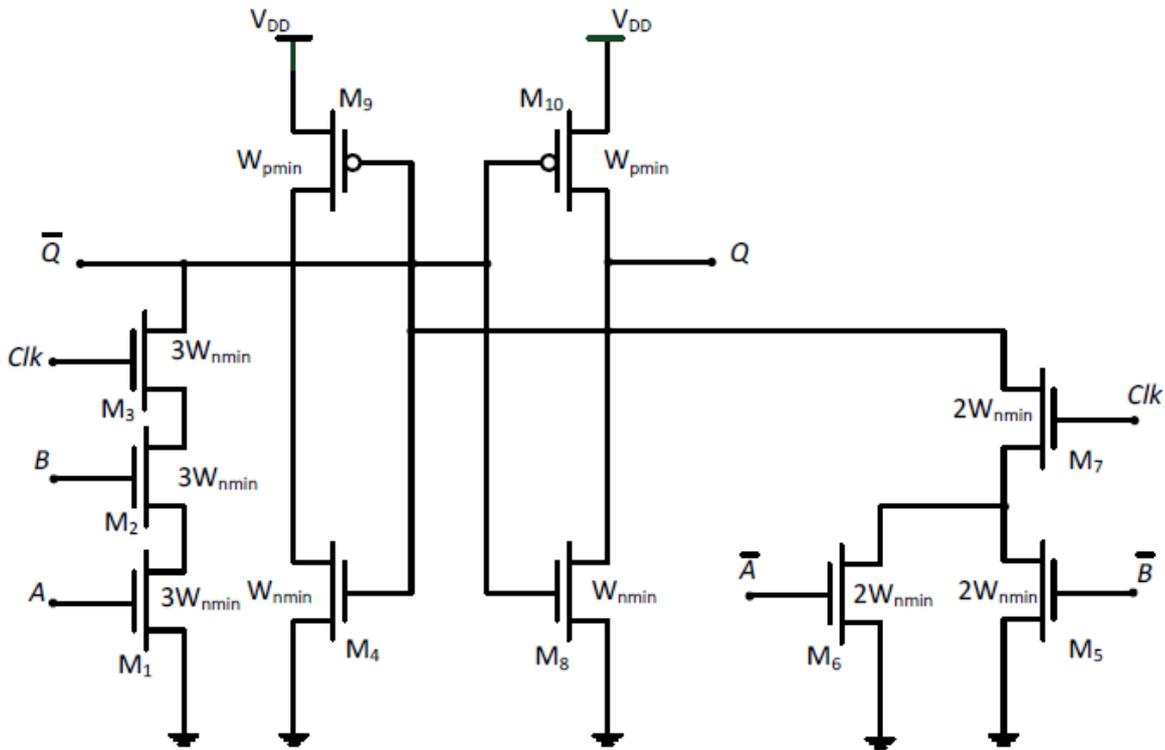


Figure 12.42: Combined AND gate with SR latch with sizing of the reference latch with minimum length

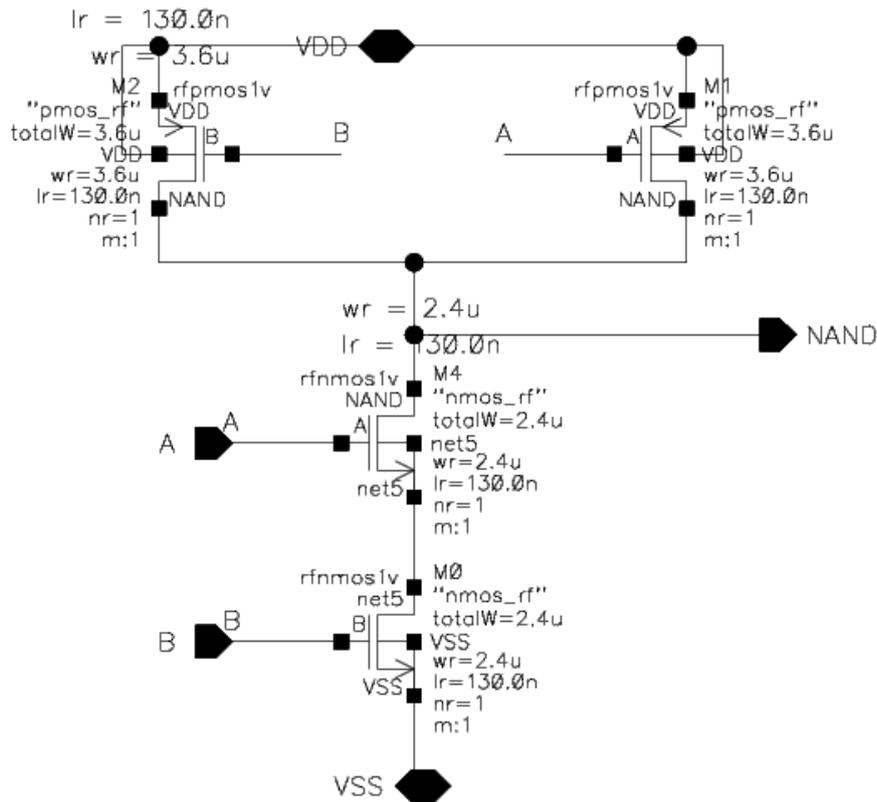


Figure 12.43: CMOS NAND

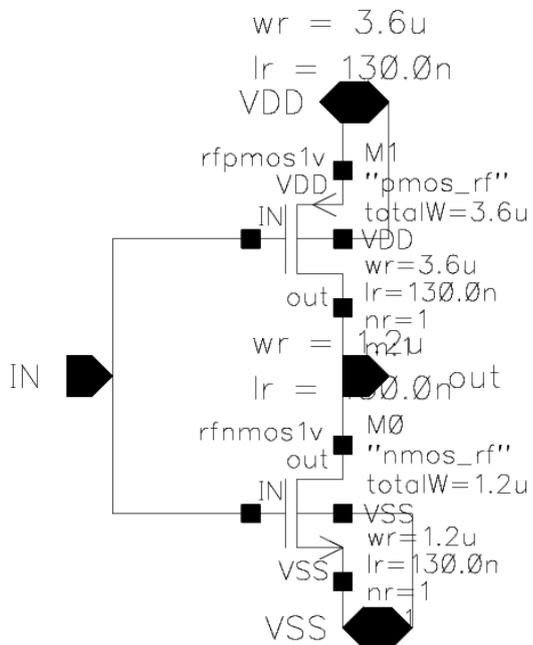


Figure 12.12.44: CMOS NOT

Table 12.08 : CMOS MMD sizing

CMOS SR LATCH	Transistor Width	Transistor Length
M3,M0	1.2 u	130 n
M4,5,6,7	2.4 u	130 n
M1,2	3.6 u	130n
CMOS NAND	Transistor Width	Transistor Length
M1,2	3.6 u	130 n
M0,M4	2.4 u	130 n
Reference inverter	Transistor Width	Transistor Length
M0	1.2 u	130 n
M1	3.6 u	130 n

12.5.2.4 Typical Simulation results

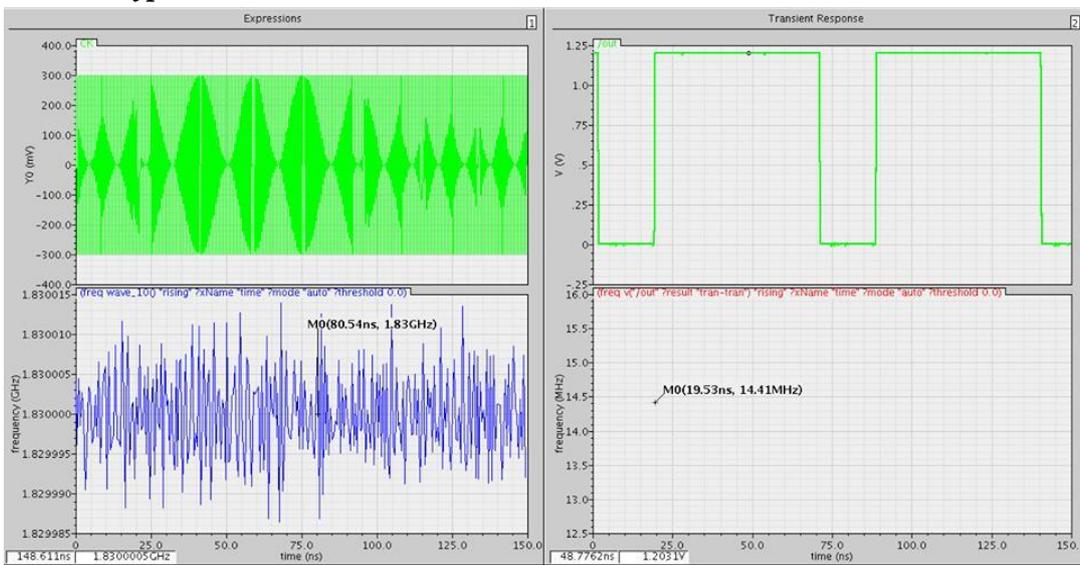


Figure 12.12.45 : VCO input , output at max Division Ratio & their Frequencies

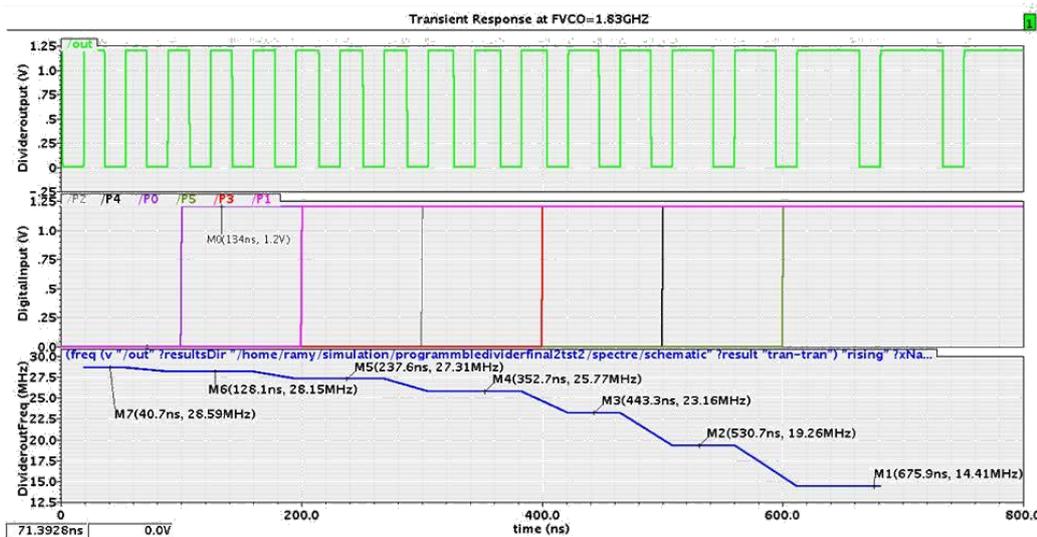


Figure 12.46: Variable Division Ratio by Varying Digital Input at VCO = 1.83GHZ

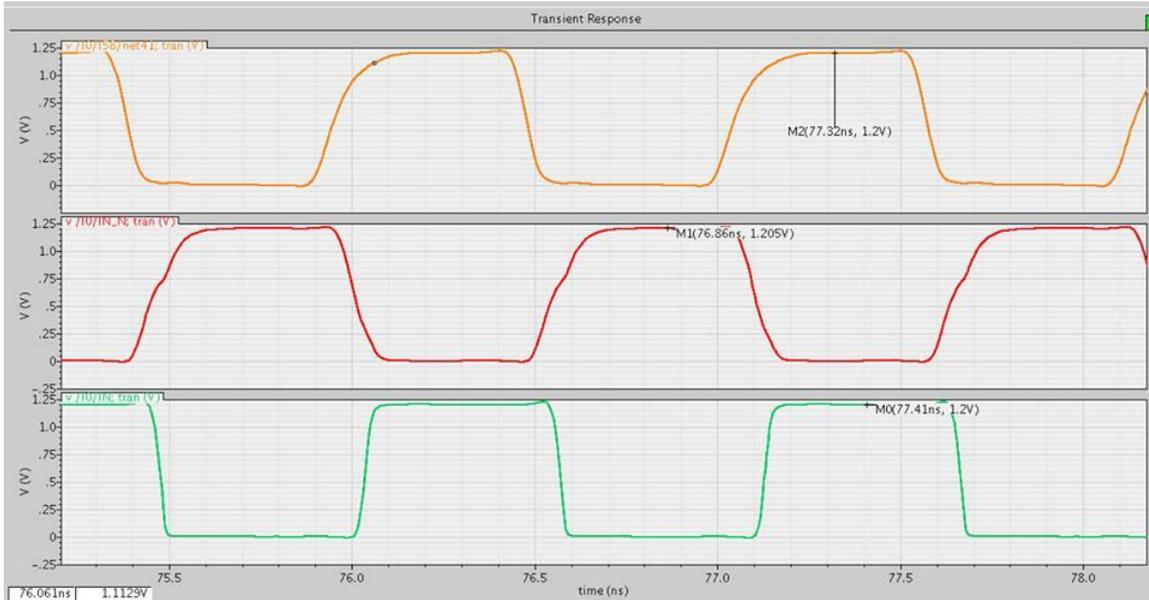


Figure 12.49: outputs after stages at CMOS buffer

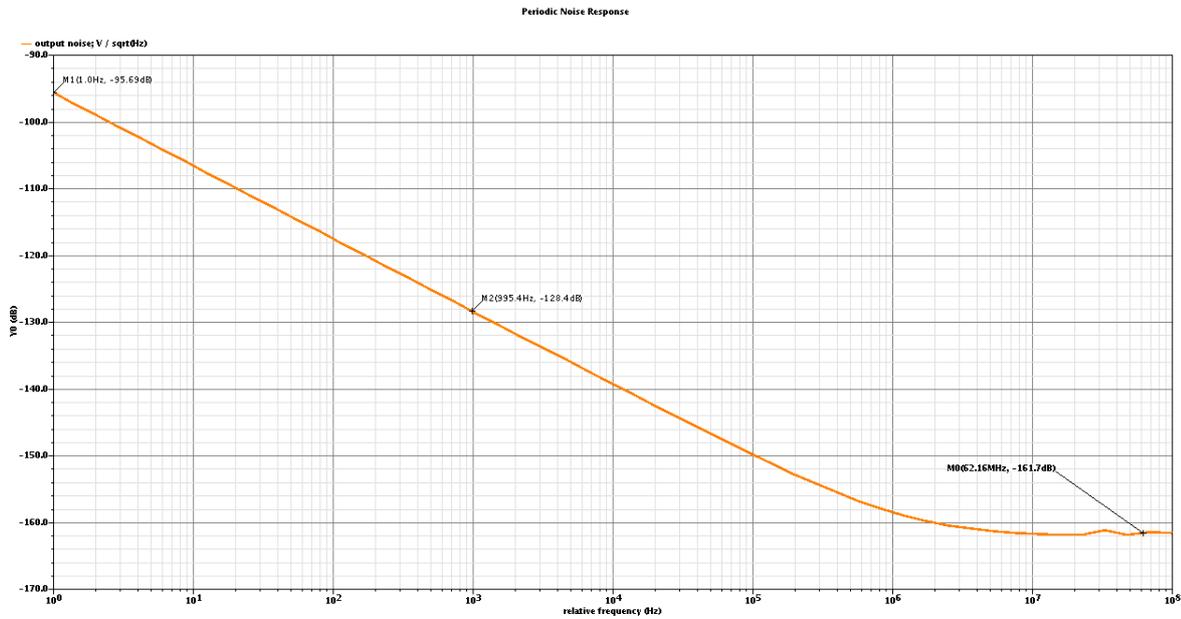


Figure 12.50: phase noise simulation

12.5.2 Required Vs. Achieved

Table 12.0 9 : Overall specs of Feedback Divider

SPECS	Required	Achieved
Current	2 mA	1.9 mA
Noise floor	< -135 dBc/HZ	-161.7 dBc/HZ
Works up to	2.3 GHZ	2.3 GHZ
Division Ratio	69.38 -> 71.38	64 -> 127

12.5.3 Corner simulation

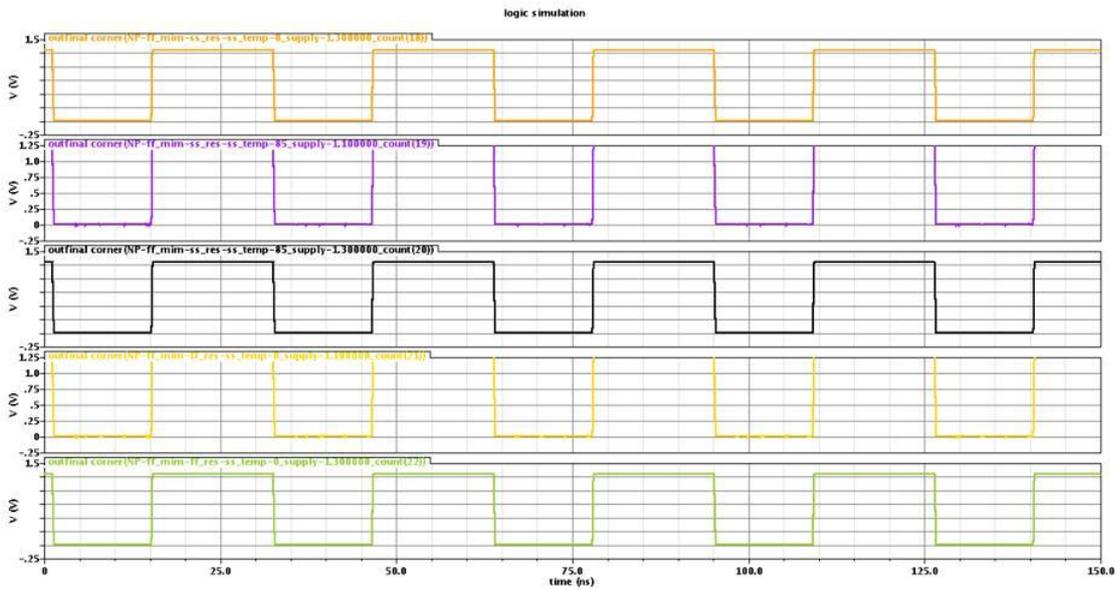


Figure 12.51: output waveforms across corner

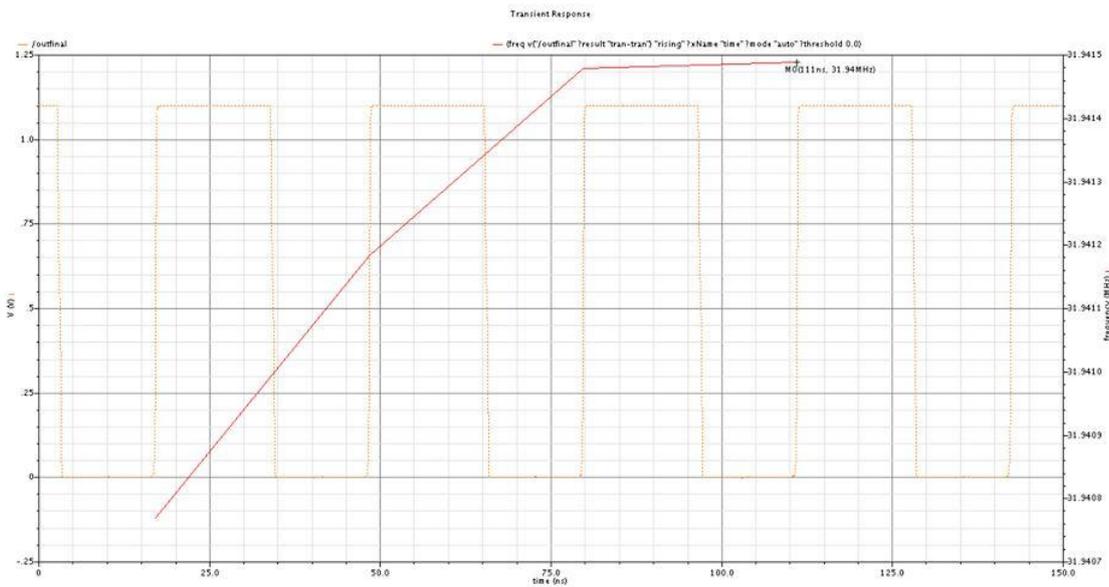


Figure 12.47: Division Ratio of 72 worst-case Corner VCO=2.3GHZ output=31.9444MHZ

12.6 Conclusion

Frequency Divider one of the hardest high frequency Minimum size power hungry blocks for the ISM band transceiver that serves & interfaces with many other blocks. Its interfaces with **VCO**, **mixer**, **PA**, **PFD**, **sigma-delta** needs to be adjusted at common mode and needs to be integerable with them across their worst case corner simulations . The main tradeoff in any frequency divider design that is operating at several GHz range is maximum operating frequency

References:

- 1) B. RAZAVI, RF MICROELECTRONICS, 2ND EDITION. PRENTICE-HALL
- 2) JOHN ROGERS & CALVEN PLETT. RADIO FREQUENCY INTEGRATED CIRCUIT.DESIGN. ARTECH HOUSE, INC.,2003.
- 3) VAUCHER, C. S., ET AL., "A FAMILY OF LOW-POWER TRULY MODULAR PROGRAMMABLE DIVIDERS IN STANDARD 0.35 MM CMOS TECHNOLOGY," IEEE J. SOLID-STATE CIRCUITS, VOL. 35,JULY 2000. SYNTHESIS "
- 4) RAJA K. K. R. SANDIREDDY, FOSTER F. DAI, RICHARD C.JAGGER 2004"A GENERIC ARCHITECTURE FOR MULTI-MODULUS DIVIDERS IN LOW-POWER AND HIGH-SPEED FREQUENCY
- 5) CRANINCKX, J., AND M. S. J. STEYAERT, "A 1.75-GHz/3-V DUAL-MODULUS DIVIDE BY-128/129 PRESCALER IN 0.7-μm CMOS," IEEE J. SOLID-STATE CIRCUITS,VOL. 31, JULY 1996.
- 6) TECHNIQUES FOR HIGH DATA RATE MODULATION AND LOW POWER OPERATION OF FRACTIONAL-N FREQUENCY SYNTHESIZERS BY MICHAEL HENDERSON PERROTT
- 7) M.H. PERROTT, "HIGH-SPEED COMMUNICATION CIRCUIT AND SYSTEMS," MIT OCW 2003.
- 8) JAN M. RABAEEY, ANANTHA CHANDARKASAN & BORIVOJIE NIKOLIC, "DIGITAL INTEGRATED CIRCUITS", 2ND EDITION.
- 9) J.JR. NAVARRO SOARES & W.A.M. VAN NOIJE. A 1.6-GHz DUAL MODULUS PRESCALER USING THE EXTENDED TRUE-SINGLE-PHASE-CLOCK CMOS CIRCUIT TECHNIQUE (E-TSPC). IEEE J. SOLID-STATE CIRCUITS, VOL. 34, NO. 1, JAN. 1999
- 10) A.WAFA AND A.AHMED "HIGH-SPEED RF MULTI-MODULUS PRESCALER ARCHITECTURE FOR Σ - Δ FRACTIONAL N PLL FREQUENCY SYNTHESIZERS".
- 11) Q. HUANG AND R. ROGENMOSER, "SPEED OPTIMIZATION OF EDGE-TRIGGERED CMOS CIRCUITS FOR GIGAHERTZ SINGLE-PHASE CLOCKS," IEEE J. SOLID-STATE CIRCUITS, VOL. 31, PP. 456–465, MAR. 1996.
- 12) THE DESIGNER'S GUIDE COMMUNITY BY MANOLIS TERROVITIS
ATHEROS COMMUNICATIONS "SIMULATING THE PHASE NOISE CONTRIBUTION OF THE DIVIDER IN A PHASE LOCK LOOP"
- 13) BIAS CIRCUIT DESIGN FOR LOW-VOLTAGE CASCODE TRANSISTORS

13 Sigma Delta

13.1 Introduction

The large computer and memory markets drive a “state-of-the-art” deep sub-micron CMOS fabrication technology. Traditional analogue and mixed mode radio electronics is constantly being pushed towards implementation in digital CMOS. The trend is to pack in more and more transistors of ever smaller size, and with rather poorer analogue properties, so that the emphasis is on developing digital techniques, which scale better with technology and can be reused easily and ported from process to process. These trends converge in the concept of a system on a chip (SOC), where all the components of a computer system and various electronic sub-systems are integrated into a single chip [1].

Analogue DSMs have received much more attention in the literature. The situation has changed fairly recently, however, with the introduction of digital DSMs for radio transceiver applications. Digital DSMs are now found in applications which contribute to greater integration and digitization of the radio front-end, including oversampled digital-to-analogue converters (DACs), mismatch shaping converters and, most notably, delta-sigma fractional-N frequency synthesis [1].

In this work a Digital Sigma Delta modulator was designed and implemented for a fractional-N frequency synthesizer, using the MASH 1-1-1 architecture. The different architectures were studied and compared using MATLAB Simulink software, the topology of choice was selected based on performance requirements and hardware and power optimization, a VHDL behavioral model was then written on Mentor’s Modelsim software, and the testbench verified the MATLAB results, Xilinx software was then used for the hardware synthesis and a VerilogA model was written for the sake of system integration with the analog components of the PLL in the Cadence environment.

13.2 Basics Concepts of the Sigma Delta

The concept of SD modulation was first employed in oversampling analog-to digital (A/D) and digital-to-analog (D/A) converters [1, 2]. Oversampling data converters operate at a much higher sampling rate than the Nyquist rate, which is twice the signal bandwidth.

13.3 quantization noise and over sampling effects

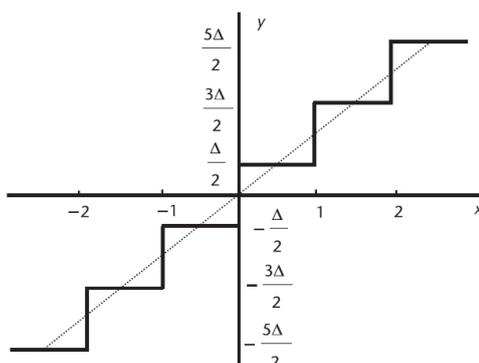


Figure 13.13.1: Transfer Characteristics of a multi bit quantizer

Suppose the analog input is x and the quantized output is y and a step size Δ , then in representing the input. This error e will be bounded over one quantizer level by a value of

$$-\frac{\Delta}{2} \leq e \leq \frac{\Delta}{2} \quad (1)$$

and the quantized output can be represented by a linear function with error e

$$y = \Delta \cdot x + e \quad (2)$$

Where the step size Δ corresponds to the slope of the straight line shown in Figure 1. The quantization error as a function of the input is given in Figure 2.

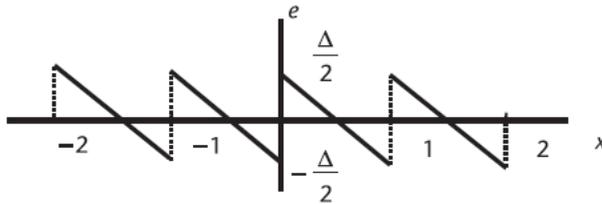


Figure 13.13.2: The quantization error as a function of the input.

Note

Δ . If the input is “random” in nature, then the instantaneous error will also be random. The error is thus uncorrelated from sample to sample and can hence be treated as “noise.” Quantization and the resultant quantization noise can be modeled as a linear circuit including an additive quantization error source, as shown in Figure 9.3. Thus, the quantization noise for a random signal can be treated as additive white noise having a value anywhere in the range from $-\Delta/2$ to $\Delta/2$. The quantization noise has equal probability with a probability density of

$$p(e) = \begin{cases} \frac{1}{\Delta} & \text{if } -\frac{\Delta}{2} \leq e \leq \frac{\Delta}{2} \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

The normalization factor is needed to guarantee that

$$\int_{-\Delta/2}^{\Delta/2} p(e) de = 1 \quad (4)$$

Integrating the square of the error and dividing by the step size the mean rms error voltage is found to be

$$e_{\text{rms}}^2 = \int_{-\infty}^{+\infty} p(e) e^2 de = \frac{1}{\Delta} \int_{-\Delta/2}^{\Delta/2} e^2 de = \frac{\Delta^2}{12} \quad (5)$$

From control theory we know that the frequency spectrum of a sampled system repeats every sampling frequency, thus the quantization noise in a sampled system will be centered at half the sampling frequency and repeated, assuming all the noise power lies in the positive frequency band and assuming white noise the PSD of the quantization noise is given by

$$E^2(f) = \frac{e_{\text{rms}}^2}{f_s/2} = 2T e_{\text{rms}}^2 \quad (6)$$

For a band limited signal with bandwidth f_0 , the quantization noise power that falls into the signal bandwidth can thus be found to be

$$n_0^2 = \int_0^{f_0} E^2(f) df = 2f_0 T e_{\text{rms}}^2 = \frac{\Delta^2 f_0}{6 \cdot f_s} = \frac{\Delta^2}{12 \cdot \text{OSR}} \quad (7)$$

Where the oversampling ratio (OSR) is defined as the ratio of the sampling frequency f_s to the Nyquist frequency $2f_0$

$$\text{OSR} = \frac{f_s}{2f_0} \quad (8)$$

for an N-bit sampled system, if the quantizer has 2^N quantization levels equally spaced by Δ , the maximum peak to peak amplitude is give $v_{\max} = (2^N - 1) \cdot \Delta$

$$(9)$$

for a sinusoidal signal, the associated signal power is

$$P = \frac{1}{8} (2^N - 1)^2 \cdot \Delta^2 \quad (10)$$

Thus the SNR due to quantization noise that falls into the signal band becomes

$$\text{SNR} = 10 \log \left(\frac{\frac{1}{8} (2^N - 1)^2 \Delta^2}{n_0^2} \right) \approx 10 \log \left(\frac{3 \cdot 2^{2N} \text{OSR}}{2} \right) \quad (11)$$

Noting that $\log_{10}(x) = \log_{10}(2) \cdot \log_2(x)$, the above equation becomes

$$\text{SNR} \approx 6.02 \cdot N + 3 \cdot \log_2(\text{OSR}) + 1.76 \quad (12)$$

The above equation shows that the SNR improves by 6 dB for every bit added to the quantizer for the same amount of total quantization noise power doubling the sampling frequency reduces the in-band quantization noise by 3 dB. As illustrated in Figure 3, oversampling reduces the in-band rms quantization noise since the total noise is spread across the entire sampling bandwidth. Hence, doubling the oversampling ratio is equivalent to increasing the quantizer levels by a half-bit as far as the quantization noise is concerned. Oversampling allows the use of a lower-resolution converter without sacrificing noise performance. Another way of looking at this effect is that by doubling the sampling rate

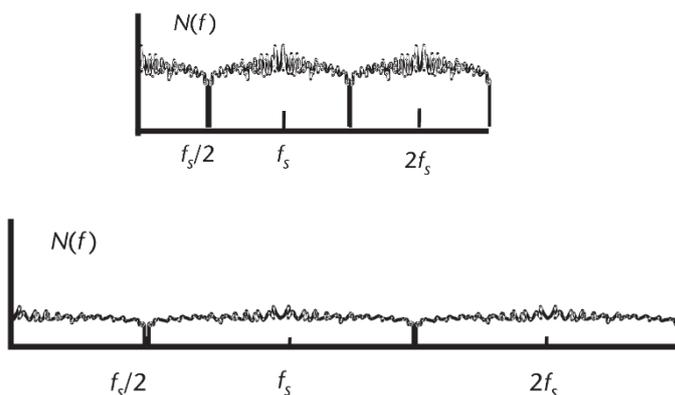


Figure 13.13.3: Quantization noise reduction by increasing OSR: (up) low sampling frequency, and (down) higher sampling frequency.

13.4 Noise-Shaping Effect

While noise oversampling can reduce the random quantization by averaging it over a wider sampling bandwidth, another useful scheme for quantization noise reduction is noise shaping using feedback. Consider a feedback model shown in Figure 4. In this figure, an additive noise model of the quantizer follows a filter with transfer function of $H(s)$. Then, negative feedback is added to stabilize the system.

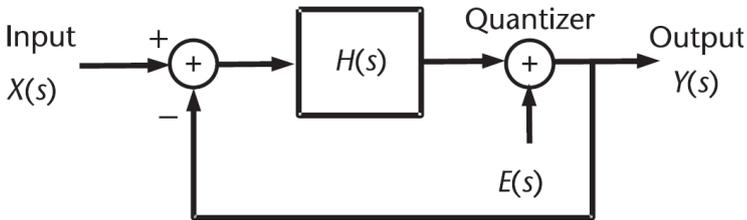


Figure 13.13.4: Feedback model of a noise-shaping system

From control theory, the output is given by

$$Y(s) = \frac{H(s)}{1 + H(s)} X(s) + \frac{1}{1 + H(s)} E(s) \quad (13)$$

If $|H(s)| \gg 1$, we can approximate the signal transfer function as

$$Y(s) = \frac{H(s)}{1 + H(s)} X(s) \approx X(s) \quad (14)$$

Thus the signal transfer function is unity at low frequencies, similarly we find the noise transfer function (NTF) to be

$$\text{NTF}(s) = \frac{1}{1 + H(s)} E(s) \approx 0 \quad (15)$$

From this analysis we can see that the output quantization noise can be reduced in certain frequency bands if $H(s)$ is designed to have high gain in those frequency bands. The most troublesome noise is the close-in quantization noise since it is hard to remove using a LPF.

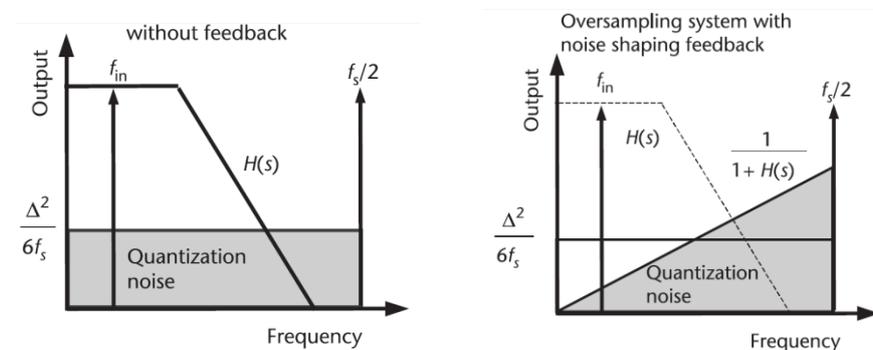


Figure 13.13.5: Output of an oversampling system with and without feedback

Assuming the quantization noise to be white noise uniformly distributed from $0 \leq f \leq f_s/2$ with noise power density $\Delta^2/6f_s$. In a feedback oversampling system with a low pass transfer function for $H(s)$, the in-band quantization noise can be shaped as illustrated in Figure 5. The resultant noise transfer function of $1/[1 + H(s)]$ has a high pass effect. Hence, the quantization noise is highpass-shaped by the feedback loop, leading to lower in-band noise. Note that the total quantization noise energy for both systems with and without feedback is the same. In other words, the total shaded area in each of Figure 5 plots is $\Delta^2/12$. The only difference is that the quantization noise with feedback is shifted to the higher frequency band, where it can be easily filtered by the low pass shaped loop filter. It is important to note here that modeling the quantization noise as white noise is subject to some conditions, The error sequence is a sample sequence of a stationary random process. The error should be uncorrelated with the input. The variance corresponds to the mean square value of the signal e . the error is a white noise process. The probability of distribution of the error is uniform over the quantization error.

13.5 Sigma Delta Modulators in Fractional-N frequency synthesizers

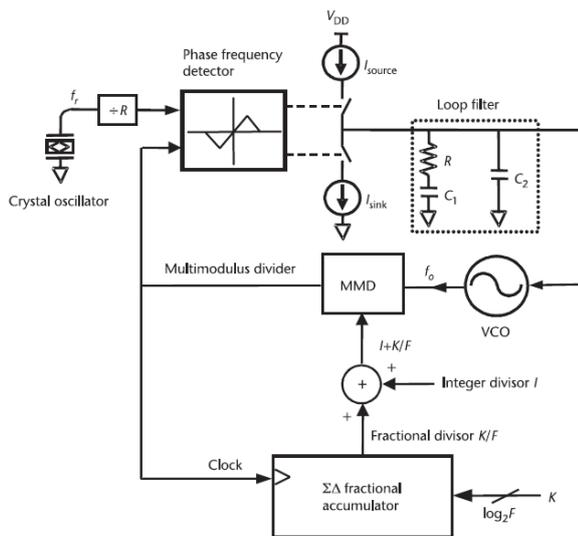


Figure 13.13.6: Fractional-N frequency synthesizer with a SD modulator

In order to achieve fine frequency resolution get that resolution using a conventional integer-N PLL, a low reference frequency, f_{ref} , is needed. The low f_{ref} in turn has several drawbacks; firstly, the reference spur and its harmonics are located at low offset frequencies which will require the PLL loop bandwidth(BW) to be very small (about $f_{ref}/10$) and hence the lock time increases. Secondly, a large divide ratio, N, is required to get the desired output frequency, and this increases the in band phase noise. Finally, the phase noise of the VCO will not be sufficiently suppressed at low offset frequencies.

13.6 Spur Reduction Techniques in Fractional-N Frequency Synthesis

Due to the generation of unwanted spurs in addition to the reference spur, fractional-N frequency synthesis is not useful in practical applications unless the fractional spurs are suppressed. Therefore, additional circuitry must be added to suppress those fractional spurs. Various techniques have been proposed. A summarized comparison is

found in the Table 1, this work focuses on the Sigma Delta method, however more details on other methods could be found in [3]

Table 13.1: Spur reduction techniques in fractional-N synthesizers

Technique	Feature	Problem
DAC estimation	Cancels spurs by DAC	Analog mismatches
Random Jittering	Randomizes divider control	Frequency jitter
Sigma Delta modulation	Modulates divider control with noise shaping	Quantization noise at high frequency
Phase Interpolation	Inherent fractional division	Multi-phase VCO
Phase compensation	Time-domain compensation	Analog mismatches
Phase insertion	Frequency multiplier using pulse insertion	Analog mismatches

A fractional-N PLL can overcome the above-mentioned disadvantages of an integer-N PLL by decoupling the resolution from the reference frequency. The divider is dynamically modulated using a sigma delta modulator to give, on the average, a fractional value. A sigma delta modulator is a non-linear system due to its quantizer. The quantization error is statistically dependent on the input. Assuming the input signal is sufficiently busy, the error from sample to sample is largely uncorrelated; it looks like white noise [7]. The white noise approximation is accepted when using higher order sigma delta modulators. Besides, higher order modulators provide much better noise shaping with fewer spurs, as there is much more complex interaction between input and quantization noise. However, for constant inputs, the white noise assumption is invalid. Strong tones appear even for higher order modulators. Therefore, dithering techniques are required to remove these tones [6].

13.7 Architectures of Sigma Delta Modulators

Sigma Delta modulators are basically divided into two types: single loop and cascaded (or MASH, Multi-stage noise Shaping). AMASH modulator is the cascading of lower order stages. For example a MASH 1-1-1 modulator is composed of three cascaded 1st order Sigma Delta modulators as shown in Fig. 3. It has an unconditionally stable input range normalized to the integer value from 0 to 1. The pass band gain of a MASH 1-1-1 noise transfer function (NTF) equals 8. The output has eight levels and spreads from -3 to 4 with an average between 0 and 1. As a result, the standard deviation of the normalized phase error is relatively high.

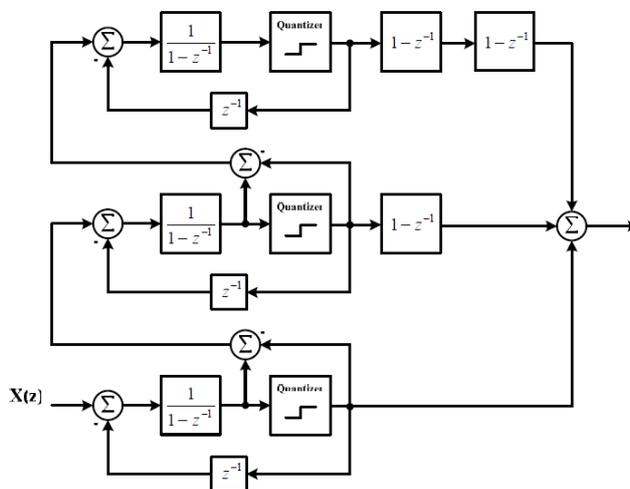


Figure 13.13.7: Discrete time model of the MASH 1-1-1 $\Sigma\Delta$ modulator

On the other hand, the single-loop modulator consists of a single, discrete time filter, which determines the NTF. The order of the filter defines the order of the modulator. The loop filter can be realized using one of the two architectures: Cascaded Integrators with Distributed Feedback (CIFB) as shown in Fig.8 or Cascaded Integrators with Distributed Feed forward (CIFF) as shown in Fig. 9. Although the single-loop modulator is more complex than the MASH modulator, it offers a much higher flexibility in terms of noise shaping. As the NTF of the modulator becomes more aggressive (higher pass band gain), the stable input range decreases.

The use of a multi-bit quantizer can solve this problem at the expense of increasing the number of output levels.

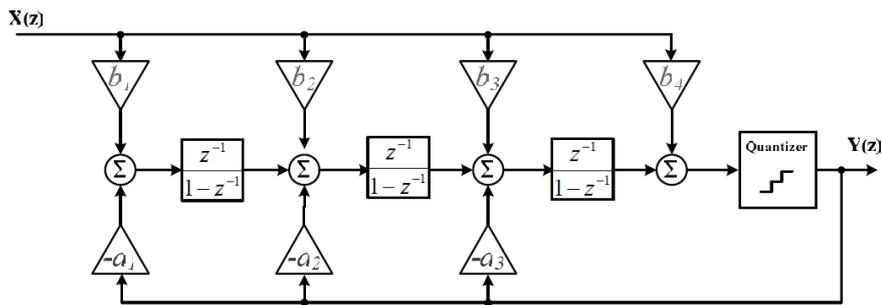


Figure 13.13.8 : Discrete time model of the 3rd order single loop modulator using CIFB

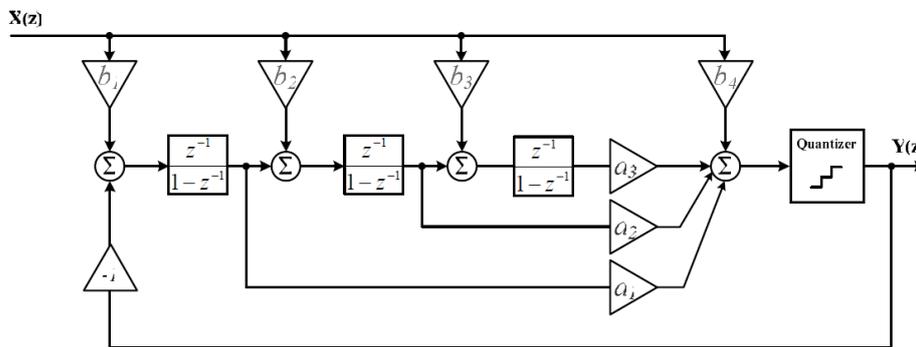
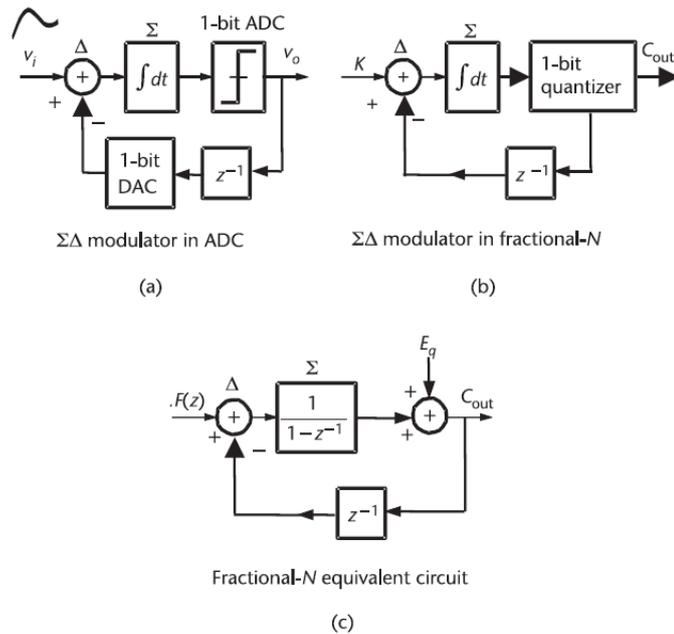


Figure 13.13.9: Discrete time model of the 3rd order single loop modulator using CIFF

13.7.1 First Order Sigma Delta Modulator

In fractional N-Synthesis, the fractional spurs come from the fractional accumulator where the input is a constant frequency word in digital format, and the output is the 1-bit carry-out used to control the modulus of the divider, adding the SD blocks to a fractional $-N$ accumulator, we obtain a SD modulated accumulator, as shown in Figure 10



The SD fractional-N output is found to be

$$C_{out}(z) = \frac{1}{1 - z^{-1}} \cdot \frac{z^{-1}}{1 + \frac{z^{-1}}{1 - z^{-1}}} \cdot F(z) + \frac{1}{1 + \frac{z^{-1}}{1 - z^{-1}}} E_q(z) \quad (16)$$

$$= F(z) + (1 - z^{-1}) E_q(z)$$

Where $F(z)$ is the input to the fractional accumulator, and $E_q(z)$ is the quantization error introduced, note that $E_q(z)$ is high pass filtered while $F(z)$ remains constant.

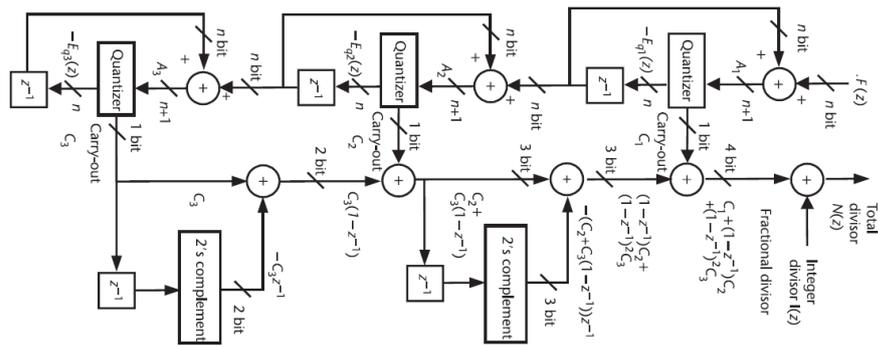
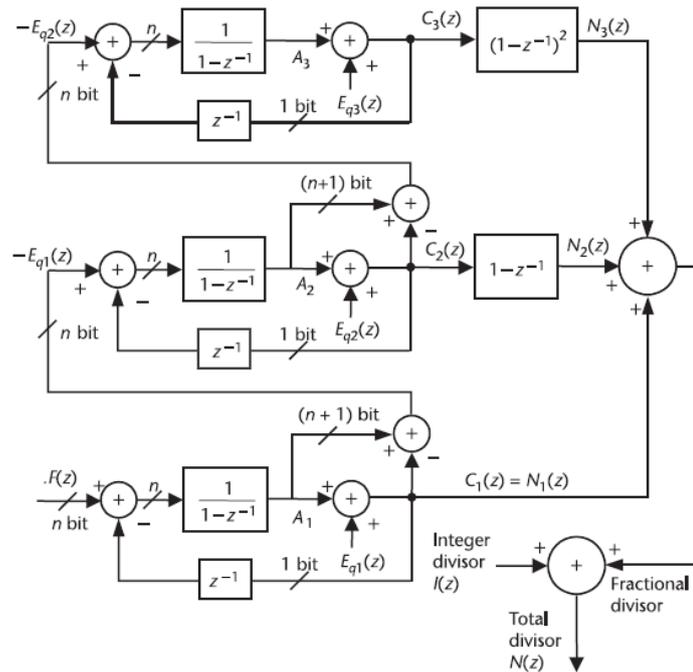
13.7.2 MASH 1-1-1 Modulator

The MASH 1-1-1 consists of three cascaded first order Modulators to achieve a third order modulator, the three loops are identical, only the first loop is fed the constant fractional division ratio. The output of the second loop is taken from the quantized error E_{q1} of the first loop, while the input of the third loop is taken from the quantized error E_{q2} of the second loop

Its transfer function is given by [1]

$$N(z) = I(z) + N_1(z) + N_2(z) + N_3(z) = I(z) + F(z) + (1 - z^{-1})^3 E_{q3}(z) \quad (17)$$

Where $I(z)$ and $F(z)$ are the integer and fractional portions of the division ratios respectively, the Integer and fractional parts are not affected by the modulator while the quantization noise generated by the last loop E_{q3} is modulated by a third order high pass function $(1 - z^{-1})^3$, the quantization error generated by the first and second loop are totally cancelled. The multi loop architecture provides high order noise shaping without adding additional noise.



The MASH 1-1-1 structure is a widely used topology due to its stability, high order in-band noise-shaping characteristic, and easy implementation. As shown, a SD modulator dithers the loop division ratio around its average value. Instantaneously, there are always small phase errors for a SD modulated PLL. However, the average phase error ought to be zero in order for the loop to lock to the desired frequency. Unfortunately, for a multiloop

MASH SD modulator, the higher the order of the modulator, the larger the phase error it causes. To improve the phase error distribution without degrading the noise-shaping slope, alternative topologies need to be investigated. The same noise shaping characteristics could be achieved using a combination of lower order modulators, like the MASH 1-2. Shown in the Figure below.

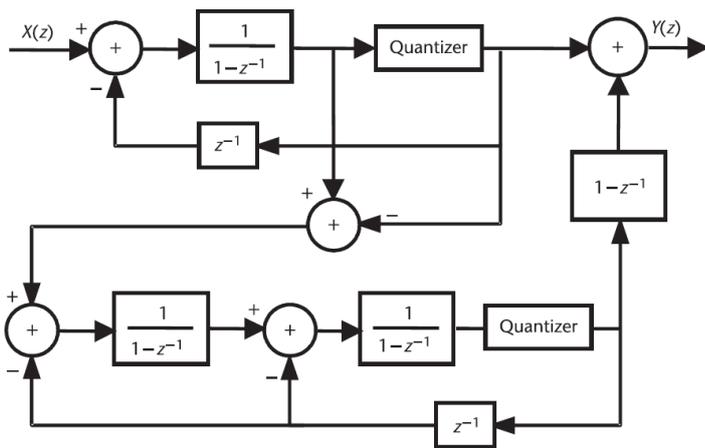


Figure 13.13.13: a third order MASH 1-2 topology

13.7.3 Single Stage Modulator

A single-stage SD modulator is proposed in [1]. Conceptually, by inserting a block with a transfer function of $H(z) = 1 - H_e(z)$ in an accumulator as shown in Figure 14, the accumulator output becomes,

$$Y(z) = X(z) + A(z)H(z) - A(z) = X(z) - E(z)H_e(z) \quad (18)$$

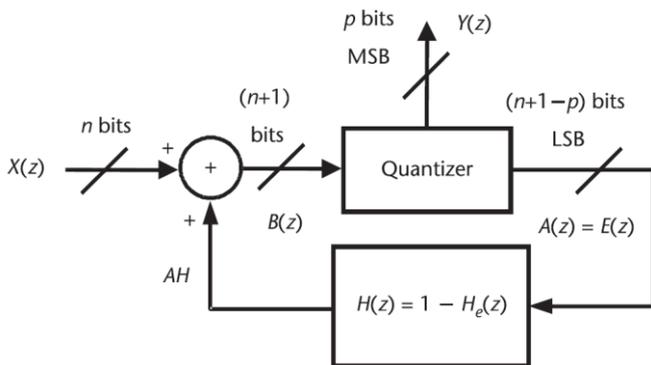


Figure 13.13.14: Conceptual drawing of the single-stage SD modulator.

Where $Y(z)$ is the most significant p bits of the adder output $B(z)$, and $A(z)$ is the remaining $(n + 1 - p)$ bits of the adder output $B(z)$. It is evident that the input signal $X(z)$ is not affected by the modulator, while the quantization noise $E(z)$, which is a truncated word $A(z)$, is filtered by the noise transfer function of $H_e(z)$. If the noise transfer function $H_e(z)$ is the high pass transfer function $(1 - z^{-1})^m$, If an input frequency word $X(z)$ has n bits, $B(z)$ should have $(n + 1)$ bits to include the carry-out, and $A(z)H(z)$ cannot exceed n bits. The modulator output (z) can be of any number of bits, offering flexibility in choosing the number of output bits. However, the maximum number of bits for $A(z)H(z)$ should be carefully calculated to prevent overflow of the adder.

For $m = 3$, $H(z) = 1 - (1 - z^{-1})^3 = z^{-1}(3 - 3z^{-1} + z^{-2})$, the implementation of the third-order, single-stage SD modulator is given in Figure 9.26. Again, the subtraction is implemented using two's complement format. Multiplication by

By dividing by 2 to get the SSB and taking 10log result we can get the output phase noise in dBc/Hz

$$PN(f) \text{ [dBc/Hz]} = 10 \log \left\{ \frac{(2\pi)^2}{24f_r} \cdot \left[2 \sin \left(\frac{\pi f}{f_r} \right) \right]^{2(m-1)} \right\} \quad (25)$$

13.9 Calculation of the Required Resolution

The most straightforward and most common approach to realizing a fractional-N frequency synthesizer is to use a high-resolution SD modulator and to approximate the required fraction with sufficient accuracy [2]. This approach is based on the most hardware- efficient truncation. For such an implementation, the SD resolution is a power of two, $Q = 2^k$, and the step size of the synthesizer can be expressed as:

$$\Delta f = \frac{f_r}{2^k} \quad (26)$$

This leads to minimum required width of the SD modulator of

$$k \geq \left\lceil \log_2 \frac{f_r}{2\Delta f} \right\rceil = \left\lceil \log_2 \frac{z}{z'} \right\rceil = 17 \text{ bits} \quad (27)$$

13.10 Dithering Techniques

The phase error PSD given in (9.80) is derived based on a uniform quantization model. For the first-order SD modulator, the accumulator output spectrum is highly dependent upon its dc input. When only input bits near the MSB are nonzero (e.g., $K/F = 1/4, 1/2, 3/4$), the accumulator output cycle repeats often, resulting in insufficient randomness to decorrelate the quantization error. In this case, the uniform quantization model is not quite appropriate. Thus we need to randomize the input of the accumulator.

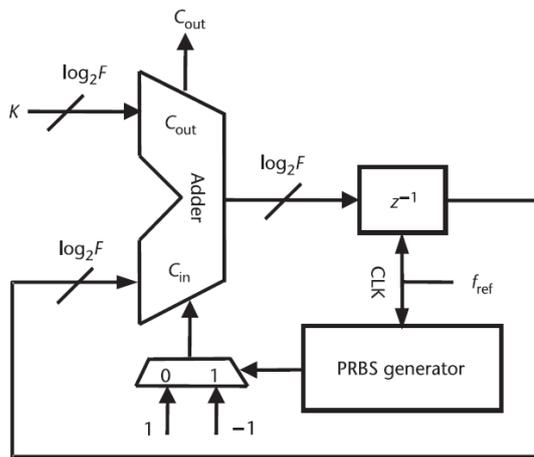
13.11 Deterministic techniques

Applying an initial condition to the accumulator of the first stage by setting its value to an odd number guarantees a minimum sequence length of 2^{n+1} where n represents the accumulator size.

Another method would be setting the LSB of the input to 1 to guarantee that the input is a prime number, which also results in a sequence length of a minimum value of 2^{n-1} .

13.11.1 Stochastic techniques

To randomize the tonal fractional accumulator output, a *pseudo random bit sequence* (PRBS) generator, with equal probability of ones and zeros, can be employed to dither the accumulator input value, as demonstrated in Figure 16. The PRBS outputs selectively add an equal number of 1's and -1's to the carry-in of the adder. While it does not affect the average accumulator input value, it does decorrelate the accumulator output. Without dithering, the quantizer produces highly correlated errors that create harmonic-distortion components without additional energy in between.



Additional hardware (as we need minimum size for the LFSR for efficient dithering normally we use LFSR size equals to accumulator size) , also LFSR is a finite state machine which will has a repeated output so to get an ideal dithering we need a very large LFSR (linear feedback shift register).

It also degrade the low noise component of the quantization noise by adding its noise floor at the input. This problem can be solved by passing the random number through a high pass filter of $(1 - z^{-1})$, or by entering the random generator inside the loop before the quantizer so that its noise floor can be shaped by using the loop but this may affect quantizer dynamic range . For these reasons we opted to set the LSB of the input to 1 to dither our modulator.

13.12 Types of Quantizers

13.12.1 Truncation Quantizer SDM

In Truncation SDM we can only have a modulus of which is implemented by simply truncation the most significant bit to be the quantizer output ,then for certain ratios we can't get it with perfect accuracy , so we are limited to minimum number of bits to achieve the required accuracy.

13.12.2 Variable Modulus SDM

Adding some logic circuits give us the ability to get any variable quantizer modulus. As shown we have 2 inputs x & Q where x is the input value while Q is the value of the arbitrary modulus by adding adder and Mux we make the output y equals 0 as long as accumulated input $< Q$, And 1 when accumulated input $> Q$. The main disadvantage that we need very efficient dithering technique so we can suppress the output spurs.

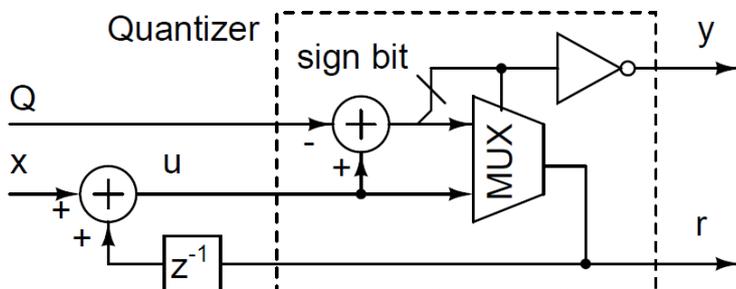


Figure 13.13.17 : A first-order DSM with a single-bit variable modulus quantizer

13.13 Simulations & Results

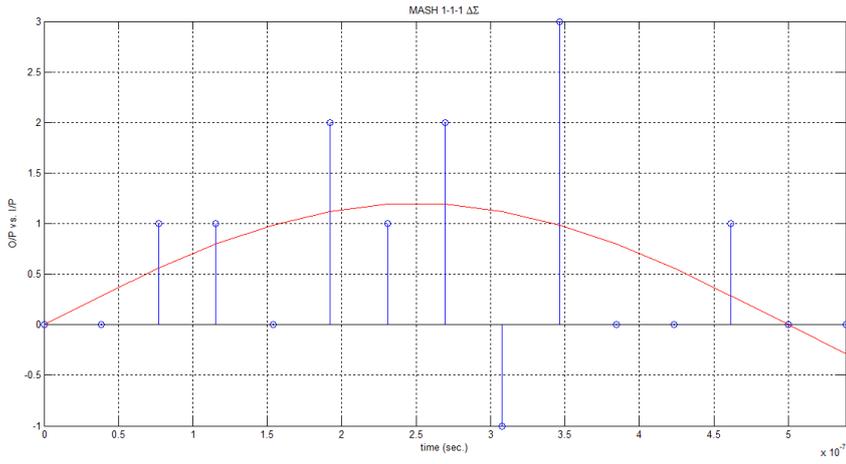


Figure 13.13.18: Output levels for a sinusoidal input of amplitude 1.2 V

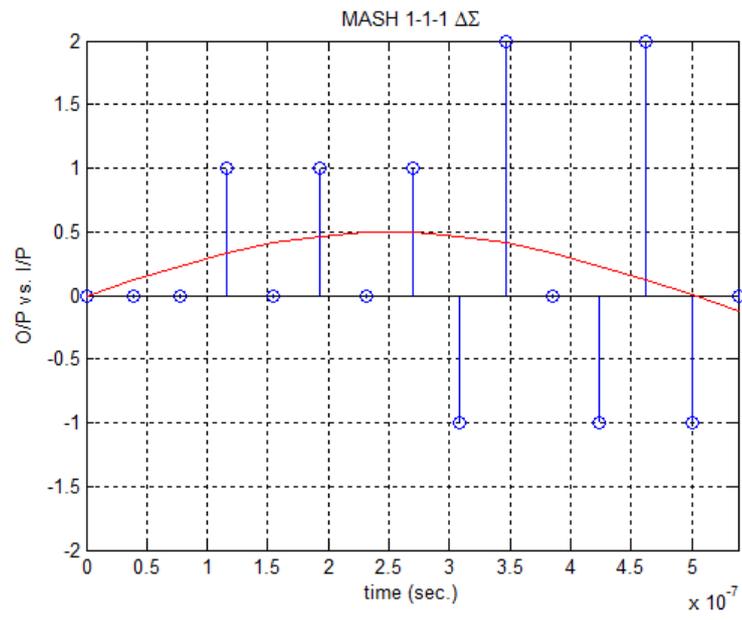


Figure 13.13.22 : Output Levels for a sinusoidal input of 0.5 V

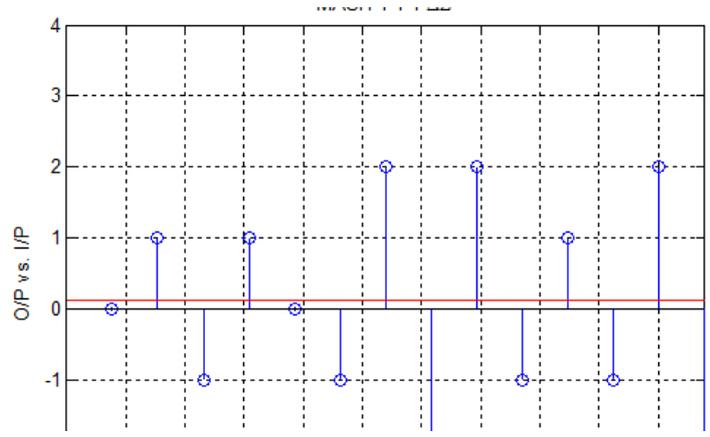
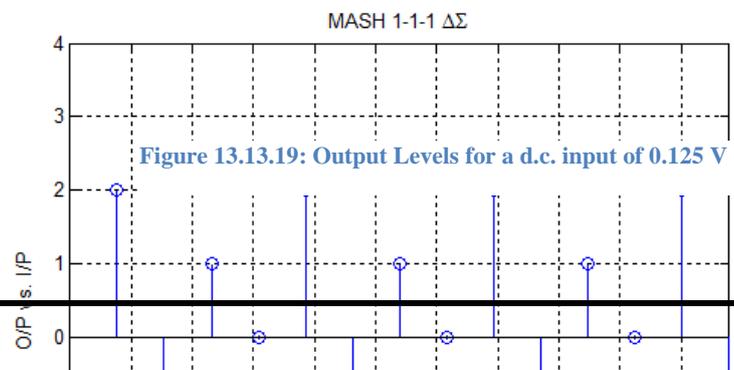


Figure 13.13.19: Output Levels for a d.c. input of 0.125 V



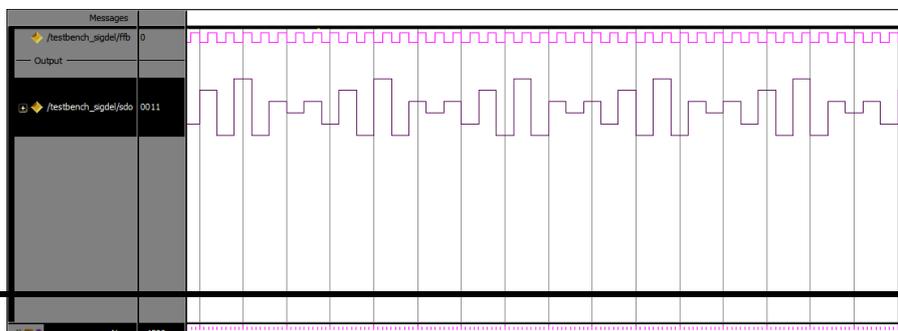
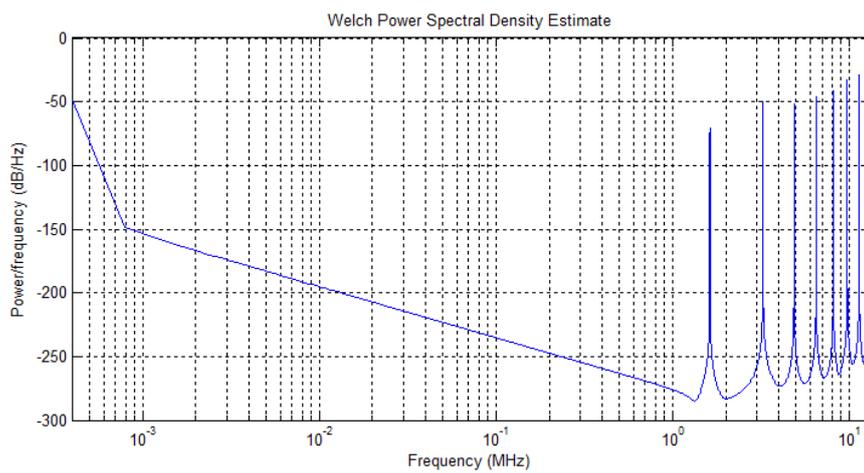
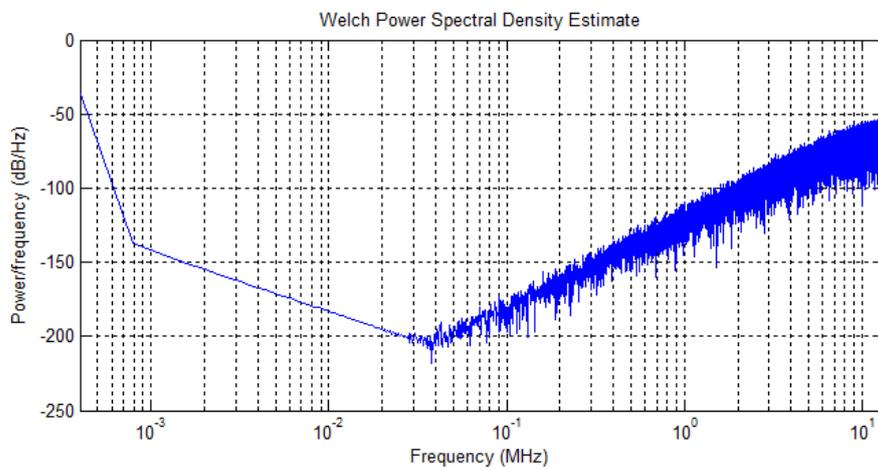
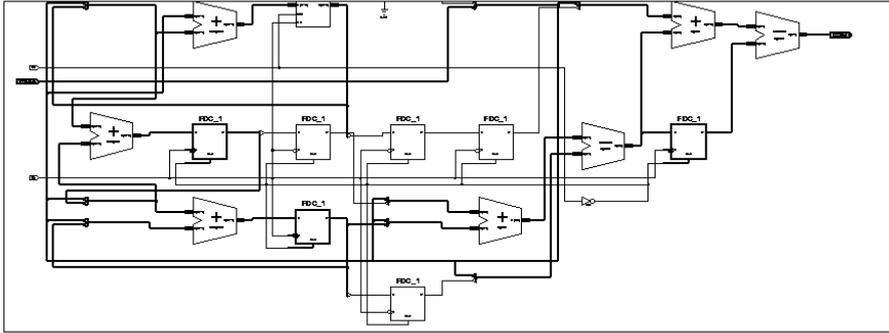


Figure 1313.23: Output power spectral density of MASH 1-1-1 without dithering



13.14 Conclusion

In this work, the main issues concerning the design of Sigma Delta Modulators for fractional-N PLL are summarized. A comparison between different architectures is presented, a 17 bit MASH 1-1-1, LSB dithered, single bit truncation quantizer is designed and successfully implemented for a fractional-N frequency synthesizer with a reference frequency of 26 MHz, and channel spacing 750 Hz, and was designed for an even finer resolution, 200 Hz than required to compensate for any imperfections in the PLL's Crystal oscillator

13.15 Acknowledgements

I would like to thank everyone who has assisted me during the course of the project and dedicated time for my questions, I especially appreciate the assistance of Eng. Ahmed Saad, Eng. Mohamed Mohsen & Eng. Sherif Daa from Silicon Vison, our project sponsor, I would also like to thank Eng. Ahmed El Kholy and Eng. Samer Bahr for their helpful insights, along with all the staff of the IC Laboratory at Ain Shams University, Most importantly I would like to thank Prof. Hani Fikry for supervising our project and providing guidance.

1.1 References

- [1] John Rogers & Calven Plett. Radio Frequency Integrated Circuit. Design. ArtechHouse,Inc.,2003
- [2] Maciej Borkowski , DIGITAL Σ - Δ Modulation variaable modulus and tonal behaviour in a fixed pint digital enviroment , OULU 2008
- [3] Woogeun Rhee,multi-bit delta-sigma modulation technique for fractional- n frequency synthesizers, 2000
- [4] A. El-Kholy et al, "A wide spreading range programmable spread spectrum clock generator using Σ 7 fractional-N PLL", *IEEE Northeast Workshop on Circuits and Systems*, June 2009
- [5] R. Schreier, and G. Temes, *Understanding Delta-Sigma Data Converters*. New York: Wiley, 2005.
- [6] K. Shu and E. Sanchez-Sinencio, *CMOS PLL Synthesizers: Analysis and Design*, Springer, 2006.
- [7] B. De Muer and M. Steyaert, *CMOS Fractional-N Synthesizers Design for High Spectral Purity and Monolithic Integration*. Norwell, MA: Kluwer, 2003

14 Power amplifier

14.1 Introduction

Power amplifiers are the most power-hungry building block of RF transceivers & pose difficult design challenges. ^[1] PA is one of the most critical building blocks in low power SOC integration. This work presents inverse class F power amplifier for low power applications. Using a non-linear power amplifier of inverse class F has the advantage of higher efficiency than a linear power; but then a good filter at the output is required.

In low power application a small load value is not required since the power amplifier is not expected to handle high power levels. Since low power PAs requires larger output loads than higher power PAs, larger RF choke is required; but these inductors have low quality in our design a quality factor of 30 is used. As a result, PAs designed for low-power application have considerably lower efficiency than high power PAs. For that reason this work cannot be compared to previously published inverse class F or any non-linear class, since they are all higher power PAs.

This work presents programmable inverse class F with a CMOS circuit as a driver stage operating at a center frequency of 915MHz, which provide hard switching required for our non-linear classes. A CML and CML to CMOS is used as a pre-amplifier with two main purposes to provide high gain to input signal and to convert from differential to signal ended instead of using transformer to reduce circuit complexity.

The required specification is to provide +11dBm at fundamental frequency of 915MHz with a minimum attenuation for second harmonic of -34 dBm and other harmonics -50 dBm. The current budget is 25 mAmpere.

The specification required is achieved across all corners, in typical-typical case: output power at fundamental frequency of +11.44 dBm, Second harmonic of -37.29 dBm, and Third harmonic of -61.58 dBm with a current 23.58 mAmpere and PAE of 23.62%. The DC programmability provides 256 different levels for output power, but it is recommendable to use 8 levels only.

The sections are organized as follows: section 1.2 general consideration discussing concepts that are used in any general power amplifier, section 1.3 Topology survey, section 1.4 comparisons between topologies, section 1.5 Design methodology showing all designs schematic and used optimization methods and section 1.6 Simulated results and test benches.

14.2 General considerations

14.2.1 Efficiency

Since PAs are the most power- hungry block in RF transceivers, their efficiency is critical. The PA draws much more than the rest of transceiver do ^[1]. The efficiency is defined by two figure of merits Drain efficiency and power-added efficiency.

1. Drain efficiency $\eta = \frac{P_L}{P_{supply}}$ Where P_L denotes average power delivered to the load and P_{supply} denotes average power drawn from the load.
2. Power-added efficiency "PAE" = $\frac{P_L - P_{in}}{P_{supply}}$ Where P_{in} denotes average power input power
PAE $\leq \eta$.

PAE is more accurate measure for efficiency, as in some cases the output stage has low gain and some of the input leak to the load through parasitic capacitance at high frequency.

14.2.2 Linearity

PA linearity doesn't mean PA circuit is linear as output stage of PA exhibits a large swing on input leading a non-linear gm in all classes. PA nonlinearity must be characterized with respect to the modulation scheme of interest.^[1] One of the most famous trade-offs is Linearity, efficiency trade off "discussed later".

PA nonlinearity will leads in two effects

1. High adjacent channel power as a result of spectral regrowth
2. Amplitude compression

Modulation schemes is divided into two main categories

1. Constant envelope
2. Variable envelope

14.2.3 Single-Ended and Differential PAs

Most stand-alone PAs have been designed as a cascade of single ended stages 2 reasons for this choice

1. Antenna is typical single ended
2. Single RF circuits are much simpler to be tested then differential counterparts

Draw backs of single ended amplifier

2. Wasted half of the transmitter voltage gain solved by balun
3. large transient currents alters the resonance freq. or impedance transformation of the output network if it is comparable to L_D
4. L_{B1} allows some of the output stage signal to travel back to the preceding stage(s) through V_{DD} line causing ripple in frequency response or instability
5. L_{B2} degenerates the output stage & introduce feed back

Differential realization greatly eases these 2 issues

1. Draws small transient currents from V_{DD} to ground lines
2. Less sensitive to L_{B1} & L_{B2} & less feed back
3. Enhances voltage gain & packing issues
4. Lower coupling to LO which reducing pulling

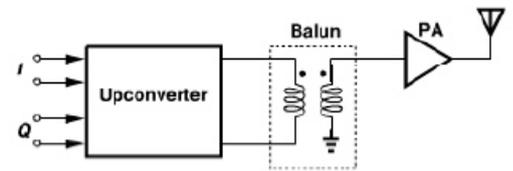


Figure 14.14.1 : Upconverter with balun connection^[1]

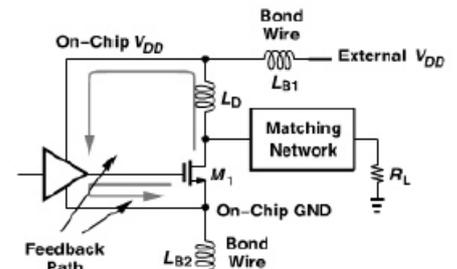


Figure 14.14.2 : Feedback in a single ended PA due to bond wire^[1]

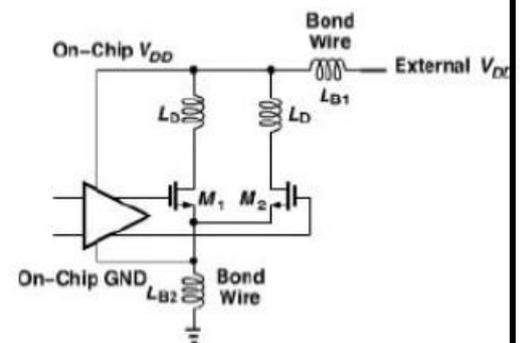


Figure 14.14.3 : less problematic situation in a differential PA^[1]

14.2.4 Cascode output stage VS common source stage

All classes' efficiency calculation, the drain waveform is assumed to have peak to peak nearly $2V_{DD}$ (Breakdown issue & substantial stress). One can choose V_{DD} equal to half maximum tolerable voltage with 2 penalties.

1. Lower headroom limits the linear voltage range of circuit
2. The proportionality higher output circuit (for a given output power) leads to a greater loss in the output matching network reducing the efficiency

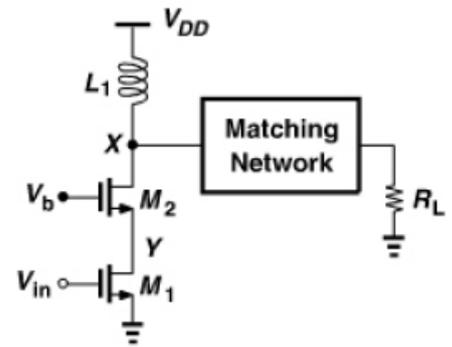


Figure 14.14.4 : Cascoded PA [1]

Cascode shields the input transistor as V_x rises, If $V_b = V_{DD}$ which leads to $V_p \leq V_{DD} - V_{th2}$ peak to peak voltage is limited to $2V_{DD} - 2V_{th2}$.

The common source stage remains linear across a wider output voltage range than cascode circuit, at low voltage cascode offers a slightly voltage swing advantage over their CS counterparts by using M2 of low threshold Mosfet.

Cost of cascode is efficiency & linearity, Cascode have small ($|S_{12}|$) reverse isolation, cascode stage experience less feedback “improving stability”, while a common source stage may suffers from a negative resistance [1].

14.3 Topology survey

Power amplifiers are divided into two main categories linear classes and non-linear classes. The calculated efficiency in all classes is maximum theoretical efficiency.

14.3.1 Linear classes

The linear operation of power amplifiers is region where the transistor is always on, which is different from the general definition of linearity (expressing nearly constant g_m). In linear classes the transistor acts as VCCS.

14.3.1.1 Class A PAs

Class A amplifiers are defined as circuits in which the transistors(s) remain on and operate linearly across the full input & output range. The transistor is chosen to be higher than peak signal current, to ensure that transistor doesn't turn off at any point during the signal excursion. [1]

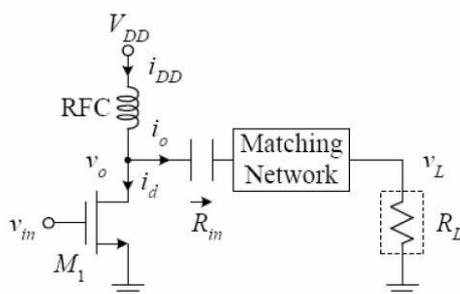


Figure 14.14.6 : General power amplifier model

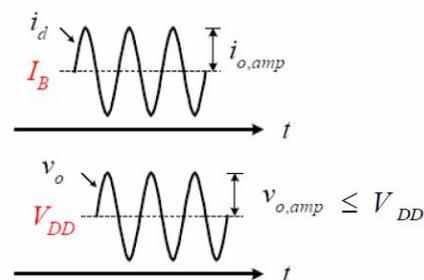


Figure 14.14.5 : Voltage & current waveform

Efficiency calculation:

The current flowing into transistor is divided in two currents constant current (dc) drawn from RFC & RF current from load.

$$\text{Power delivered to load } P_L = V_{L\text{ rms}} I_{RF\text{ rms}} = \frac{V_{DD}}{\sqrt{2}} \times \frac{I_{RF}}{\sqrt{2}} = \frac{V_{DD} \cdot I_{RF}}{2}$$

$$\text{Power supply} = V_{DD} \cdot I_B$$

$$\text{For maximum efficiency } I_{RF} = I_B, \eta = \frac{P_L}{P_{\text{supply}}} = 50\%.$$

Power handling capability:

Power output capability is obtained by dividing power output by peak voltage & current.

$$\text{Power output capability} = \frac{P_{\text{load}}}{V_{\text{peak}} I_{\text{peak}}} = \frac{\frac{V_{DD} \cdot I_{RF}}{2}}{2V_{DD} \cdot 2I_{RF}} = \frac{1}{8}$$

Conduction angle:

It is defined as % of signal period during which the transistor(s) remains on multiplied by 360°. Conduction angle is sometimes helpful to distinguish PA classes from the output transistor(s).

Main features in class A PAs:

- Transistor is always conducting which translates into poor drain Efficiency (< 50%)
- Typically the maximum efficiency is less than 50% due to the effect of finite $V_{ds,sat}$
- Excellent linearity, but linearity is not maintained for hard-driven class A
- Normalized power capability = 1/8
- Conduction angle of 360°

Assumptions in η calculation in class A which lead to 50%

1. The drain peak-peak voltage swing is equal to twice the supply voltage (“Drain can withstand a drain-source voltage of $2V_{DD}$ with no reliability issue or breakdown issue”)
2. The transistor barely turns off (i.e. nonlinearity resulting from the very large change in the trans-conductance(g_m) of device is tolerable)
3. The matching network interposed between the output transistor & the antenna is lossless

14.3.1.2 Class B PAs

A clue to how one might achieve higher efficiency than a class A amplifier is actually implicit in the waveforms of Figure 4. It should be clear that if the bias were arranged to reduce the fraction of a cycle over which the drain current and drain voltage are simultaneously nonzero, transistor dissipation would diminish. [2]

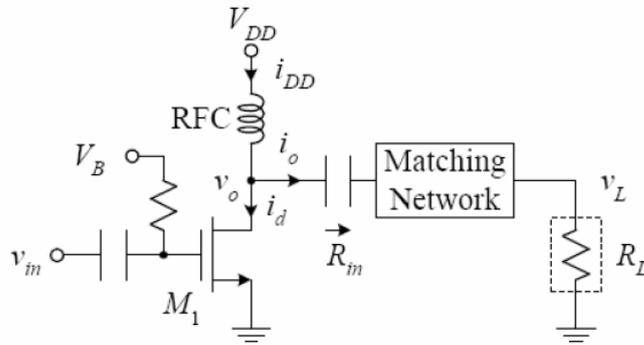


Figure 14.14.8 : Class B power amplifier

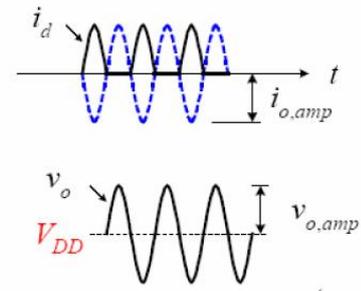


Figure 14.14.7 : Current & Voltage waveforms

Setting the value of $V_B = V_{th}$, while lead to $I_B = 0$. If $V > V_{th}$ the transistor(s) conducts otherwise the transistor(s) is off. Device is on for 50% (conduction angle of 180°).

The circuit creates a lot of harmonics, a tank circuit will be used to allow fundamental to pass & reject other harmonics.

Average $I_d = I_{avg}$, I_{avg} : average current drawn from supply $P_{consumed} = V_{DD}I_{avg}$

Efficiency calculation:

Using Fourier series

$$I_{d\ avg} = \frac{1}{2\pi} \int_0^{2\pi} I_d\ d\theta = \frac{1}{2\pi} \int_0^\pi I_{max}\ \sin\ \theta\ d\theta = \frac{I_{max}}{2\pi} (1 - -1)$$

$$I_{d\ avg} = \frac{I_{max}}{\pi}$$

1st Harmonic component of current

$$I_{1st} = \frac{1}{\pi} \int_0^{2\pi} I_d\ \sin\ \theta\ d\theta = \frac{1}{\pi} \int_0^\pi I_{max}\ \sin\ \theta\ \sin\ \theta\ d\theta$$

$$I_{1st} = \frac{1}{2\pi} \int_0^\pi I_{max}(1 - \cos\ 2\theta)\ d\theta$$

$$I_{1st} = \frac{I_{max}}{2}$$

$$P_L = \frac{1}{2} I_{1st}^2 \times R_{out} = \frac{1}{8} I_{max}^2 \times R_{out}$$

In order to provide max swing $\frac{I_{max}}{2} = V_{DD} \times R_{out}$

$$P_L = \frac{1}{4} V_{DD} I_{max}$$

$$P_{supply} = V_{DD} I_{avg} = \frac{V_{DD} I_{max}}{\pi}$$

$$\eta = \frac{P_L}{P_{supply}} = \frac{\pi}{4} = 78.5\%$$

Power handling capability:

$$\text{Power output capability} = \frac{P_{load}}{V_{peak} I_{peak}} = \frac{V_{DD} \cdot I_{RF}}{2V_{DD} \cdot I_{RF}} = \frac{1}{8}$$

Main Features of Class B PAs:

- Transistor is conducting for half cycle
- Typically the maximum efficiency is 78.5%
- Class B is less linear compared to class A
- Normalized power capability is same as class A = 1/8
- Conduction angle of 180°

14.3.1.3 Class C PAs

In class C PAs the gate bias is arranged to cause the transistor to conduct less than half the time, It is clear that as the conduction angle gets smaller the efficiency increases and the circuit becomes more non-linear.

Setting the value of $V_B < V_{th}$, while lead to $I_B = 0$ If $V > V_{th}$ the transistor(s) conducts otherwise the transistor(s) is off. Device is on for less than 50% (conduction angle is less than 180°).

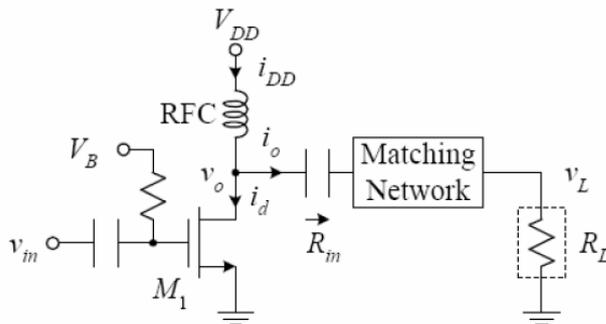


Figure 14.14.9 : Class C power amplifier

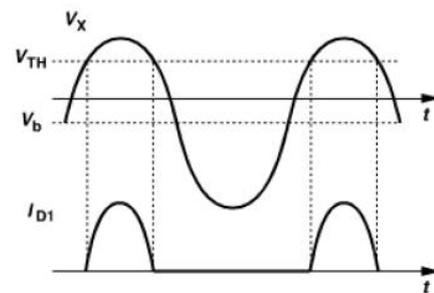


Figure 14.14.10 : Current & Voltage waveforms

$$\eta = \frac{\text{Fundamental power}}{\text{Total power}}$$

$$I_D = \begin{cases} I_{RF} \cos \theta - I_{RF} \cos \phi & \text{for } -\phi < \theta < \phi \\ 0 & \text{for all other } \theta \end{cases}$$

Where 2ϕ is region when Mosfet is conducting current ($V_g > V_{th}$)

Efficiency calculation:

Using Fourier series:

$$I_{d \text{ avg}} = \frac{1}{2\pi} \int_0^{2\pi} I_d d\theta = \frac{1}{2\pi} \int_{-\phi}^{\phi} I_{RF} \cos \theta - I_{RF} \cos \phi d\theta$$

$$I_{d \text{ avg}} = \frac{I_{RF}}{\pi} [\sin \phi - \phi \cos \phi]$$

$$\text{Current 1}^{\text{st}} \text{ Harmonic component } I_{1st} = \frac{1}{\pi} \int_0^{2\pi} I_d \cos \theta d\theta$$

$$= \frac{1}{\pi} \int_{-\phi}^{\phi} (I_{RF} \cos \theta - I_{RF} \cos \phi) \cos \theta d\theta$$

$$I_{1st} = \frac{I_{RF}}{2\pi} [2\phi - \sin 2\phi]$$

$$P_{delivered} = 0.5 V_{DD} I_{d1st} \quad P_{consumed} = V_{DD} I_{d avg}$$

$$\eta = \frac{P_{delivered}}{P_{consumed}} = \frac{1}{4} \frac{[2\phi - \sin 2\phi]}{[\sin \phi - \phi \cos \phi]}$$

If $\phi = \frac{\pi}{4}$: $\eta = 93.73$ % class C

If $\phi = \frac{\pi}{2}$: $\eta = 78.5$ % class B

If $\phi = \pi$: $\eta = 50$ % class A

As the conduction angle decrease, the average current decreases

$I_{d avg} = \frac{I_{RF}}{\pi} [\sin \phi - \phi \cos \phi]$ at $\frac{\pi}{4} = 0.0484 I_{RF}$ which means that the current delivered to the load is going down as conduction angle decreases. $P_{consumed}$ & $P_{delivered}$ both decreased as conduction angle decreases until everything is off.

Drain current contains many harmonics that have to be filtered out before transmission (Tank circuit)

Main Features of Class C PAs:

- Transistor is conducting for less than half cycle depending on bias
- Typically the maximum efficiency is bigger than 78.5%
- Class C is less linear compared to class B
- Conduction angle is chosen between 180° & 0°
- Normalized power capability less than 1/8 depending on conduction angle
- Requires smaller load resistances than class A that translates into bigger devices which is harder to realize and effect of parasitics is more critical

14.3.1.4 Class AB PAs

As seen that Class A amplifiers conduct 100% of the time, Class B amplifiers 50% of the time, and Class C PAs somewhere between 0 and 50% of the time. The Class AB amplifiers, as its name suggests conducts somewhere between 50% and 100%, depending on the bias levels chosen. [2]
Typically AB class is best comprise between A, B, C.

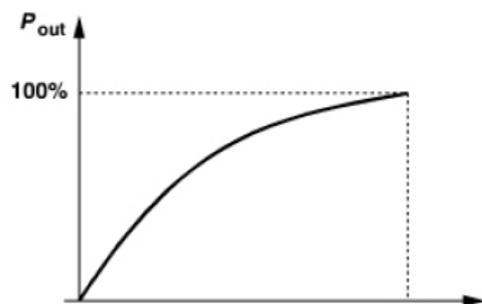
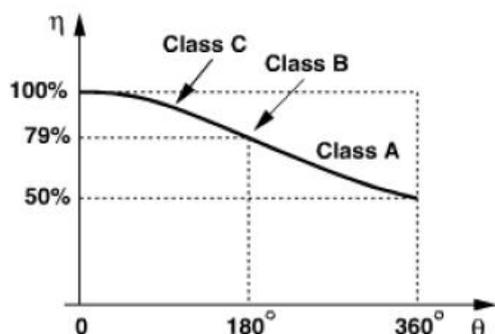


Figure 14.14.12 : Efficiency VS condition angle [1]

Figure 14.14.11 : Output power VS condition angle [1]

Main Features of Class AB PAs:

- Conduction angle is chosen between 360° & 180°
- Linearity is acceptable in many applications, while efficiency is between that of class A and class B

14.3.2 Non-linear classes

The transistor(s) is hard-driven i.e. transistor(s) behaves like a switch or voltage controlled resistance (VCR). Non-linear classes tend to achieve efficiency approaching 100% while delivering full power, a remarkable advantage over class C amplifiers.^[1] The non-linear classes is divided into two main categories zero voltage switching (ZVS) and zero current switching (ZCS).

In order for transistor to be hard-driven it requires 3 conditions

1. M1 sustains a small voltage when it carries current
2. M1 carries a small current when it sustain a finite voltage
3. The transition times between the on & off are minimized

We conclude from 3 previous conditions

1. Overlap between current & voltage is equal Zero
2. Transistor power consumption equal Zero
3. The On-resistance of the switch must be minimized

14.3.2.1 Class D PAs

In class D PAs the transistor experience a switching action on gate leading to the generation of all harmonics at output, which requires a tank circuit to filter the fundamental component from all other harmonics.

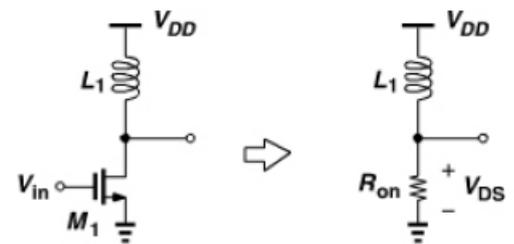


Figure 14.14.13 : Output stage with switching transistor^[1]

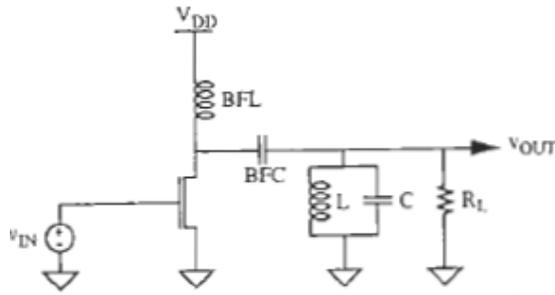
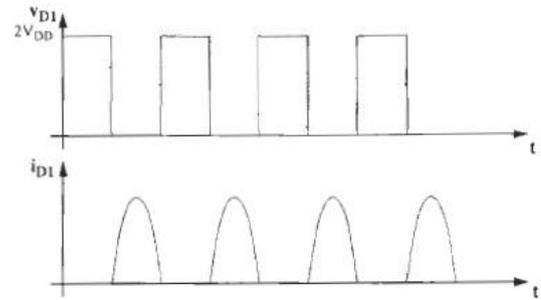


Figure 14.14.15 : Class D power amplifiers [2]



Efficiency calculation:

Figure 14.14.14 : Voltage & current waveforms [2]

At first instant efficiency is consider 100%, but let us take a close look

$$\eta = \frac{\text{Fundamental power}}{\text{Total power}}$$

$$I_D = \left[\begin{array}{ll} I_{max} & \frac{\pi}{2} < \theta < \frac{\pi}{2} \\ 0 & \frac{\pi}{2} < \theta < \frac{3\pi}{2} \end{array} \right]$$

$$I_{d1st} = \frac{1}{\pi} \int_0^{2\pi} I_d d\theta = \frac{1}{\pi} \int_{-\pi/2}^{\pi/2} I_{max} \cos \theta d\theta = \frac{I_{max}}{\pi} (1 - -1) = \frac{2I_{max}}{\pi}$$

$$P_{load} = \frac{I_{d1st} V_{DD}}{2} = \frac{I_{max} V_{DD}}{\pi} \rightarrow \text{Power delivered at desired freq.}$$

$$P_{supplied} = \frac{V_{DD} I_{max}}{2}$$

$$\eta = \frac{P_{load}}{P_{supplied}} = \frac{2}{\pi} = 63.69 \%$$

Efficiency is 100% for all harmonic's (1st, 2nd, 3rd,), but we desire only the fundamental harmonic.

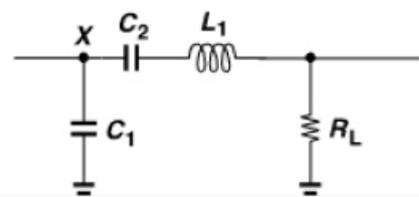
Power handling capability:

$$\text{Power output capability} = \frac{P_{load}}{V_{peak} I_{peak}} = \frac{\frac{V_{DD} \cdot I_{max}}{\pi}}{2V_{DD} \cdot I_{max}} = \frac{1}{2\pi}$$

Main features in class D PAs:

- Transistor acts a VCR
- Transistor experience a hard switching
- Highly non-linear
- All harmonics is generated at output
- Requires a tank circuit at output
- Max theoretical efficiency equal 63.69 %
- Power output capability equal $\frac{1}{2\pi}$ which is higher than class A PAs

14.3.2.2 Class E PAs



As seen using transistor as switch has the potential for providing greatly improved efficiency, but it's not always trivial to realize that potential in practice due to imperfections in real switches. The associated dissipation degrades efficiency to prevent gross losses; the switch must be quite fast relative to the frequency of operation. [2]

Class E amplifiers force a zero switch voltage (ZVS) for a nonzero interval of time about the instant of switching, Class E amplifiers deals with transitions time with proper load.

Class E load must satisfy 3 conditions

1. As the switch turns off , V_x remains low enough for current to drop to zero (i.e. V_x & I_d have no overlapping)
2. V_x reaches zero just before the switch turns on
3. $\frac{dV_x}{dt}$ is also near zero when the switch turns on

When M1 is on, it shorts node x to ground but carries little current because V_x is already near zero at this time (condition 2) if R_{on1} is small V_x remains near zero & L_D sustains a relatively constant voltage, thus carrying current of $I_d = (1/L_D)$

$$\int [V_{DD} - V_x] dt = [(V_{DD} - V_x)/L_D] t, V_x \text{ is nearly constant} = 0.$$

In other words, one cycle is to charge L_D with minimal drop across M1. When M1 is off the inductor current begins to fall approaching zero with rising V_x , as V_x begins to fall approaching zero with zero slope at the end of the second half cycle condition (2, 3).

An optimal design minimizes the total loss in the load network and the subsequent filter (if any) while meeting a specified maximum limit for harmonic output at load. Equation will be given for the elements in Figure 15. The derivations are too long to give here; there are available from the authors. [5]& [6]

$$R = 0.577 \frac{(V_{cc} - V_{ds \text{ min}})^2}{\text{Output power}}$$

Impedance transformation can be used if the load resistance is not equal to this value of R.

$$L = \frac{Q_l R}{2\pi f}$$

$$C_1 = \frac{1}{2\pi f R * 5.447}$$

$$C_2 = C_1 \left(\frac{5.447}{Q_l} \right) \left(1 + \frac{1.42}{Q_l - 2.08} \right)$$

$$P_{out \text{ max}} = 0.577 \frac{V_{dd}^2}{R_L}$$

$$\text{Power handling capability} = \frac{P_{load}}{V_{peak} I_{peak}} = \frac{0.577 \frac{V_{dd}^2}{R}}{3.562 V_{DD} \cdot \frac{1.77 V_{DD}}{R}} = 0.09$$

The choice of Q_l involves a trade-off among:

Figure 14.14.16 : Class E load [1]

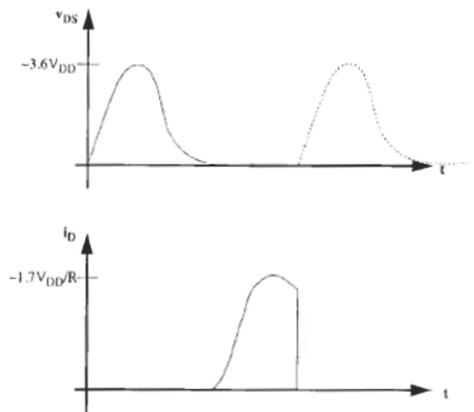


Figure 14.14.17 : Voltage & Current waveforms [2]

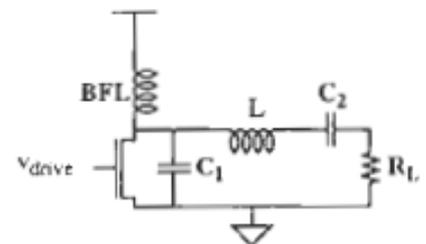


Figure 14.14.18 : Class E power amplifier

1. Low harmonic content of the power delivered to R (high Q_l)
2. High efficiency (low Q_l)
3. Complexity of the filter used for additional harmonic suppressions

Ideal operation of class E requires duty cycle of 50% for maximum efficiency. It could be shown that $V_{D_s peak} = 3.562 V_{dd}$ [7].

As seen the drain voltage rises nearly to $3.562 V_{dd}$. Class E doesn't scale gracefully with the trend towards lower power technology with breakdown voltage. [2]

Main features in class E PAs:

- Design depends main on proper load design
- $V_{D_s peak} = 3.562 V_{dd}$ Which makes it harder for low power technology
- Efficiency is nearly 90%
- Class E is considered as ZVS

14.3.2.3 Class F PAs

A class F power amplifier uses an output filter to control the harmonic content of its drain voltage or the drain current waveforms, there by shaping them to reduce power dissipation

by the transistor and thus increase efficiency.

An ideal class F shapes the waveform of voltage into a square wave, while the current is shaped as a half current waveform. The wave shaping is done by adding odd harmonics to voltage and adding even number of harmonics to current waveform.

Adding infinite number of harmonic makes the design of output network too complex, most designers uses finite number of harmonics and sacrifice some losses as using transmission line may be inconveniently long or even inapplicable in fully on-chip integration . Class F main trade is between output network complexity and efficiency.

The maximum efficiency of an ideal PA increase from 50% of class A to 70.7, 81.6, 86.6 and 90.4 as harmonics added. [8]

It can be shown from author [8] that drain voltage reaches $2 V_{DD}$ and the current reaches πI_{dc} .

The efficiency of class F depends on the number of harmonics added, as the number of harmonics increase the efficiency increase.

Table 14.18: Max efficiency and Output power capability

M	η			
	N= 1	N= 3	N= 5	N = ∞
1	0.5	0.5774	0.6033	0.637

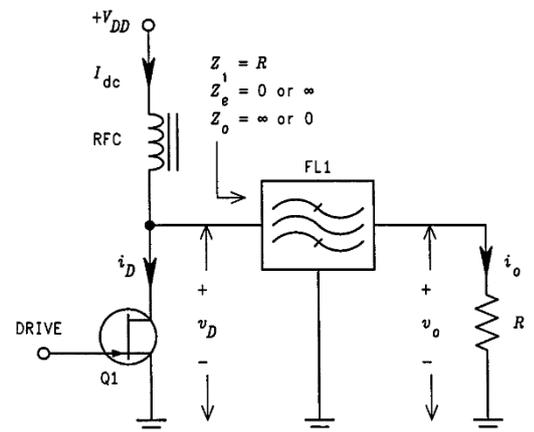


Figure 14.14.19 : Class F power amplifier

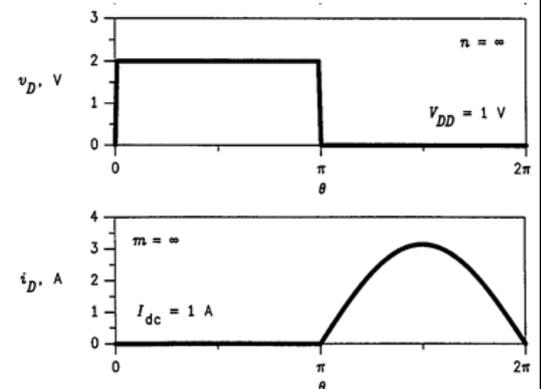


Figure 14.14.20 : Voltage and current waveforms

2	0.7071	0.8165	0.8532	0.9003
4	0.7497	0.8656	0.9045	0.9545
∞	0.785	0.9069	0.9477	1
	P_{max}			
	N= 1	N= 3	N= 5	N = ∞
	0.125	0.1443	0.1508	0.159

M represents number of even harmonics added to the wave, while N numbers of odd harmonics added to the wave.

Main features in class F PAs:

- Depends mainly on harmonic termination to shape voltage and current waveform
- Efficiency is 100% for infinite number of harmonics
- Class F is considered as ZCS
- $V_{Ds} = 2 V_{dd}$ and $i_{D max} = \pi I_{Dc}$ for infinite number of harmonics

14.3.2.4 Inverse class F PAs (F^{-1})

The dual of the class F is itself a power amplifier with the same theoretical bounds on efficiency as its cousin. [9] Whereas the class F amplifier's termination appears as an open circuit at odd harmonics of the carrier beyond the fundamental, and as a short circuit at even harmonics, the inverse class F employs a termination that appears as an open circuit. [2]

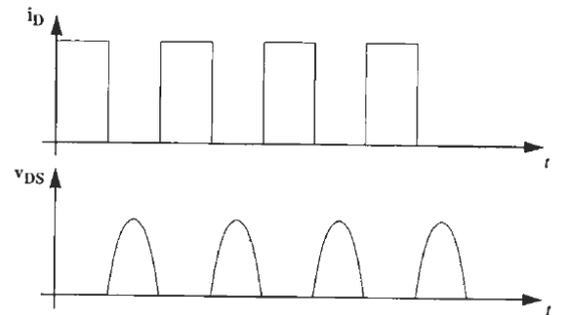


Figure 14.14.21 : Inverse class F current & voltage waveform [2]

It is shown in author [10] for class F power amplifiers with finite harmonic termination for adding 2nd even waveform $i_{D max} = (8/3) I_{Dc}$, using duality between class F and inverse class F then $V_{Ds max} = (8/3) V_{DD}$.

Main features in class F^{-1} PAs:

- Depends mainly on harmonic termination to shape voltage and current waveform
- Efficiency is 100% for infinite number of harmonics
- Class F is considered as ZVS
- $V_{Ds} = \pi V_{dd}$ and $i_{D max} = 2 I_{Dc}$ for infinite number of harmonics

14.4 Comparison between topologies

Table 14.19: Comparison between different topologies

	Linear classes-				Non-linear classes			
Point of comp.	Class A	Class B	Class C	Class AB	Class D	Class E	Class F	Class F^{-1}

Efficiency	50%	78.5%	78.5-100%	50-78.5%	63.69 %	~90%-100%	78.5%-100%	78.5%-100%
Power handling capability	1/8	1/8	1/8-0	1/8	$\frac{1}{2\pi}$	0.09	1/8-0.159	1/8-0.159
Linearity	Best	Good	Bad	Best-good	Worst	Worst	Worst	Worst
Supported modulation tech.	Constant & variable envelop	Constant envelop	Constant envelop	Constant envelop	Constant envelop			
Conduction angle(°)	360	180	180-0	360-180	-	-	-	-
Switching type	-	-	-	-	-	ZVS	ZCS	ZVS
Drain voltage	$2V_{dd}$	$2V_{dd}$	$2V_{dd}$	$2V_{dd}$	$2V_{dd}$	$3.562V_{dd}$	$2V_{dd}$	πV_{dd}
Drain current	$2I_B$	I_B	$I_B - 0$	$2I_B - I_B$	I_{max}	$1.77V_{dd}/R$	πI_{dc}	$2I_{dc}$

A simple approach to make non-linear class supports variable envelope modulation is back-off technique but it decreases the efficiency which translates to very large power consumption.

Non-linear classes don't support variable envelope modulation techniques, however due to their high efficiency researches a great deal of effort has been expended on linearization techniques that offers a higher overall efficiency than back-off from the compression point does.

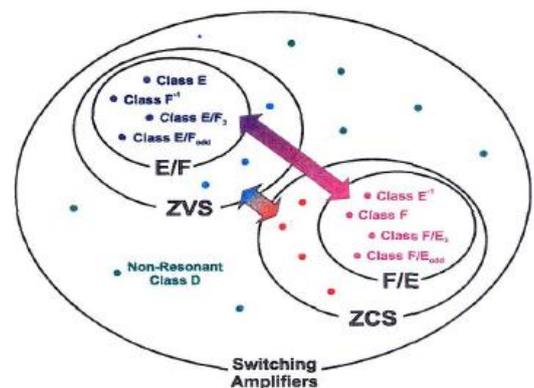


Figure 14.14.22 : Switching Amplifiers

From above comparison class E seems to be best choice in non-linear class, but it face the problem of high drain voltage that reaches to $3.56V_{dd}$, which raises the concerns about breakdown and reliability issues. Class F drain voltage reaches $2V_{dd}$, but practical implementation uses only finite number of harmonic termination sacrificing efficiency. In next section hand analysis and simulations will show the reason behind choosing inverse class F.

14.5 Design methodology

The required specification is to provide +11dBm at fundamental frequency 915MHz with a minimum attenuation for second harmonic of -34 dBm and other harmonics -50 dBm. The current budget is 25 mAmpere. The modulation type is a constant envelope. PA block is divided into 5 sub-blocks bias

circuit, pre-amplifier, driver, switches, output stage & Termination. Each sub-block is discussed alone then PA will be discussed as a signal block.

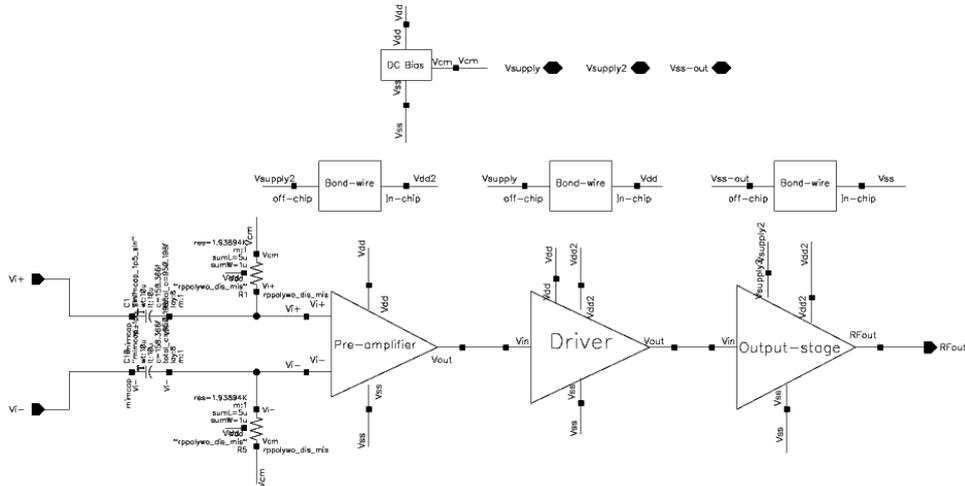


Figure 14.14.23 : PA sub-blocks

14.5.1 Bias Circuit

The circuit shown is used to bias input NMOS of the CML, the circuit uses the concept of replica bias to generate a nearly constant DC voltage across PVT variations. This circuit provide a DC voltage of nearly 600 mv. All PMOS have same size to mirror constant current 10uA in all branches, All NMOS is sized the same to compensate any process variations, the resistance is added to compensate any supply voltage variation.

Bias circuit schematic

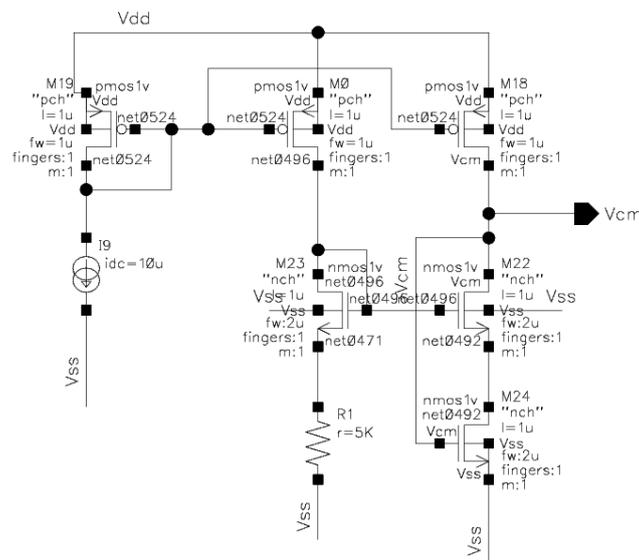


Figure 14.14.24 : Bias Circuit

14.5.2 Pre-amplifier

The pre-amplifier input is a sine wave 600mv peak to peak. The pre-amplifier is designed with three main goals

1. Convert the signal from differential into a signal ended
2. Decrease the loading capacitance on divider
3. Provide gain to signal

A CML with CML to CMOS is chosen as pre-amplifier because it nearly converts small input signal into a rail-rail signal with large swing, which is desirable for non-linear PAs. A small overdrive is chosen so input signal could steer the current in one branch. The input swing is 600mV differential from divider, the linear range of CML is $2\sqrt{2}V_{od}$ then a minimum $V_{od} = 212\text{mv}$ is required to steer current in one branch and provide a signal of $V_{dd} - V_{thp}$ and $V_{odn} + V_{od}$ current source. Taking the effect of compression & non-linearity assume $V_{od} = 120\text{mv}$.

14.5.2.1 Optimization curves

The test-bench used is making MOSFET diode connected and sweeping the input DC voltage.

Using $\frac{g_m}{I_d}$ method we assume required $V_{od} = 120\text{mv}$. Using curves below, we find that $\frac{g_m}{I_d} = 9$, $\frac{I_d}{(W/L)} = 6.42 \text{ uA}/m^2$, Assuming a current of 60uA . $\frac{W}{L} = 9.23$, taking the minimum $L = 130 \text{ nm}$ to decrease the parasitic capacitance. Then $W = 1.2 \text{ um}$.

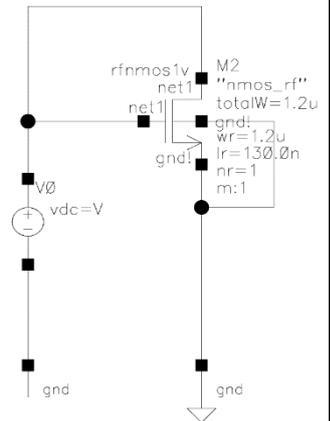


Figure 14.14.25 : Test-bench

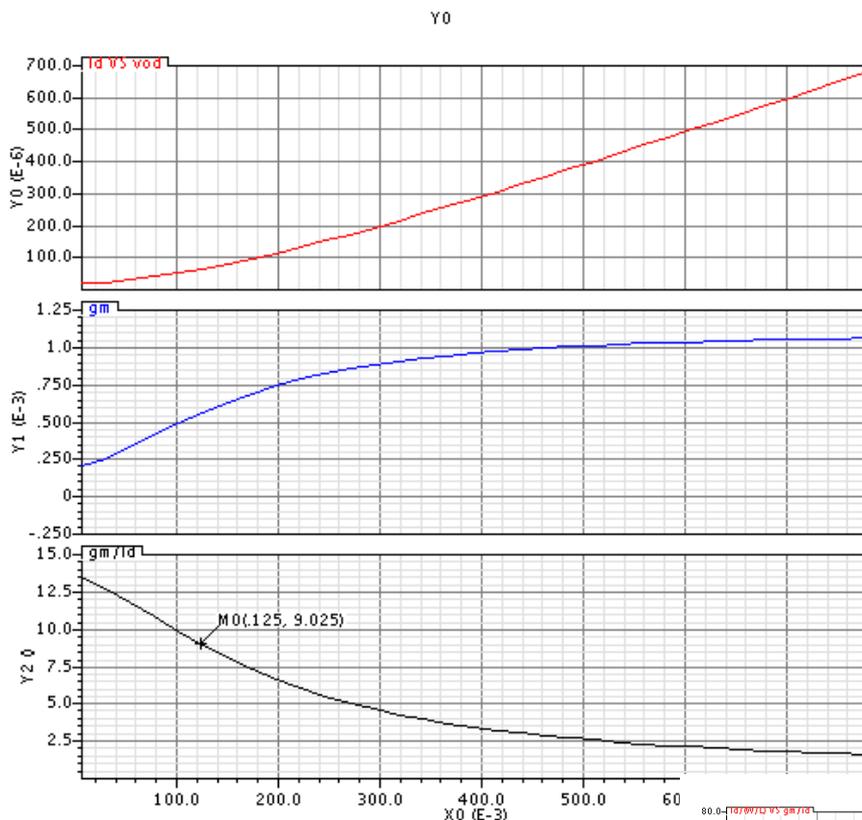
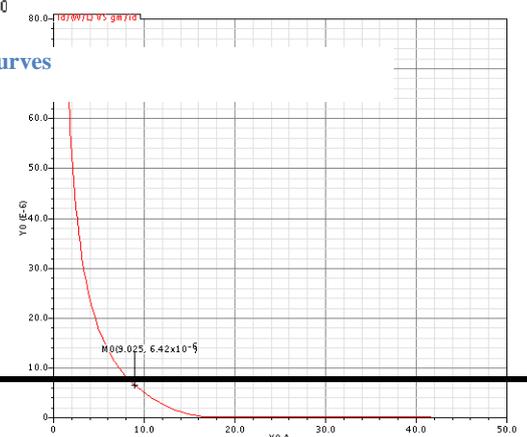


Figure 14.14.26 : Optimization curves

The first curve shows I_d VS V_{od} , second curves shows g_m VS V_{od} and third curves shows g_m/I_d VS V_{od} . After simulating on cadence the high input swing makes the CML



makes MOS experience non-constant gm leading to small output swing. Using try & error method the current is increased to 184 uAmpere.

The designing of CML to CMOS stage have two main purposes:

1. Change the swing of CML to rail-rail signal
2. Adjust the duty cycle to 50% which is preferable for non-linear classes PAs

Figure 14.14.27 : Id/(W/L) / (gm/Id) Curve

Pre-amplifier schematic

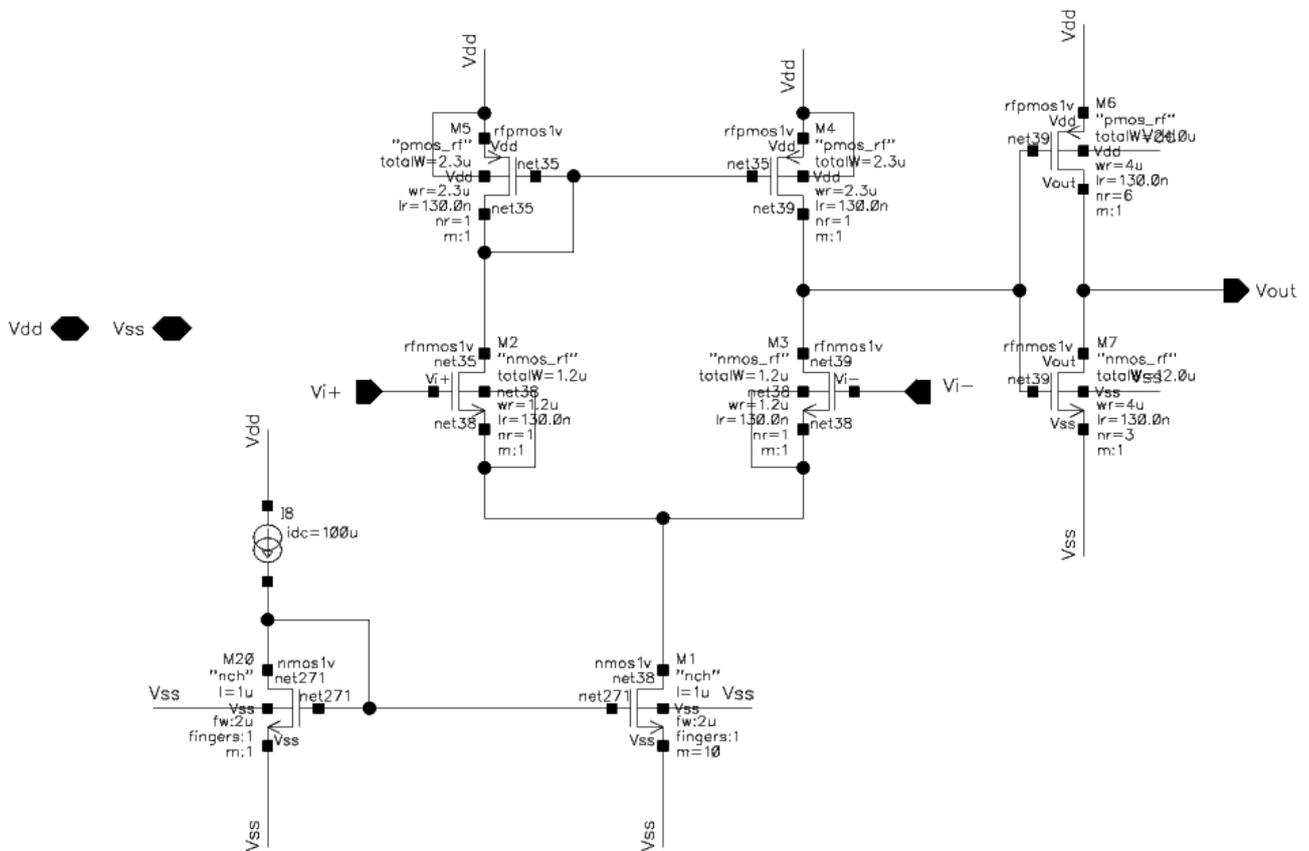


Figure 14.14.28 : Pre-amplifier

Adjusting the duty cycle to 50% is preferable for non-linear PAs as it provides maximum efficiency and maximum second harmonic rejection. [7]

Pre-amplifier test bench:

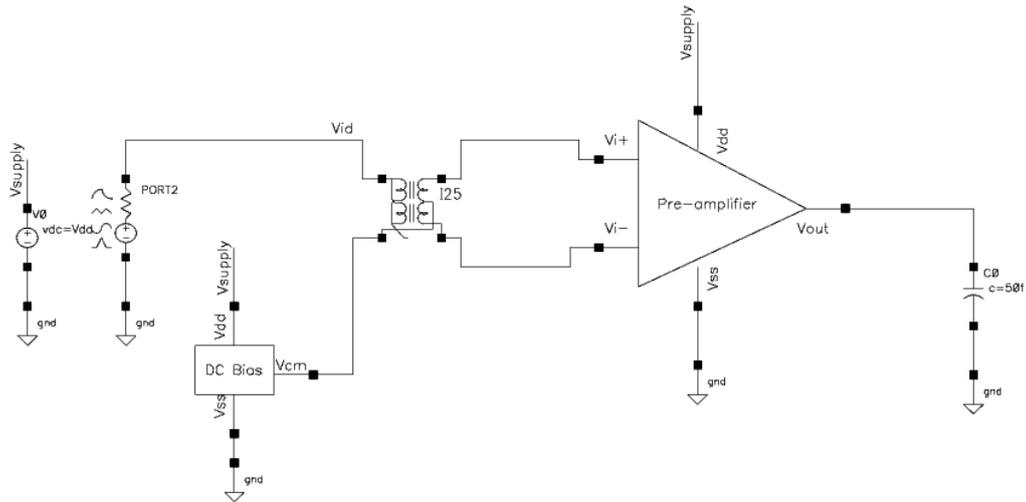


Figure 14.14.29 : Preamplifier test bench

The test purpose is to measure the gain of fundamental frequency using PSS analysis, the input signal is 600mv differential. Simulated results show a voltage gain of 1.77 dB. The input capacitance of pre-amplifier is 1.46 fentoFarad.

The current consumption

1. CML = 184 uAmpere
2. CMOS = 702 uAmpere

Total current of pre-amplifier = 0.886 mA

14.5.3 Driver

14.5.3.1 Induction tuned load PA driver

This driver is conventional for PAs as it provides higher voltage than given supply, however it faces some problems with reduced conduction angles for higher efficiency. Under this condition, the output sine wave amplitude of driver must be increased for maximum output current at output stage. In conventional class F PA using inductor tuned driver will result in negative voltage swing that is not an issue for GaAs MESFET, but a major problem for CMOS. The negative swing can potentially forward bias the drain junction diode. The usages of tuned driver have another serious drawback as the output of driver is sine wave the transistor is not hard driven raising serious doubts that output stage transistor would experience knee effect, which decrease the efficiency of PA.

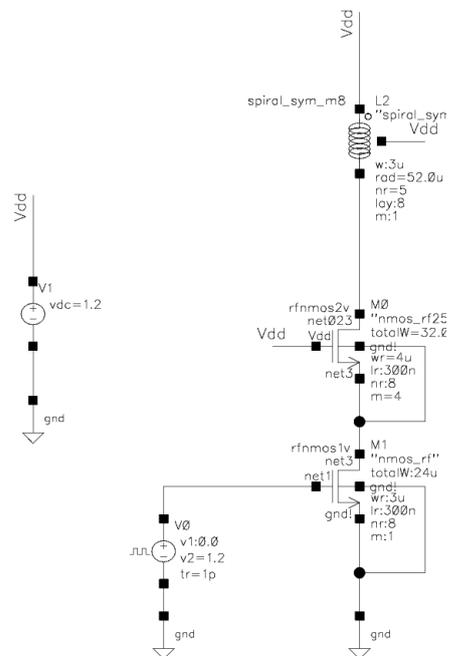


Figure 14.14.30 : Induction tuned load PA driver

14.5.3.2 Class F driver

The theory of class F PAs is based on peaking of odd harmonics and attenuating the even ones in order to a square-shaped waveform. This results in less overlap between voltage and current signals, which leads to less power consumption. Class F is known to provide nearly a square with output voltage of $2V_{dd}$.

The design of class F driver is divided into 2 sections

1. Core
2. Termination

The core is chosen for minimum current consumption, while the termination is designed to provide maximum flat response to output by setting the ratio of $\frac{L}{C} = 10^3$. For third and fifth harmonic termination $L_3 = 1.8 \text{ nH}$, $L_5 = 1.1 \text{ nH}$, $C_3 = 1.8 \text{ pF}$, $C_5 = 1.1 \text{ pF}$.

Although Class F seems to be pretty satisfactory as a level shifter between low supply (1.2) to higher supply (2.5) and it's also more immune to supply voltage variation, it requires the uses of inductors which bring the issue of cost and area. If the coils is to be implanted on chip it will consume a large area on chip, else if off chip component is used the cost will too high.

Also class F driver will consume large current compared to CMOS inverter circuit.

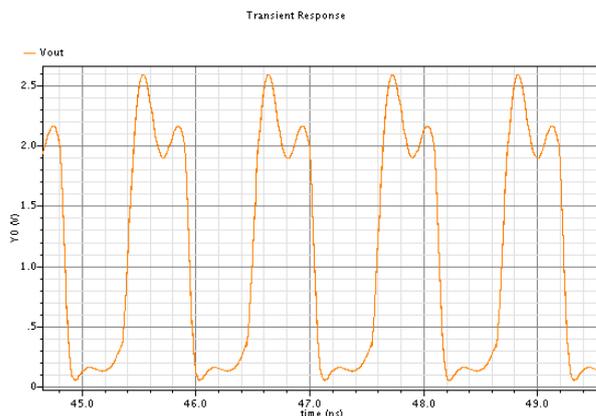


Figure 14.14.32 : Output voltage waveform

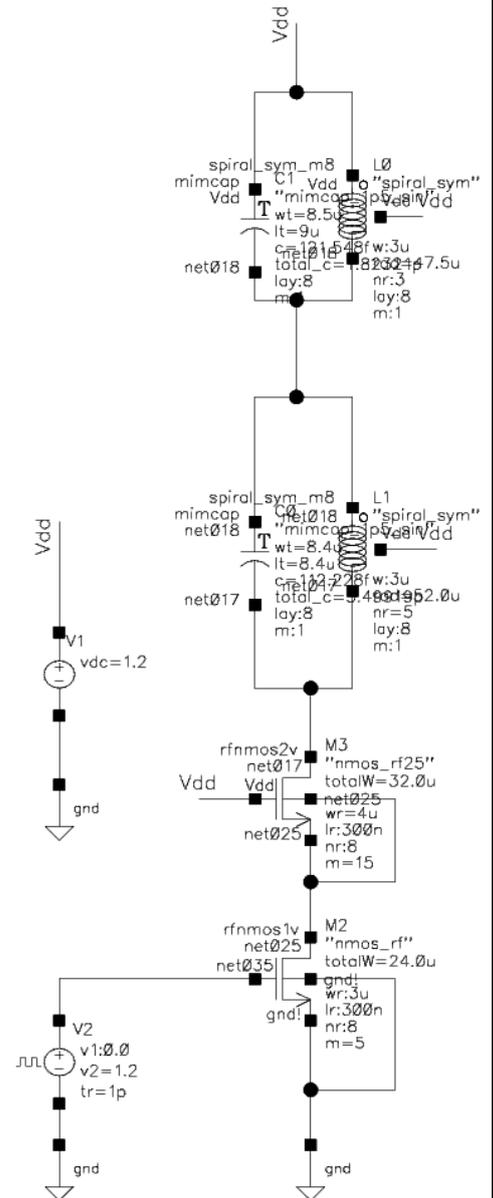


Figure 14.14.31 : Class F driver

14.5.3.3 CMOS Driver

The final choice of driver is CMOS driver which is easily implanted on chip and doesn't require large area on chip compared to inductors. The circuit consists of four cascaded CMOS inverters and RC section. The cascading of CMOS circuits to provide with three main goals

1. Decreasing rise and fall time of input signal
2. Convert from low supply voltage (1.2) to higher supply voltage (2.5) required at output stage
3. Support the output capacitance load from switches

The capacitor at the input of divider is used to couple the input DC voltage from the previous stage (pre-amplifier), while the resistance is used to bias the CMOS inverter with a voltage of V_{dd} , thus it set the input at a DC voltage of V_{dd} with a swing of V_{dd} .

The first stage is CMOS inverter with a resistive degeneration, the use of the resistance is to provide stability across corners, as the input signal although its input is from 0 to 1.2volt rail to rail for 1volt MOSFET it is considered small compared to input for 2volt MOSFET.

The driver most important trade-off is that number of stage which consume large current with decreasing rise and fall time of input signal to increase PA core efficiency. After some trial and errors a four stages of inverter is chosen.

The driver consumes a current of 2.7mA.

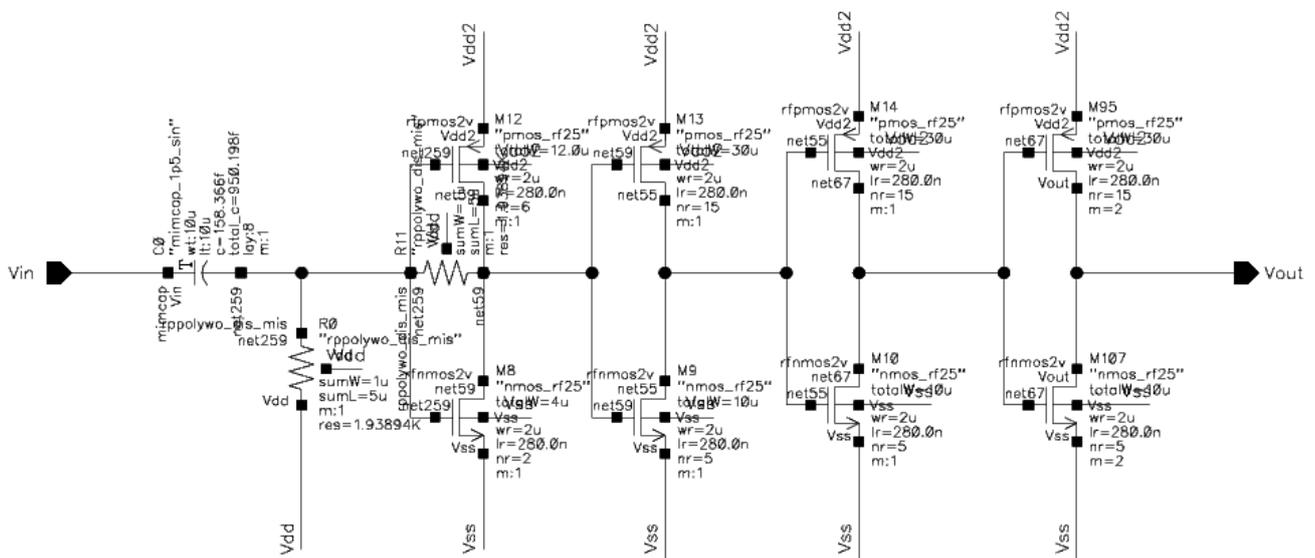


Figure 14.14.33 : Driver

14.5.4 Switch

At first the switch contained a buffer to restore the shape of signal after transmission gate, but it required another switches at the supply to decrease the power consumption when this path is not used. The final choice of switch is divided into two parts

1. Transmission gate switch
2. Pull down network

The size of transmission gate has a trade-off between loading capacitance on driver and the effective transmission of the RF signal through switch. The pull down network function is to set the gate of core transistor to zero when the switch is closed. As the gate of core transistor will be a high impedance node which could take any value, thus to save current pull down network set the gate core value to zero when the path of the signal is closed.

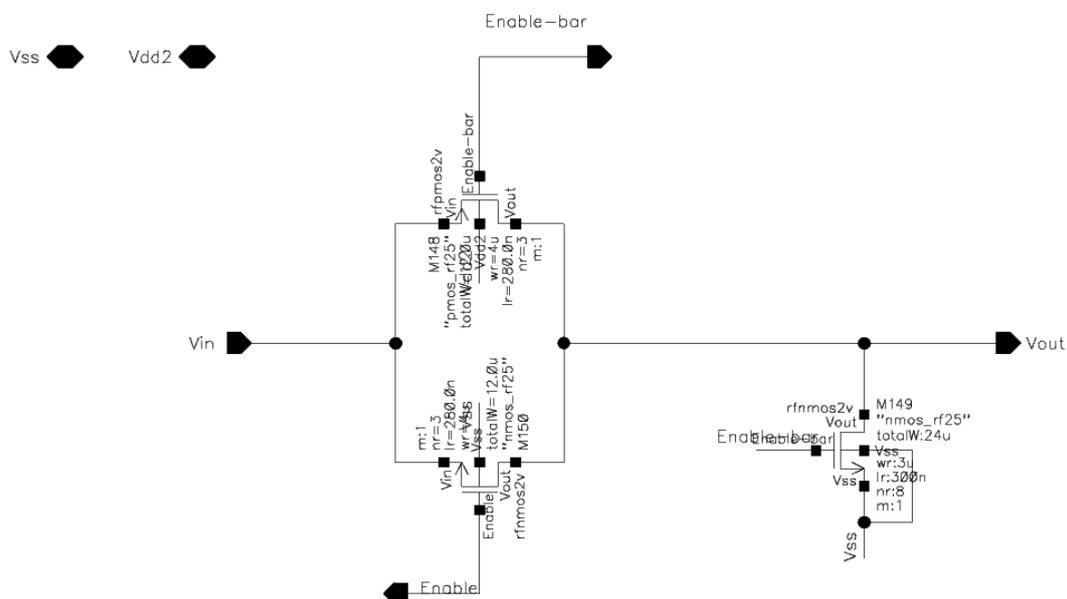


Figure 14.14.34: Switch

14.5.5 Output stage

The output stage is divided into four parts

1. 8 switch to provide DC programmability
2. PA core containing all transistor(s)
3. Bond wire model
4. Termination which includes harmonics termination, RF choke, DC coupling capacitor, matching network & Filter

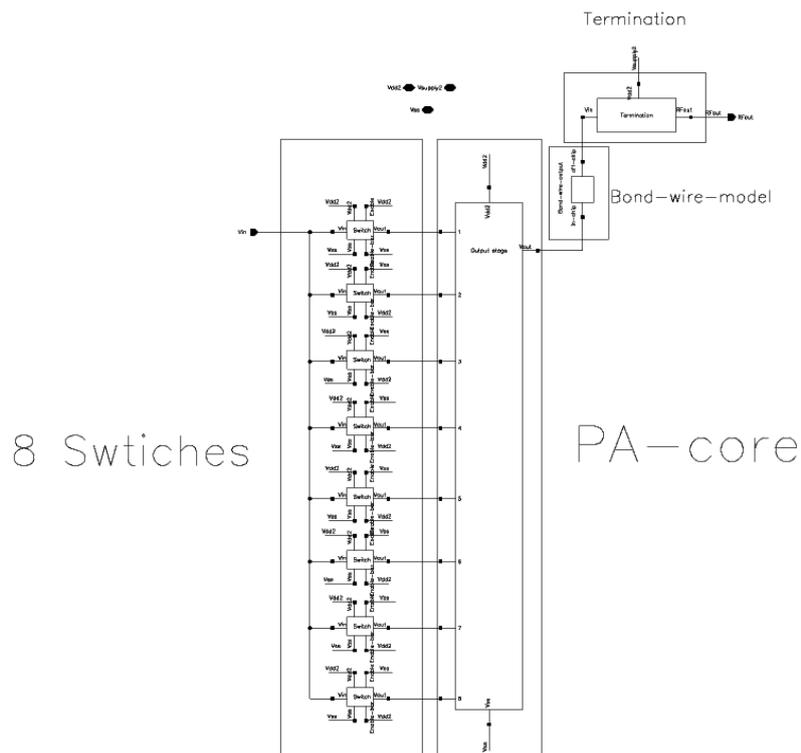


Figure 14.14.35 : Output stage

14.5.5.1 Termination

The termination determine the class of PA, non-linear class uses different harmonic termination to shape the voltage and current waveforms at the drain. Thus decrease voltage and current overlap which translates into a higher efficiency for PA. All off chip components is assumed to have a quality factor of 30.

14.5.5.1.1 RF Choke

Since low power PAs requires larger output loads than higher power PAs a large DC –feed RF inductors is required, but that inductor is assumed to have quality of 30 only. As a result PAs designed for low power applications have considerably lower efficiency than higher power PAs.

14.5.5.1.2 Coupling capacitor

The coupling capacitor value is determined to set the resonance frequency of RF choke and coupling capacitor away from fundamental frequency

14.5.5.1.3 Class F PA

The theory of class F PAs is based on peaking of odd harmonics and attenuating the even ones in order to a square-shaped waveform. This results in less overlap between voltage and current signals, which leads to less power consumption. Class F is known to provide nearly a square with output voltage of $2V_{dd}$. In most practical designs a finite number of harmonics is used, In this case third harmonic peaking is used. Assuming that tank circuit is lossless and without any filter the maximum theoretical efficiency is equal to 88%.

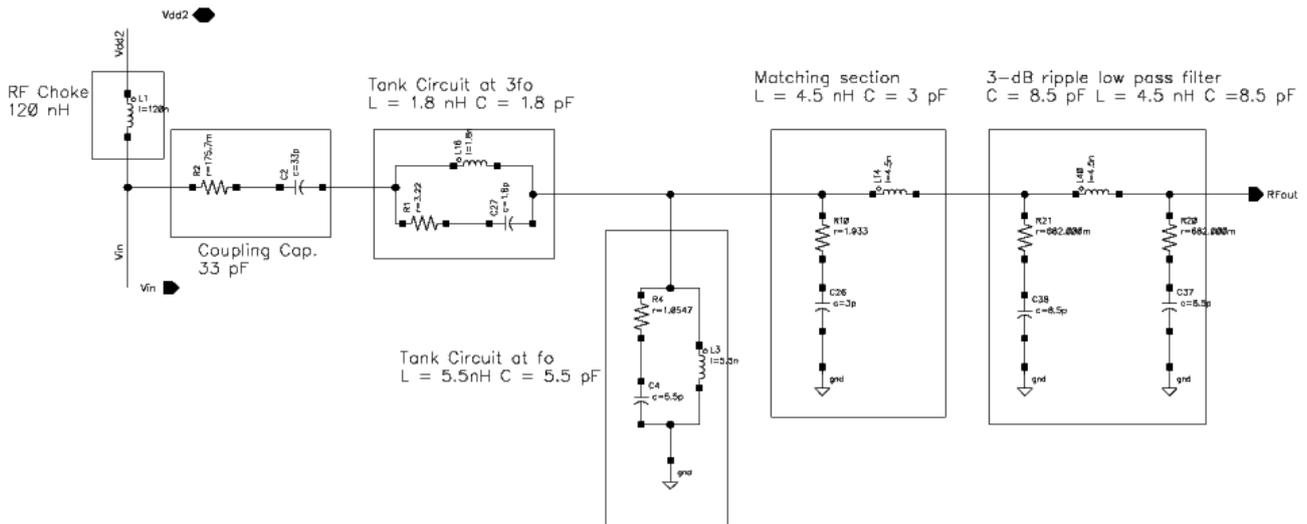


Figure 14.14.36: Class F termination network

The design of Class F seems convenient for low power application, but there is a major issue across corner simulations the second harmonic variation is big which requires to add more attenuation at the typical – typical case which leads to a decrease in the overall efficiency.

Harmonic termination is designed to provide maximum flat response to output by setting the ratio of $\frac{L}{C} = 10^3$. For third harmonic $L_3 = 1.8 \text{ nH}$, $C_3 = 1.8 \text{ pF}$, fundamental frequency $L_1 = 5.5 \text{ nH}$, $C_1 = 5.5 \text{ pF}$. The tank circuit at f_0 purpose is to make core sees finite impedance only at fundamental frequency, else core will see infinite impedance away from the fundamental frequency.

14.5.5.1.4 Inverse class F PA

The theory of class F PAs is based on peaking of even harmonics and attenuating the odd ones in order to a square-shaped waveform. This results in less overlap between voltage and current signals, which leads to less power consumption. In most practical designs a finite number of harmonics is used, In this case second harmonic peaking is used. Assuming that tank circuit is lossless and without any filter the maximum theoretical efficiency is equal to 88%.

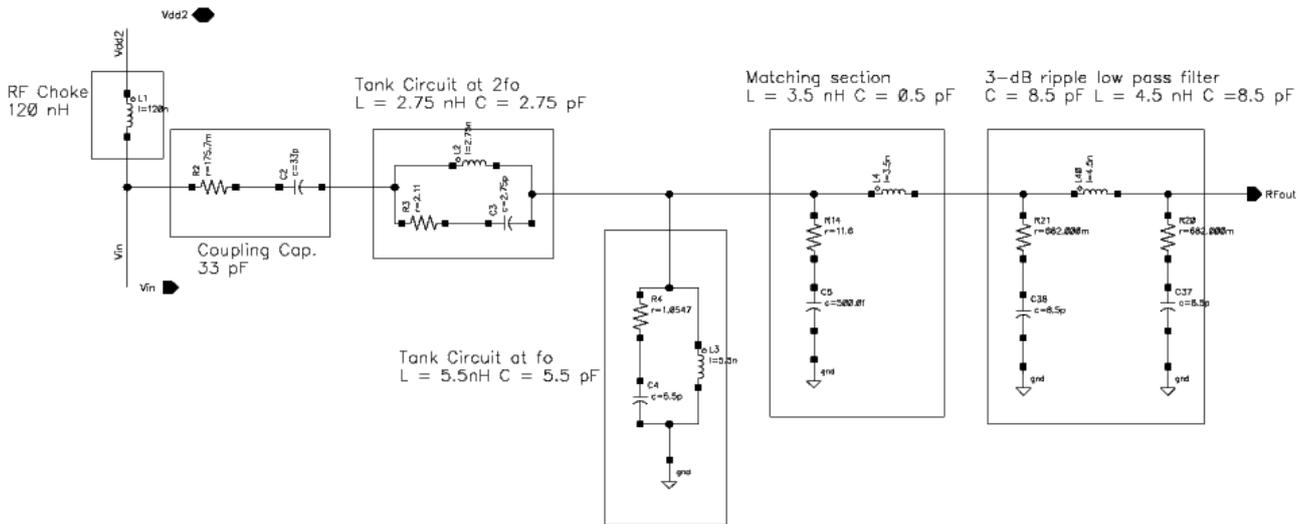


Figure 14.14.37 : Inverse class F termination

The main advantage of class F^{-1} is tank circuit at $2f_0$ that makes the variation of second harmonics across small when compared to class F.

It can be shown from author [10] that second harmonic peaking in class F will lead to a current of $2.6 I_d$. From duality between class F and class F^{-1} the drain voltage will reach $2.6 V_{dd}$.

Harmonic termination is designed to provide maximum flat response to output by setting the ratio of $\frac{L}{C} = 10^3$. For second harmonic $L_2 = 2.75$ nH, $C_2 = 2.75$ pF.

14.5.5.1.5 Matching section

The matching is used to transform the antenna impedance (50 ohm) to required impedance to be seen at PA which translates into delivering the required output power to the antenna.

Although the load-pull measurements are used to determine the value of the matching network it have three main drawbacks. [1]

1. The measured results for one device size can't be directly applied to different size
2. The contours and impedance levels are measured at single frequency , failing to predict the behaviour at other frequency at other frequency (stability)
3. Optimum impedance doesn't provide peaking at higher frequency which means it can't predict efficiency and output power in presence of harmonic termination.

Therefore to determine the value of matching network is done by a two dimensional sweep to determine optimum operating point.

A trade-off between low pass characteristics of matching and maximum efficiency, the technique used here is co-designing the filter and the matching network. The matching could be by two methods

1. Series inductor then parallel capacitor
2. Series capacitor then parallel Series inductor

After comparing between two methods Series inductor, then parallel capacitor is chosen as it have lower value for inductor and capacitor. Final an optimum point is chosen with the values of

$$L = 3.5 \text{ nH}, C = 0.5 \text{ pF}.$$

14.5.5.1.6 3-dB ripple Filter

The filter purpose is to attenuate higher order harmonics a 3-dB ripple low pass filter is used, as it provides a higher attenuation than butter worth low pass filter. The filter consists of 3 elements only to decrease the number of off-chip component which will decrease in the overall area and cost of the chip.

The cut-off frequency $f_c = 1.2 \text{ GHz}$ which is away from fundamental frequency (915 MHz), the reason behind choosing cut-off frequency away from fundamental frequency to provide nearly no losses at fundamental frequency. Using chebyshev polynomials the values of pi-section is determined $L = 4.5\text{nH}$, $C = 8.5\text{pF}$.

Filter trade-off is between number of elements and the type of filter needed to provide certain attenuation for second harmonic.

14.5.5.2 PA core

The PA core consist of 8 parallel cascoded stages each stage provides a specific power level this 8 parallel stages could provide 256 different power levels at harmonics, but it's recommend to use 8 levels only. The DC programmability employs to open switch at a time, while the other switch is closed.

The advantage of using cascoded stage is to provide high isolation between input and output of PA. The value of the low transistor device gate width must be large enough to handle the required current and also ensure a low DC output curves for knee voltage. On the other hand a large gate width associated large parasitic capacitance. An optimum value of 96 μm for first stage was found by trial and error.

The upper transistor is usually a thicker oxide than lower transistor to make it more immune to breakdown and reliability issues, but in our case the maximum swing on transistor is equal to 1.67 which only requires transistor of 1v, but to ensure that transistor doesn't suffer any across corners a Mosfet of 2v is used.

The upper transistor gate width size

1. Large gate width provides large gain
2. Medium gate width provides medium gain
3. Small gate width provides small gain

A high gain is preferable in PA although large gate width size is associated large parasitic capacitance. An optimum value of 704 μm for first stage was found by trial and error. The length is minimum for both upper a lower transistors.

The DC programmability is provided by switching between 8 different stages. The sizes of rest 7 cascaded stage are done by proper scaling to provide a certain power levels.

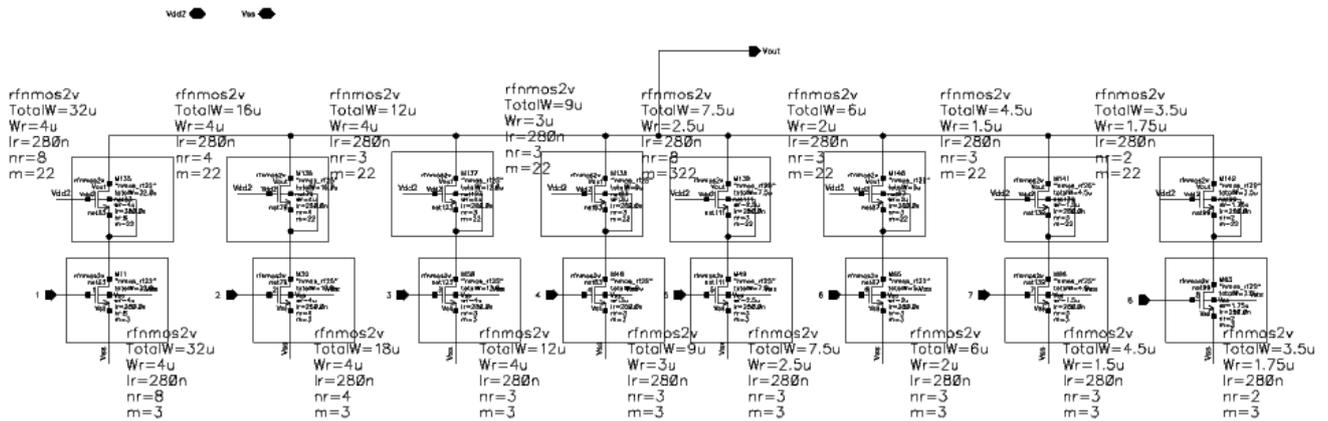


Figure 14.14.38 : Output stage core

14.5.5.3 Bond wire model

The bond wire is divided into two sections

1. Supplies and ground bond wire models (Figure 39)
2. Output node bond wire model (Figure 38)

The inductance is assumed to have a value of 250pH is the same for both model, the capacitance on the node off-chip is assumed to have a value of 2pF the

The difference between the two models is value of capacitance on node of in-chip for output node model the capacitance is assumed to have a value of 100 fF while for supplies and ground model the capacitance is assumed to a value of 500 fF.

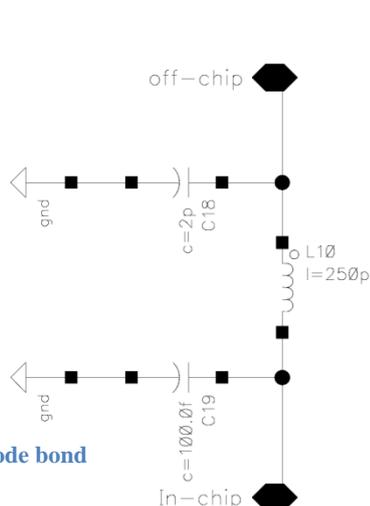


Figure 14.14.39 : Output node bond wire model

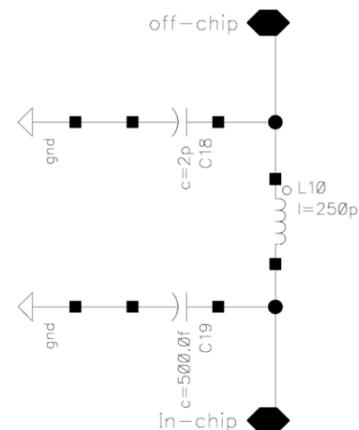


Figure 14.14.40 : Supplies and ground bond wire model

The PA output stage consumes 20 mA delivering full power of 11.44 dBm.

14.6 Simulated results and test benches

The test-bench used is the PA driven by sinusoidal sources. The use of this test-bench to make the general measurements, which include Power related measurements (input power, output power, supply voltage, supply current, power gain and power dissipation)

1. Efficiency measurements (drain efficiency and power added efficiency)
2. Linearity measurements (1 dB compression point)
3. Stability measurements (K-factors and B1f)

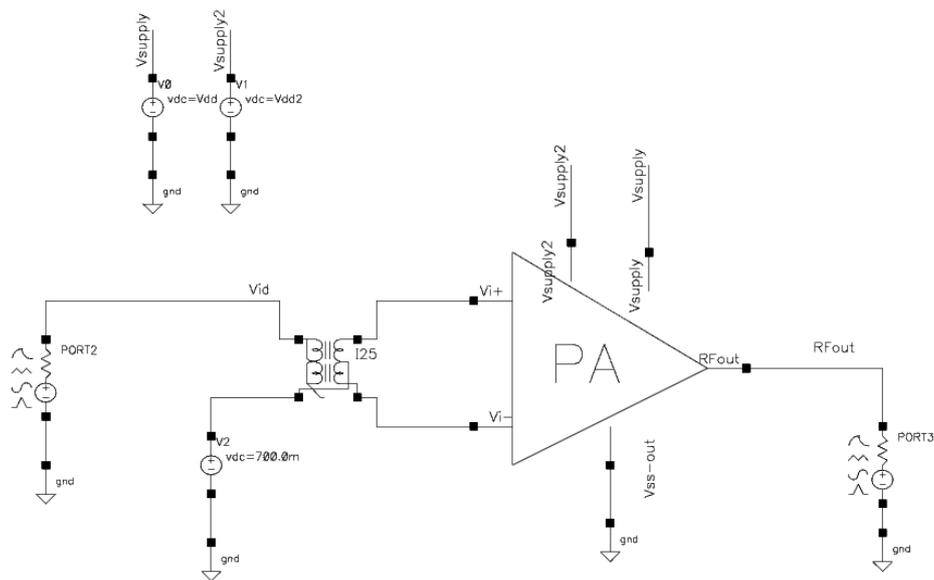


Figure 14.14.41 : PA test-bench

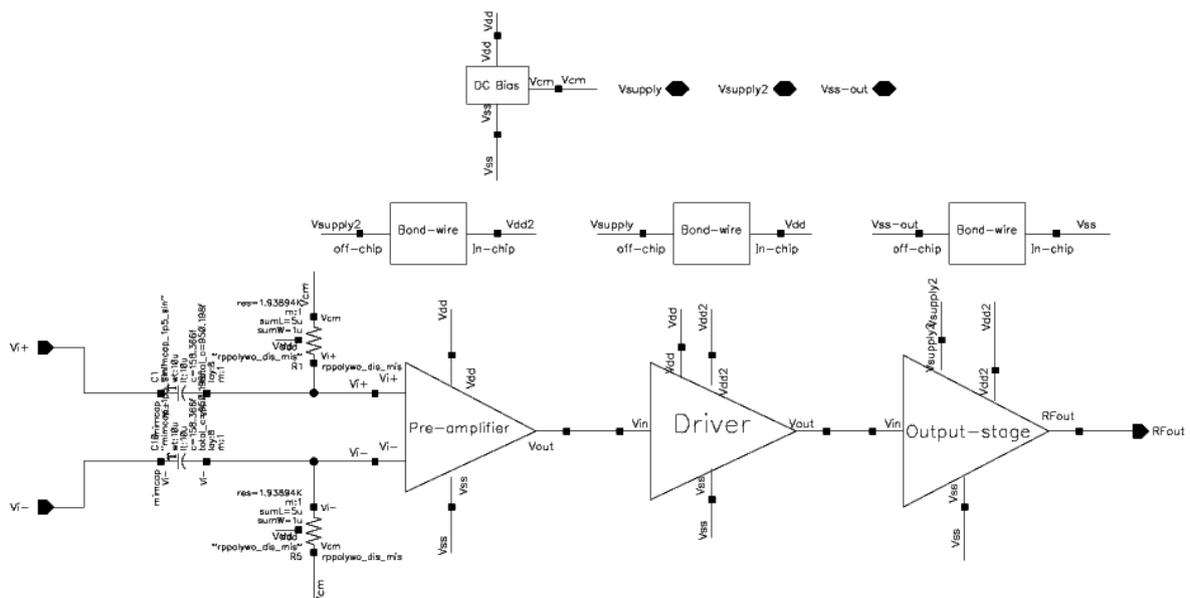


Figure 14.14.42 : PA schematic

- Add an input port that uses a sinusoidal source. Add a PORT component with
 - ❖ source type = sine
 - ❖ frequency parameterized as frf = 915 MHz
 - ❖ amplitude parameterized as pin
- Add an output port that uses a dc source that serves as a load. Add a PORT component with
 - ❖ source type = dc
 - ❖ dc = 0

Power Related Measurements (Swept PSS)

Power related measurements include input power, output power, supply voltage, supply current, power gain and power dissipation. To make these measurements, use a swept PSS analysis to sweep the input power level.

Simulated results:

Pout Vs Pin

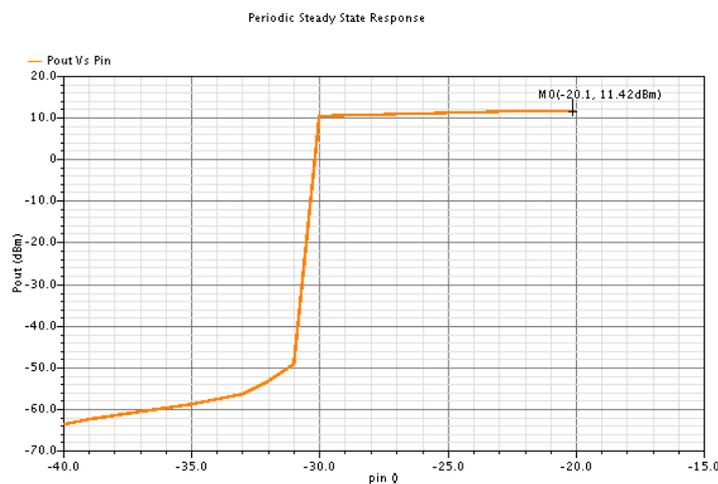


Figure 14.14.43: Pout Vs Pin

Shows if input power become more than -32dBm the output power is nearly constant.

Power spectrum measurements

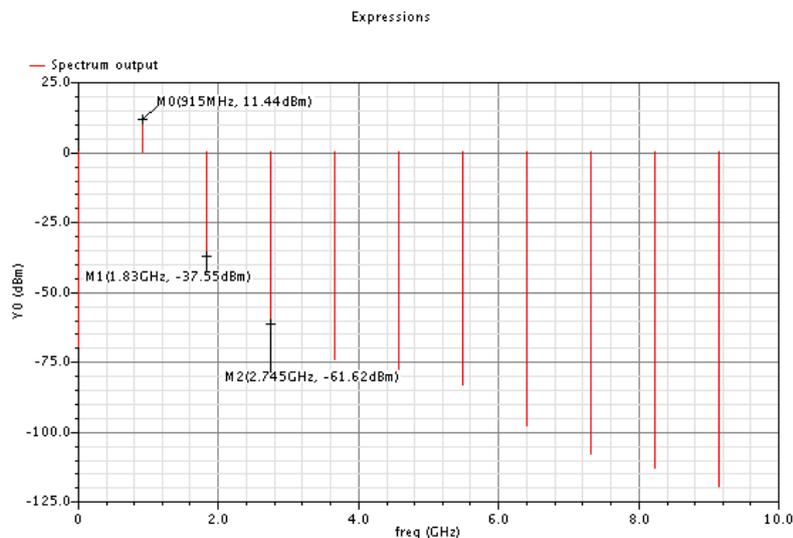


Figure 14.14.44 : Power spectrum measurements

Drain efficiency

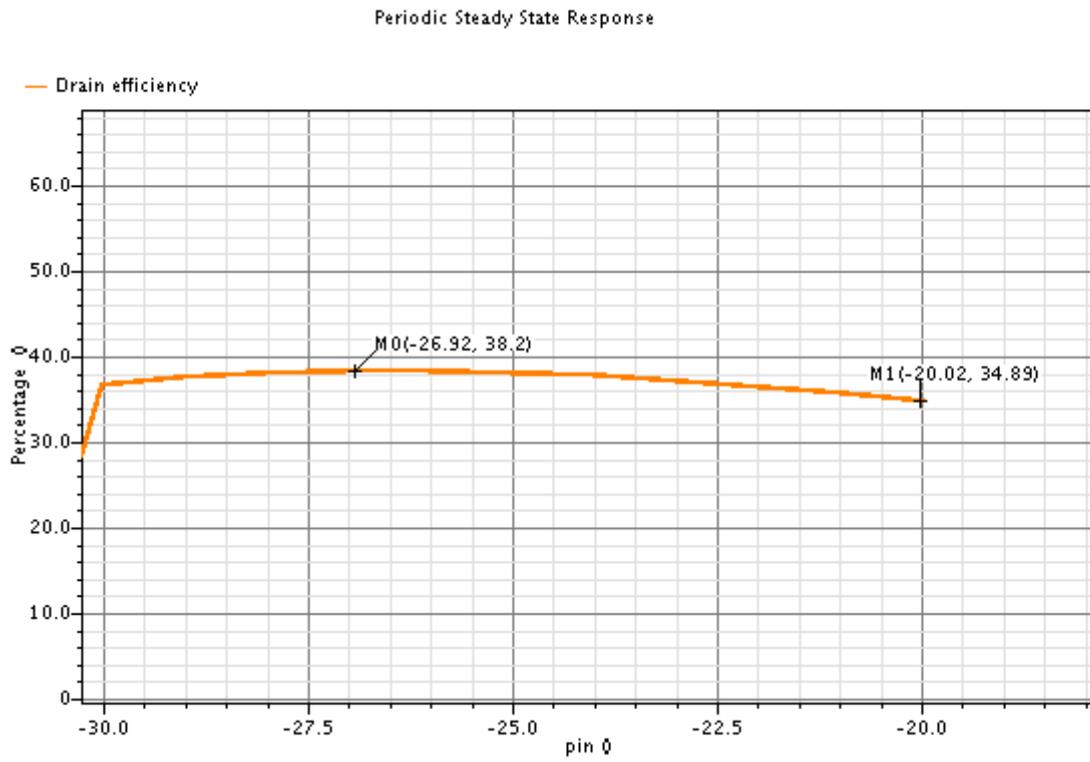


Figure 14.14.45 : Drain efficiency

Power aided efficiency

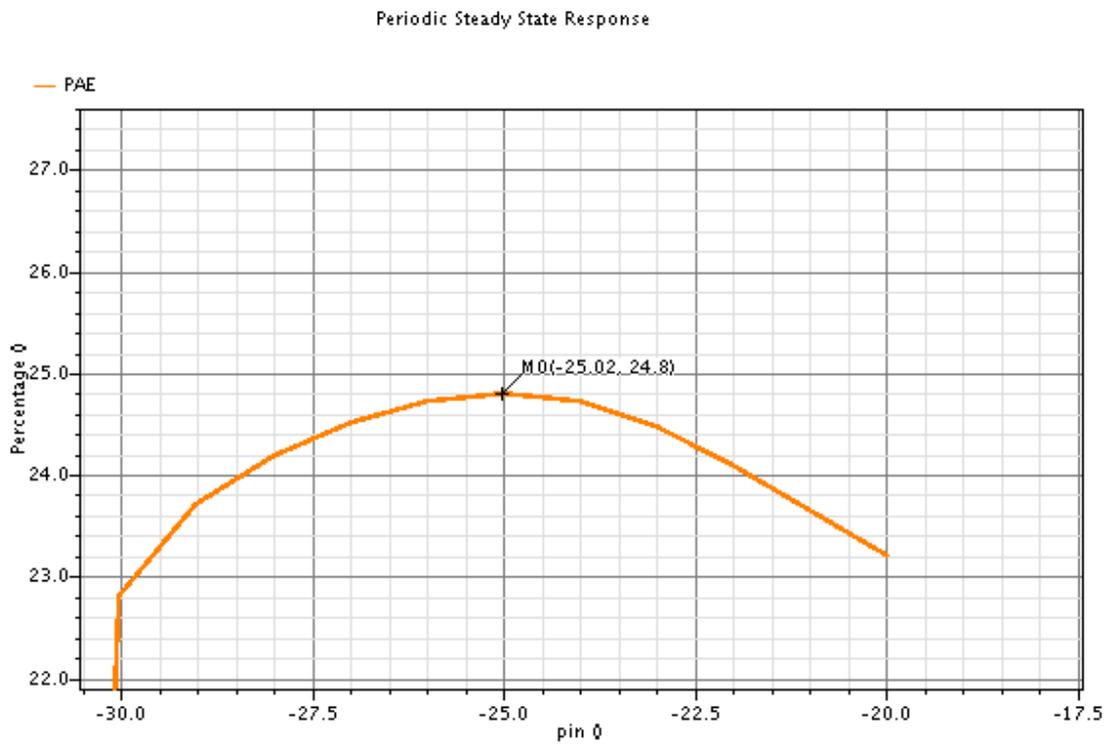


Figure 14.14.46 : Power aided efficiency

Power gain efficiency measurements

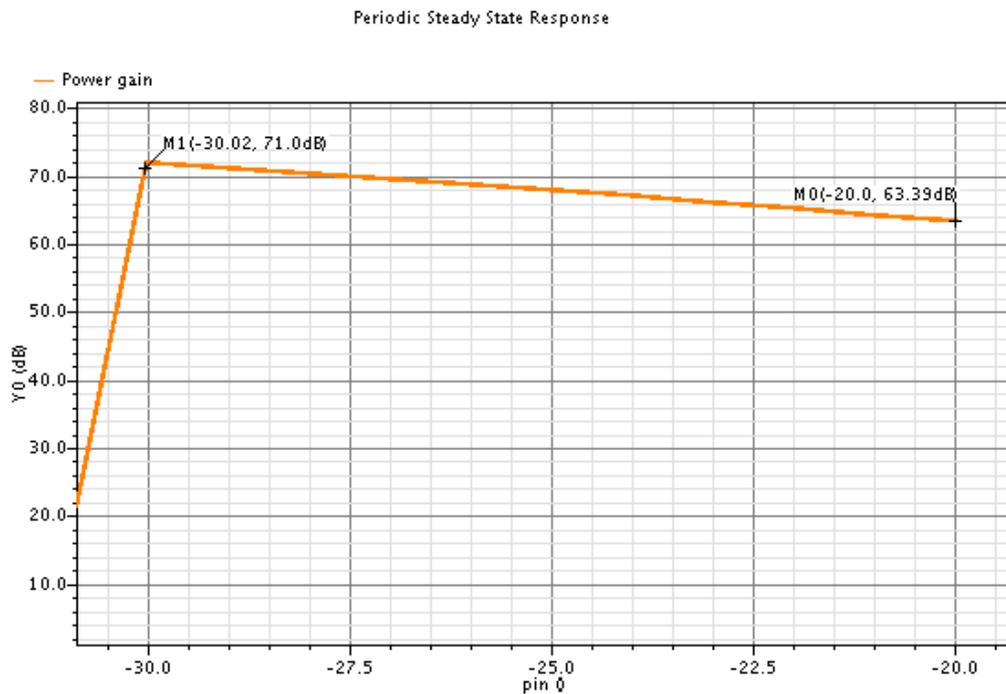


Figure 14.14.47 : Power gain

1-dB compression point

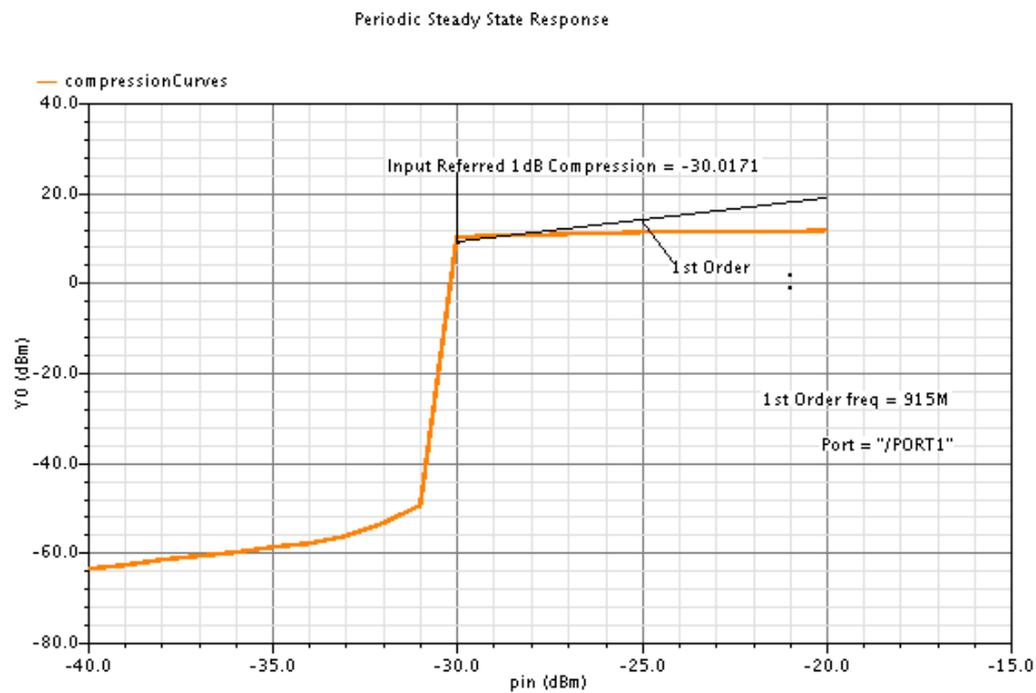


Figure 14.14.48 : 1-dB compression point

Shows 1-dB compression point which is equal to -30 dBm which is desirable in non-linear PAs

Stability of any PA is determined by 3 factors

1. S-parameters ($S_{11}, S_{22} < 0$ across all freq.)
2. Load variations expressed in $K_f (>1)$ & $B_{1f} (<1)$ factors
3. Input power

As pointed out by Gonzalez in [11], stability is guaranteed for the following conditions

$$K_f > 1, \Delta < 1$$

$$K_f > 1, B_{1f} = 1 + |S_{11}|^2 - |S_{22}|^2 - \Delta^2 > 0$$

Kf Stability measurements

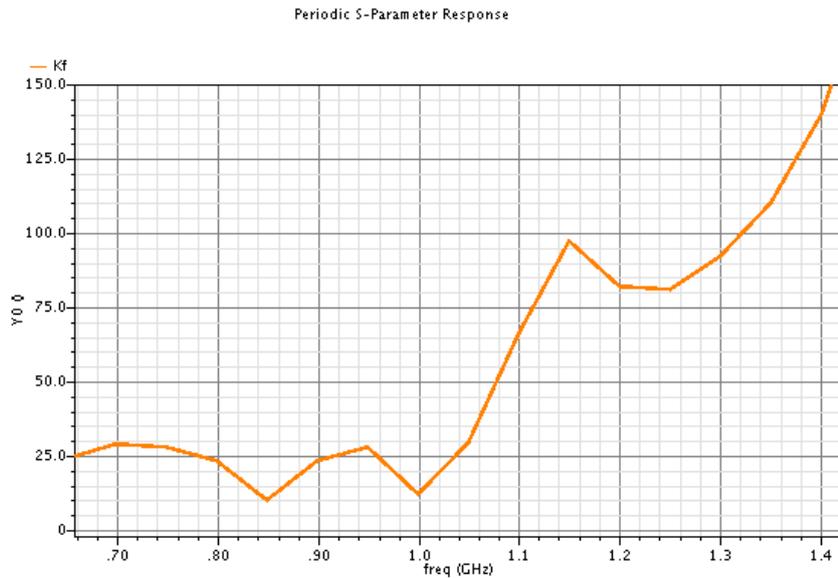


Figure 14.14.49 : Kf stability factor

B1f stability measurements

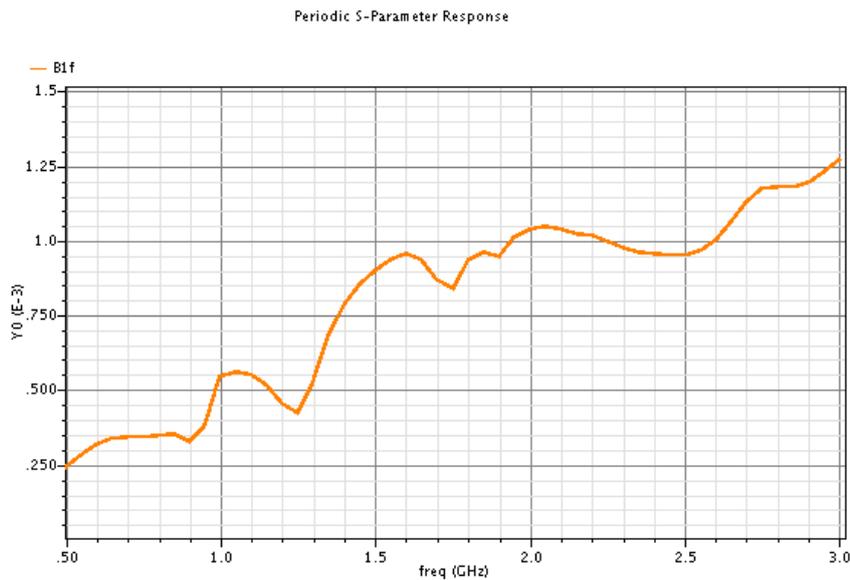


Figure 14.14.50 : B1f stability measurements

Displays the stability of PA across frequency

Voltage and current waveforms

The figure shows V_{ds} of lower transistor of cascode with drain current.

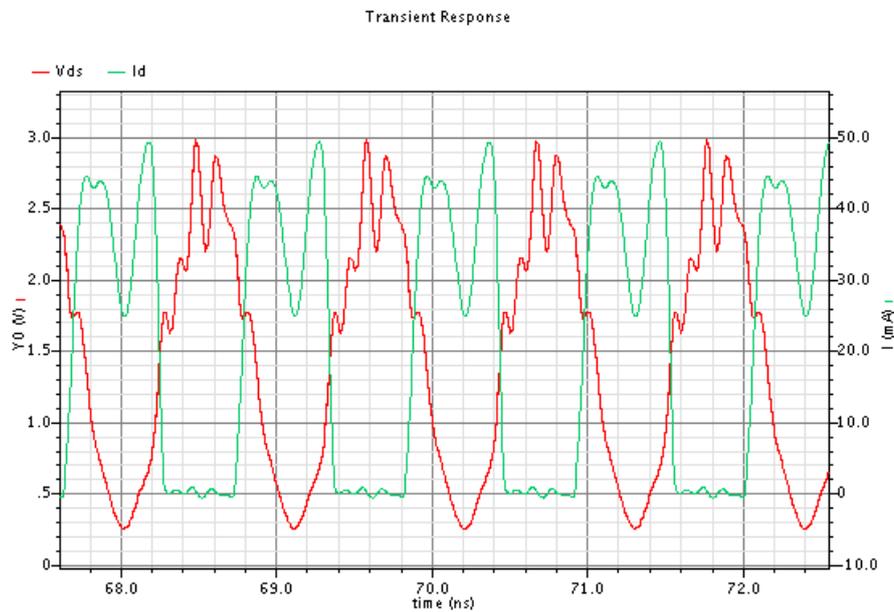


Figure 14.14.51: Voltage and current waveforms

Displays the Voltage & current waveforms of lower transistor as the upper transistor of cascoded have a gate voltage of 2.5volt (supply 2 volt), then maximum voltage on drain of lower is ($V_{dd2} - V_{th}$). The results also show the ring effect of the bond wire model. The drain current is nearly equal $2I_d$ as core draws a 20mA which agrees with theoretical data.

Output power & PAE VS second supply variation (2.5volt)

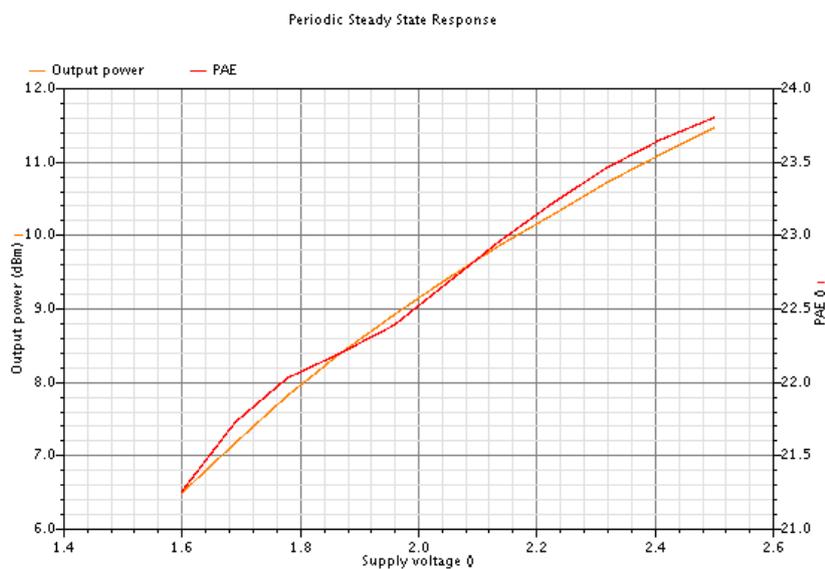


Figure 14.14.52 : Output power & PAE VS supplies variation

Shows the output power and PAE with second supply from 1.6 to 2.5 as for low power application it required for PA to operate correctly at smaller voltage than supply.

Output power VS frequency

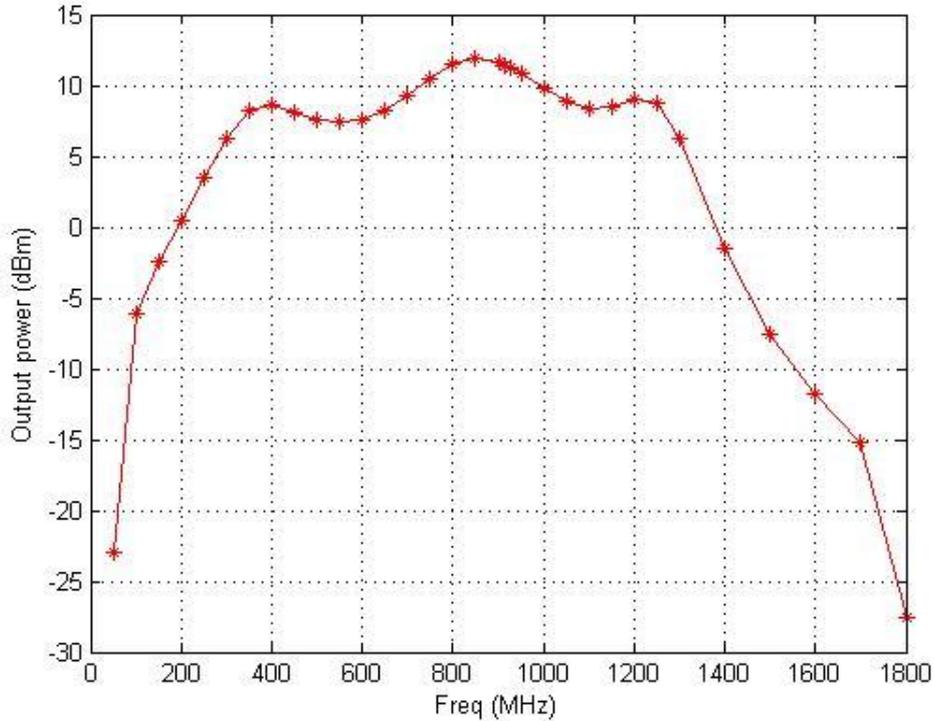


Figure 14.14.53 : Output power VS frequency

Shows output power VS frequency that curve is done on Matlab and the result is obtained using ocean scripting.

Summary of results:

Table 14.20: Specification VS achieved

	Specification	Achieved
Output power (dBm)	+11	11.44
2 nd harmonic (dBm)	< -34	-37.29
3 rd harmonic (dBm)	< -50	-61.58
Current consumption (mA)	< 25	23.586
PAE (%)	-	23.62
Power gain (dB)	-	24.45
1-dB compression point (dBm)	-	-30

Table 14.21: Current consumption of each stage

Stage	Current (mA)

Pre-amplifier	0.886
Driver	2.7
Output stage (max power)	20

All the following data is typical-typical case:

Table 14.22 : DC programmability power levels (8 levels recommend)

Number	Fundamental power (dBm)	2 nd Harmonic (dBm)	3 rd Harmonic (dBm)	Current (mAmp)	PAE (%)
1	11.44	-37.29	-61.58	23.58	23.62
2	8.919	-41.13	-65.26	17.27	18.89
3	7.042	-43.48	-65.74	14.27	15.02
4	4.841	-47	-66.75	11.63	11.28
5	3.337	-50.08	-67.92	10.25	9.155
6	1.436	-55.06	-70.18	8.83	6.967
7	-1.151	-65.43	-73.24	7.39	4.69
8	-3.26	-75.27	-75.2	6.5	3.345

1. The extreme corners for output power

Table 14.23 : Corners simulation

	MOS1v	MOS2v	Cap	res	temp	Supply 1	Supply 2	Pout(dBm)
Max	ff	ff	ss	ss	0	1.1	2.6	12.2
Min	ss	ss	ss	ff	85	1.1	2.4	10.1

2. The extreme corners for second harmonic

	MOS1v	MOS2v	Cap	res	temp	Supply 1	Supply 2	2 nd harmonic (dBm)
Max	ff	sf	ss	ff	85	1.1	2.4	-40.2
Min	sf	ff	ff	ss	0	1.3	2.6	-34.4

3. The extreme corners for Third harmonic

	MOS1v	MOS2v	Cap	res	temp	Supply 1	Supply 2	3 rd harmonic (dBm)
Max	ss	fs	ff	ff	85	1.1	2.4	-67.3
Min	fs	ff	ss	ss	0	1.1	2.6	-59.1

4. The extreme corners for Current

	MOS1v	MOS2v	Cap	res	temp	Supply 1	Supply 2	Current(mA)
Max	sf	ff	ff	ff	0	1.3	2.6	28.97
Min	ff	ss	ss	ss	85	1.1	2.4	17.92

5. The extreme corners for PAE

	MOS1v	MOS2v	Cap	res	temp	Supply 1	Supply 2	PAE (%)
Max	fs	ff	ff	ss	0	1.1	2.6	25.95
Min	ss	ss	ff	ff	85	1.3	2.4	19.96

14.7 Conclusion

An inverse class F is with CMOS driver is designed and simulated. The circuit can operate with supply voltages ranging from 1.6-2.5 while maintaining high efficiency. The circuit achieves the specification required through all corners. The achieved results in typical-typical case shows an output power of +11.44dBm is provided at fundamental frequency (915 MHz) with an attenuation of -37.29dBm at second harmonic. A total current of 23.58 mA is drawn from the supply at the fundamental frequency (915 MHz) providing a PAE of 23.62%. The work demonstrates the feasibility of use inverse class F for short, low power application.

14.8 References

- [1] Behzad Razavi, "RF microelectronics" second edition.
- [2] Thomas H. Lee, "The design of CMOS Radio-Frequency integrated circuits" second edition.
- [3] Steve Cripps "RF Power Amplifiers for Wireless Communications" Second Edition
- [4] Scott Kee Thesis "The Class E/F Family of Harmonic-Tuned Switching Power Amplifiers"
- [5] Nathan O. Sokal and Alan D. Sokal "Class E – A new class of high efficiency tuned signal-ended switching power amplifiers " IEEE J. Solid-State Circuits , vol. SSC-10, pp.168-176, June 1975.
- [6] N. S. Fuzik, E. A. Sadykov, and V. I. Serguchev, "Electrical design of the oscillatory circuits if the final stage of a radio transmitter operating in the biharmonic mode," Radio Eng., vol. 25 , no. 1, pp. 141-145, 1970.
- [7] Frederick H. Raab, "Idealized operation of the Class E tuned power amplifier" IEEE tranactions on circuits and system, Vol. CAS-24, No.12, pp.725-735, 1977.
- [8] Frederick H. Raab, "Maximum efficiency and output of class- F power amplifiers" IEEE tranactions on circuits and system, Vol. 49, No.6, pp.1162-1166, 2001.
- [9] See e.g. S. Kee et al.. "The class E/F family of ZVS switching amplifiers, " IEEE tranactions Microwave theory and Tech., Vol. 51, May 2003.
- [10] Frederick H. Raab, "Class- F power amplifiers with maximally flat waveforms" IEEE tranactions on circuits and system, Vol. 45, No.11, pp.2007-2012, November 1997.
- [11] G. Gonzalez, "Microwave Transistor Amplifiers: Analysis and Design", Prentice Hall, 1984.

15 Transmitter & PLL System Verification

15.1 VCO with Divider

15.1.1 Circuit schematic

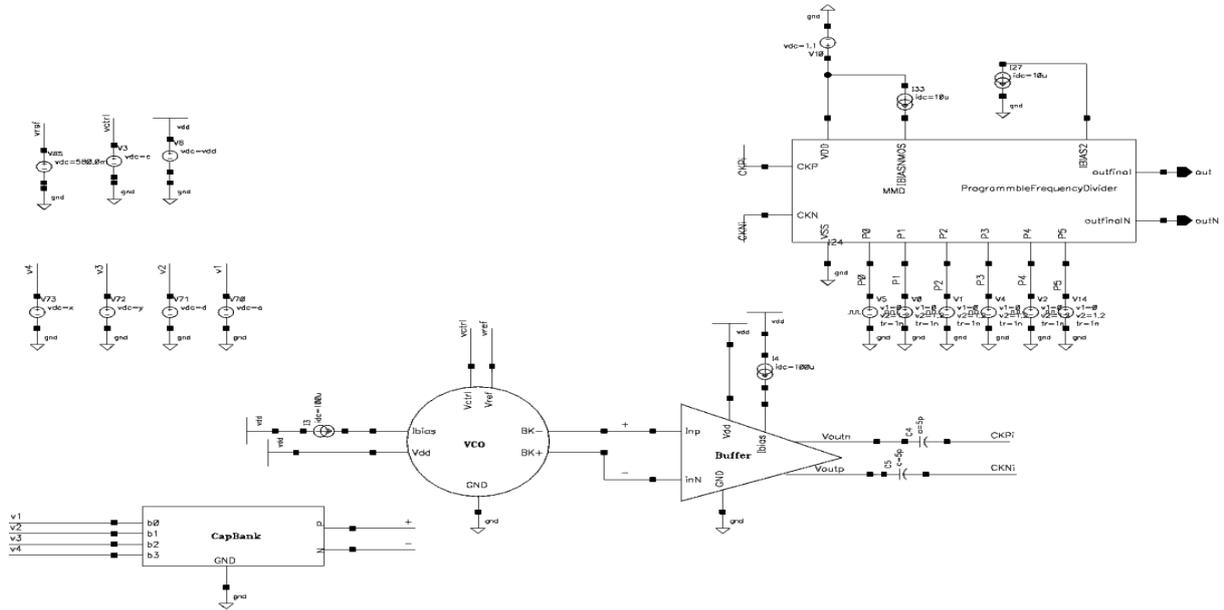
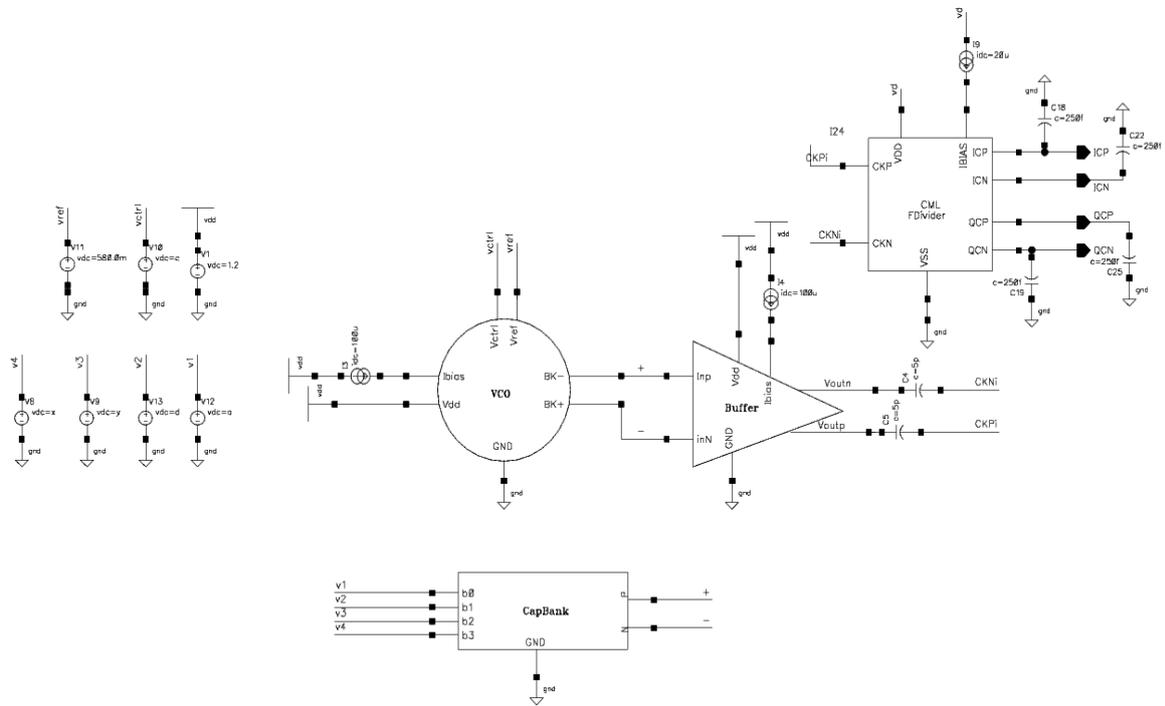


Figure 15.15.1 : VCO with Feedback Divider



-Figure 15.2 : VCO with quadrature generator (IQ) divide by 2

15.1.2 Simulations

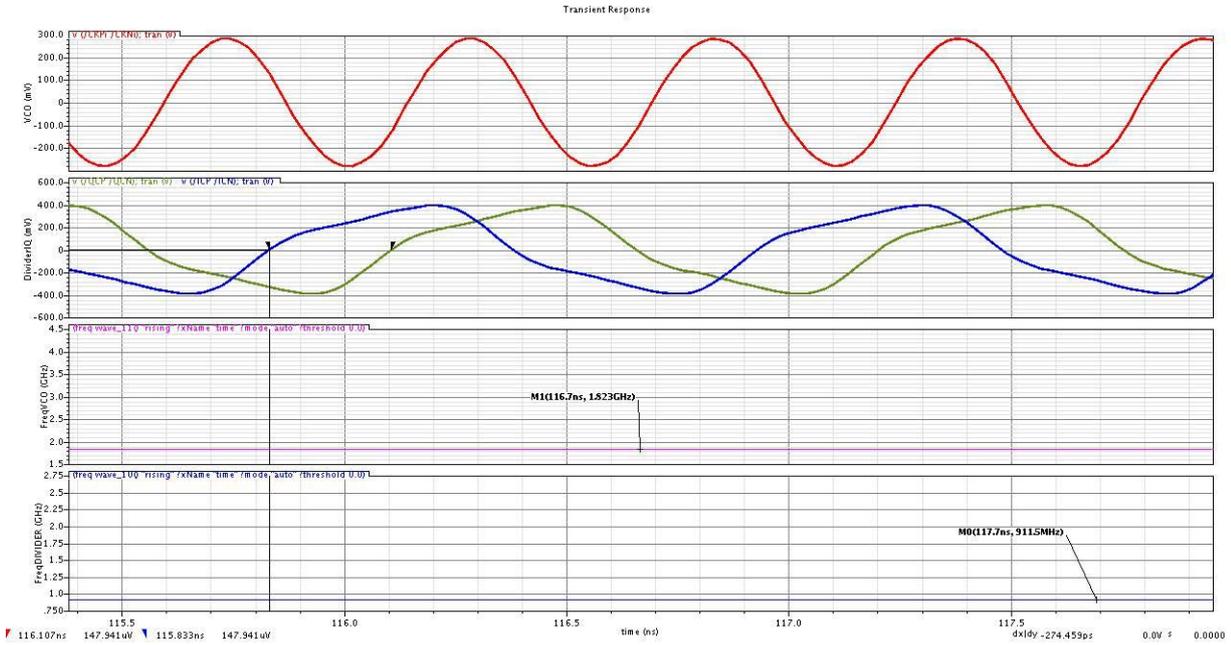


Figure 15.15.3 Transient simulation of VCO and IQ divide by 2 and their Freq

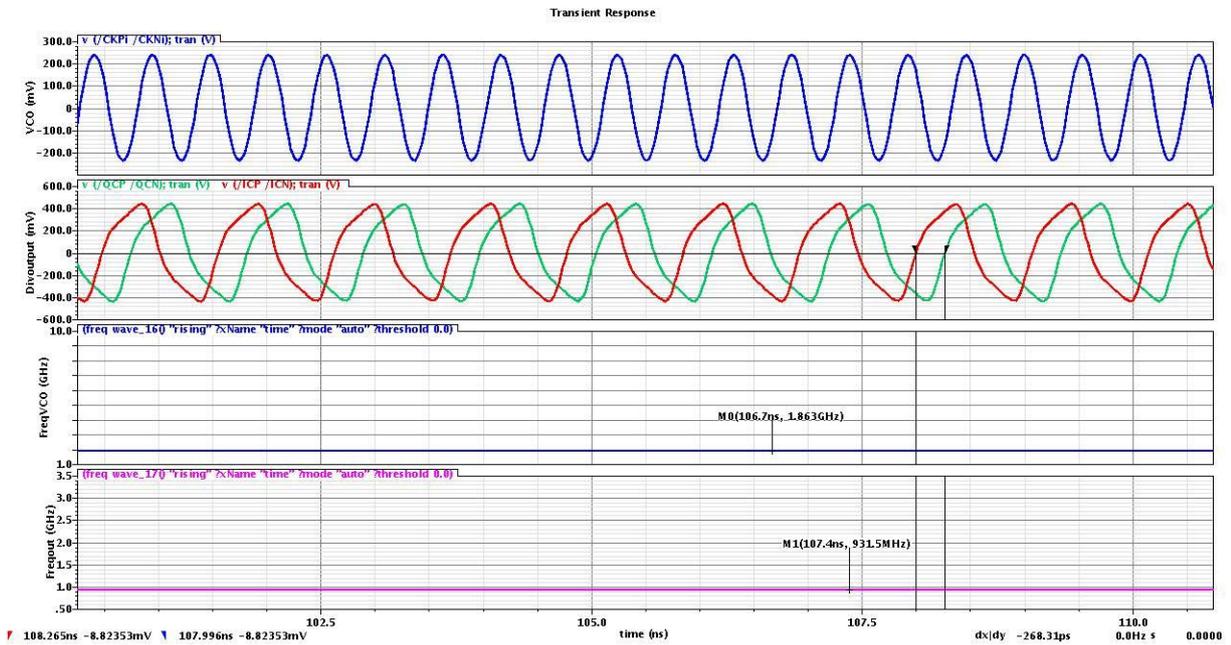


Figure 15.15.4 : Transient simulation of VCO and IQ divide by 2 and their Freq at worst case corner

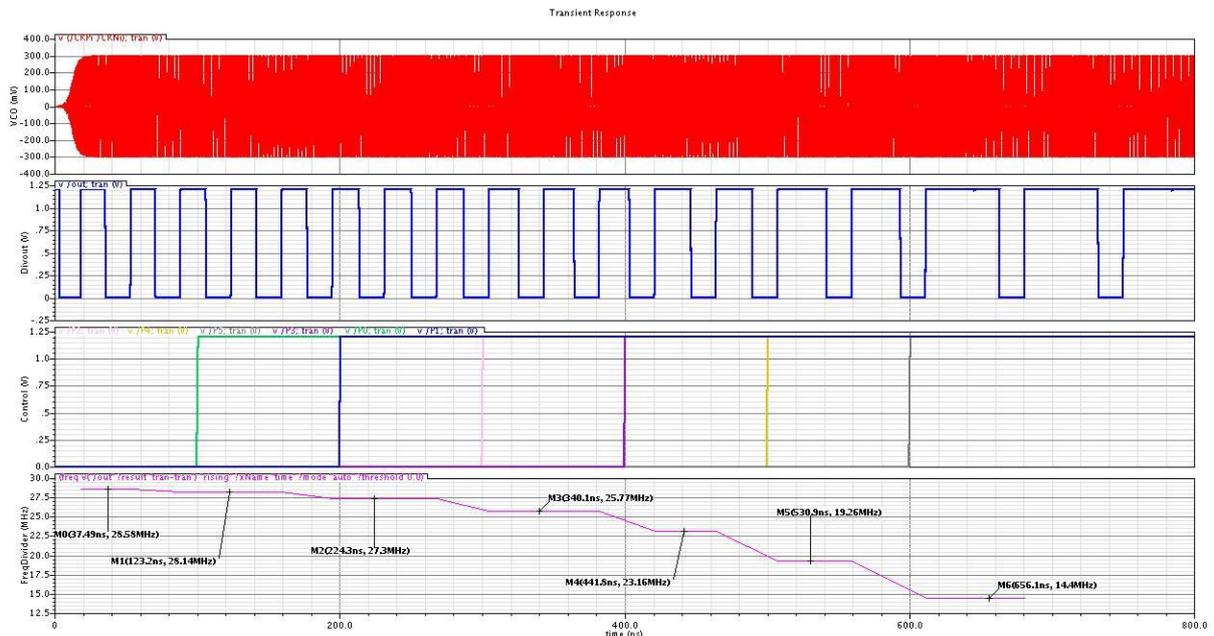


Figure 15.15.5 : Transient simulation of VCO and Feedback divider with variable division ratio

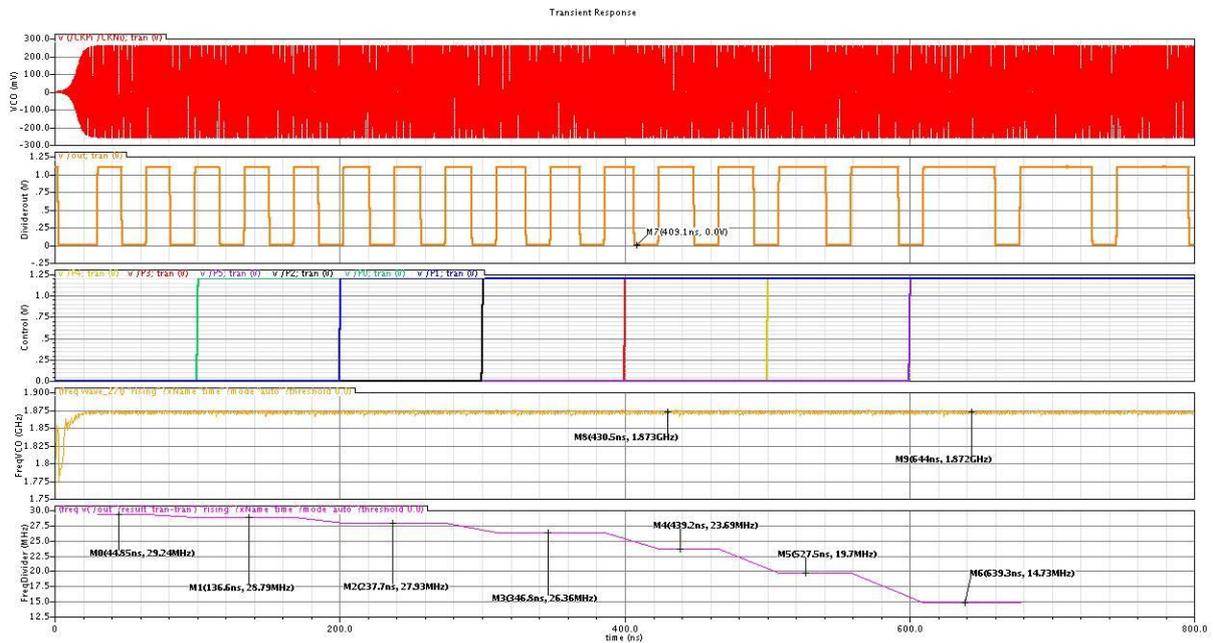


Figure 15.6 : Transient simulation of VCO and Feedback divider with variable division ratio At worst case corner

15.1.3 PSS result

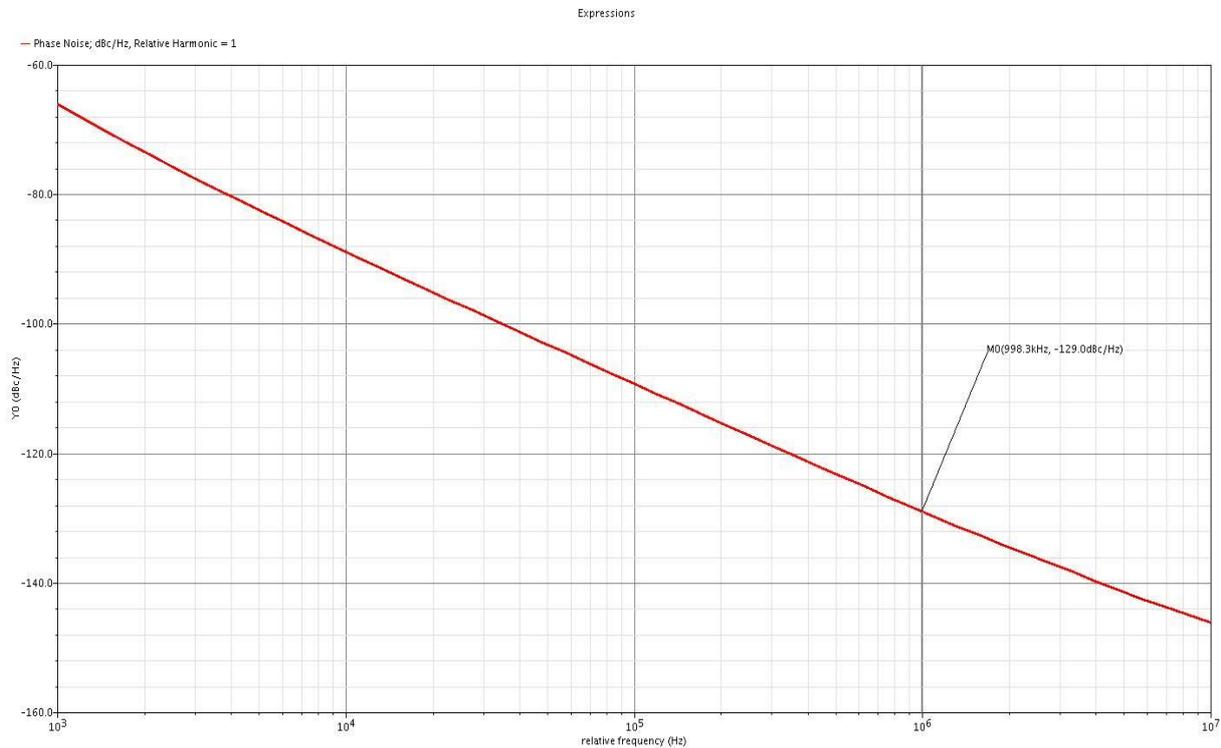


Figure 1515.7 : Phase noise at divider output
PN=-129dBc/Hz

15.1.4 Divider with PA

15.1.5 Introduction

This section discuss the assembly of two blocks divider and power through this section we will discuss the test bench used for both blocks, simulated results and corners.

15.1.6 Divider-PA test bench

The test-bench used in both blocks

1. Input is differential signal 600 mv peak to peak
2. Input frequency is 1.83 GHz
3. I-channel is only used while Q-channel is load with a capacitance of 250 pF
4. I-channel is the input of PA and Mixer to model the capacitance of Mixer and routing capacitance a 70 pF is assumed as load to I-channel with PA
5. The output of PA is port with impedance of 50 ohm that uses a dc source that serves as a load. with source type = dc and dc = 0
6. Two supplies are provided one is 1.2 volt that serves Divider and part of PA and other is 2.5 volt that serves the only part of PA.

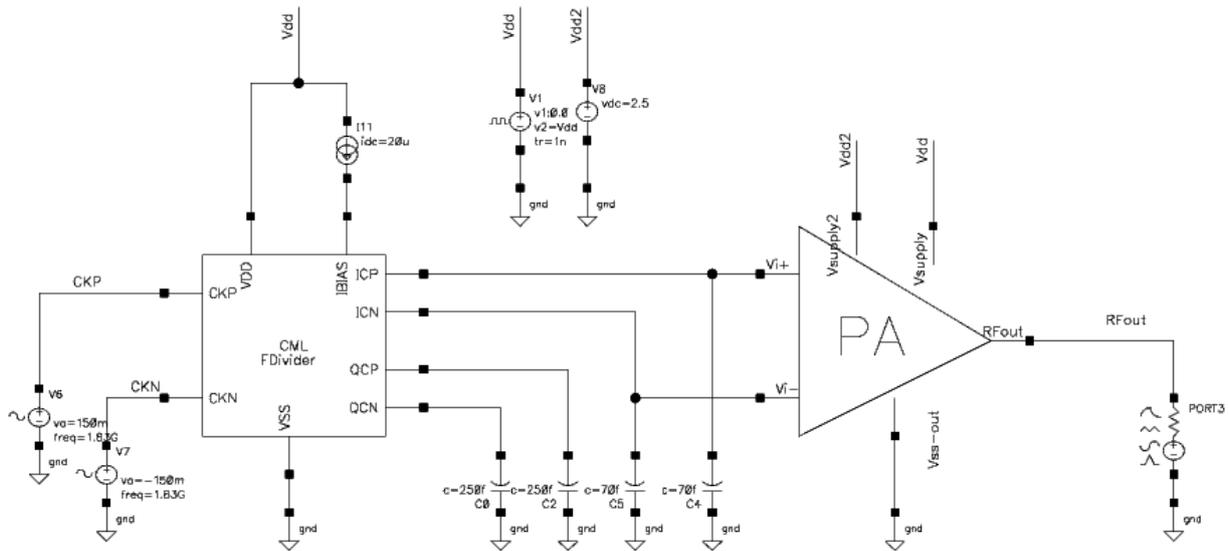


Figure 1515.8 : Divider-PA test bench

15.1.7 Coupling DC of Divider

In order to provide maximum attenuation for second harmonic at antenna the wave supplied must be core of PA must be a perfectly symmetric sine wave. Pre-amplifier is the first stage of PA which requires a DC of 600 mv to give a symmetrical square at output, but the divider provide a DC of 700mv. Form the above it now clear that the solution is coupling the DC of the divider and biasing a DC value of 600mv.

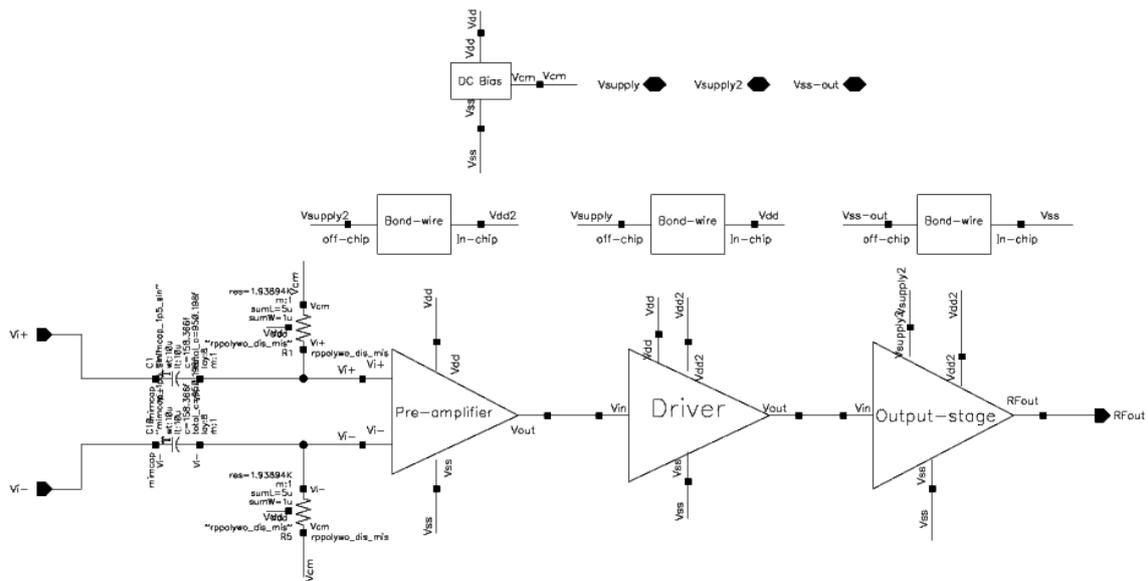


Figure 1515.9 : Power amplifier

15.1.8 Pre-amplifier

The pre-amplifier is adjusted by changing CMOS circuit into a CMOS circuit with a resistive. This is done to make circuit achieves all specification all corners. Resistive feed-back decrease the variations across corners, thus pre-amplifier achieves acceptable duty cycle variation which translate into 2nd harmonic.

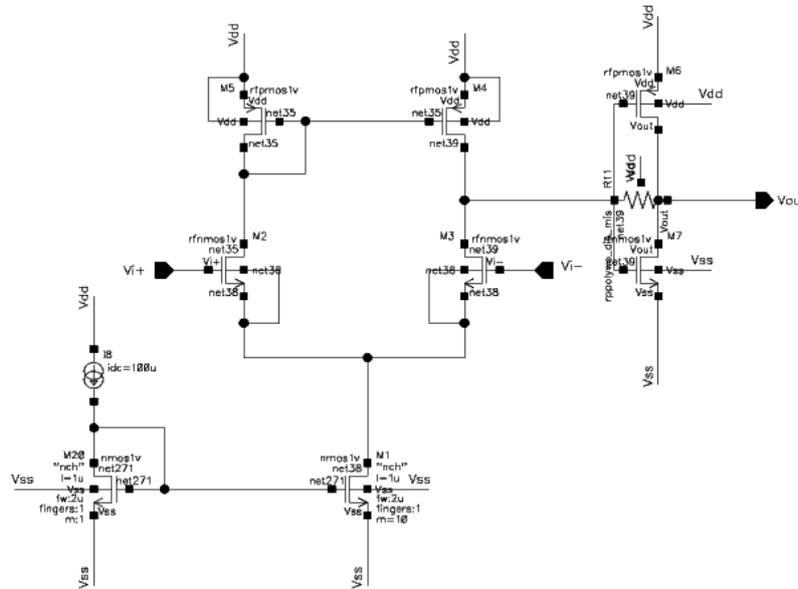


Figure 1515.15.70 : Pre-amplifier

15.1.9 Simulated results

15.1.9.1 Transient analysis

The figure shows the voltage across different nodes in both circuits. The first curves output of PA to the antenna having a frequency of 915MHz, the second curves shows the input to the divider a sine wave having a frequency of 1.83 GHz and the last curves shows the output of divider to PA.

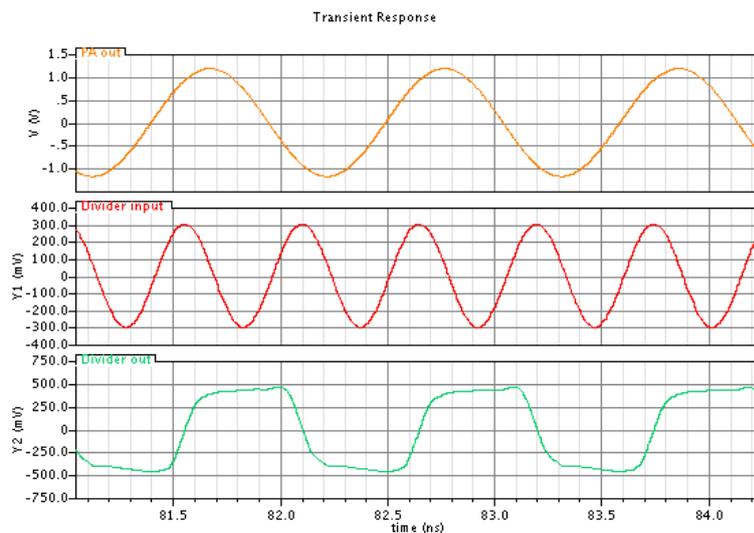


Figure 15.11 : Transient analysis 1

This figure shows the time required for divider and PA to settle nearly equal 35 nsec is required for both blocks to

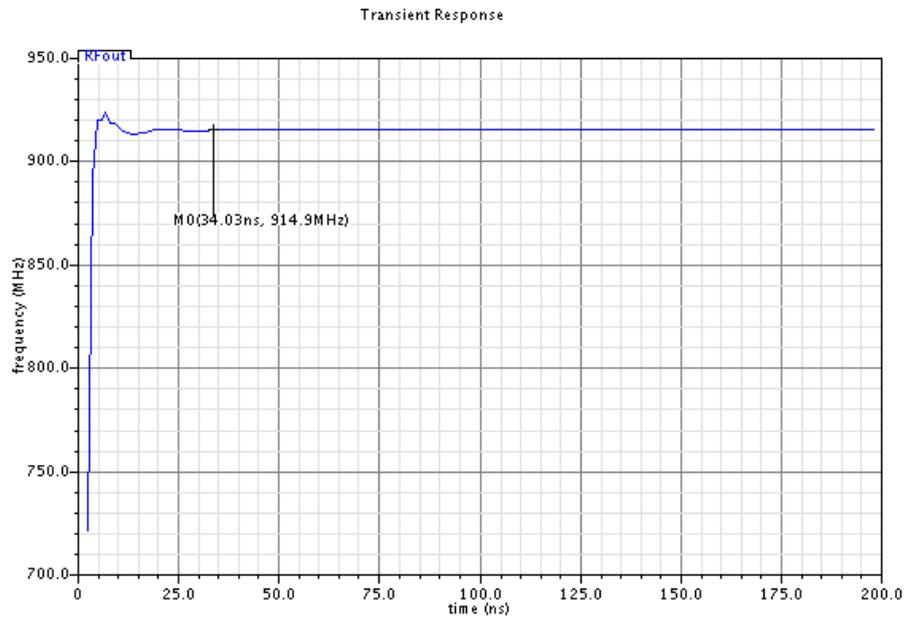


Figure 1515.15.82 : Transient analysis 2

15.1.9.2 PSS analysis

15.1.9.3 Output spectrum

The analysis is done in typical-typical case shows an output +11.43dBm, 2nd harmonic of -38.57dBm and -61.89dBm.

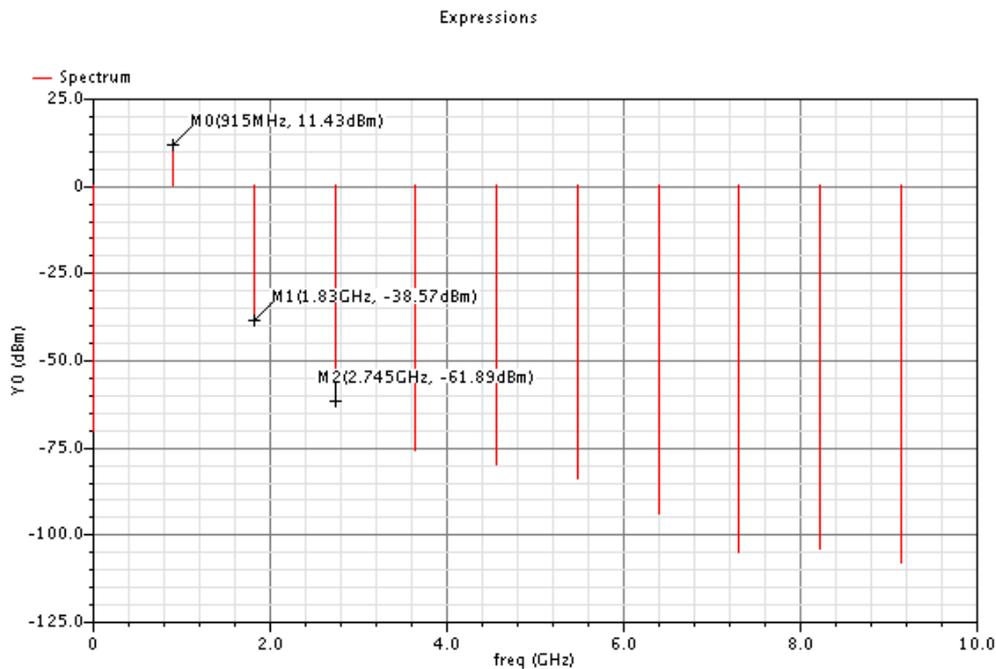


Figure 1515.15.93 : Power spectrum

15.1.9.4 Pout VS Pin

The output figure shows that compression point of both blocks is shifted by 5 dB than PA alone. PA has 1-dB compression -30 dBm while both blocks have -25 dBm.

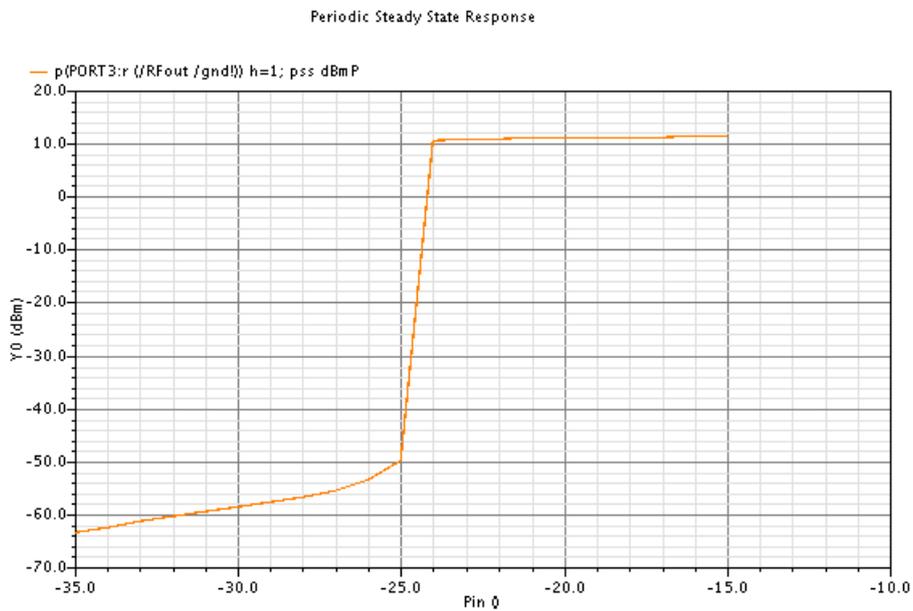


Figure 1515.15.10 : Pout Vs Pin

15.1.10 Corners simulation

6. The extreme corners for output power

	MOS1v	MOS2v	Cap	res	temp	Supply 1	Supply 2	Pout(dBm)
Max	ss	ff	ss	ss	0	1.1	2.4	11.8
Min	ss	ss	ss	ff	85	1.1	2.4	9.6

7. The extreme corners for second harmonic

	MOS1v	MOS2v	Cap	res	temp	Supply 1	Supply 2	2 nd harmonic (dBm)
Max	ss	ff	ss	ff	85	1.1	2.4	-43
Min	ss	fs	ff	ff	0	1.3	2.4	-34.8

8. The extreme corners for Third harmonic

	MOS1v	MOS2v	Cap	res	temp	Supply 1	Supply 2	3 rd harmonic (dBm)
Max	ss	sf	ss	ff	85	1.1	2.4	-73.3
Min	ss	ff	ss	ss	0	1.1	2.6	-59.9

15.2 VCO with Divider and PA

15.2.1 Circuit schematic

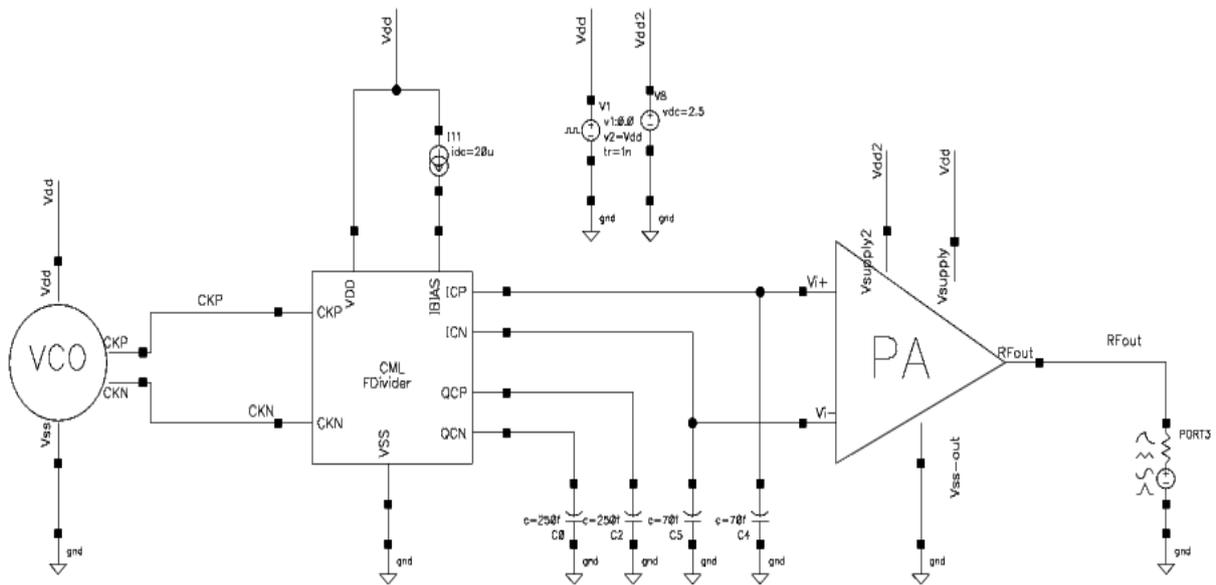


Figure 1515.15.115 : Circuit schematic for VCO with Divider & PA

15.2.2 Transient result

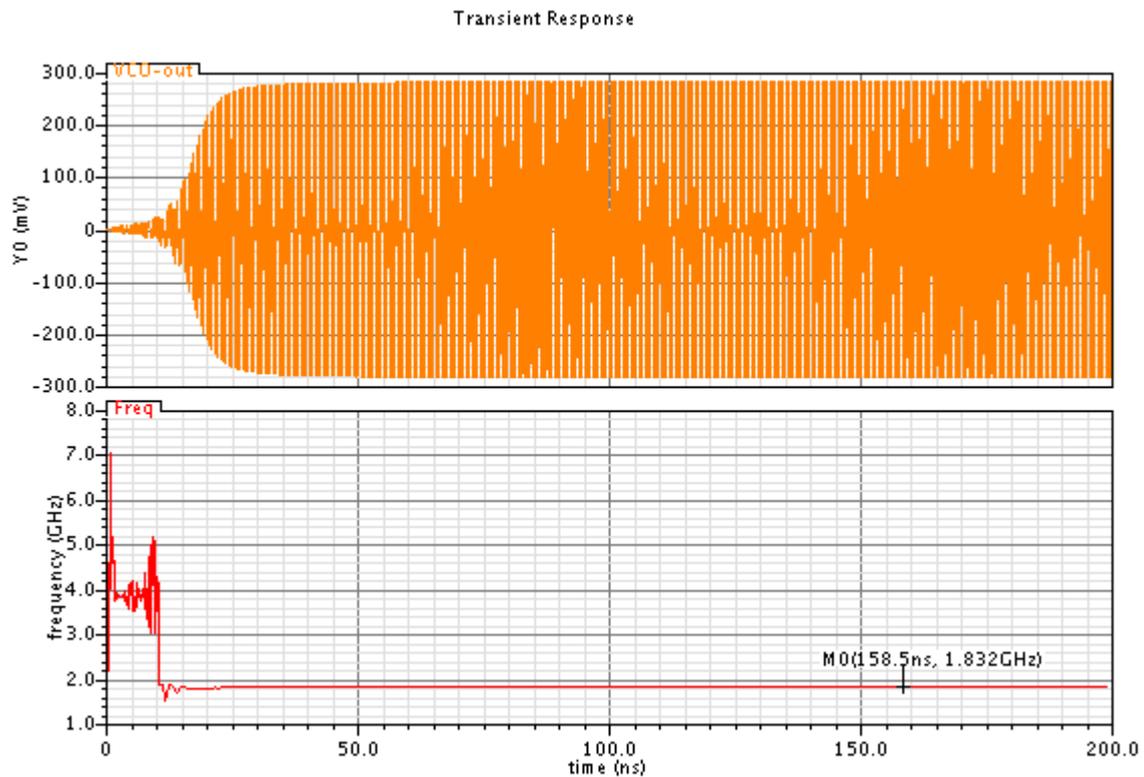


Figure 15.15.124 : VCO transient output

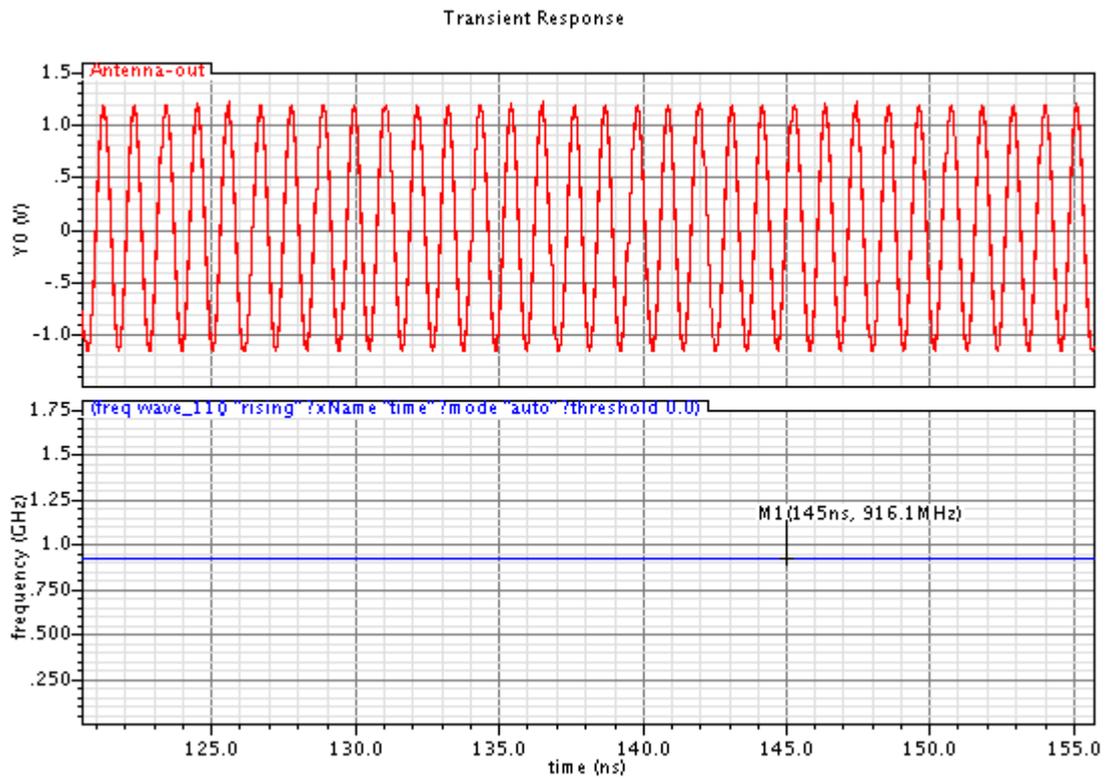


Figure 15.15.137 : PA Output

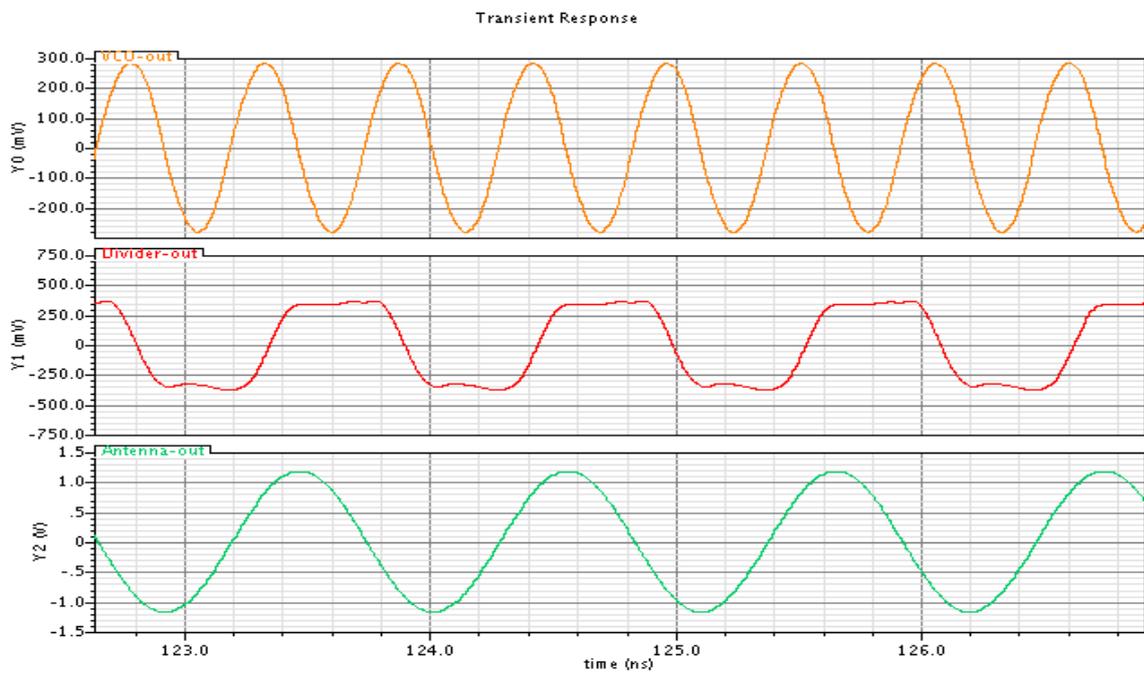


Figure 15.15.148 : Transient output of VCO, Divider & PA

15.2.3 PSS result

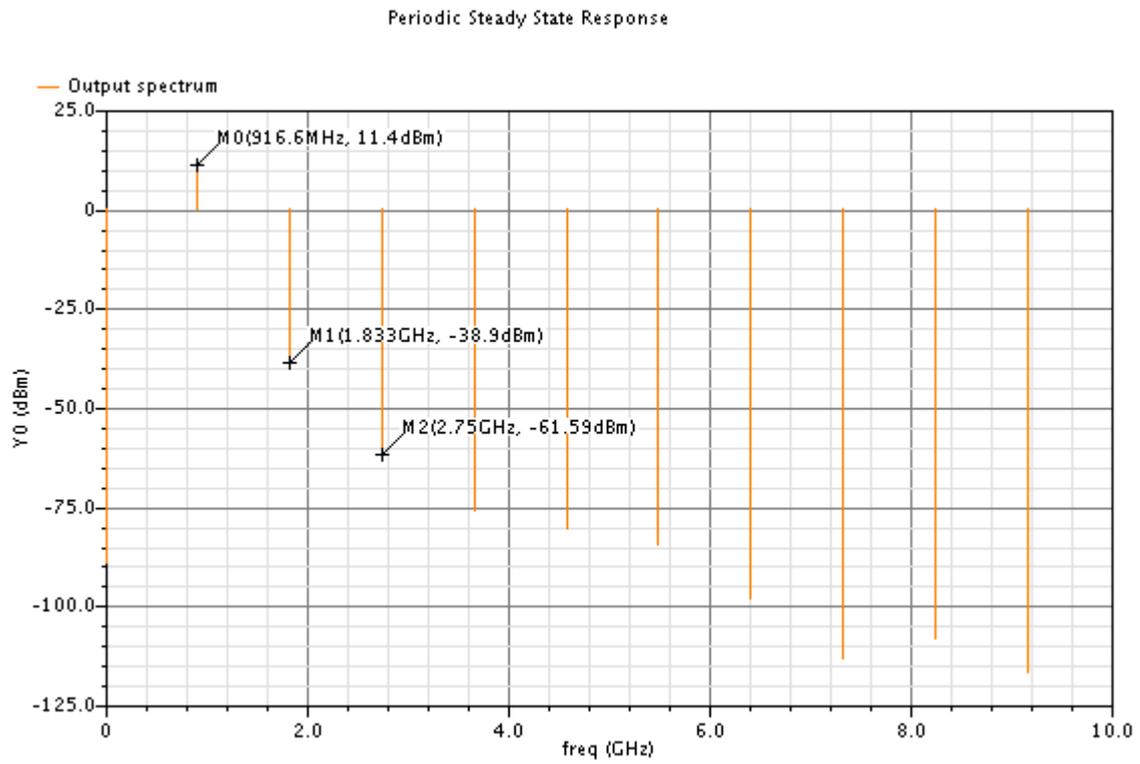


Figure 1515.19 : PA output spectrum

I. Appendix A: Ocean Script Codes

Ocean Script code for Open loop gain of folded cascode OTA "PGA"

```
ocnWaveformTool( 'wavescan )
simulator( 'spectre )
design( "/home/omar/simulation/dasdasda_4/spectre/schematic/netlist/netlist")
resultsDir( "/home/omar/simulation/dasdasda_4/spectre/schematic" )
desVar( "vin" 1u      )
desVar( "pin" -30     )
desVar( "v1" 0       )
desVar( "v2" 0       )
desVar( "v3" 0       )
desVar( "v4" 0       )
desVar( "v5" 0       )
desVar( "v6" 0       )
desVar( "v7" 0       )
desVar( "v8" 0       )
desVar( "w" 100u     )
desVar( "v" 1.2      )
desVar( "rr" 250K    )
for( corner 1 4
for( res 1 2
for( cap 1 2
if( corner == 1 res == 1 && cap == 1
modelFile( ('/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/rf013.scs" "ff")
list("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/ResModel.scs" ff_disres")
list("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/rf013.scs" "ff_mim"))))
if( corner == 1 res == 1 && cap == 2
modelFile( ('/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/rf013.scs" "ff")
list("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/ResModel.scs" ff_disres")
list("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/rf013.scs" "ss_mim"))))
if( corner == 1 res == 2 && cap == 1
```

```

modelFile(("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/rf013.scs" "ff")
list("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/ResModel.scs" "ss_disres")
list("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/rf013.scs" "ff_mim"))))
if( corner == 1 res == 2 && cap == 2
modelFile(("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/rf013.scs" "ff")
list("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/ResModel.scs" "ss_disres")
list("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/rf013.scs" "ss_mim"))))
if( corner == 2 res == 1 && cap == 1
modelFile(("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/rf013.scs" "ss")
list("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/ResModel.scs" "ff_disres")
list("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/rf013.scs" "ff_mim"))))
if( corner == 2 res == 1 && cap == 2
modelFile(("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/rf013.scs" "ss")
list("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/ResModel.scs" "ff_disres")
list("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/rf013.scs" "ss_mim"))))
if( corner == 2 res == 2 && cap == 1
modelFile(("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/rf013.scs" "ss")
list("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/ResModel.scs" "ss_disres")
list("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/rf013.scs" "ff_mim"))))
if( corner == 2 res == 2 && cap == 2
modelFile(("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/rf013.scs" "ss")
list("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/ResModel.scs" "ss_disres")
list("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/rf013.scs" "ss_mim"))))
if( corner == 3
modelFile(("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/rf013.scs" "sf")
list("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/ResModel.scs" "ss_disres")
list("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/rf013.scs" "ss_mim"))))
if( corner == 4
modelFile(("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/rf013.scs" "fs")
list("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/ResModel.scs" "ff_disres")

```

```

list("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/rf013.scs" "ff_min"))
for( vdd 1 2
for( temperature 1 2
if( vdd == 1 desVar( "v" 1.08  ) )
if( vdd == 2 desVar( "v" 1.32  ) )
if( temperature == 1 temp(127) )
if( temperature == 2 temp(-40) )
analysis('ac ?start "1" ?stop "10G" )
run()
v\ \(\net0299\ \net307\); ac\ dB20(V) = db((v("/net0299" ?result "ac") - v("/net307" ?result "ac")))
plot( v\ \(\net0299\ \net307\); ac\ dB20(V) ?expr '( "v (/net0299 /net307); ac dB20(V)" ) ) )
)
)
)
)

```

Ocean Script code for CMFB loop gain of folded cascode OTA "PGA"

```
ocnWaveformTool( 'wavescan )
simulator( 'spectre )
design( "/home/omar/simulation/Final_Op_Amp_First_Stage_3/spectre/schematic/netlist/netlist")
resultsDir( "/home/omar/simulation/Final_Op_Amp_First_Stage_3/spectre/schematic" )
for( corner 1 4
for( res 1 2
for( cap 1 2
if( corner == 1 res == 1 && cap == 1
modelFile( ('/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/rf013.scs" "ff")
list("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/ResModel.scs" "ff_disres")
list("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/rf013.scs" "ff_mim"))
)
if( corner == 1 res == 1 && cap == 2
modelFile( ('/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/rf013.scs" "ff")
list("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/ResModel.scs" "ff_disres")
list("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/rf013.scs" "ss_mim"))
)
if( corner == 1 res == 2 && cap == 1
modelFile( ('/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/rf013.scs" "ff")
list("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/ResModel.scs" "ss_disres")
list("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/rf013.scs" "ff_mim"))
)
if( corner == 1 res == 2 && cap == 2
modelFile( ('/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/rf013.scs" "ff")
list("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/ResModel.scs" "ss_disres")
list("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/rf013.scs" "ss_mim"))
)
;////////////////////////////////////
if( corner == 2 res == 1 && cap == 1
```

```

modelFile(("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/rf013.scs" "ss")
list("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/ResModel.scs" "ff_disres")
list("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/rf013.scs" "ff_mim"))
)
if( corner == 2 res == 1 && cap == 2
modelFile(("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/rf013.scs" "ss")
list("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/ResModel.scs" "ff_disres")
list("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/rf013.scs" "ss_mim"))
)
if( corner == 2 res == 2 && cap == 1
modelFile(("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/rf013.scs" "ss")
list("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/ResModel.scs" "ss_disres")
list("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/rf013.scs" "ff_mim"))
)
if( corner == 2 res == 2 && cap == 2
modelFile(("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/rf013.scs" "ss")
list("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/ResModel.scs" "ss_disres")
list("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/rf013.scs" "ss_mim"))
)
;////////////////////////////////////
if( corner == 3
modelFile(("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/rf013.scs" "sf")
list("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/ResModel.scs" "ss_disres")
list("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/rf013.scs" "ss_mim"))
)
if( corner == 4
modelFile(("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/rf013.scs" "fs")
list("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/ResModel.scs" "ff_disres")
list("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/rf013.scs" "ff_mim"))
)

```


Ocean Script for PM of CMFB loop gain of folded cascode OTA "PGA"

```
ocnWaveformTool( 'wavescan )

simulator( 'spectre )

design( "/home/omar/simulation/Final_Op_Amp_First_Stage_3/spectre/schematic/netlist/netlist")

resultsDir( "/home/omar/simulation/Final_Op_Amp_First_Stage_3/spectre/schematic" )

p = outfile( "/home/omar/Desktop/scr1out.txt" "a" )

fprintf( p "%s \t %s \t %s \t %s \t %s \t %s \t %s \n\n" "mos_thin" "mim" "vddVal" "tempVal" "GBW" "DC_gain"
"Phase_Margin")

for( corner 1 4

for( res 1 2

for( cap 1 2

if( corner == 1 res == 1 && cap == 1

modelFile( ("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/rf013.scs" "ff")

list("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/ResModel.scs" "ff_disres")

list("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/rf013.scs" "ff_mim"))

)

if( corner == 1 res == 1 && cap == 2

modelFile( ("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/rf013.scs" "ff")

list("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/ResModel.scs" "ff_disres")

list("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/rf013.scs" "ss_mim"))

)

if( corner == 1 res == 2 && cap == 1

modelFile( ("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/rf013.scs" "ff")

list("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/ResModel.scs" "ss_disres")

list("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/rf013.scs" "ff_mim"))

)

if( corner == 1 res == 2 && cap == 2

modelFile( ("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/rf013.scs" "ff")

list("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/ResModel.scs" "ss_disres")

list("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/rf013.scs" "ss_mim"))
```

```

)
if( corner == 2 res == 1 && cap == 1
modelFile( ('/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/rf013.scs" "ss")
list("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/ResModel.scs" "ff_disres")
list("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/rf013.scs" "ff_mim"))
)
if( corner == 2 res == 1 && cap == 2
modelFile( ('/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/rf013.scs" "ss")
list("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/ResModel.scs" "ff_disres")
list("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/rf013.scs" "ss_mim"))
)
if( corner == 2 res == 2 && cap == 1
modelFile( ('/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/rf013.scs" "ss")
list("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/ResModel.scs" "ss_disres")
list("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/rf013.scs" "ff_mim"))
)
if( corner == 2 res == 2 && cap == 2
modelFile( ('/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/rf013.scs" "ss")
list("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/ResModel.scs" "ss_disres")
list("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/rf013.scs" "ss_mim"))
)
if( corner == 3
modelFile( ('/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/rf013.scs" "sf")
list("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/ResModel.scs" "ss_disres")
list("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/rf013.scs" "ss_mim"))
)
if( corner == 4
modelFile( ('/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/rf013.scs" "fs")
list("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/ResModel.scs" "ff_disres")
list("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/rf013.scs" "ff_mim"))

```

```
)  
for( vdd 1 2  
for( temperature 1 2  
if( vdd == 1 desVar( "v" 1.08 ))  
if( vdd == 2 desVar( "v" 1.32 ))  
if( temperature == 1 temp(127) )  
if( temperature == 2 temp(-40) )  
analysis('stb ?start "1" ?stop "10G" ?probe "/V0" )  
run()  
Phase_Margin = getData("phaseMargin" ?result "stb_margin")  
fprintf( p "%1.3n \n" Phase_Margin )  
)  
)  
)  
)  
)  
)
```

Ocean Script code for NF of PGA

```
ocnWaveformTool( 'wavescan )
simulator( 'spectre )

design( "/home/omar/simulation/dasdasda_4/spectre/schematic/netlist/netlist")

resultsDir( "/home/omar/simulation/dasdasda_4/spectre/schematic" )

desVar( "w" 100u      )
desVar( "pin" -30     )
desVar( "v3" 01.2     )
desVar( "v2" 0       )
desVar( "v1" 01.2     )
desVar( "v7" 0       )
desVar( "v8" 01.2     )
desVar( "v5" 0       )
desVar( "v6" 0       )
desVar( "v4" 0       )

for( corner 1 4
for( res 1 2
for( cap 1 2

if( corner == 1 res == 1 && cap == 1

modelFile( ('/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/rf013.scs" "ff")
list("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/ResModel.scs" "ff_disres")
list("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/rf013.scs" "ff_mim"))))
if( corner == 1 res == 1 && cap == 2

modelFile( ('/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/rf013.scs" "ff")
list("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/ResModel.scs" "ff_disres")
list("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/rf013.scs" "ss_mim"))))
if( corner == 1 res == 2 && cap == 1

modelFile( ('/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/rf013.scs" "ff")
list("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/ResModel.scs" "ss_disres")
list("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/rf013.scs" "ff_mim"))))
if( corner == 1 res == 2 && cap == 2

modelFile( ('/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/rf013.scs" "ff")
```

```

list("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/ResModel.scs" "ss_disres")
list("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/rf013.scs" "ss_mim"))
if( corner == 2 res == 1 && cap == 1
modelFile( ("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/rf013.scs" "ss")
list("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/ResModel.scs" "ff_disres")
list("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/rf013.scs" "ff_mim"))
if( corner == 2 res == 1 && cap == 2
modelFile( ("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/rf013.scs" "ss")
list("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/ResModel.scs" "ff_disres")
list("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/rf013.scs" "ss_mim"))
if( corner == 2 res == 2 && cap == 1
modelFile( ("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/rf013.scs" "ss")
list("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/ResModel.scs" "ss_disres")
list("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/rf013.scs" "ff_mim"))
if( corner == 2 res == 2 && cap == 2
modelFile( ("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/rf013.scs" "ss")
list("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/ResModel.scs" "ss_disres")
list("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/rf013.scs" "ss_mim"))
if( corner == 3
modelFile( ("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/rf013.scs" "sf")
list("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/ResModel.scs" "ss_disres")
list("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/rf013.scs" "ss_mim"))
if( corner == 4
modelFile( ("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/rf013.scs" "fs")
list("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/ResModel.scs" "ff_disres")
list("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/rf013.scs" "ff_mim"))
for( vdd 1 2
for( temperature 1 2
if( vdd == 1 desVar( "v" 1.08 ))
if( vdd == 2 desVar( "v" 1.32 ))
if( temperature == 1 temp(127) )
if( temperature == 2 temp(-40) )

```

```
analysis('noise ?start "80K" ?stop "2000K" ?p "/net0299"  
        ?n "/net307" ?oprobe "" ?iprobe "/PORT0" )  
  
run()  
  
noise\ figure = getData("NF" ?result "noise")  
  
plot( noise\ figure ?expr '( "noise figure" ) )  
  
)  
  
)  
  
)  
  
)  
  
)
```

Ocean Script code for Open loop gain of folded cascode OTA "PD 1&2"

```
ocnWaveformTool( 'wavescan )
simulator( 'spectre )
design( "/home/omar/simulation/PD_4/spectre/schematic/netlist/netlist")
resultsDir( "/home/omar/simulation/PD_4/spectre/schematic" )
p = outfile( "/home/omar/Desktop/PM_PD.txt" "a" )
fprintf( p "%s \t %s \t %s \t %s \t %s \t %s \t %s \n\n" "mos_thin" "mim" "vddVal" "tempVal" "GBW" "DC_gain"
"Phase_Margin")
for( corner 1 4
for( res 1 2
for( cap 1 2
if( corner == 1 res == 1 && cap == 1
modelFile( ("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/rf013.scs" "ff")
list("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/ResModel.scs" "ff_disres")
list("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/rf013.scs" "ff_mim"))))
if( corner == 1 res == 1 && cap == 2
modelFile( ("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/rf013.scs" "ff")
list("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/ResModel.scs" "ff_disres")
list("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/rf013.scs" "ss_mim"))))
if( corner == 1 res == 2 && cap == 1
modelFile( ("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/rf013.scs" "ff")
list("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/ResModel.scs" "ss_disres")
list("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/rf013.scs" "ff_mim"))))
if( corner == 1 res == 2 && cap == 2
modelFile( ("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/rf013.scs" "ff")
list("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/ResModel.scs" "ss_disres")
list("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/rf013.scs" "ss_mim"))))
if( corner == 2 res == 1 && cap == 1
modelFile( ("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/rf013.scs" "ss")
list("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/ResModel.scs" "ff_disres"))
```

```

list("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/rf013.scs" "ff_mim"))
if( corner == 2 res == 1 && cap == 2
modelFile( ('/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/rf013.scs" "ss")
list("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/ResModel.scs" "ff_disres")
list("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/rf013.scs" "ss_mim"))
if( corner == 2 res == 2 && cap == 1
modelFile( ('/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/rf013.scs" "ss")
list("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/ResModel.scs" "ss_disres")
list("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/rf013.scs" "ff_mim"))
if( corner == 2 res == 2 && cap == 2
modelFile( ('/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/rf013.scs" "ss")
list("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/ResModel.scs" "ss_disres")
list("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/rf013.scs" "ss_mim"))
if( corner == 3
modelFile( ('/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/rf013.scs" "sf")
list("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/ResModel.scs" "ss_disres")
list("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/rf013.scs" "ss_mim"))
if( corner == 4
modelFile( ('/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/rf013.scs" "fs")
list("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/ResModel.scs" "ff_disres")
list("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/rf013.scs" "ff_mim"))
for( vdd 1 2
for( temperature 1 2
if( vdd == 1 desVar( "v" 1.08 ))
if( vdd == 2 desVar( "v" 1.32 ))
if( temperature == 1 temp(127) )
if( temperature == 2 temp(-40) )
analysis('ac ?start "1" ?stop "10G" )
run()
selectResult( 'ac )

```

```
v\ \net0158;\ ac\ dB20(V) = db(v("/net0158" ?result "ac"))  
plot( v\ \net0158;\ ac\ dB20(V) ?expr '( "v /net0158; ac dB20(V)" ) )  
)  
)  
)  
)  
)  
close(p)
```

Ocean Script for CMFB loop gain of folded cascode OTA "PD 1&2"

```
ocnWaveformTool( 'wavescan )

simulator( 'spectre )

design( "/home/omar/simulation/PD_4/spectre/schematic/netlist/netlist")

resultsDir( "/home/omar/simulation/PD_4/spectre/schematic" )

p = outfile( "/home/omar/Desktop/PM_PD.txt" "a" )

fprintf( p "%s \t %s \t %s \t %s \t %s \t %s \t %s \n\n" "mos_thin" "mim" "vddVal" "tempVal" "GBW" "DC_gain"
"Phase_Margin")

for( corner 1 4

for( res 1 2

for( cap 1 2

if( corner == 1 res == 1 && cap == 1

modelFile( ("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/rf013.scs" "ff")

list("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/ResModel.scs" "ff_disres")

list("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/rf013.scs" "ff_mim"))

)

if( corner == 1 res == 1 && cap == 2

modelFile( ("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/rf013.scs" "ff")

list("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/ResModel.scs" "ff_disres")

list("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/rf013.scs" "ss_mim"))

if( corner == 1 res == 2 && cap == 1

modelFile( ("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/rf013.scs" "ff")

list("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/ResModel.scs" "ss_disres")

list("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/rf013.scs" "ff_mim"))

if( corner == 1 res == 2 && cap == 2

modelFile( ("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/rf013.scs" "ff")

list("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/ResModel.scs" "ss_disres")

list("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/rf013.scs" "ss_mim"))

if( corner == 2 res == 1 && cap == 1

modelFile( ("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/rf013.scs" "ss")
```

```

list("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/ResModel.scs" "ff_disres")
list("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/rf013.scs" "ff_mim"))))
if( corner == 2 res == 1 && cap == 2
modelFile( ("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/rf013.scs" "ss")
list("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/ResModel.scs" "ff_disres")
list("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/rf013.scs" "ss_mim"))))
if( corner == 2 res == 2 && cap == 1
modelFile( ("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/rf013.scs" "ss")
list("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/ResModel.scs" "ss_disres")
list("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/rf013.scs" "ff_mim"))))
if( corner == 2 res == 2 && cap == 2
modelFile( ("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/rf013.scs" "ss")
list("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/ResModel.scs" "ss_disres")
list("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/rf013.scs" "ss_mim"))))
if( corner == 3
modelFile( ("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/rf013.scs" "sf")
list("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/ResModel.scs" "ss_disres")
list("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/rf013.scs" "ss_mim"))))
if( corner == 4
modelFile( ("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/rf013.scs" "fs")
list("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/ResModel.scs" "ff_disres")
list("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/rf013.scs" "ff_mim"))))
for( vdd 1 2
for( temperature 1 2
if( vdd == 1 desVar( "v" 1.08 ) )
if( vdd == 2 desVar( "v" 1.32 ) )
if( temperature == 1 temp(127) )
if( temperature == 2 temp(-40) )
analysis('stb ?start "1" ?stop "10G" ?probe "/V0" )
run()

```

```
selectResult( 'stb )  
  
Loop\ Gain\ dB20 = db(mag(getData("loopGain" ?result "stb")))  
  
plot( Loop\ Gain\ dB20 ?expr '( "Loop Gain dB20" ) )  
  
)  
  
)  
  
)  
  
)  
  
)  
  
close(p)
```

Ocean Script for PM of CMFB loop gain of OTA "PD 1&2"

```
ocnWaveformTool( 'wavescan )

simulator( 'spectre )

design( "/home/omar/simulation/PD_4/spectre/schematic/netlist/netlist")

resultsDir( "/home/omar/simulation/PD_4/spectre/schematic" )

p = outfile( "/home/omar/Desktop/PM_PD.txt" "a" )

fprintf( p "%s \t %s \t %s \t %s \t %s \t %s \t %s \n\n" "mos_thin" "mim" "vddVal" "tempVal" "GBW" "DC_gain"
"Phase_Margin")

for( corner 1 4

for( res 1 2

for( cap 1 2

if( corner == 1 res == 1 && cap == 1

modelFile( ("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/rf013.scs" "ff")
list("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/ResModel.scs" "ff_disres")
list("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/rf013.scs" "ff_mim"))))

if( corner == 1 res == 1 && cap == 2

modelFile( ("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/rf013.scs" "ff")
list("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/ResModel.scs" "ff_disres")
list("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/rf013.scs" "ss_mim"))))

if( corner == 1 res == 2 && cap == 1

modelFile( ("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/rf013.scs" "ff")
list("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/ResModel.scs" "ss_disres")
list("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/rf013.scs" "ff_mim"))))

if( corner == 1 res == 2 && cap == 2

modelFile( ("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/rf013.scs" "ff")
list("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/ResModel.scs" "ss_disres")
list("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/rf013.scs" "ss_mim"))))

if( corner == 2 res == 1 && cap == 1

modelFile( ("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/rf013.scs" "ss")
list("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/ResModel.scs" "ff_disres")
```

```

list("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/rf013.scs" "ff_mim"))
if( corner == 2 res == 1 && cap == 2
modelFile( ("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/rf013.scs" "ss")
list("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/ResModel.scs" "ff_disres")
list("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/rf013.scs" "ss_mim"))
if( corner == 2 res == 2 && cap == 1
modelFile( ("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/rf013.scs" "ss")
list("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/ResModel.scs" "ss_disres")
list("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/rf013.scs" "ff_mim"))
if( corner == 2 res == 2 && cap == 2
modelFile( ("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/rf013.scs" "ss")
list("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/ResModel.scs" "ss_disres")
list("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/rf013.scs" "ss_mim"))
if( corner == 3
modelFile( ("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/rf013.scs" "sf")
list("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/ResModel.scs" "ss_disres")
list("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/rf013.scs" "ss_mim"))
if( corner == 4
modelFile( ("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/rf013.scs" "fs")
list("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/ResModel.scs" "ff_disres")
list("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/rf013.scs" "ff_mim"))
for( vdd 1 2
for( temperature 1 2
if( vdd == 1 desVar( "v" 1.08 ) )
if( vdd == 2 desVar( "v" 1.32 ) )
if( temperature == 1 temp(127) )
if( temperature == 2 temp(-40) )
analysis('stb ?start "1" ?stop "10G" ?probe "/V0" )
run()
selectResult( 'stb )

```

```
Phase_Margin = getData("phaseMargin" ?result "stb_margin")
```

```
fprintf( p "%1.3f \n" Phase_Margin )
```

```
)
```

```
)
```

```
)
```

```
)
```

```
)
```

```
close(p)
```

ADC Ocean script

Reference buffer ocean script:

```
ocnWaveformTool( 'wavescan )
simulator( 'spectre )
design( "/home/amrhema/simulation/ref_buffer_with_switch/spectre/schematic/netlist/netlist")
resultsDir( "/home/amrhema/simulation/ref_buffer_with_switch/spectre/schematic" )
desVar(      "Vdd" 1.2   )
counter=0
tmp='(0 85)
corner_txt=('ss" "ff" "sf" "fs")
cap_res_txt=('ss" "ff")
Vss=(1.1 1.3)
fp=outfile("/home/amrhema/Desktop/out.txt" "a")
fprintf(fp "phase_margin corner res cap temp supply\n")
close(fp)
foreach(corner corner_txt

foreach(res cap_res_txt

foreach(cap cap_res_txt
    modelFile(
        list("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/rf013.scs"
    sprintf( nil "%s" corner ))

        list("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/rf013.scs"
    sprintf( nil "%s_rfmos" corner ))

        list("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/ResModel.sc
s" sprintf( nil "%s_disres" res ))
```

```

list("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/rf013.scs"
sprintf( nil "%s_mim" cap ))
)

foreach(temp tmp
temp( temp )
foreach(supply Vss

desVar( "Vdd" supply)
analysis('tran ?stop "1u" ?errpreset "moderate" )
analysis('dc ?saveOppoint t )
analysis('stb ?start "1" ?stop "1G" ?step "10M"
?probe "/IPRB0" )

run()

sprintf(label "corner(NP-%s_mim-%s_res-%s_temp-%n_supply-%n_count(%n))" corner cap res
temp supply counter)

Phase\ Margin = getData("phaseMargin" ?result "stb_margin")
plot( Phase\ Margin ?expr list(label) '( "Phase Margin" ) )
fp=outfile("/home/amrhema/Desktop/out.txt" "a")
fprintf(fp "%n %s %s %s %n %n \n" Phase\ Margin corner res cap temp supply)
close(fp)
)

```

Results :

```

phase_margin corner res cap temp supply
80.824109 ss ss ss 0 1.100000
80.124197 ss ss ss 0 1.300000

```

81.365326 ss ss ss 85 1.100000
80.622467 ss ss ss 85 1.300000
80.824109 ss ss ff 0 1.100000
80.124197 ss ss ff 0 1.300000
81.365326 ss ss ff 85 1.100000
80.622467 ss ss ff 85 1.300000
80.824109 ss ff ss 0 1.100000
80.124197 ss ff ss 0 1.300000
81.365326 ss ff ss 85 1.100000
80.622467 ss ff ss 85 1.300000
80.824109 ss ff ff 0 1.100000
80.124197 ss ff ff 0 1.300000
81.365326 ss ff ff 85 1.100000
80.622467 ss ff ff 85 1.300000
80.999799 ff ss ss 0 1.100000
80.050359 ff ss ss 0 1.300000
81.623770 ff ss ss 85 1.100000
80.653976 ff ss ss 85 1.300000
80.999799 ff ss ff 0 1.100000
80.050359 ff ss ff 0 1.300000
81.623770 ff ss ff 85 1.100000
80.653976 ff ss ff 85 1.300000
80.999799 ff ff ss 0 1.100000
80.050359 ff ff ss 0 1.300000
81.623770 ff ff ss 85 1.100000
80.653976 ff ff ss 85 1.300000
80.999799 ff ff ff 0 1.100000

80.050359 ff ff ff 0 1.300000
81.623770 ff ff ff 85 1.100000
80.653976 ff ff ff 85 1.300000
81.373480 sf ss ss 0 1.100000
80.542732 sf ss ss 0 1.300000
81.951736 sf ss ss 85 1.100000
81.083924 sf ss ss 85 1.300000
81.373480 sf ss ff 0 1.100000
80.542732 sf ss ff 0 1.300000
81.951736 sf ss ff 85 1.100000
81.083924 sf ss ff 85 1.300000
81.373480 sf ff ss 0 1.100000
80.542732 sf ff ss 0 1.300000
81.951736 sf ff ss 85 1.100000
81.083924 sf ff ss 85 1.300000
81.373480 sf ff ff 0 1.100000
80.542732 sf ff ff 0 1.300000
81.951736 sf ff ff 85 1.100000
81.083924 sf ff ff 85 1.300000
80.464861 fs ss ss 0 1.100000
79.658936 fs ss ss 0 1.300000
81.046443 fs ss ss 85 1.100000
80.207940 fs ss ss 85 1.300000
80.464861 fs ss ff 0 1.100000
79.658936 fs ss ff 0 1.300000
81.046443 fs ss ff 85 1.100000
80.207940 fs ss ff 85 1.300000

80.464861 fs ff ss 0 1.100000

79.658936 fs ff ss 0 1.300000

81.046443 fs ff ss 85 1.100000

80.207940 fs ff ss 85 1.300000

80.464861 fs ff ff 0 1.100000

79.658936 fs ff ff 0 1.300000

81.046443 fs ff ff 85 1.100000

80.207940 fs ff ff 85 1.300000

Input buffer ocean script:

```
ocnWaveformTool( 'wavescan' )
```

```
simulator( 'spectre' )
```

```
design( "/home/amrhema/simulation/folded_cascode_inp_Buffer5/spectre/schematic/netlist/netlist" )
```

```
resultsDir( "/home/amrhema/simulation/folded_cascode_inp_Buffer5/spectre/schematic" )
```

```
desVar( "Vdd" 1.2 )
```

```
counter=0
```

```
tmp='(0 85)
```

```
corner_txt=('"ss" "ff" "sf" "fs")
```

```
cap_res_txt=('"ss" "ff")
```

```
Vss='(1.1 1.3)
```

```
fp=outfile("/home/amrhema/Desktop/out_buf.txt" "a")
```

```
fprintf(fp "phase_margin corner res cap temp supply\n")
```

```
close(fp)
```

```
foreach(corner corner_txt
```

```
foreach(res cap_res_txt
```

```

foreach(cap cap_res_txt
    modelFile(
        list("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/rf013.scs"
        sprintf( nil "%s" corner ))

        list("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/rf013.scs"
        sprintf( nil "%s_rfmos" corner ))

        list("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/ResModel.sc
s" sprintf( nil "%s_disres" res ))

        list("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/rf013.scs"
        sprintf( nil "%s_mim" cap ))
    )

foreach(temp tmp
    temp( temp )

foreach(supply Vss

desVar(      "Vdd" supply)

counter=counter+1

analysis('dc ?saveOppoint t )

analysis('stb ?start "1" ?stop "1G" ?step "10M"
        ?probe "/I25/vinj" )

run()

sprintf(label "corner(NP-%s_mim-%s_res-%s_temp-%n_supply-%n_count(%n))" corner cap res
temp supply counter)

Phase\ Margin = getData("phaseMargin" ?result "stb_margin")

plot( Phase\ Margin ?expr '( "Phase Margin" ) )

fp=outfile("/home/amrhema/Desktop/out_buf.txt" "a")

```

```
fprintf(fp "%n %s %s %s %n %n \n" Phase\ Margin corner res cap temp supply)
```

```
close(fp)
```

```
)
```

```
)
```

```
)
```

```
)
```

```
)
```

Result :

```
phase_margin corner res cap temp supply
```

```
93.231455 ss ss ss 0 1.100000
```

```
53.782745 ss ss ss 0 1.300000
```

```
106.159941 ss ss ss 85 1.100000
```

```
64.203672 ss ss ss 85 1.300000
```

```
71.110530 ss ss ff 0 1.100000
```

```
50.351604 ss ss ff 0 1.300000
```

```
80.740454 ss ss ff 85 1.100000
```

```
54.426342 ss ss ff 85 1.300000
```

```
93.231455 ss ff ss 0 1.100000
```

```
53.782745 ss ff ss 0 1.300000
```

```
106.159941 ss ff ss 85 1.100000
```

```
64.203672 ss ff ss 85 1.300000
```

```
71.110530 ss ff ff 0 1.100000
```

```
50.351604 ss ff ff 0 1.300000
```

```
80.740454 ss ff ff 85 1.100000
```

```
54.426342 ss ff ff 85 1.300000
```

```
77.336911 ff ss ss 0 1.100000
```

```
72.209303 ff ss ss 0 1.300000
```

77.671059 ff ss ss 85 1.100000
65.809384 ff ss ss 85 1.300000
74.378610 ff ss ff 0 1.100000
68.387871 ff ss ff 0 1.300000
67.785019 ff ss ff 85 1.100000
70.519231 ff ss ff 85 1.300000
77.336911 ff ff ss 0 1.100000
72.209303 ff ff ss 0 1.300000
77.671059 ff ff ss 85 1.100000
65.809384 ff ff ss 85 1.300000
74.378610 ff ff ff 0 1.100000
68.387871 ff ff ff 0 1.300000
67.785019 ff ff ff 85 1.100000
70.519231 ff ff ff 85 1.300000
90.050494 sf ss ss 0 1.100000
55.864109 sf ss ss 0 1.300000
102.411736 sf ss ss 85 1.100000
65.411795 sf ss ss 85 1.300000
69.685120 sf ss ff 0 1.100000
61.086348 sf ss ff 0 1.300000
78.034949 sf ss ff 85 1.100000
56.482840 sf ss ff 85 1.300000
90.050494 sf ff ss 0 1.100000
55.864109 sf ff ss 0 1.300000
102.411736 sf ff ss 85 1.100000
65.411795 sf ff ss 85 1.300000
69.685120 sf ff ff 0 1.100000

61.086348 sf ff ff 0 1.300000
78.034949 sf ff ff 85 1.100000
56.482840 sf ff ff 85 1.300000
69.533508 fs ss ss 0 1.100000
70.202493 fs ss ss 0 1.300000
81.148252 fs ss ss 85 1.100000
63.982539 fs ss ss 85 1.300000
73.467615 fs ss ff 0 1.100000
66.405093 fs ss ff 0 1.300000
68.299269 fs ss ff 85 1.100000
68.555940 fs ss ff 85 1.300000
69.533508 fs ff ss 0 1.100000
70.202493 fs ff ss 0 1.300000
81.148252 fs ff ss 85 1.100000
63.982539 fs ff ss 85 1.300000
73.467615 fs ff ff 0 1.100000
66.405093 fs ff ff 0 1.300000
68.299269 fs ff ff 85 1.100000
68.555940 fs ff ff 85 1.300000

Ocean script for VCO corners

```
ocnWaveformTool( 'wavescan )  
simulator( 'spectre )  
design( "/home/kumite/simulation/LC/spectre/schematic/netlist/netlist")  
resultsDir( "/home/kumite/simulation/LC/spectre/schematic" )  
for(i 0 1  
when(i==0  
desVar( "vdd" 1.2 )  
desVar( "x" 0 )
```

```

desVar( "c" 0.4 )
desVar( "a" 0 )
desVar( "d" 0 )
desVar( "y" 0 )
envOption(
    'firstRun t
)
counter=0
tmp='(0 85)
corner_txt='("ss" "ff" "sf" "fs")
cap_res_txt='("ss" "ff")
Vss='(1.1 1.3)
fp=outfile("/home/kumite/Desktop/swingH3.xlsx" "w")
fprintf(fp "Cnt N/P Res cap Ind Var Temp Vdd P.N Sw freq\n")
close(fp)
foreach(corner corner_txt
foreach(res cap_res_txt
foreach(cap cap_res_txt
foreach(ind cap_res_txt
foreach(var cap_res_txt
modelFile(
    list("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/rf013.scs" sprintf( nil
"%s" corner ))
    list("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/rf013.scs"
sprintf( nil "%s_rfmos" corner ))
    list("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/ResModel.scs" sprintf( nil
"%s_disres" res ))
    list("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/rf013.scs" sprintf( nil
"%s_mim" cap ))
    list("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/rf013.scs"
sprintf( nil "%s_rfind" ind ))
    list("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/rf013.scs" sprintf( nil
"%s_mos_cap" var ))
)
foreach(temp tmp
temp( temp )
foreach(supply Vss

```

```

desVar( "Vdd" supply )

counter=counter+1

analysis('pnoise ?relharmnum "1" ?start "1K" ?stop "10M"
        ?maxsideband "15" ?p "/net146" ?n "/net103" ?oprobe ""
        ?iprobe "" ?refsideband "" )

analysis('pss ?fund "1.8e9" ?harms "3" ?errpreset "liberal"
        ?tstab "120n" ?p "/net146" ?n "/gnd!" )

run()

sprintf(label "corner(NP-%s_mim-%s_res-%s_temp-%n_supply-%n_count(%n))" corner cap res temp supply
counter)

mm=Phase\ Noise\; dBc\Hz\ Relative\ Harmonic\ \= 1 = value(phaseNoise(1 "pss_fd" ?result "pnoise")
1000000)

sw=v\ \(\vnet103\ \vnet146)\; pss\ \ (V) = ymax((v("/net103" ?result "pss_td") - v("/net146" ?result "pss_td")))

freq=harm\=1\ freq\; pss\ \ (Hz) = harmonic(xval(getData("/+" ?result "pss_fd")) '1)

fp=outfile("/home/kumite/Desktop/swingH3.xlsx" "a")

fprintf(fp "%n %s %s %s %s %s %n %n % 1.1f % 1.1f % 1.1f\n" counter corner cap res ind var temp supply mm sw*2
freq/100000000)

close(fp)

)

)

)

)

)

)

)

)

)

when(i==1

desVar( "vdd" 1.2 )

desVar( "x" 1.2 )

desVar( "c" 0.9 )

```

```

desVar( "a" 1.2 )
desVar( "d" 1.2 )
desVar( "y" 1.2 )
envOption(
    'firstRun t
)
counter=0
tmp=(0 85)
corner_txt=('ss" "ff" "sf" "fs")
cap_res_txt=('ss" "ff")
Vss=(1.1 1.3)
fp=outfile("/home/kumite/Desktop/swingL3.xlsx" "w")

fprintf(fp "Cnt N/P Res cap Ind Var Temp Vdd P.N Sw freq\n")
close(fp)
foreach(corner corner_txt
foreach(res cap_res_txt
foreach(cap cap_res_txt
foreach(ind cap_res_txt
foreach(var cap_res_txt
modelFile(    list("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/rf013.scs" sprintf( nil
"%s" corner ))
    list("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/rf013.scs" sprintf( nil
"%s_rfmos" corner ))
    list("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/ResModel.scs" sprintf( nil
"%s_disres" res ))
    list("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/rf013.scs" sprintf( nil
"%s_mim" cap ))

    list("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/rf013.scs" sprintf( nil
"%s_rfind" ind ))    list("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/rf013.scs"
sprintf( nil "%s_mos_cap" var ))

)
foreach(temp tmp
    temp( temp )

```



```

ocnWaveformTool( 'wavescan )

simulator( 'spectre )

design( "/home/ramy/simulation/segma1.41pbias/spectre/schematic/netlist/netlist")

resultsDir( "/home/ramy/simulation/segma1.41pbias/spectre/schematic" )

counter=0

tmp='(0 85)

corner_txt=('ss" "ff" "sf" "fs")

cap_res_txt=('ss" "ff")

Vss=(1.1 1.3)

desVar( "Vdd" 1.2      )

fp=outfile("/home/ramy/Desktop/DelayN.xlsx" "w")

fprintf(fp "Counter NMOSPMOS Resistance Mimcap Temp Vdd Delay(ps)\n")

close(fp)

foreach(corner corner_txt

foreach(res cap_res_txt

foreach(cap cap_res_txt

        modelFile(      list("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/rf013.scs"
sprintf( nil "%s" corner ))

                list("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/rf013.scs"
sprintf( nil "%s_rfmos" corner ))

                list("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/ResModel.scs" sprintf( nil
"%s_disres" res ))

                list("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/rf013.scs"
sprintf( nil "%s_mim" cap ))

                )

foreach(temp tmp

        temp( temp )

foreach(supply Vss

```



```

design( "/home/ramy/simulation/collect12plus2V5fixed/spectre/schematic/netlist/netlist")

resultsDir( "/home/ramy/simulation/collect12plus2V5fixed/spectre/schematic" )

counter=0

tmp='(0 85)

corner_txt=('ss" "ff" "sf" "fs")

cap_res_txt=('ss" "ff")

Vss=(1.1 1.3)

desVar( "Vdd" 1.2      )

foreach(corner corner_txt

foreach(res cap_res_txt

foreach(cap cap_res_txt

      modelFile(      list("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/rf013.scs"
sprintf( nil "%s" corner ))
      list("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/rf013.scs" sprintf( nil
"%s_rfmos" corner ))

      list("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/ResModel.scs" sprintf( nil
"%s_disres" res ))

      list("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/rf013.scs"
sprintf( nil "%s_mim" cap ))

      )

foreach(temp tmp

      temp( temp )

foreach(supply Vss

desVar( "Vdd" supply  )

counter=counter+1

analysis('tran ?stop "150n" ?errpreset "moderate" )

save( 'v "/outfinal" )

run()

      sprintf(label "corner(NP-%s_mim-%s_res-%s_temp-%n_supply-%n_count(%n))" corner cap res temp supply
counter)

```

```

text1=strcat("outfinal " label )

selectResult( 'tran )

plot(getData("/outfinal") ?expr list(text1)))

)

)

)

)

)

)

```

Ocean Results :

Results For Delay Function across Corners

IQ phase Mismatch across corners at Vco freq=1.83 GHZ ,Typical T/4 = 273.2 ps

Counter	NMOSPMOS	Resistance	Mimcap	Temp	Vdd	Delay(ps)
1	Ss	ss	ss	0	1.100000	273.2
2	Ss	ss	ss	0	1.300000	273.2
3	Ss	ss	ss	85	1.100000	273.2

Ocean script code for Power amplifier

Ocean code for corner simulation

```

ocnWaveformTool( 'wavescan )
simulator( 'spectre )
design( "/home/khaled/simulation/Programmable-PA/spectre/schematic/netlist/netlist")
resultsDir( "/home/khaled/simulation/Programmable-PA/spectre/schematic" )
desVar(      "frf" 915M  )
counter=0
tmp='(0 85)
corner_txt='("ss" "ff" "sf" "fs")
cap_res_txt='("ss" "ff")
Vss=(1.1 1.3)
Vss2=(2.4 2.6)
fp=outfile("/home/khaled/Desktop/PA.xlsx" "a")
fprintf(fp "counter MOS1v MOS2v cap res temp supply supply2 Pout 2nd-harmonic 3rd-harmonic
Iavg PAE\n")
close(fp)
foreach(corner corner_txt
foreach(corner25 corner_txt
foreach(res cap_res_txt
foreach(cap cap_res_txt

```

```

    modelFile(
        list("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/rf013.scs"
sprintf( nil "%s" corner ))

        list("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/rf013.scs"
sprintf( nil "%s_rfmos" corner ))

        list("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/rf013.scs"
sprintf( nil "%s_rfmos25" corner25 ))
        list("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/ResModel.sc
s" sprintf( nil "%s_disres" res ))
        list("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/rf013.scs"
sprintf( nil "%s_mim" cap ))
    )
foreach(temp tmp
    temp( temp )
foreach(supply Vss
    desVar( "Vdd" supply)
foreach(supply2 Vss2
    desVar( "Vdd2" supply2 )
counter=counter+1
save( 'i "/PORT3/PLUS" "/PORT2/PLUS" "/V3/PLUS" "/V2/PLUS" )
analysis('tran ?stop "100n" ?errpreset "conservative" )
analysis('pss ?fund "915M" ?harms "5" ?errpreset "conservative"
        ?tstab "100n" ?saveinit "no" ?maxstep "21.85e-12" )

run()
selectResult( 'tran )
I = average(getData("/V2/MINUS"))
Iavg = I*1e3
Spectrum = dbm(spectralPower(((v("/RFout" ?result "pss_fd") - 0.0) / resultParam("PORT3:r" ?result
"pss_fd")) (v("/RFout" ?result "pss_fd") - 0.0)))
PAE = (((100.0 * harmonic((let((vn (v("/RFout" ?result "pss_fd") - 0.0))) spectralPower((vn /
resultParam("PORT3:r" ?result "pss_fd")) vn)) + spectralPower(i("/PORT2/PLUS" ?result "pss_fd")
(v("/Vid" ?result "pss_fd") - 0.0))) '(1))) / -harmonic((spectralPower(i("/V3/PLUS" ?result "pss_fd")
(v("/Vx" ?result "pss_fd") - 0.0)) + spectralPower(i("/V2/PLUS" ?result "pss_fd") (v("/Vy" ?result
"pss_fd") - 0.0))) '(0)))
Pout = value(Spectrum = dbm(spectralPower(((v("/RFout" ?result "pss_fd") - 0.0) /
resultParam("PORT3:r" ?result "pss_fd")) (v("/RFout" ?result "pss_fd") - 0.0))) 915000000)
\2nd\ harmonic = value(Spectrum = dbm(spectralPower(((v("/RFout" ?result "pss_fd") - 0.0) /
resultParam("PORT3:r" ?result "pss_fd")) (v("/RFout" ?result "pss_fd") - 0.0))) 1.83e+09)
\3rd\ harmonic = value(Spectrum = dbm(spectralPower(((v("/RFout" ?result "pss_fd") - 0.0) /
resultParam("PORT3:r" ?result "pss_fd")) (v("/RFout" ?result "pss_fd") - 0.0))) 2.745e+09)

fp=outfile("/home/khaled/Desktop/PA.xlsx" "a")
fprintf(fp "%n %s %s %s %s %s %n %n %n % 1.1f % 1.1f % 1.1f % 1.2f % 1.2f \n" counter corner
corner25 cap res temp supply supply2 Pout \2nd\ harmonic \3rd\ harmonic Iavg PAE)
close(fp)
)
)
)

```

```
)  
)  
)  
)
```

Code for drawing output power VS frequency

```
ocnWaveformTool( 'wavescan )  
simulator( 'spectre )  
design( "/home/khaled/simulation/Programmable-PA-try/spectre/schematic/netlist/netlist")  
resultsDir( "/home/khaled/simulation/Programmable-PA-try/spectre/schematic" )  
modelFile(  
    ("/tools/pdks/tsmc13rf/tsmc13rf/./models/RF_12_25_FSG.scs" "")  
)  
desVar(      "Vdd2" 2.5  )  
desVar(      "Vdd"  1.2  )  
temp( 27 )  
frequeny=(50M 100M 150M 200M 250M 300M 350M 400M 450M 500M 550M 600M 650M  
700M 750M 800M 850M 900M 915M 925M 950M 1G 1.05G 1.1G 1.15G 1.2G 1.25G 1.3G 1.4G  
1.5G 1.6G 1.7G 1.8G 1.9G 2G 2.2G 2.4G 2.6G 2.8G 3G)  
fp=outfile("/home/khaled/Desktop/Pout.xlsx" "a")  
fprintf(fp "Freq PAE Pout\n")  
close(fp)  
foreach(Freq frequeny  
desVar(      "frf" Freq )  
analysis('pss ?fund "frf" ?harms "10" ?errpreset "conservative"  
        ?tstab "100n" ?saveinit "no" ?maxstep "21.85e-12" )  
save( 'i "/PORT3/PLUS" "/PORT2/PLUS" "/V3/PLUS" "/V2/MINUS" "/M1/D" )  
run()  
Power\ Vs\ Freq = dbm(harmonic(spectralPower(((v("/RFout" ?result "pss_fd") - 0.0) /  
resultParam("PORT3:r" ?result "pss_fd")) (v("/RFout" ?result "pss_fd") - 0.0)) '(1)))  
fp=outfile("/home/khaled/Desktop/Pout.xlsx" "a")  
fprintf(fp "%n %1.2f %1.2f \n" Freq PAE Power\ Vs\ Freq)  
close(fp)  
)
```

Ocean results

Counter	MOS1v	MOS2v	cap	res	temp	supply	supply2	Pout	2nd-harmonic	3rd-harmonic	Iavg	PAE
1	ss	ss	ss	ss	0	1.1	2.4	11	-37.3	-61.7	21.03	24.05
2	ss	ss	ss	ss	0	1.1	2.6	11.8	-36.8	-60.9	23.11	24.47
3	ss	ss	ss	ss	0	1.3	2.4	11	-37.1	-61.8	22	23.66
4	ss	ss	ss	ss	0	1.3	2.6	11.8	-36.5	-61	24.2	24.09
5	ss	ss	ss	ss	85	1.1	2.4	10.2	-38.8	-63.4	18.94	22.11
6	ss	ss	ss	ss	85	1.1	2.6	11	-38	-62.9	21.17	22.53
7	ss	ss	ss	ss	85	1.3	2.4	10.2	-38.5	-63.6	19.78	21.75
8	ss	ss	ss	ss	85	1.3	2.6	11.1	-37.7	-63	22.06	22.21
9	ss	ss	ff	ss	0	1.1	2.4	11	-37.3	-61.7	21.04	24.05
10	ss	ss	ff	ss	0	1.1	2.6	11.8	-36.8	-61	23.13	24.46
11	ss	ss	ff	ss	0	1.3	2.4	11	-37.1	-61.6	21.97	23.68
12	ss	ss	ff	ss	0	1.3	2.6	11.8	-36.5	-61.1	24.18	24.11
13	ss	ss	ff	ss	85	1.1	2.4	10.2	-38.7	-63.6	19	22.06
14	ss	ss	ff	ss	85	1.1	2.6	11.1	-38	-63	21.22	22.49
15	ss	ss	ff	ss	85	1.3	2.4	10.2	-38.5	-63.7	19.79	21.74
16	ss	ss	ff	ss	85	1.3	2.6	11.1	-37.7	-62.9	22.07	22.21
17	ss	ss	ss	ff	0	1.1	2.4	11	-37.4	-61.6	21.08	23.98
18	ss	ss	ss	ff	0	1.1	2.6	11.8	-37	-60.9	23.11	24.46
19	ss	ss	ss	ff	0	1.3	2.4	11	-37.1	-61.6	22.08	23.5
20	ss	ss	ss	ff	0	1.3	2.6	11.8	-36.7	-61	24.2	24.05
21	ss	ss	ss	ff	85	1.1	2.4	10.1	-38.9	-63.4	19.02	21.99
22	ss	ss	ss	ff	85	1.1	2.6	11	-38.3	-62.8	21.15	22.5
23	ss	ss	ss	ff	85	1.3	2.4	10.2	-38.5	-63.7	19.94	21.53
24	ss	ss	ss	ff	85	1.3	2.6	11.1	-37.8	-62.9	22.17	22.06
25	ss	ss	ff	ff	0	1.1	2.4	11	-37.4	-61.7	21.07	23.99
26	ss	ss	ff	ff	0	1.1	2.6	11.8	-37	-61	23.12	24.46
27	ss	ss	ff	ff	0	1.3	2.4	11	-37.1	-61.6	22.06	23.53
28	ss	ss	ff	ff	0	1.3	2.6	11.8	-36.7	-61.1	24.18	24.07
29	ss	ss	ff	ff	85	1.1	2.4	10.2	-38.8	-63.5	19.08	21.94
30	ss	ss	ff	ff	85	1.1	2.6	11	-38.2	-62.9	21.2	22.47
31	ss	ss	ff	ff	85	1.3	2.4	10.2	-38.5	-63.7	19.96	21.51
32	ss	ss	ff	ff	85	1.3	2.6	11.1	-37.8	-62.7	22.17	22.06
33	ss	ff	ss	ss	0	1.1	2.4	11.4	-35.4	-60.7	24.96	22.31
34	ss	ff	ss	ss	0	1.1	2.6	12.1	-35.4	-59.3	26.77	23.02
35	ss	ff	ss	ss	0	1.3	2.4	11.4	-35.3	-60.8	25.93	22.16
36	ss	ff	ss	ss	0	1.3	2.6	12.1	-34.7	-59.6	28.25	22.43
37	ss	ff	ss	ss	85	1.1	2.4	10.9	-36.2	-64.5	23.98	20.58
38	ss	ff	ss	ss	85	1.1	2.6	11.7	-36.1	-62.5	25.88	21.29
39	ss	ff	ss	ss	85	1.3	2.4	10.9	-36.3	-64.3	24.55	20.72
40	ss	ff	ss	ss	85	1.3	2.6	11.7	-35.8	-62.6	25.86	21.16
41	ss	ff	ff	ss	0	1.1	2.4	11.4	-35.4	-60.7	24.99	22.29
42	ss	ff	ff	ss	0	1.1	2.6	12.1	-35.4	-59.3	26.8	23
43	ss	ff	ff	ss	0	1.3	2.4	11.4	-35.3	-60.8	25.95	22.14
44	ss	ff	ff	ss	0	1.3	2.6	12.1	-34.7	-59.6	28.29	22.4
45	ss	ff	ff	ss	85	1.1	2.4	10.9	-36.2	-64.5	24.02	20.55
46	ss	ff	ff	ss	85	1.1	2.6	11.7	-36	-62.5	25.93	21.26
47	ss	ff	ff	ss	85	1.3	2.4	10.9	-36.4	-64.3	24.56	20.72
48	ss	ff	ff	ss	85	1.3	2.6	11.7	-35.8	-62.6	25.89	21.14
49	ss	ff	ss	ff	0	1.1	2.4	11.4	-35.6	-60.6	25.09	22.23
50	ss	ff	ss	ff	0	1.1	2.6	12.1	-35.6	-59.3	26.94	22.89
51	ss	ff	ss	ff	0	1.3	2.4	11.4	-35.2	-60.9	26.15	21.91

52	ss	ff	ss	ff	0	1.3	2.6	12.1	-34.7	-59.7	28.53	22.17
53	ss	ff	ss	ff	85	1.1	2.4	10.9	-36.4	-64.4	24.13	20.48
54	ss	ff	ss	ff	85	1.1	2.6	11.7	-36.3	-62.3	26	21.23
55	ss	ff	ss	ff	85	1.3	2.4	10.9	-36.4	-64.2	24.74	20.54
56	ss	ff	ss	ff	85	1.3	2.6	11.7	-35.9	-62.6	26.07	20.95
57	ss	ff	ff	ff	0	1.1	2.4	11.4	-35.6	-60.6	25.11	22.22
58	ss	ff	ff	ff	0	1.1	2.6	12.1	-35.6	-59.4	26.97	22.87
59	ss	ff	ff	ff	0	1.3	2.4	11.4	-35.2	-60.9	26.2	21.87
60	ss	ff	ff	ff	0	1.3	2.6	12.1	-34.7	-59.7	28.6	22.11
61	ss	ff	ff	ff	85	1.1	2.4	10.9	-36.4	-64.4	24.16	20.46
62	ss	ff	ff	ff	85	1.1	2.6	11.7	-36.3	-62.2	26.03	21.22
63	ss	ff	ff	ff	85	1.3	2.4	10.9	-36.4	-64.2	24.76	20.52
64	ss	ff	ff	ff	85	1.3	2.6	11.7	-35.9	-62.6	26.11	20.93
65	ss	sf	ss	ss	0	1.1	2.4	11.2	-37.5	-60.7	21.69	24.48
66	ss	sf	ss	ss	0	1.1	2.6	12	-37.1	-59.7	23.68	24.91
67	ss	sf	ss	ss	0	1.3	2.4	11.2	-37.2	-60.8	22.62	24.18
68	ss	sf	ss	ss	0	1.3	2.6	12	-36.6	-59.8	23.97	24.46
69	ss	sf	ss	ss	85	1.1	2.4	10.5	-38.6	-62.7	20.1	22.51
70	ss	sf	ss	ss	85	1.1	2.6	11.3	-38.2	-62	22.07	23.06
71	ss	sf	ss	ss	85	1.3	2.4	10.5	-38.4	-62.9	20.81	22.3
72	ss	sf	ss	ss	85	1.3	2.6	11.3	-37.8	-62.2	22.18	22.8
73	ss	sf	ff	ss	0	1.1	2.4	11.2	-37.5	-60.7	21.71	24.47
74	ss	sf	ff	ss	0	1.1	2.6	12	-37.1	-59.7	23.7	24.89
75	ss	sf	ff	ss	0	1.3	2.4	11.2	-37.2	-60.8	22.62	24.19
76	ss	sf	ff	ss	0	1.3	2.6	12	-36.6	-59.8	23.98	24.45
77	ss	sf	ff	ss	85	1.1	2.4	10.5	-38.6	-62.8	20.13	22.49
78	ss	sf	ff	ss	85	1.1	2.6	11.3	-38.2	-62.1	22.11	23.03
79	ss	sf	ff	ss	85	1.3	2.4	10.5	-38.4	-63	20.81	22.31
80	ss	sf	ff	ss	85	1.3	2.6	11.3	-37.9	-62.1	22.19	22.8
81	ss	sf	ss	ff	0	1.1	2.4	11.2	-37.6	-60.7	21.76	24.4
82	ss	sf	ss	ff	0	1.1	2.6	12	-37.4	-59.7	23.73	24.84
83	ss	sf	ss	ff	0	1.3	2.4	11.2	-37.3	-60.8	22.67	24.07
84	ss	sf	ss	ff	0	1.3	2.6	12	-36.7	-59.8	24.07	24.31
85	ss	sf	ss	ff	85	1.1	2.4	10.5	-38.8	-62.7	20.14	22.44
86	ss	sf	ss	ff	85	1.1	2.6	11.3	-38.6	-62	22.05	23.04
87	ss	sf	ss	ff	85	1.3	2.4	10.5	-38.5	-62.9	20.91	22.15
88	ss	sf	ss	ff	85	1.3	2.6	11.3	-38	-62.1	22.28	22.66
89	ss	sf	ff	ff	0	1.1	2.4	11.2	-37.6	-60.7	21.77	24.4
90	ss	sf	ff	ff	0	1.1	2.6	12	-37.4	-59.7	23.72	24.85
91	ss	sf	ff	ff	0	1.3	2.4	11.2	-37.4	-60.9	22.67	24.08
92	ss	sf	ff	ff	0	1.3	2.6	12	-36.7	-59.8	24.09	24.29
93	ss	sf	ff	ff	85	1.1	2.4	10.5	-38.8	-62.8	20.16	22.43
94	ss	sf	ff	ff	85	1.1	2.6	11.3	-38.6	-62	22.07	23.02
95	ss	sf	ff	ff	85	1.3	2.4	10.5	-38.5	-63	20.9	22.16
96	ss	sf	ff	ff	85	1.3	2.6	11.3	-38	-62.1	22.29	22.66
97	ss	fs	ss	ss	0	1.1	2.4	11.2	-35.3	-63.1	25.43	21.91
98	ss	fs	ss	ss	0	1.1	2.6	12	-35	-61.3	26.44	22.52
99	ss	fs	ss	ss	0	1.3	2.4	11.2	-35.3	-63.2	25.39	21.8
100	ss	fs	ss	ss	0	1.3	2.6	12	-34.9	-61.5	27.53	22.33
101	ss	fs	ss	ss	85	1.1	2.4	10.6	-36.3	-66.8	23.8	20.17
102	ss	fs	ss	ss	85	1.1	2.6	11.5	-35.9	-65.1	25.13	20.84
103	ss	fs	ss	ss	85	1.3	2.4	10.6	-36.4	-66.9	23.72	20.23

104	ss	fs	ss	ss	85	1.3	2.6	11.5	-35.9	-65.1	25.89	20.84
105	ss	fs	ff	ss	0	1.1	2.4	11.2	-35.3	-63.1	25.45	21.91
106	ss	fs	ff	ss	0	1.1	2.6	12	-35	-61.3	26.46	22.51
107	ss	fs	ff	ss	0	1.3	2.4	11.2	-35.3	-63.2	25.38	21.82
108	ss	fs	ff	ss	0	1.3	2.6	12	-34.9	-61.5	27.52	22.35
109	ss	fs	ff	ss	85	1.1	2.4	10.6	-36.3	-67	23.82	20.17
110	ss	fs	ff	ss	85	1.1	2.6	11.5	-35.9	-65.1	25.16	20.83
111	ss	fs	ff	ss	85	1.3	2.4	10.6	-36.5	-66.9	23.7	20.27
112	ss	fs	ff	ss	85	1.3	2.6	11.5	-36	-65.1	25.87	20.88
113	ss	fs	ss	ff	0	1.1	2.4	11.2	-35.5	-63	25.49	21.88
114	ss	fs	ss	ff	0	1.1	2.6	12	-35.2	-61.2	26.49	22.51
115	ss	fs	ss	ff	0	1.3	2.4	11.2	-35.4	-63.1	25.43	21.74
116	ss	fs	ss	ff	0	1.3	2.6	12	-34.9	-61.4	27.64	22.23
117	ss	fs	ss	ff	85	1.1	2.4	10.6	-36.3	-67.1	23.39	19.96
118	ss	fs	ss	ff	85	1.1	2.6	11.5	-36.1	-64.9	25.22	20.79
119	ss	fs	ss	ff	85	1.3	2.4	10.6	-36.4	-67.1	23.94	19.99
120	ss	fs	ss	ff	85	1.3	2.6	11.5	-36.1	-65	26	20.74
121	ss	fs	ff	ff	0	1.1	2.4	11.2	-35.5	-63.1	25.51	21.88
122	ss	fs	ff	ff	0	1.1	2.6	12	-35.2	-61.1	26.5	22.52
123	ss	fs	ff	ff	0	1.3	2.4	11.2	-35.4	-63.1	25.44	21.74
124	ss	fs	ff	ff	0	1.3	2.6	12	-34.9	-61.4	27.65	22.22
125	ss	fs	ff	ff	85	1.1	2.4	10.6	-36.3	-67.3	23.4	19.96
126	ss	fs	ff	ff	85	1.1	2.6	11.5	-36.1	-64.9	25.24	20.79
127	ss	fs	ff	ff	85	1.3	2.4	10.6	-36.4	-67	23.93	20.02
128	ss	fs	ff	ff	85	1.3	2.6	11.5	-36.1	-65	26	20.76
129	ff	ss	ss	ss	0	1.1	2.4	10.9	-38.8	-61.5	19.67	25.46
130	ff	ss	ss	ss	0	1.1	2.6	11.8	-38.2	-60.9	21.75	25.75
131	ff	ss	ss	ss	0	1.3	2.4	11	-38.4	-61.7	21.03	24.79
132	ff	ss	ss	ss	0	1.3	2.6	11.8	-37.6	-60.9	23.23	25.05
133	ff	ss	ss	ss	85	1.1	2.4	10.1	-39.8	-63.5	17.92	22.93
134	ff	ss	ss	ss	85	1.1	2.6	11	-39.2	-62.9	19.9	23.5
135	ff	ss	ss	ss	85	1.3	2.4	10.1	-39.5	-63.5	18.91	22.42
136	ff	ss	ss	ss	85	1.3	2.6	11	-38.8	-63	21.06	22.99
137	ff	ss	ff	ss	0	1.1	2.4	10.9	-38.8	-61.5	19.66	25.47
138	ff	ss	ff	ss	0	1.1	2.6	11.8	-38.2	-60.9	21.77	25.74
139	ff	ss	ff	ss	0	1.3	2.4	11	-38.4	-61.7	21.02	24.81
140	ff	ss	ff	ss	0	1.3	2.6	11.8	-37.6	-60.9	23.22	25.05
141	ff	ss	ff	ss	85	1.1	2.4	10.1	-39.8	-63.6	17.93	22.93
142	ff	ss	ff	ss	85	1.1	2.6	11	-39.2	-63	19.92	23.49
143	ff	ss	ff	ss	85	1.3	2.4	10.1	-39.5	-63.6	18.9	22.43
144	ff	ss	ff	ss	85	1.3	2.6	11	-38.8	-63	21.04	23
145	ff	ss	ss	ff	0	1.1	2.4	10.9	-38.9	-61.6	19.66	25.4
146	ff	ss	ss	ff	0	1.1	2.6	11.8	-38.3	-61	21.8	25.64
147	ff	ss	ss	ff	0	1.3	2.4	11	-38.5	-61.7	21.17	24.7
148	ff	ss	ss	ff	0	1.3	2.6	11.8	-37.7	-60.9	23.25	24.94
149	ff	ss	ss	ff	85	1.1	2.4	10.1	-39.9	-63.6	17.93	22.83
150	ff	ss	ss	ff	85	1.1	2.6	11	-39.4	-62.9	19.87	23.41
151	ff	ss	ss	ff	85	1.3	2.4	10.1	-39.5	-63.5	18.99	22.25
152	ff	ss	ss	ff	85	1.3	2.6	11	-38.9	-63	21.07	22.87
153	ff	ss	ff	ff	0	1.1	2.4	10.9	-39	-61.6	19.65	25.41
154	ff	ss	ff	ff	0	1.1	2.6	11.8	-38.3	-61	21.81	25.63
155	ff	ss	ff	ff	0	1.3	2.4	11	-38.5	-61.7	21.26	24.71

156	ff	ss	ff	ff	0	1.3	2.6	11.8	-37.7	-61	23.26	24.93
157	ff	ss	ff	ff	85	1.1	2.4	10.1	-39.9	-63.6	17.94	22.83
158	ff	ss	ff	ff	85	1.1	2.6	11	-39.4	-62.9	19.88	23.4
159	ff	ss	ff	ff	85	1.3	2.4	10.1	-39.5	-63.6	18.99	22.26
160	ff	ss	ff	ff	85	1.3	2.6	11	-38.9	-63	21.06	22.88
161	ff	ff	ss	ss	0	1.1	2.4	11.4	-36.3	-60.2	23.99	23.25
162	ff	ff	ss	ss	0	1.1	2.6	12.2	-35.9	-59.2	26.16	23.48
163	ff	ff	ss	ss	0	1.3	2.4	11.4	-35.7	-60.5	25.59	22.51
164	ff	ff	ss	ss	0	1.3	2.6	12.1	-35	-59.4	27.98	22.62
165	ff	ff	ss	ss	85	1.1	2.4	10.9	-37.7	-63.1	22.5	22.05
166	ff	ff	ss	ss	85	1.1	2.6	11.7	-37.3	-61.8	24.59	22.41
167	ff	ff	ss	ss	85	1.3	2.4	10.9	-37.2	-63.5	23.74	21.47
168	ff	ff	ss	ss	85	1.3	2.6	11.7	-36.5	-62.2	25.21	21.7
169	ff	ff	ff	ss	0	1.1	2.4	11.4	-36.3	-60.2	24.02	23.23
170	ff	ff	ff	ss	0	1.1	2.6	12.2	-35.8	-59.2	26.21	23.45
171	ff	ff	ff	ss	0	1.3	2.4	11.4	-35.7	-60.5	25.61	22.49
172	ff	ff	ff	ss	0	1.3	2.6	12.1	-35	-59.5	27.99	22.6
173	ff	ff	ff	ss	85	1.1	2.4	10.9	-37.7	-63.1	22.53	22.03
174	ff	ff	ff	ss	85	1.1	2.6	11.7	-37.3	-61.8	24.63	22.38
175	ff	ff	ff	ss	85	1.3	2.4	10.9	-37.2	-63.5	23.74	21.47
176	ff	ff	ff	ss	85	1.3	2.6	11.7	-36.5	-62.1	25.23	21.69
177	ff	ff	ss	ff	0	1.1	2.4	11.4	-36.3	-60.3	24.17	23.06
178	ff	ff	ss	ff	0	1.1	2.6	12.2	-35.9	-59.3	26.4	23.26
179	ff	ff	ss	ff	0	1.3	2.4	11.4	-35.6	-60.6	25.8	22.28
180	ff	ff	ss	ff	0	1.3	2.6	12.1	-35	-59.6	28.2	22.38
181	ff	ff	ss	ff	85	1.1	2.4	10.9	-37.9	-63.1	22.68	21.87
182	ff	ff	ss	ff	85	1.1	2.6	11.7	-37.5	-61.9	24.81	22.21
183	ff	ff	ss	ff	85	1.3	2.4	10.9	-37.3	-63.4	23.9	21.28
184	ff	ff	ss	ff	85	1.3	2.6	11.7	-36.5	-62.1	25.4	21.47
185	ff	ff	ff	ff	0	1.1	2.4	11.4	-36.3	-60.3	24.22	23.03
186	ff	ff	ff	ff	0	1.1	2.6	12.2	-35.9	-59.3	26.46	23.21
187	ff	ff	ff	ff	0	1.3	2.4	11.4	-35.6	-60.6	25.87	22.23
188	ff	ff	ff	ff	0	1.3	2.6	12.1	-34.9	-59.6	28.25	22.34
189	ff	ff	ff	ff	85	1.1	2.4	10.9	-37.9	-63.1	22.71	21.84
190	ff	ff	ff	ff	85	1.1	2.6	11.7	-37.5	-62	24.84	22.18
191	ff	ff	ff	ff	85	1.3	2.4	10.9	-37.3	-63.4	23.92	21.25
192	ff	ff	ff	ff	85	1.3	2.6	11.7	-36.5	-62.1	25.43	21.44
193	ff	sf	ss	ss	0	1.1	2.4	11.2	-38.8	-60.5	20.53	25.69
194	ff	sf	ss	ss	0	1.1	2.6	11.9	-38.2	-59.6	22.71	25.77
195	ff	sf	ss	ss	0	1.3	2.4	11.2	-38.1	-60.6	21.93	24.91
196	ff	sf	ss	ss	0	1.3	2.6	12	-37.2	-59.7	23.48	24.86
197	ff	sf	ss	ss	85	1.1	2.4	10.4	-40	-62.7	18.8	23.66
198	ff	sf	ss	ss	85	1.1	2.6	11.3	-39.5	-62.1	20.78	24.09
199	ff	sf	ss	ss	85	1.3	2.4	10.5	-39.5	-62.7	19.89	23.08
200	ff	sf	ss	ss	85	1.3	2.6	11.3	-38.8	-62.1	21.37	23.44
201	ff	sf	ff	ss	0	1.1	2.4	11.2	-38.8	-60.5	20.55	25.66
202	ff	sf	ff	ss	0	1.1	2.6	11.9	-38.1	-59.7	22.75	25.73
203	ff	sf	ff	ss	0	1.3	2.4	11.2	-38.1	-60.6	21.93	24.9
204	ff	sf	ff	ss	0	1.3	2.6	12	-37.2	-59.7	23.49	24.84
205	ff	sf	ff	ss	85	1.1	2.4	10.4	-40	-62.7	18.82	23.65
206	ff	sf	ff	ss	85	1.1	2.6	11.3	-39.5	-62	20.81	24.06
207	ff	sf	ff	ss	85	1.3	2.4	10.5	-39.5	-62.7	19.89	23.09

208	ff	sf	ff	ss	85	1.3	2.6	11.3	-38.8	-62.1	21.38	23.43
209	ff	sf	ss	ff	0	1.1	2.4	11.2	-38.9	-60.5	20.63	25.52
210	ff	sf	ss	ff	0	1.1	2.6	11.9	-38.3	-59.7	22.83	25.61
211	ff	sf	ss	ff	0	1.3	2.4	11.2	-38.2	-60.6	22	24.75
212	ff	sf	ss	ff	0	1.3	2.6	12	-37.2	-59.7	23.58	24.67
213	ff	sf	ss	ff	85	1.1	2.4	10.4	-40.2	-62.8	18.86	23.51
214	ff	sf	ss	ff	85	1.1	2.6	11.3	-39.7	-62.1	20.87	23.89
215	ff	sf	ss	ff	85	1.3	2.4	10.5	-39.6	-62.7	19.95	22.91
216	ff	sf	ss	ff	85	1.3	2.6	11.3	-38.9	-62.1	21.48	23.24
217	ff	sf	ff	ff	0	1.1	2.4	11.2	-38.8	-60.4	20.65	25.49
218	ff	sf	ff	ff	0	1.1	2.6	11.9	-38.2	-59.8	22.87	25.57
219	ff	sf	ff	ff	0	1.3	2.4	11.2	-38.2	-60.6	22.02	24.72
220	ff	sf	ff	ff	0	1.3	2.6	12	-37.2	-59.7	23.61	24.65
221	ff	sf	ff	ff	85	1.1	2.4	10.4	-40.2	-62.8	18.87	23.5
222	ff	sf	ff	ff	85	1.1	2.6	11.3	-39.7	-62.1	20.9	23.87
223	ff	sf	ff	ff	85	1.3	2.4	10.5	-39.6	-62.7	19.95	22.91
224	ff	sf	ff	ff	85	1.3	2.6	11.3	-38.9	-62.1	21.5	23.23
225	ff	fs	ss	ss	0	1.1	2.4	11.3	-36.7	-61.9	24.01	23.4
226	ff	fs	ss	ss	0	1.1	2.6	12	-36.1	-60.8	25.24	23.64
227	ff	fs	ss	ss	0	1.3	2.4	11.2	-36.2	-62.2	24.82	22.72
228	ff	fs	ss	ss	0	1.3	2.6	12	-35.5	-61	26.98	22.93
229	ff	fs	ss	ss	85	1.1	2.4	10.7	-37.9	-65	22.18	21.8
230	ff	fs	ss	ss	85	1.1	2.6	11.5	-37.4	-63.7	23.59	22.25
231	ff	fs	ss	ss	85	1.3	2.4	10.7	-37.6	-65.3	22.63	21.34
232	ff	fs	ss	ss	85	1.3	2.6	11.5	-36.9	-63.9	24.91	21.72
233	ff	fs	ff	ss	0	1.1	2.4	11.3	-36.7	-61.9	24.02	23.39
234	ff	fs	ff	ss	0	1.1	2.6	12	-36.1	-60.7	25.26	23.63
235	ff	fs	ff	ss	0	1.3	2.4	11.2	-36.2	-62.3	24.96	22.72
236	ff	fs	ff	ss	0	1.3	2.6	12	-35.5	-61.1	26.99	22.91
237	ff	fs	ff	ss	85	1.1	2.4	10.7	-37.9	-64.9	22.18	21.81
238	ff	fs	ff	ss	85	1.1	2.6	11.5	-37.4	-63.8	23.6	22.25
239	ff	fs	ff	ss	85	1.3	2.4	10.7	-37.6	-65.2	22.6	21.37
240	ff	fs	ff	ss	85	1.3	2.6	11.5	-36.9	-64	24.89	21.74
241	ff	fs	ss	ff	0	1.1	2.4	11.3	-36.8	-61.9	24.07	23.31
242	ff	fs	ss	ff	0	1.1	2.6	12	-36.2	-60.7	25.36	23.51
243	ff	fs	ss	ff	0	1.3	2.4	11.2	-36.3	-62.2	24.99	22.64
244	ff	fs	ss	ff	0	1.3	2.6	12	-35.5	-61.1	27.14	22.77
245	ff	fs	ss	ff	85	1.1	2.4	10.7	-38.1	-64.9	21.64	21.72
246	ff	fs	ss	ff	85	1.1	2.6	11.5	-37.6	-63.8	23.67	22.16
247	ff	fs	ss	ff	85	1.3	2.4	10.7	-37.7	-65.2	22.69	21.22
248	ff	fs	ss	ff	85	1.3	2.6	11.5	-37.1	-63.9	24.98	21.62
249	ff	fs	ff	ff	0	1.1	2.4	11.3	-36.8	-61.9	24.09	23.3
250	ff	fs	ff	ff	0	1.1	2.6	12	-36.2	-60.6	25.39	23.49
251	ff	fs	ff	ff	0	1.3	2.4	11.2	-36.3	-62.2	25.01	22.61
252	ff	fs	ff	ff	0	1.3	2.6	12	-35.5	-61.1	27.2	22.73
253	ff	fs	ff	ff	85	1.1	2.4	10.7	-38.1	-64.9	21.63	21.73
254	ff	fs	ff	ff	85	1.1	2.6	11.5	-37.6	-63.8	23.69	22.15
255	ff	fs	ff	ff	85	1.3	2.4	10.7	-37.7	-65.1	22.68	21.24
256	ff	fs	ff	ff	85	1.3	2.6	11.5	-37.1	-63.9	24.99	21.62
257	sf	ss	ss	ss	0	1.1	2.4	11	-37.8	-61.6	20.62	24.5
258	sf	ss	ss	ss	0	1.1	2.6	11.8	-37.2	-60.9	22.7	24.87
259	sf	ss	ss	ss	0	1.3	2.4	11	-37.2	-61.8	22.15	23.74

260	sf	ss	ss	ss	0	1.3	2.6	11.8	-36.6	-61.1	24.23	24.14
261	sf	ss	ss	ss	85	1.1	2.4	10.1	-39.1	-63.4	18.58	22.42
262	sf	ss	ss	ss	85	1.1	2.6	11	-38.4	-62.7	20.72	22.9
263	sf	ss	ss	ss	85	1.3	2.4	10.2	-38.7	-63.6	19.6	21.87
264	sf	ss	ss	ss	85	1.3	2.6	11.1	-37.9	-62.9	21.87	22.35
265	sf	ss	ff	ss	0	1.1	2.4	11	-37.8	-61.6	20.62	24.5
266	sf	ss	ff	ss	0	1.1	2.6	11.8	-37.2	-60.9	22.72	24.87
267	sf	ss	ff	ss	0	1.3	2.4	11	-37.3	-61.9	22.25	23.77
268	sf	ss	ff	ss	0	1.3	2.6	11.8	-36.6	-61.1	24.21	24.16
269	sf	ss	ff	ss	85	1.1	2.4	10.1	-39.1	-63.5	18.63	22.39
270	sf	ss	ff	ss	85	1.1	2.6	11	-38.4	-62.8	20.76	22.87
271	sf	ss	ff	ss	85	1.3	2.4	10.2	-38.7	-63.7	19.61	21.87
272	sf	ss	ff	ss	85	1.3	2.6	11.1	-37.9	-63	21.87	22.36
273	sf	ss	ss	ff	0	1.1	2.4	11	-37.8	-61.6	20.65	24.42
274	sf	ss	ss	ff	0	1.1	2.6	11.8	-37.4	-60.9	22.73	24.81
275	sf	ss	ss	ff	0	1.3	2.4	11	-37.2	-61.8	22.41	23.58
276	sf	ss	ss	ff	0	1.3	2.6	11.8	-36.7	-61.1	24.23	24.08
277	sf	ss	ss	ff	85	1.1	2.4	10.1	-39.2	-63.4	18.66	22.29
278	sf	ss	ss	ff	85	1.1	2.6	11	-38.6	-62.7	20.72	22.83
279	sf	ss	ss	ff	85	1.3	2.4	10.2	-38.7	-63.7	19.76	21.64
280	sf	ss	ss	ff	85	1.3	2.6	11	-38	-63	21.96	22.2
281	sf	ss	ff	ff	0	1.1	2.4	11	-37.9	-61.7	20.65	24.44
282	sf	ss	ff	ff	0	1.1	2.6	11.8	-37.4	-61	22.74	24.81
283	sf	ss	ff	ff	0	1.3	2.4	11	-37.3	-61.9	22.38	23.62
284	sf	ss	ff	ff	0	1.3	2.6	11.8	-36.8	-61.1	24.21	24.11
285	sf	ss	ff	ff	85	1.1	2.4	10.1	-39.2	-63.5	18.71	22.26
286	sf	ss	ff	ff	85	1.1	2.6	11	-38.6	-62.9	20.75	22.81
287	sf	ss	ff	ff	85	1.3	2.4	10.2	-38.7	-63.7	19.77	21.63
288	sf	ss	ff	ff	85	1.3	2.6	11	-38	-63.1	21.96	22.21
289	sf	ff	ss	ss	0	1.1	2.4	11.4	-35.6	-60.6	24.68	22.58
290	sf	ff	ss	ss	0	1.1	2.6	12.1	-35.3	-59.4	26.79	22.95
291	sf	ff	ss	ss	0	1.3	2.4	11.4	-35.2	-61	26.22	21.99
292	sf	ff	ss	ss	0	1.3	2.6	12.1	-34.5	-59.8	28.61	22.12
293	sf	ff	ss	ss	85	1.1	2.4	10.9	-36.8	-63.7	23.38	21.19
294	sf	ff	ss	ss	85	1.1	2.6	11.7	-36.5	-62.1	25.41	21.72
295	sf	ff	ss	ss	85	1.3	2.4	10.9	-36.5	-64.1	24.42	20.8
296	sf	ff	ss	ss	85	1.3	2.6	11.7	-35.8	-62.7	25.86	21.14
297	sf	ff	ff	ss	0	1.1	2.4	11.4	-35.6	-60.6	24.73	22.54
298	sf	ff	ff	ss	0	1.1	2.6	12.1	-35.2	-59.4	26.85	22.9
299	sf	ff	ff	ss	0	1.3	2.4	11.4	-35.1	-61	26.26	21.96
300	sf	ff	ff	ss	0	1.3	2.6	12.1	-34.4	-59.9	28.66	22.07
301	sf	ff	ff	ss	85	1.1	2.4	10.9	-36.8	-63.7	23.42	21.15
302	sf	ff	ff	ss	85	1.1	2.6	11.7	-36.5	-62.2	25.47	21.67
303	sf	ff	ff	ss	85	1.3	2.4	10.9	-36.5	-64	24.44	20.8
304	sf	ff	ff	ss	85	1.3	2.6	11.7	-35.8	-62.7	25.89	21.11
305	sf	ff	ss	ff	0	1.1	2.4	11.4	-35.7	-60.7	24.9	22.37
306	sf	ff	ss	ff	0	1.1	2.6	12.1	-35.3	-59.4	27.05	22.72
307	sf	ff	ss	ff	0	1.3	2.4	11.4	-35.1	-61.1	26.51	21.74
308	sf	ff	ss	ff	0	1.3	2.6	12.1	-34.4	-60	28.89	21.84
309	sf	ff	ss	ff	85	1.1	2.4	10.9	-36.9	-63.6	23.57	21.03
310	sf	ff	ss	ff	85	1.1	2.6	11.7	-36.7	-62.2	25.62	21.54
311	sf	ff	ss	ff	85	1.3	2.4	10.9	-36.5	-64	24.61	20.62

312	sf	ff	ss	ff	85	1.3	2.6	11.7	-35.9	-62.7	26.08	20.91
313	sf	ff	ff	ff	0	1.1	2.4	11.4	-35.6	-60.7	24.96	22.33
314	sf	ff	ff	ff	0	1.1	2.6	12.1	-35.3	-59.4	27.12	22.67
315	sf	ff	ff	ff	0	1.3	2.4	11.4	-35.1	-61.2	26.64	21.68
316	sf	ff	ff	ff	0	1.3	2.6	12.1	-34.4	-60	28.97	21.77
317	sf	ff	ff	ff	85	1.1	2.4	10.9	-36.9	-63.6	23.61	20.99
318	sf	ff	ff	ff	85	1.1	2.6	11.7	-36.6	-62.3	25.68	21.5
319	sf	ff	ff	ff	85	1.3	2.4	10.9	-36.5	-64	24.64	20.59
320	sf	ff	ff	ff	85	1.3	2.6	11.7	-35.9	-62.8	26.13	20.87
321	sf	sf	ss	ss	0	1.1	2.4	11.2	-37.9	-60.6	21.29	24.92
322	sf	sf	ss	ss	0	1.1	2.6	12	-37.4	-59.7	23.39	25.17
323	sf	sf	ss	ss	0	1.3	2.4	11.2	-37.3	-60.9	22.68	24.19
324	sf	sf	ss	ss	0	1.3	2.6	12	-36.6	-59.8	24.07	24.36
325	sf	sf	ss	ss	85	1.1	2.4	10.5	-39.1	-62.7	19.61	22.97
326	sf	sf	ss	ss	85	1.1	2.6	11.3	-38.7	-61.9	21.58	23.48
327	sf	sf	ss	ss	85	1.3	2.4	10.5	-38.6	-63	20.63	22.44
328	sf	sf	ss	ss	85	1.3	2.6	11.3	-38	-62.1	22.05	22.89
329	sf	sf	ff	ss	0	1.1	2.4	11.2	-37.9	-60.5	21.31	24.9
330	sf	sf	ff	ss	0	1.1	2.6	12	-37.4	-59.7	23.44	25.13
331	sf	sf	ff	ss	0	1.3	2.4	11.2	-37.3	-60.9	22.67	24.2
332	sf	sf	ff	ss	0	1.3	2.6	12	-36.5	-59.8	24.08	24.33
333	sf	sf	ff	ss	85	1.1	2.4	10.5	-39.1	-62.8	19.64	22.96
334	sf	sf	ff	ss	85	1.1	2.6	11.3	-38.7	-61.9	21.62	23.45
335	sf	sf	ff	ss	85	1.3	2.4	10.5	-38.6	-62.9	20.63	22.45
336	sf	sf	ff	ss	85	1.3	2.6	11.3	-38	-62.1	22.06	22.88
337	sf	sf	ss	ff	0	1.1	2.4	11.2	-38	-60.5	21.4	24.76
338	sf	sf	ss	ff	0	1.1	2.6	12	-37.5	-59.7	23.54	25
339	sf	sf	ss	ff	0	1.3	2.4	11.2	-37.4	-60.8	22.73	24.07
340	sf	sf	ss	ff	0	1.3	2.6	12	-36.6	-59.8	24.18	24.18
341	sf	sf	ss	ff	85	1.1	2.4	10.5	-39.3	-62.8	19.68	22.85
342	sf	sf	ss	ff	85	1.1	2.6	11.3	-38.9	-61.9	21.64	23.35
343	sf	sf	ss	ff	85	1.3	2.4	10.5	-38.7	-62.9	20.72	22.28
344	sf	sf	ss	ff	85	1.3	2.6	11.3	-38.1	-62.2	22.17	22.72
345	sf	sf	ff	ff	0	1.1	2.4	11.2	-38	-60.6	21.42	24.75
346	sf	sf	ff	ff	0	1.1	2.6	12	-37.5	-59.7	23.57	24.97
347	sf	sf	ff	ff	0	1.3	2.4	11.2	-37.4	-60.8	22.74	24.06
348	sf	sf	ff	ff	0	1.3	2.6	12	-36.6	-59.8	24.21	24.14
349	sf	sf	ff	ff	85	1.1	2.4	10.5	-39.3	-62.8	19.69	22.84
350	sf	sf	ff	ff	85	1.1	2.6	11.3	-38.9	-61.9	21.67	23.33
351	sf	sf	ff	ff	85	1.3	2.4	10.5	-38.7	-62.9	20.72	22.29
352	sf	sf	ff	ff	85	1.3	2.6	11.3	-38.1	-62.1	22.18	22.71
353	sf	fs	ss	ss	0	1.1	2.4	11.2	-35.7	-62.7	24.97	22.38
354	sf	fs	ss	ss	0	1.1	2.6	12	-35.3	-61	26.07	22.86
355	sf	fs	ss	ss	0	1.3	2.4	11.2	-35.3	-63.2	25.88	21.81
356	sf	fs	ss	ss	0	1.3	2.6	12	-34.8	-61.5	27.78	22.26
357	sf	fs	ss	ss	85	1.1	2.4	10.6	-36.9	-66.2	23.18	20.81
358	sf	fs	ss	ss	85	1.1	2.6	11.5	-36.4	-64.7	24.55	21.39
359	sf	fs	ss	ss	85	1.3	2.4	10.6	-36.6	-66.7	23.53	20.4
360	sf	fs	ss	ss	85	1.3	2.6	11.5	-36.1	-65	25.73	20.97
361	sf	fs	ff	ss	0	1.1	2.4	11.2	-35.7	-62.7	24.98	22.38
362	sf	fs	ff	ss	0	1.1	2.6	12	-35.3	-61	26.1	22.84
363	sf	fs	ff	ss	0	1.3	2.4	11.2	-35.4	-63.2	25.85	21.83

364	sf	fs	ff	ss	0	1.3	2.6	12	-34.8	-61.4	27.79	22.26
365	sf	fs	ff	ss	85	1.1	2.4	10.6	-36.9	-66.2	23.19	20.81
366	sf	fs	ff	ss	85	1.1	2.6	11.5	-36.5	-64.5	24.57	21.39
367	sf	fs	ff	ss	85	1.3	2.4	10.6	-36.6	-66.6	23.5	20.44
368	sf	fs	ff	ss	85	1.3	2.6	11.5	-36.1	-64.9	25.71	21
369	sf	fs	ss	ff	0	1.1	2.4	11.2	-35.9	-62.6	25.03	22.34
370	sf	fs	ss	ff	0	1.1	2.6	12	-35.4	-61	26.21	22.74
371	sf	fs	ss	ff	0	1.3	2.4	11.2	-35.4	-63.1	25.89	21.76
372	sf	fs	ss	ff	0	1.3	2.6	12	-34.9	-61.4	27.98	22.13
373	sf	fs	ss	ff	85	1.1	2.4	10.6	-36.9	-66.3	22.73	20.65
374	sf	fs	ss	ff	85	1.1	2.6	11.5	-36.6	-64.3	24.63	21.33
375	sf	fs	ss	ff	85	1.3	2.4	10.6	-36.6	-66.8	23.71	20.19
376	sf	fs	ss	ff	85	1.3	2.6	11.5	-36.2	-64.8	25.82	20.88
377	sf	fs	ff	ff	0	1.1	2.4	11.2	-35.8	-62.5	25.05	22.32
378	sf	fs	ff	ff	0	1.1	2.6	12	-35.4	-61	26.25	22.72
379	sf	fs	ff	ff	0	1.3	2.4	11.2	-35.5	-63	25.88	21.76
380	sf	fs	ff	ff	0	1.3	2.6	12	-34.9	-61.4	28.24	22.1
381	sf	fs	ff	ff	85	1.1	2.4	10.6	-36.9	-66.3	22.73	20.66
382	sf	fs	ff	ff	85	1.1	2.6	11.5	-36.6	-64.3	24.65	21.32
383	sf	fs	ff	ff	85	1.3	2.4	10.6	-36.6	-66.7	23.7	20.22
384	sf	fs	ff	ff	85	1.3	2.6	11.5	-36.2	-64.8	25.82	20.89
385	fs	ss	ss	ss	0	1.1	2.4	10.9	-38.5	-61.6	19.91	25.26
386	fs	ss	ss	ss	0	1.1	2.6	11.8	-38	-60.8	21.92	25.63
387	fs	ss	ss	ss	0	1.3	2.4	11	-38.3	-61.7	20.92	24.84
388	fs	ss	ss	ss	0	1.3	2.6	11.8	-37.7	-60.8	23.12	25.18
389	fs	ss	ss	ss	85	1.1	2.4	10.1	-39.5	-63.4	18.24	22.72
390	fs	ss	ss	ss	85	1.1	2.6	11	-38.9	-63	20.25	23.27
391	fs	ss	ss	ss	85	1.3	2.4	10.1	-39.3	-63.2	19	22.39
392	fs	ss	ss	ss	85	1.3	2.6	11	-38.6	-63	21.14	22.96
393	fs	ss	ff	ss	0	1.1	2.4	10.9	-38.6	-61.6	19.9	25.28
394	fs	ss	ff	ss	0	1.1	2.6	11.8	-38	-60.8	21.92	25.63
395	fs	ss	ff	ss	0	1.3	2.4	11	-38.3	-61.6	20.89	24.86
396	fs	ss	ff	ss	0	1.3	2.6	11.8	-37.7	-60.8	23.1	25.19
397	fs	ss	ff	ss	85	1.1	2.4	10.1	-39.5	-63.3	18.26	22.7
398	fs	ss	ff	ss	85	1.1	2.6	11	-38.9	-63	20.27	23.27
399	fs	ss	ff	ss	85	1.3	2.4	10.1	-39.3	-63.2	18.99	22.4
400	fs	ss	ff	ss	85	1.3	2.6	11	-38.7	-63	21.13	22.97
401	fs	ss	ss	ff	0	1.1	2.4	10.9	-38.7	-61.6	19.88	25.23
402	fs	ss	ss	ff	0	1.1	2.6	11.8	-38.2	-60.9	21.92	25.56
403	fs	ss	ss	ff	0	1.3	2.4	11	-38.4	-61.6	20.91	24.76
404	fs	ss	ss	ff	0	1.3	2.6	11.8	-37.8	-60.8	23.13	25.07
405	fs	ss	ss	ff	85	1.1	2.4	10.1	-39.6	-63.4	18.26	22.62
406	fs	ss	ss	ff	85	1.1	2.6	11	-39.1	-63	20.2	23.22
407	fs	ss	ss	ff	85	1.3	2.4	10.1	-39.3	-63.2	19.09	22.22
408	fs	ss	ss	ff	85	1.3	2.6	11	-38.8	-63	21.16	22.85
409	fs	ss	ff	ff	0	1.1	2.4	10.9	-38.7	-61.7	19.87	25.24
410	fs	ss	ff	ff	0	1.1	2.6	11.8	-38.2	-60.9	21.91	25.58
411	fs	ss	ff	ff	0	1.3	2.4	11	-38.5	-61.7	20.89	24.78
412	fs	ss	ff	ff	0	1.3	2.6	11.8	-37.8	-60.8	23.12	25.08
413	fs	ss	ff	ff	85	1.1	2.4	10.1	-39.6	-63.3	18.28	22.61
414	fs	ss	ff	ff	85	1.1	2.6	11	-39.1	-63.1	20.21	23.22
415	fs	ss	ff	ff	85	1.3	2.4	10.1	-39.4	-63.3	19.09	22.23

416	fs	ss	ff	ff	85	1.3	2.6	11	-38.8	-63	21.15	22.86
417	fs	ff	ss	ss	0	1.1	2.4	11.4	-36.5	-60.2	23.84	23.47
418	fs	ff	ss	ss	0	1.1	2.6	12.2	-36.2	-59.1	25.9	23.76
419	fs	ff	ss	ss	0	1.3	2.4	11.4	-35.9	-60.5	25.28	22.82
420	fs	ff	ss	ss	0	1.3	2.6	12.1	-35.3	-59.2	27.65	22.96
421	fs	ff	ss	ss	85	1.1	2.4	10.9	-37.5	-63.3	22.75	21.84
422	fs	ff	ss	ss	85	1.1	2.6	11.7	-37.3	-62	24.7	22.35
423	fs	ff	ss	ss	85	1.3	2.4	10.9	-37.2	-63.4	23.68	21.56
424	fs	ff	ss	ss	85	1.3	2.6	11.7	-36.6	-62.2	25.08	21.86
425	fs	ff	ff	ss	0	1.1	2.4	11.4	-36.5	-60.2	23.87	23.45
426	fs	ff	ff	ss	0	1.1	2.6	12.2	-36.2	-59.1	25.95	23.73
427	fs	ff	ff	ss	0	1.3	2.4	11.4	-35.9	-60.4	25.3	22.8
428	fs	ff	ff	ss	0	1.3	2.6	12.1	-35.3	-59.2	27.67	22.94
429	fs	ff	ff	ss	85	1.1	2.4	10.9	-37.5	-63.3	22.77	21.83
430	fs	ff	ff	ss	85	1.1	2.6	11.7	-37.3	-62	24.73	22.34
431	fs	ff	ff	ss	85	1.3	2.4	10.9	-37.3	-63.4	23.68	21.57
432	fs	ff	ff	ss	85	1.3	2.6	11.7	-36.6	-62.2	25.09	21.85
433	fs	ff	ss	ff	0	1.1	2.4	11.4	-36.6	-60.2	24.02	23.28
434	fs	ff	ss	ff	0	1.1	2.6	12.2	-36.3	-59.2	26.14	23.54
435	fs	ff	ss	ff	0	1.3	2.4	11.4	-35.9	-60.5	25.48	22.57
436	fs	ff	ss	ff	0	1.3	2.6	12.1	-35.3	-59.3	27.89	22.7
437	fs	ff	ss	ff	85	1.1	2.4	10.9	-37.7	-63.2	22.89	21.7
438	fs	ff	ss	ff	85	1.1	2.6	11.7	-37.5	-62	24.86	22.21
439	fs	ff	ss	ff	85	1.3	2.4	10.9	-37.3	-63.4	23.86	21.37
440	fs	ff	ss	ff	85	1.3	2.6	11.7	-36.7	-62.2	25.27	21.64
441	fs	ff	ff	ff	0	1.1	2.4	11.4	-36.6	-60.1	24.05	23.25
442	fs	ff	ff	ff	0	1.1	2.6	12.2	-36.2	-59.2	26.2	23.5
443	fs	ff	ff	ff	0	1.3	2.4	11.4	-35.8	-60.6	25.52	22.52
444	fs	ff	ff	ff	0	1.3	2.6	12.1	-35.2	-59.3	27.94	22.66
445	fs	ff	ff	ff	85	1.1	2.4	10.9	-37.7	-63.3	22.91	21.69
446	fs	ff	ff	ff	85	1.1	2.6	11.7	-37.5	-62	24.87	22.21
447	fs	ff	ff	ff	85	1.3	2.4	10.9	-37.3	-63.4	23.88	21.35
448	fs	ff	ff	ff	85	1.3	2.6	11.7	-36.7	-62.2	25.3	21.61
449	fs	sf	ss	ss	0	1.1	2.4	11.2	-38.8	-60.5	20.55	25.7
450	fs	sf	ss	ss	0	1.1	2.6	11.9	-38.4	-59.7	22.55	25.95
451	fs	sf	ss	ss	0	1.3	2.4	11.2	-38.3	-60.6	21.77	25.1
452	fs	sf	ss	ss	0	1.3	2.6	12	-37.4	-59.8	23.28	25.11
453	fs	sf	ss	ss	85	1.1	2.4	10.4	-39.7	-62.6	19.11	23.43
454	fs	sf	ss	ss	85	1.1	2.6	11.3	-39.3	-62	21.02	23.93
455	fs	sf	ss	ss	85	1.3	2.4	10.5	-39.4	-62.7	19.92	23.1
456	fs	sf	ss	ss	85	1.3	2.6	11.3	-38.8	-62	21.35	23.5
457	fs	sf	ff	ss	0	1.1	2.4	11.2	-38.8	-60.5	20.56	25.7
458	fs	sf	ff	ss	0	1.1	2.6	11.9	-38.4	-59.7	22.58	25.93
459	fs	sf	ff	ss	0	1.3	2.4	11.2	-38.3	-60.6	21.76	25.1
460	fs	sf	ff	ss	0	1.3	2.6	12	-37.4	-59.7	23.28	25.1
461	fs	sf	ff	ss	85	1.1	2.4	10.4	-39.7	-62.6	19.12	23.43
462	fs	sf	ff	ss	85	1.1	2.6	11.3	-39.3	-62	21.04	23.92
463	fs	sf	ff	ss	85	1.3	2.4	10.5	-39.4	-62.7	19.92	23.11
464	fs	sf	ff	ss	85	1.3	2.6	11.3	-38.8	-62	21.35	23.49
465	fs	sf	ss	ff	0	1.1	2.4	11.2	-38.9	-60.6	20.6	25.58
466	fs	sf	ss	ff	0	1.1	2.6	11.9	-38.5	-59.7	22.62	25.82
467	fs	sf	ss	ff	0	1.3	2.4	11.2	-38.3	-60.6	21.83	24.94

468	fs	sf	ss	ff	0	1.3	2.6	12	-37.5	-59.7	23.37	24.94
469	fs	sf	ss	ff	85	1.1	2.4	10.4	-39.9	-62.6	19.14	23.31
470	fs	sf	ss	ff	85	1.1	2.6	11.3	-39.6	-62.2	21.05	23.79
471	fs	sf	ss	ff	85	1.3	2.4	10.5	-39.6	-62.6	19.99	22.94
472	fs	sf	ss	ff	85	1.3	2.6	11.3	-38.9	-62.1	21.45	23.31
473	fs	sf	ff	ff	0	1.1	2.4	11.2	-38.9	-60.6	20.6	25.59
474	fs	sf	ff	ff	0	1.1	2.6	11.9	-38.5	-59.7	22.65	25.8
475	fs	sf	ff	ff	0	1.3	2.4	11.2	-38.3	-60.6	21.84	24.93
476	fs	sf	ff	ff	0	1.3	2.6	12	-37.5	-59.7	23.4	24.92
477	fs	sf	ff	ff	85	1.1	2.4	10.4	-39.9	-62.6	19.14	23.31
478	fs	sf	ff	ff	85	1.1	2.6	11.3	-39.6	-62.1	21.06	23.79
479	fs	sf	ff	ff	85	1.3	2.4	10.5	-39.6	-62.6	19.99	22.94
480	fs	sf	ff	ff	85	1.3	2.6	11.3	-38.9	-62.1	21.46	23.3
481	fs	fs	ss	ss	0	1.1	2.4	11.3	-36.6	-61.9	24.11	23.33
482	fs	fs	ss	ss	0	1.1	2.6	12	-36.3	-60.8	25.14	23.78
483	fs	fs	ss	ss	0	1.3	2.4	11.2	-36.3	-62	24.37	22.92
484	fs	fs	ss	ss	0	1.3	2.6	12	-35.7	-60.9	26.65	23.21
485	fs	fs	ss	ss	85	1.1	2.4	10.7	-37.5	-65.4	22.55	21.47
486	fs	fs	ss	ss	85	1.1	2.6	11.5	-37.1	-63.8	23.88	22.01
487	fs	fs	ss	ss	85	1.3	2.4	10.7	-37.5	-65.4	22.67	21.31
488	fs	fs	ss	ss	85	1.3	2.6	11.5	-36.9	-63.9	24.89	21.75
489	fs	fs	ff	ss	0	1.1	2.4	11.3	-36.6	-61.9	24.11	23.34
490	fs	fs	ff	ss	0	1.1	2.6	12	-36.3	-60.8	25.14	23.8
491	fs	fs	ff	ss	0	1.3	2.4	11.3	-36.3	-62	24.35	22.94
492	fs	fs	ff	ss	0	1.3	2.6	12	-35.7	-60.9	26.64	23.21
493	fs	fs	ff	ss	85	1.1	2.4	10.7	-37.5	-65.5	22.54	21.49
494	fs	fs	ff	ss	85	1.1	2.6	11.5	-37.1	-63.7	23.89	22.01
495	fs	fs	ff	ss	85	1.3	2.4	10.7	-37.5	-65.4	22.64	21.35
496	fs	fs	ff	ss	85	1.3	2.6	11.5	-36.9	-63.8	24.87	21.78
497	fs	fs	ss	ff	0	1.1	2.4	11.3	-36.8	-61.9	24.16	23.28
498	fs	fs	ss	ff	0	1.1	2.6	12	-36.4	-60.8	25.21	23.7
499	fs	fs	ss	ff	0	1.3	2.4	11.3	-36.4	-62	24.4	22.84
500	fs	fs	ss	ff	0	1.3	2.6	12	-35.8	-60.9	26.77	23.05
501	fs	fs	ss	ff	85	1.1	2.4	10.7	-37.7	-65.3	22.01	21.38
502	fs	fs	ss	ff	85	1.1	2.6	11.5	-37.3	-63.7	23.94	21.95
503	fs	fs	ss	ff	85	1.3	2.4	10.7	-37.6	-65.3	22.75	21.19
504	fs	fs	ss	ff	85	1.3	2.6	11.5	-37.1	-63.7	24.98	21.65
505	fs	fs	ff	ff	0	1.1	2.4	11.3	-36.8	-61.9	24.16	23.29
506	fs	fs	ff	ff	0	1.1	2.6	12	-36.4	-60.9	25.22	23.71
507	fs	fs	ff	ff	0	1.3	2.4	11.3	-36.4	-62	24.41	22.84
508	fs	fs	ff	ff	0	1.3	2.6	12	-35.8	-60.9	26.8	23.02
509	fs	fs	ff	ff	85	1.1	2.4	10.7	-37.7	-65.3	22.01	21.4
510	fs	fs	ff	ff	85	1.1	2.6	11.5	-37.4	-63.7	23.96	21.94
511	fs	fs	ff	ff	85	1.3	2.4	10.7	-37.6	-65.3	22.74	21.22
512	fs	fs	ff	ff	85	1.3	2.6	11.5	-37.1	-63.8	24.98	21.66

II. Appendix B: Mat Lab Codes

ADC Matlab code:

```
sampling_rate=1e5;
ext_sampling_rate=10000;
dc_shift = 1;
n = 10; %number of bits of ADC
ADC_bits = 5;
v_ref = 1.2;
bit_duration = 1/(ADC_bits*ext_sampling_rate); %??????
t=0:1/sampling_rate:0.01;
signal=sin(2*pi*1000*t);
figure()
plot(t,signal);
axis([0,0.01,-1,1]);
title('Signal');
xlabel('Time (sec)');
ylabel('Amplitude (mV)');
grid on;

%-----DC_Shift-----
signal = signal + dc_shift;
%-----SAMPLING-----
pulse_rate = 1* ext_sampling_rate;
sample_hold=square(2*pi*ext_sampling_rate*t);
signal_sampled = zeros(size(signal));
for i=1:length(t)*ext_sampling_rate
    signal_sampled = signal_sampled + signal(i*sampling_rate/ext_sampling_rate).*rectpuls((t-i/ext_sampling_rate-
2/pulse_rate)*2*pulse_rate-0.5);
end
figure();
plot(t,signal_sampled);
axis([0,0.01,0,2]);
title('Sampled Signal');
xlabel('Time (sec)');
ylabel('Amplitude (V)');
grid on;

%-----Quantization-----
signal_quantized = ceil(2^ADC_bits*signal_sampled /v_ref)*v_ref/2^ADC_bits-v_ref/(2*2^ADC_bits);
figure();
plot(t,signal_quantized);
axis([0,0.01,0,2]);
title('Quantized Signal');
xlabel('Time (sec)');
ylabel('Amplitude (V)');
grid on;
q_vin=0:1e-5:1.2;
q_vout=ceil(2^ADC_bits*q_vin/v_ref)*v_ref/2^ADC_bits-v_ref/(2*2^ADC_bits);
figure()
hold on;
plot(q_vin,q_vin);
plot(q_vin,q_vout,'r');
title('Quantizer Response');
xlabel('Vin (V)');
ylabel('Vout (V)');
axis([0,1.2,0,1.2]);
legend('Y = X', 'Quantizer Response');
signal_quantized_cont = ceil(2^ADC_bits*signal /v_ref)*v_ref/2^ADC_bits - v_ref/(2*2^ADC_bits);
quantization_noise = signal - signal_quantized_cont;
```

```

figure();
plot(t,quantization_noise);
title('Quantization Noise');
xlabel('Time (sec)');
ylabel('Amplitude (V)');
%axis([0,0.02,0,0.02]);

signal_power = sum(signal.^2)*sampling_rate;
quantization_noise_power = sum(quantization_noise.^2)*sampling_rate;
SQNR = signal_power / quantization_noise_power;

%-----Bits_Generation-----
signal_bit_value = signal_quantized * 2 ^ ADC_bits / v_ref - 0.5;

for i=1:length(signal_bit_value)*ext_sampling_rate/sampling_rate-1
j=floor(i*sampling_rate/ext_sampling_rate+sampling_rate/(ext_sampling_rate*2));
signal_binary(i,:) = bitget(signal_bit_value(j),ADC_bits:-1:1);
end

%-----Serialization-----
signal_binary_length = size(signal_binary);
signal_binary_length = signal_binary_length(1);
for i=1:signal_binary_length
    signal_serialized(1+(i-1)*ADC_bits:i*ADC_bits) = signal_binary(i,:);
end

%%-----Digital_Signal_Generation-----
%%rectpuls((t-start_time)/duration-0.5);

signal_digital = zeros([1 2*length(signal)]+1);
t2=linspace(0,t(end),length(signal_digital));
% t2=linspace(0,t(end),length(ecg_digital));
for i=1:length(signal_serialized)
    if(signal_serialized(i)==0)
        signal_digital(floor(1+(i-1)*bit_duration*2*sampling_rate):i*bit_duration*2*sampling_rate)=0;
    end
end
figure()
plot(t2,signal_digital);
title('Digital Signal');
xlabel('Time (sec)');
ylabel('Value');

```

Dnl & Inl code:

```

function [dnl,inl] = dnl_inl_sin(y)
%DNL_INL_SIN
% dnl and inl ADC output
% input y contains the ADC output
% vector obtained from quantizing a
% sinusoid
% Boris Murmann, Aug 2002
% Bernhard Boser, Sept 2002
% histogram boundaries
minbin=min(y);

```

```

maxbin=max(y);
% histogram
h = hist(y, minbin:maxbin);
% cumulative histogram
ch = cumsum(h);
% transition levels
T = -cos(pi*ch/sum(h));
% linearized histogram
hlin = T(2:end) - T(1:end-1);
% truncate at least first and last
% bin, more if input did not clip ADC
trunc=2;
hlin_trunc = hlin(1+trunc:end-trunc);
% calculate lsb size and dnl
lsb= sum(hlin_trunc) / (length(hlin_trunc));
dnl= [0 hlin_trunc/lsb-1];
misscodes = length(find(dnl<-0.9));
% calculate inl
inl= cumsum(dnl);

```

Reference buffer ocean script:

```

ocnWaveformTool( 'wavescan )

simulator( 'spectre )

design( "/home/amrhema/simulation/ref_buffer_with_switch/spectre/schematic/netlist/netlist")

resultsDir( "/home/amrhema/simulation/ref_buffer_with_switch/spectre/schematic" )

desVar(      "Vdd" 1.2   )

counter=0

tmp='(0 85)

corner_txt=('"ss" "ff" "sf" "fs")

cap_res_txt=('"ss" "ff")

Vss=(1.1 1.3)

fp=outfile("/home/amrhema/Desktop/out.txt" "a")

fprintf(fp "phase_margin corner res cap temp supply\n")

close(fp)

foreach(corner corner_txt

foreach(res cap_res_txt

```

```

foreach(cap cap_res_txt
    modelFile(
        list("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/rf013.scs"
        sprintf( nil "%s" corner ))

        list("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/rf013.scs"
        sprintf( nil "%s_rfmos" corner ))

        list("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/ResModel.sc
s" sprintf( nil "%s_disres" res ))

        list("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/rf013.scs"
        sprintf( nil "%s_mim" cap ))
    )

foreach(temp tmp
    temp( temp )

foreach(supply Vss

desVar(      "Vdd" supply)
analysis('tran ?stop "1u" ?errpreset "moderate" )
analysis('dc ?saveOppoint t )
analysis('stb ?start "1" ?stop "1G" ?step "10M"
        ?probe "/IPRB0" )

run()

sprintf(label "corner(NP-%s_mim-%s_res-%s_temp-%n_supply-%n_count(%n))" corner cap res
temp supply counter)

Phase\ Margin = getData("phaseMargin" ?result "stb_margin")

plot( Phase\ Margin ?expr list(label) '( "Phase Margin" ) )

```

```
fp=outfile("/home/amrhema/Desktop/out.txt" "a")
fprintf(fp "%n %s %s %s %n %n \n" Phase\ Margin corner res cap temp supply)
close(fp)
)
```

Ocean script used to test Noise Figure of LNA

```
ocnWaveformTool( 'wavescan' )
simulator( 'spectre' )
design( "/home/amgad/simulation/last_onchip/spectre/schematic/netlist/netlist" )
resultsDir( "/home/amgad/simulation/last_onchip/spectre/schematic" )

counter=0
corner_txt=('ss" "ff" "sf" "fs")
cap_res_txt=('ss" "ff")
ind_res_txt=('ss" "ff")
temp_=(0 85)
Vss=(1.1 1.3)

foreach(corner corner_txt
foreach(res cap_res_txt
foreach(cap cap_res_txt
foreach(ind ind_res_txt
    modelFile(
        list("/tools/pdks/tsmc13rf/tsmc13rf/../../models/RF_12_25_FSG/T013MMSP001_1_3/spectre/rf013.scs" sprintf( nil "%s" corner ))

        list("/tools/pdks/tsmc13rf/tsmc13rf/../../models/RF_12_25_FSG/T013MMSP001_1_3/spectre/rf013.scs" sprintf( nil "%s_rfmos" corner ))

        list("/tools/pdks/tsmc13rf/tsmc13rf/../../models/RF_12_25_FSG/T013MMSP001_1_3/spectre/ResModel.scs" sprintf( nil "%s_disres" res ))

        list("/tools/pdks/tsmc13rf/tsmc13rf/../../models/RF_12_25_FSG/T013MMSP001_1_3/spectre/rf013.scs" sprintf( nil "%s_mim" cap ))
list("/tools/pdks/tsmc13rf/models/RF_12_25_FSG/T013MMSP001_1_3/spectre/rf013.scs"
sprintf( nil "%s_rfind" ind ))
    )

foreach(tmp temp_
temp( tmp )
foreach(supply Vss

desVar(          "Vdd" supply)
counter=counter+1
analysis('noise ?start "800M" ?stop "1.05G" ?oprobe "/PORT12"
        ?iprobe "/PORT13" )
desVar(          "prf" -40      )
run()
sprintf(label "corner(NP-%s_mim-%s_res-%s_temp-%n_supply-%n_count(%n)_ind-%s)" corner cap
res tmp supply counter ind)
text1=strcat("NF" label )

noise\ figure = getData("NF" ?result "noise")
```


23	Ss	ss	ff	ff	85	1.1	3.7
24	Ss	ss	ff	ff	85	1.3	3.6
25	Ss	ff	ff	ss	0	1.1	2.6
26	Ss	ff	ff	ss	0	1.3	2.5
27	Ss	ff	ff	ss	85	1.1	3.4
28	Ss	ff	ff	ss	85	1.3	3.4
29	Ss	ff	ff	ff	0	1.1	2.4
30	Ss	ff	ff	ff	0	1.3	2.4
31	Ss	ff	ff	ff	85	1.1	3.2
32	Ss	ff	ff	ff	85	1.3	3.2
33	Ff	ss	ss	ss	0	1.1	2.4
34	Ff	ss	ss	ss	0	1.3	2.4
35	Ff	ss	ss	ss	85	1.1	3.2
36	Ff	ss	ss	ss	85	1.3	3.2
37	Ff	ss	ss	ff	0	1.1	2.2
38	Ff	ss	ss	ff	0	1.3	2.1
39	Ff	ss	ss	ff	85	1.1	2.9
40	Ff	ss	ss	ff	85	1.3	2.9
41	Ff	ff	ss	ss	0	1.1	2.1
42	Ff	ff	ss	ss	0	1.3	2
43	Ff	ff	ss	ss	85	1.1	2.8
44	Ff	ff	ss	ss	85	1.3	2.7
45	Ff	ff	ss	ff	0	1.1	1.9
46	Ff	ff	ss	ff	0	1.3	1.9
47	Ff	ff	ss	ff	85	1.1	2.6
48	Ff	ff	ss	ff	85	1.3	2.6
49	Ff	ss	ff	ss	0	1.1	2.6
50	Ff	ss	ff	ss	0	1.3	2.5
51	Ff	ss	ff	ss	85	1.1	3.4
52	Ff	ss	ff	ss	85	1.3	3.3
53	Ff	ss	ff	ff	0	1.1	2.3
54	Ff	ss	ff	ff	0	1.3	2.3
55	Ff	ss	ff	ff	85	1.1	3.1
56	Ff	ss	ff	ff	85	1.3	3
57	Ff	ff	ff	ss	0	1.1	2.2
58	Ff	ff	ff	ss	0	1.3	2.1
59	Ff	Ff	ff	ss	85	1.1	2.9
60	Ff	Ff	ff	ss	85	1.3	2.9
61	Ff	Ff	ff	ff	0	1.1	2
62	Ff	Ff	ff	ff	0	1.3	2
63	Ff	Ff	ff	ff	85	1.1	2.7
64	Ff	Ff	ff	ff	85	1.3	2.7
65	Sf	Ss	ss	ss	0	1.1	2.9
66	Sf	Ss	ss	ss	0	1.3	2.9
67	Sf	Ss	ss	ss	85	1.1	3.8
68	Sf	Ss	ss	ss	85	1.3	3.7
69	Sf	Ss	ss	ff	0	1.1	2.6
70	Sf	Ss	ss	ff	0	1.3	2.6

71	Sf	Ss	ss	ff	85	1.1	3.4
72	Sf	Ss	ss	ff	85	1.3	3.4
73	Sf	Ff	ss	ss	0	1.1	2.4
74	Sf	Ff	ss	ss	0	1.3	2.4
75	Sf	Ff	ss	ss	85	1.1	3.2
76	sf	Ff	ss	ss	85	1.3	3.1
77	sf	Ff	ss	ff	0	1.1	2.3
78	sf	Ff	ss	ff	0	1.3	2.3
79	sf	Ff	ss	ff	85	1.1	3
80	sf	Ff	ss	ff	85	1.3	3
81	sf	Ss	ff	ss	0	1.1	3.1
82	sf	Ss	ff	ss	0	1.3	3
83	sf	Ss	ff	ss	85	1.1	3.9
84	sf	Ss	ff	ss	85	1.3	3.9
85	sf	Ss	ff	ff	0	1.1	2.8
86	sf	Ss	ff	ff	0	1.3	2.7
87	sf	Ss	ff	ff	85	1.1	3.6
88	sf	Ss	ff	ff	85	1.3	3.5
89	sf	Ff	ff	ss	0	1.1	2.6
90	sf	Ff	ff	ss	0	1.3	2.5
91	sf	Ff	ff	ss	85	1.1	3.3
92	sf	Ff	ff	ss	85	1.3	3.3
93	sf	Ff	ff	ff	0	1.1	2.4
94	sf	Ff	ff	ff	0	1.3	2.4
95	sf	Ff	ff	ff	85	1.1	3.2
96	sf	Ff	ff	ff	85	1.3	3.1
97	fs	Ss	ss	ss	0	1.1	2.5
98	fs	Ss	ss	ss	0	1.3	2.5
99	fs	Ss	ss	ss	85	1.1	3.3
100	fs	Ss	ss	ss	85	1.3	3.3
101	fs	Ss	ss	ff	0	1.1	2.3
102	fs	Ss	ss	ff	0	1.3	2.2
103	fs	Ss	ss	ff	85	1.1	3
104	fs	Ss	ss	ff	85	1.3	3
105	fs	Ff	ss	ss	0	1.1	2.1
106	fs	Ff	ss	ss	0	1.3	2.1
107	fs	Ff	ss	ss	85	1.1	2.9
108	fs	Ff	ss	ss	85	1.3	2.8
109	fs	Ff	ss	ff	0	1.1	2
110	fs	Ff	ss	ff	0	1.3	2
111	fs	Ff	ss	ff	85	1.1	2.7
112	fs	Ff	ss	ff	85	1.3	2.7
113	fs	Ss	ff	ss	0	1.1	2.6
114	fs	Ss	ff	ss	0	1.3	2.6
115	fs	Ss	ff	ss	85	1.1	3.5
116	fs	Ss	ff	ss	85	1.3	3.4
117	fs	Ss	ff	ff	0	1.1	2.4
118	fs	Ss	ff	ff	0	1.3	2.3

119	fs	Ss	ff	ff	85	1.1	3.2
120	fs	Ss	ff	ff	85	1.3	3.1
121	fs	Ff	ff	ss	0	1.1	2.2

2-S11 across corners

Counter	NMOSPMOS	Mimcap	Resistance	ind	Temp	Vdd	NF
1	ss	Ss	ss	ss	0	1.1	-13
2	ss	Ss	ss	ss	0	1.3	-13
3	ss	Ss	ss	ss	85	1.1	-13
4	ss	Ss	ss	ss	85	1.3	-13
5	ss	Ss	ss	ff	0	1.1	-13
6	ss	Ss	ss	ff	0	1.3	-14
7	ss	Ss	ss	ff	85	1.1	-14
8	ss	Ss	ss	ff	85	1.3	-14
9	ss	Ff	ss	ss	0	1.1	-17
10	ss	Ff	ss	ss	0	1.3	-18
11	ss	Ff	ss	ss	85	1.1	-18
12	ss	Ff	ss	ss	85	1.3	-18
13	ss	Ff	ss	ff	0	1.1	-12
14	ss	Ff	ss	ff	0	1.3	-12
15	ss	Ff	ss	ff	85	1.1	-12
16	ss	Ff	ss	ff	85	1.3	-12
17	ss	Ss	ff	ss	0	1.1	-13
18	ss	Ss	ff	ss	0	1.3	-13
19	ss	Ss	ff	ss	85	1.1	-13
20	ss	Ss	ff	ss	85	1.3	-13
21	ss	Ss	ff	ff	0	1.1	-13
22	ss	Ss	ff	ff	0	1.3	-14
23	ss	Ss	ff	ff	85	1.1	-14
24	ss	Ss	ff	ff	85	1.3	-14
25	ss	Ff	ff	ss	0	1.1	-18
26	ss	Ff	ff	ss	0	1.3	-18
27	ss	Ff	ff	ss	85	1.1	-18
28	ss	Ff	ff	ss	85	1.3	-18
29	ss	Ff	ff	ff	0	1.1	-12
30	ss	Ff	ff	ff	0	1.3	-12
31	ss	Ff	ff	ff	85	1.1	-12
32	ss	Ff	ff	ff	85	1.3	-12
33	ff	Ss	ss	ss	0	1.1	-15
34	ff	Ss	ss	ss	0	1.3	-16
35	ff	Ss	ss	ss	85	1.1	-15
36	ff	Ss	ss	ss	85	1.3	-16
37	ff	Ss	ss	ff	0	1.1	-17
38	ff	Ss	ss	ff	0	1.3	-17
39	ff	Ss	ss	ff	85	1.1	-17
40	ff	Ss	ss	ff	85	1.3	-18
41	ff	Ff	ss	ss	0	1.1	-20

42	ff	Ff	ss	ss	0	1.3	-20
43	ff	Ff	ss	ss	85	1.1	-20
44	ff	Ff	ss	ss	85	1.3	-20
45	ff	Ff	ss	ff	0	1.1	-13
46	ff	Ff	ss	ff	0	1.3	-13
47	ff	Ff	ss	ff	85	1.1	-13
48	ff	Ff	ss	ff	85	1.3	-13
49	ff	Ss	ff	ss	0	1.1	-15
50	ff	Ss	ff	ss	0	1.3	-16
51	ff	Ss	ff	ss	85	1.1	-15
52	ff	Ss	ff	ss	85	1.3	-16
53	ff	Ss	ff	ff	0	1.1	-17
54	ff	Ss	ff	ff	0	1.3	-17
55	ff	Ss	ff	ff	85	1.1	-17
56	ff	Ss	ff	ff	85	1.3	-18
57	ff	Ff	ff	ss	0	1.1	-20
58	ff	Ff	ff	ss	0	1.3	-20
59	ff	Ff	ff	ss	85	1.1	-20
60	ff	Ff	ff	ss	85	1.3	-20
61	ff	Ff	ff	ff	0	1.1	-13
62	ff	Ff	ff	ff	0	1.3	-13
63	ff	Ff	ff	ff	85	1.1	-13
64	ff	Ff	ff	ff	85	1.3	-13
65	sf	Ss	ss	ss	0	1.1	-12
66	sf	Ss	ss	ss	0	1.3	-12
67	sf	Ss	ss	ss	85	1.1	-13
68	sf	Ss	ss	ss	85	1.3	-13
69	sf	Ss	ss	ff	0	1.1	-13
70	sf	Ss	ss	ff	0	1.3	-13
71	sf	Ss	ss	ff	85	1.1	-14
72	sf	Ss	ss	ff	85	1.3	-14
73	sf	Ff	ss	ss	0	1.1	-17
74	sf	Ff	ss	ss	0	1.3	-17
75	sf	Ff	ss	ss	85	1.1	-18
76	sf	Ff	ss	ss	85	1.3	-18
77	sf	Ff	ss	ff	0	1.1	-12
78	sf	Ff	ss	ff	0	1.3	-12
79	sf	Ff	ss	ff	85	1.1	-12
80	sf	Ff	ss	ff	85	1.3	-12
81	sf	Ss	ff	ss	0	1.1	-12
82	sf	Ss	ff	ss	0	1.3	-12
83	sf	Ss	ff	ss	85	1.1	-13
84	sf	Ss	ff	ss	85	1.3	-13
85	sf	Ss	ff	ff	0	1.1	-13
86	sf	Ss	ff	ff	0	1.3	-13
87	sf	Ss	ff	ff	85	1.1	-14
88	sf	Ss	ff	ff	85	1.3	-14
89	sf	Ff	ff	ss	0	1.1	-17

90	sf	Ff	ff	ss	0	1.3	-17
91	sf	Ff	ff	ss	85	1.1	-18
92	sf	Ff	ff	ss	85	1.3	-18
93	sf	Ff	ff	ff	0	1.1	-12
94	sf	Ff	ff	ff	0	1.3	-12
95	sf	Ff	ff	ff	85	1.1	-12
96	sf	Ff	ff	ff	85	1.3	-12
97	fs	Ss	ss	ss	0	1.1	-15
98	fs	Ss	ss	ss	0	1.3	-16
99	fs	Ss	ss	ss	85	1.1	-15
100	fs	Ss	ss	ss	85	1.3	-16
101	fs	Ss	ss	ff	0	1.1	-17
102	fs	Ss	ss	ff	0	1.3	-18
103	fs	Ss	ss	ff	85	1.1	-17
104	fs	Ss	ss	ff	85	1.3	-18
105	fs	Ff	ss	ss	0	1.1	-20
106	fs	Ff	ss	ss	0	1.3	-20
107	fs	Ff	ss	ss	85	1.1	-20
108	fs	Ff	ss	ss	85	1.3	-20
109	fs	Ff	ss	ff	0	1.1	-13
110	fs	Ff	ss	ff	0	1.3	-13
111	fs	Ff	ss	ff	85	1.1	-13
112	fs	Ff	ss	ff	85	1.3	-13
113	fs	Ss	ff	ss	0	1.1	-15
114	fs	Ss	ff	ss	0	1.3	-16
115	fs	Ss	ff	ss	85	1.1	-15
116	fs	Ss	ff	ss	85	1.3	-16
117	fs	Ss	ff	ff	0	1.1	-17
118	fs	Ss	ff	ff	0	1.3	-18
119	fs	Ss	ff	ff	85	1.1	-17
120	fs	Ss	ff	ff	85	1.3	-18
121	fs	Ff	ff	ss	0	1.1	-20
122	fs	Ff	ff	ss	0	1.3	-20
123	fs	Ff	ff	ss	85	1.1	-20
124	fs	Ff	ff	ss	85	1.3	-20
125	fs	Ff	ff	ff	0	1.1	-13
126	fs	Ff	ff	ff	0	1.3	-13
127	fs	Ff	ff	ff	85	1.1	-13
128	fs	Ff	ff	ff	85	1.3	-13

3-S21 across corners

Counter	NMOSPMOS	Mimcap	Resistance	ind	Temp	Vdd	NF
1	ss	ss	ss	ss	0	1.1	18.8
2	ss	ss	ss	ss	0	1.3	18.9
3	ss	ss	ss	ss	85	1.1	17.8
4	ss	ss	ss	ss	85	1.3	17.9
5	ss	ss	ss	ff	0	1.1	20.8

6	ss	ss	ss	ff	0	1.3	20.9
7	ss	ss	ss	ff	85	1.1	19.7
8	ss	ss	ss	ff	85	1.3	19.7
9	ss	ff	ss	ss	0	1.1	21.6
10	ss	ff	ss	ss	0	1.3	21.7
11	ss	ff	ss	ss	85	1.1	20.7
12	ss	ff	ss	ss	85	1.3	20.8
13	ss	ff	ss	ff	0	1.1	21.1
14	ss	ff	ss	ff	0	1.3	21.1
15	ss	ff	ss	ff	85	1.1	20.4
16	ss	ff	ss	ff	85	1.3	20.5
17	ss	ss	ff	ss	0	1.1	18
18	ss	ss	ff	ss	0	1.3	18
19	ss	ss	ff	ss	85	1.1	16.9
20	ss	ss	ff	ss	85	1.3	17
21	ss	ss	ff	ff	0	1.1	19.7
22	ss	ss	ff	ff	0	1.3	19.7
23	ss	ss	ff	ff	85	1.1	18.5
24	ss	ss	ff	ff	85	1.3	18.6
25	ss	ff	ff	ss	0	1.1	20.4
26	ss	ff	ff	ss	0	1.3	20.5
27	ss	ff	ff	ss	85	1.1	19.5
28	ss	ff	ff	ss	85	1.3	19.6
29	ss	ff	ff	ff	0	1.1	20
30	ss	ff	ff	ff	0	1.3	20.1
31	ss	ff	ff	ff	85	1.1	19.3
32	ss	ff	ff	ff	85	1.3	19.4
33	ff	ss	ss	ss	0	1.1	20.5
34	ff	ss	ss	ss	0	1.3	20.7
35	ff	ss	ss	ss	85	1.1	19.4
36	ff	ss	ss	ss	85	1.3	19.5
37	ff	ss	ss	ff	0	1.1	22.4
38	ff	ss	ss	ff	0	1.3	22.5
39	ff	ss	ss	ff	85	1.1	21.1
40	ff	ss	ss	ff	85	1.3	21.2
41	ff	ff	ss	ss	0	1.1	22.5
42	ff	ff	ss	ss	0	1.3	22.7
43	ff	ff	ss	ss	85	1.1	21.6
44	ff	ff	ss	ss	85	1.3	21.7
45	ff	ff	ss	ff	0	1.1	21.7
46	ff	ff	ss	ff	0	1.3	21.9
47	ff	ff	ss	ff	85	1.1	21.1
48	ff	ff	ss	ff	85	1.3	21.2
49	ff	ss	ff	ss	0	1.1	19.6
50	ff	ss	ff	ss	0	1.3	19.7
51	ff	ss	ff	ss	85	1.1	18.4
52	ff	ss	ff	ss	85	1.3	18.6
53	ff	ss	ff	ff	0	1.1	21.1

54	ff	ss	ff	ff	0	1.3	21.2
55	ff	ss	ff	ff	85	1.1	19.9
56	ff	ss	ff	ff	85	1.3	20
57	ff	ff	ff	ss	0	1.1	21.4
58	ff	ff	ff	ss	0	1.3	21.5
59	ff	ff	ff	ss	85	1.1	20.4
60	ff	ff	ff	ss	85	1.3	20.5
61	ff	ff	ff	ff	0	1.1	20.8
62	ff	ff	ff	ff	0	1.3	20.9
63	ff	ff	ff	ff	85	1.1	20
64	ff	ff	ff	ff	85	1.3	20.2
65	sf	ss	ss	ss	0	1.1	19.3
66	sf	ss	ss	ss	0	1.3	19.3
67	sf	ss	ss	ss	85	1.1	18.3
68	sf	ss	ss	ss	85	1.3	18.4
69	sf	ss	ss	ff	0	1.1	21.1
70	sf	ss	ss	ff	0	1.3	21.1
71	sf	ss	ss	ff	85	1.1	20
72	sf	ss	ss	ff	85	1.3	20.1
73	sf	ff	ss	ss	0	1.1	21.2
74	sf	ff	ss	ss	0	1.3	21.3
75	sf	ff	ss	ss	85	1.1	20.4
76	sf	ff	ss	ss	85	1.3	20.5
77	sf	ff	ss	ff	0	1.1	20.3
78	sf	ff	ss	ff	0	1.3	20.3
79	sf	ff	ss	ff	85	1.1	19.8
80	sf	ff	ss	ff	85	1.3	19.9
81	sf	ss	ff	ss	0	1.1	18.3
82	sf	ss	ff	ss	0	1.3	18.3
83	sf	ss	ff	ss	85	1.1	17.3
84	sf	ss	ff	ss	85	1.3	17.4
85	sf	ss	ff	ff	0	1.1	19.8
86	sf	ss	ff	ff	0	1.3	19.9
87	sf	ss	ff	ff	85	1.1	18.7
88	sf	ss	ff	ff	85	1.3	18.8
89	sf	ff	ff	ss	0	1.1	20
90	sf	ff	ff	ss	0	1.3	20.1
91	sf	ff	ff	ss	85	1.1	19.2
92	sf	ff	ff	ss	85	1.3	19.3
93	sf	ff	ff	ff	0	1.1	19.4
94	sf	ff	ff	ff	0	1.3	19.4
95	sf	ff	ff	ff	85	1.1	18.8
96	sf	ff	ff	ff	85	1.3	18.9
97	fs	ss	ss	ss	0	1.1	19.9
98	fs	ss	ss	ss	0	1.3	20
99	fs	ss	ss	ss	85	1.1	18.8
100	fs	ss	ss	ss	85	1.3	18.9
101	fs	ss	ss	ff	0	1.1	21.9

102	fs	ss	ss	ff	0	1.3	22
103	fs	ss	ss	ff	85	1.1	20.6
104	fs	ss	ss	ff	85	1.3	20.8
105	fs	ff	ss	ss	0	1.1	22.8
106	fs	ff	ss	ss	0	1.3	22.9
107	fs	ff	ss	ss	85	1.1	21.7
108	fs	ff	ss	ss	85	1.3	21.8
109	fs	ff	ss	ff	0	1.1	22.3
110	fs	ff	ss	ff	0	1.3	22.4
111	fs	ff	ss	ff	85	1.1	21.5
112	fs	ff	ss	ff	85	1.3	21.6
113	fs	ss	ff	ss	0	1.1	19.1
114	fs	ss	ff	ss	0	1.3	19.2
115	fs	ss	ff	ss	85	1.1	17.9
116	fs	ss	ff	ss	85	1.3	18.1
117	fs	ss	ff	ff	0	1.1	20.8
118	fs	ss	ff	ff	0	1.3	20.9
119	fs	ss	ff	ff	85	1.1	19.6
120	fs	ss	ff	ff	85	1.3	19.7
121	fs	ff	ff	ss	0	1.1	21.6
122	fs	ff	ff	ss	0	1.3	21.7
123	fs	ff	ff	ss	85	1.1	20.5
124	fs	ff	ff	ss	85	1.3	20.6
125	fs	ff	ff	ff	0	1.1	21.3
126	fs	ff	ff	ff	0	1.3	21.4
127	fs	ff	ff	ff	85	1.1	20.4
128	fs	ff	ff	ff	85	1.3	20.5

Mat Lab code for plotting PGA gains setting

```
clc;
close all;

pga = [ -0.066,-
0.066,1.96,1.96,3.89,3.89,5.832,5.832,7.93,7.93,9.962,9.962,11.65,11.65,13.65,13.65,15.55,15.55,17.45,17.45,19.49,19.4
9,21.45,21.45,24.80,24.80,26.59,26.59,28.35,28.35,30.20,30.20,31.95,31.95,36.62,36.62,38.65,38.65,40.59,40.59,42.53,4
2.53,44.63,44.63,46.05,46.05,48.07,48.07,49.99,49.99,51.91,51.91,53.99,53.99,55.99,55.99 ];
%length(pga)
steps = [
0,1,1.001,2,2.001,3,3.001,4,4.001,5,5.001,6,6.001,7,7.001,8,8.001,9,9.001,10,10.001,11,11.001,12,12.001,13,13.001,14,1
4.001,15,15.001,16,16.001,17,17.001,18,18.001,19,19.001,20,20.001,21,21.001,22,22.001,23,23.001,24,24.001,25,25.00
1,26,26.001,27,27.001,28];
%length(steps)
plot(steps,pga);
xlabel('Gain code');
ylabel('PGA dB');
title('PGA output vs, gain code');
grid on;
```

Mat Lab code for plotting PM of CMFB "PGA"

```
clc;
close all;
clear all;

x = [ 63.366, 43.194, 62.950, 41.634, 63.366, 43.194, 62.950, 41.634, 63.366, 43.194, 62.950, 41.634, 63.366, 43.194,
62.950, 41.634, 71.586, 56.456, 71.328, 55.565, 71.586, 56.456, 71.328, 55.565, 71.586, 56.456, 71.328, 55.565, 71.586,
56.456, 71.328, 55.565, 71.747, 55.178, 71.458, 53.990, 71.747, 55.178, 71.458, 53.990, 71.747, 55.178, 71.458, 53.990,
71.747, 55.178, 71.458, 53.990, 70.274, 42.888, 69.527, 42.613, 70.274, 42.888, 69.527, 42.613, 70.274, 42.888, 69.527,
42.613, 70.274, 42.888, 69.527, 42.613];

plot(x);
xlabel('Corners of MOSFET, Resistor, Capacitor, Temp & Supply');
ylabel('PM of CMFB Loop in PGA');
title('PM variations across corners');
```

Mat Lab code for plotting PM of CMFB "PD"

```
clc;
close all;
clear all;

x = [17.779 ,120.894 ,16.863 ,118.120 ,17.779 ,120.894 ,16.863 ,118.120 ,17.779 ,120.894 ,16.863 ,118.120 ,17.779
,120.894 ,16.863 ,118.120 ,8.162 ,136.726 ,7.951 ,141.330 ,8.162 ,136.726 ,7.951 ,141.330 ,8.162 ,136.726 ,7.951
,141.330 ,8.162 ,136.726 ,7.951 ,141.330 ,30.332 ,161.146 ,35.032 ,157.190 ,30.332 ,161.146,35.032 ,157.190 ,30.332
,161.146 ,35.032 ,157.190 ,30.332 ,161.146 ,35.032 ,157.190 ,14.549 ,146.031 ,12.622 ,135.675 ,14.549 ,146.031 ,12.622
,135.675 ,14.549 ,146.031 ,12.622 ,135.675 ,14.549 ,146.031 ,12.622 ,135.675];

plot(x);
xlabel('Corners of MOSFET, Resistor, Capacitor, Temp & Supply');
ylabel('PM of CMFB Loop in PD Folded Cascode');
title('PM variations across corners');
```

MATLAB code for power amplifier for drawing Output power VS frequency

```
Freq = [50 100 150 200 250 300 350 400 450 500 550 600 650 700 750 800 850 900 915 925 950
1000 1050 1100 1150 1200 1250 1300 1400 1500 1600 1700 1800];

Pout = [-22.99 -6.12 -2.38 0.52 3.49 6.27 8.28 8.68 8.12 7.58 7.4 7.56 8.18 9.26 10.46 11.48 11.86
11.62 11.44 11.3 10.86 9.81 8.84 8.33 8.48 9.05 8.69 6.31 -1.5 -7.61 -11.73 -15.16 -27.55];

PAE = [0.02 0.54 0.79 1.46 2.9 5.63 8.82 9.65 8.51 7.44 7.13 7.39 8.5 11.06 15.21 21.14 25.56 24.74
23.79 22.91 19.94 14 10.49 8.97 9.14 10.57 10.58 6.22 0.94 0.23 0.09 0.04 0];

figure(1)

plot(Freq,Pout,'-*r');

hold on;

xlabel('Freq (MHz)')

ylabel('Output power (dBm)')

Grid on ;
```

15.2.4 Matlab Code (GFSK BW calculation)

```
%GFSK parameters
```

```
h=1.016;
```

```
Rb=250e3;
```

```
BT=0.5;
```

```
messages=20;
```

```
BW_perc=0.95;
```

```
%%% %
```

```
f_3db=BT*Rb;
```

```
sigma=sqrt(log(2))/(2*pi*f_3db);
```

```
f_dev=h*Rb/2;
```

```
%%% %
```

```
fsample=10e8;
```

```
samp=1/Rb;
```

```
time=messages*samp/2;
```

```
%Signals
```

```
t=-1*time:(1/fsample):time;
```

```
%Binary data
```

```
y=square(2*pi*t*0.5*Rb+pi/2);
```

```
%gaussian filter
```

```
ht=sqrt(pi*2/log(2))*(BT/samp)*exp(-(pi^2*2*(BT/samp)^2/log(2))*power(t,2))/fsample;
```

```
%Convolution
```

```
psh_sig=conv(y, ht);
```

```
t1=(0:length(psh_sig)-1).*(1/fsample);
```

```
%calculate phi
```

```
phi=zeros(1,length(psh_sig));
```

```
phi(1)= 0;
```

```
for k=2:length(psh_sig)
```

```

    phi(k)=(psh_sig(k-1)+psh_sig(k))*0.5*(1/fsample)+phi(k-1);
end

phi=phi*2*pi*f_dev;

%passband signal
gfm=sin(2*pi*2.5e6*t1+phi);

%Frequency domain
NFFT = 2^nextpow2(length(t1));
yf= abs(fft(gfm,NFFT))/length(t);
f = fsample/2*linspace(0,1,NFFT/2+1);

%BW calculation

%fc location dtermination
h=find(yf==max(yf));
mid=h(1);
su=2*yf(mid)^2;
for k=1:mid-1
    su=su+2*(yf(mid+k))^2+2*(yf(mid-k))^2;
end
plot(f, (2*abs(yf(1:NFFT/2+1))));
axis([0 5*10^6 0 2]);
%*BW calculation
value=su;
su=2*yf(mid)^2;
for k=1:mid-1
    su=su+2*(yf(mid+k))^2+2*(yf(mid-k))^2;
    if(su>BW_perc*value)
        break;
    end
end

```

end

BW=(f(mid+k)-f(mid-k))*1e-3;

msgbox(sprintf('Bw is %1.3f Khz',BW));

break;

15.2.5

PLL system design using matlab code

```
F_min=2*902*((10)^6); F_max=2*928*((10)^6);
Fref=26*((10)^6);
F_average=((F_max+F_min))/2;
Range=F_max-F_min;
RFopt=F_average;
V_supply=1.2;
Icp=50e-6;
compliance_range=0.5;
K0=Range/compliance_range;
K_vco=2*pi*K0;
K_phi=Icp/(2*pi);
N=RFopt/Fref;
N_min=F_min/Fref; %%% min division ratio
N_max=F_max/Fref; %%% max division ratio

fm=(1e3+1):1000:(1e9+1);

w=2*pi*fm;
wref=2*pi*Fref;
s=1i*w;
K=1.38065038*((10)^(-23));
T=298;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

phi_p=(pi/3);
x=3.73;
wc=2*pi*(3e5); %%%%%%%%% loop BW
wp=wc*x;
```

```

wz=wc/x;

K_ol=(K_vco*K_phi)/N;

R1=wc/(K_ol*(1-1/x^2));

C1=72e-12;%1/(R1*wz);

C2=C1/(x^2-1);

t1=R1*C1;

t2=(t1*C2)/(C1+C2);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% ATTEN=10;

%wp3=(wref*spur_order)/sqrt(10^((ATTEN*spur_order)/10)-1);

wp3=10*wp; %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% 3rd pole

f3=wp3/(2*pi);

T3=1/wp3;

R3=50e3;

C3=1e-12;%T3/R3;

Z_2nd=(1+s.*t1)/(s.*(C1+C2).*(1+s.*t2));

%Z_2nd=(1+s.*t1)/(s.*(1+s.*t2).*(C1+C2));

Z=(Z_2nd.*(1/(s.*C3)))/(Z_2nd+R3+(1/(s.*C3)));

% Z_2nd=(1+s.*R1*C1)/(s.^2.*C1*C2*R1+s.*(C1+C2));

% Z=(Z_2nd.*(1/(s.*C3)))/(Z_2nd+R3+(1/(s.*C3)));

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

G=(K_vco.*K_phi.*Z)/s;

H_rvr=1/N;

H_ol=(H_rvr).*(G);

H_=(H_ol)/(1+H_ol);

%%

%%

stability Analysis

% %s= tf('s');

%

% Z_2nd=(1+s*t1)/(s*(C1+C2)*(1+s*t2));

% Z=(Z_2nd*(1/(s*C3)))/(Z_2nd+R3+(1/(s*C3)));

%

% G=(K_vco.*K_phi.*Z)/s;

% H_rvr=1/N;

% H_ol=(H_rvr)*(G);

% H_=(H_ol)/(1+H_ol);

%

% figure %% Phase & gain margin plot

% margin(H_ol);

% grid on

% m = allmargin(H_ol);

%%

%% spur level calculation

%%%%%%%%%

spur_order=1;

c=2e-3;

Z_2nd_at_ref=(1-1/x^2)*(wc/wref)*R1*x/spur_order;

spur_level_2nd=20*log10((wc/wref)^2*c*pi*N*x/spur_order^2);

%%%%%%%%%

%

% G=(K_vco.*K_phi*Z)/s;

% H_rvr=1/N;

% H_ol=(H_rvr)*(G);

% H_=(H_ol)/(1+H_ol);

%

% figure

% semilogx(fm,20*log10(abs(H_ol)));

% title(' transfer function')

% xlabel(' freq ')

% ylabel('unitless ')

% hold on

% grid on

%%%%%%%%%
%%%%%%%%%

%%% reference

```

const_ref= 1e-15;

Q_ref= 87.4477e3;

fc_ref=50*((10)^3);

N_ref=(1+(Fref./(2*Q_ref*fm)).^2).*const_ref.*(1+fc_ref./fm);

N_tf_ref= (N).*H_ol./(1+H_ol);

N_ref_out=(N_ref.*((abs(N_tf_ref)).^2))/(10^(-1.8));

```

```

% figure
% semilogx(fm,20*log10(abs(N_tf_ref)));
% title ('reference noise transfer function')
% xlabel(' freq ')
% ylabel('unitless ')
% hold on
% grid on

% figure
% semilogx(fm,10*log10(N_ref));
% title ('reference noise psd with offset freq')
% xlabel(' freq offset ')
% ylabel('psd(dbc/hz)')
% hold on
% grid on
%
% figure
% semilogx(fm,10*log10(N_ref_out));
% title ('reference output noise psd with offset freq')

```

```

% xlabel(' freq offset ')
% ylabel('psd(dbc/hz)')
% hold on
% grid on

% %%%%%%%%%%

% % % % % charge pump

ton=0.01/Fref;
thermal_cp=3e-24;
alpha=ton*Fref;
fc_cp=4e4;

N_cp=(1+((fc_cp)/fm)).*alpha*thermal_cp;

N_tf_cp=(N/K_phi).*(H_ol./(1+H_ol));
N_cp_out=(N_cp.*(abs(N_tf_cp).^2))/(10^(-1.8));

%
% figure
% semilogx(fm,20*log10(abs(N_tf_cp)));
% title ('CP noise transfer function')
% xlabel(' freq ')
% ylabel('unitless ')
% hold on
% grid on

```

```

%
%
% figure
% semilogx(fm,10*log10(abs(N_cp)));
% title ('CP noise psd with offset freq')
% xlabel(' freq offset ')
% ylabel('psd(dbc/hz)')
% hold on
% grid on

% figure
% semilogx(fm,10*log10(N_cp_out));
% title ('CP noise output psd with offset freq')
% xlabel(' freq offset ')
% ylabel('psd(dbc/hz)')
% hold on
% grid on

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Loop filter

%N_tf_filter=(K_vco./s).*(1./(1+H_ol));
N_filter_out=(2*K*T*R3).*(K0./w).^2.*(wp./(2.*pi.*fm)).^2.*(1+((x.*wc)./w).^2.*R1./R3);
%N_filter_out=(2*K*T*R3).*(1+((x.*wc)/(2*pi.*fm)).^2.*R1./R3).*(K0./fm).^2.*(wp./(2*pi.*fm)).^2;

```

```

% N_R3= (4*K*T*R3);
% N_R1= (4*K*T*R1);
% Z4=((R3+1./(s.*C3))./(s.*C1))./(1./(s.*C1)+R3+1./(s.*C3));
% N_tf_R3=((1./(s.*C3))./(Z_2nd+R3+1./(s.*C3))).*(1./(1+H_ol)).*(K_vco./s);
% N_tf_R1=(Z4)./(Z4+R1+1./(s.*C2)).*(1./(s.*C3))./(R3.*1./(s.*C3)).*K_vco./s.*1./(1+H_ol);
% N_R3_out=(N_R3.*((abs(N_tf_R3)).^2))./(10.^(-1.8));
% N_R1_out=(N_R1.*((abs(N_tf_R1)).^2))./(10.^(-1.8));

```

```

%N_filter_out=10*log10(N_R1_out)+10*log10(N_R1_out);

```

```

%
% figure
% semilogx(fm,20*log10(abs(N_tf_R1)));
% title ('R1 noise transfer function')
% xlabel(' freq ')
% ylabel('unitless')
% hold on
% grid on

```

```

%
%
% figure
% semilogx(fm,10*log10(N_R1_out));
% title ('R1 noise output')
% xlabel(' offset freq ')
% ylabel('dbc/hz')
% hold on
% grid on
%

```

```

% figure
% semilogx(fm,20*log10(abs(N_tf_R3)));
% title ('R3 noise transfer function')
% xlabel(' freq ')
% ylabel('unitless')
% hold on
% grid on
%
%
% figure
% semilogx(fm,10*log10(N_R3_out));
% title ('R3 noise output')
% xlabel(' offset freq ')
% ylabel('dbc/hz')
% hold on
% grid on
%

%%%%%%%%%%
                %%% % VCO

Q_vco=10;
fc_vco=1e5;

const_vco= 2e-16;

N_vco=(1+(RFopt./(2*Q_vco*fm)).^2).*const_vco.*(1+fc_vco./fm);

N_tf_vco=(1./(1+H_ol));

N_vco_out=(N_vco.*((abs(N_tf_vco)).^2))/(10^(-1.8));

```

```

%
% figure
% semilogx(fm,20*log10(abs(N_tf_vco)));
% title ('vco noise transfer function')
% xlabel(' freq ')
% ylabel('unitless')
% hold on
% grid on
%
% figure
% semilogx(fm,10*log10(N_vco));
% title ('vco noise psd')
% xlabel(' offset freq ')
% ylabel('psd(dbc/hz)')
% hold on
% grid on

% figure
% semilogx(fm,10*log10(N_vco_out));
% title ('vco noise output')
% xlabel(' offset freq ')
% ylabel('dbc/hz')
% hold on
% grid on

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

fc_div=1e5;          %%% % divider

```

```

thermal=1e-14;

N_div=((1+(fc_div./fm))*thermal);

N_tf_div=(-N).*(H_ol./(1+H_ol));

N_div_out=((N_div).*abs(N_tf_div))/(10^(-1.8));

% figure

% semilogx(fm,20*log10(abs(N_tf_div)));7

% title ('divider noise transfer function')

% xlabel(' freq ')

% ylabel('unitless ')

% hold on

% grid on

%

% figure

% semilogx(fm,10*log10(N_div));

% title ('divider noise psd with offset freq')

% xlabel(' freq offset @ fc=100khz')

% ylabel('psd(dbc/hz)')

% hold on

% grid on

%

% figure

% semilogx(fm,10*log10(N_div_out));

% title ('divider output noise psd with offset freq')

% xlabel(' freq offset @ fc=100khz')

% ylabel('psd(dbc/hz)')

% hold on

% grid on

%

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
                    %%%%%%%%% SDM

sdm_order=3;

N_sdm=((2*pi).^2./(24*Fref)).*(2*sin((pi.*fm)/Fref)).^(2.*(sdm_order-1));

N_tf_sdm=H_;

N_sdm_out=((N_sdm.*abs(N_tf_sdm).^2))/(10^(-1.8));

%

% figure

% semilogx(fm,20*log10(abs(N_tf_sdm)));

% title ('SDM noise transfer function freq')

% xlabel(' freq offset @ fc=100khz')

% ylabel('psd(dbc/hz)')

% hold on

% grid on

%

% figure

% semilogx(fm,10*log10(N_sdm));

% title ('SDM noise psd with offset freq')

% xlabel(' freq offset @ fc=100khz')

% ylabel('psd(dbc/hz)')

% hold on

% grid on

%

%

% figure

```

```

% semilogx(fm,10*log10(N_sdm_out));

% title ('SDM output noise psd with offset freq')

% xlabel(' freq offset @ fc=100khz')

% ylabel('psd(dbc/hz)')

% hold on

% grid on

% %

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% total noise

N_total_out=abs(N_cp_out)+abs(N_vco_out)+abs(N_sdm_out)+abs(N_div_out)+abs(N_ref_out);%+abs(N_filter_o
ut);

%

% figure

% semilogx(fm,10*log10(N_total_out));

% title ('total noise psd with offset freq')

% xlabel(' freq offset @ fc=100khz')

% ylabel('psd(dbc/hz)')

% hold on

% grid on

%

```

III. Appendix C: Verilog-A Codes

Verilog-AMS code for real peak detector

```
// VerilogA for Graduation_Project, PD_Modified, veriloga

`include "constants.vams"
`include "disciplines.vams"

module PD_Modified(in,out);
  input in;
  output out;
  voltage in,out;

  real vmax;
  real decay;

  analog begin
    @(initial_step)  vmax = -inf;
    decay = 0.0001;

    if (V(in) > vmax) begin vmax = V(in); V(out) <+ vmax;
      end
    if (V(in) < vmax) begin vmax = vmax - decay; V(out) <+ vmax; //vmax = V(in);
      end

  end
endmodule
```

Verilog-AMS code for real comparator

```
// VerilogA for Graduation_Project, Comparator_Ideal, veriloga
```

```
`include "constants.vams"
```

```
`include "disciplines.vams"
```

```
module Comparator_Ideal(Vout, Vp, Vn);
```

```
input Vp, Vn;
```

```
output Vout;
```

```
electrical Vout, Vp, Vn;
```

```
parameter real td = 1e-12, tr = 1e-12, tf = 1e-12;
```

```
analog begin
```

```
V(Vout) <+ transition ((V(Vp) > V(Vn)) ? 1.2 : 0, td, tr, tf);
```

```
end
```

```
endmodule
```

Verilog-AMS code for Thermometer to binary in Flash A/D

```
// VerilogA for Graduation_Project, Flash_ADC_Therometer, veriloga
`include "constants.vams"
`include "disciplines.vams"
module
Flash_ADC_Therometer_1(O1,O2,O3,O4,O5,O6,O7,O8,O9,O10,O11,O12,O13,O14,O15,O16,O17,O18,O19,O20,O21,
O22,O23,O24,O25,O26,O27,O28,O29,O30,O31,A1,A2,A3,A4,A5);
input
O1,O2,O3,O4,O5,O6,O7,O8,O9,O10,O11,O12,O13,O14,O15,O16,O17,O18,O19,O20,O21,O22,O23,O24,O25,O26,O27,
O28,O29,O30,O31;
output A1,A2,A3,A4,A5;
electrical
O1,O2,O3,O4,O5,O6,O7,O8,O9,O10,O11,O12,O13,O14,O15,O16,O17,O18,O19,O20,O21,O22,O23,O24,O25,O26,O27,
O28,O29,O30,O31,A1,A2,A3,A4,A5;
parameter real tr = 1e-12;
parameter real tf = 1e-12;
parameter real td = 1e-12;
parameter real high = 1.2;
parameter real low = 0;
real A11,A22,A33,A44,A55;
real OO1,OO2,OO3,OO4,OO5,OO6,OO7,OO8,OO9,OO10,OO11,OO12,OO13,OO14,OO15;
real OO16,OO17,OO18,OO19,OO20,OO21,OO22,OO23,OO24,OO25,OO26,OO27,OO28,OO29,OO30,OO31,x;
analog begin
OO1 = (V(O1) > 0) ? 1.2 : 0;
OO2 = (V(O2) > 0) ? 1.2 : 0;
OO3 = (V(O3) > 0) ? 1.2 : 0;
OO4 = (V(O4) > 0) ? 1.2 : 0;
OO5 = (V(O5) > 0) ? 1.2 : 0;
OO6 = (V(O6) > 0) ? 1.2 : 0;
OO7 = (V(O7) > 0) ? 1.2 : 0;
OO8 = (V(O8) > 0) ? 1.2 : 0;
OO9 = (V(O9) > 0) ? 1.2 : 0;
OO10 = (V(O10) > 0) ? 1.2 : 0;
OO11 = (V(O11) > 0) ? 1.2 : 0;
OO12 = (V(O12) > 0) ? 1.2 : 0;
OO13 = (V(O13) > 0) ? 1.2 : 0;
OO14 = (V(O14) > 0) ? 1.2 : 0;
OO15 = (V(O15) > 0) ? 1.2 : 0;
OO16 = (V(O16) > 0) ? 1.2 : 0;
OO17 = (V(O17) > 0) ? 1.2 : 0;
OO18 = (V(O18) > 0) ? 1.2 : 0;
OO19 = (V(O19) > 0) ? 1.2 : 0;
OO20 = (V(O20) > 0) ? 1.2 : 0;
OO21 = (V(O21) > 0) ? 1.2 : 0;
OO22 = (V(O22) > 0) ? 1.2 : 0;
OO23 = (V(O23) > 0) ? 1.2 : 0;
OO24 = (V(O24) > 0) ? 1.2 : 0;
OO25 = (V(O25) > 0) ? 1.2 : 0;
OO26 = (V(O26) > 0) ? 1.2 : 0;
OO27 = (V(O27) > 0) ? 1.2 : 0;
OO28 = (V(O28) > 0) ? 1.2 : 0;
OO29 = (V(O29) > 0) ? 1.2 : 0;
OO30 = (V(O30) > 0) ? 1.2 : 0;
OO31 = (V(O31) > 0) ? 1.2 : 0;
if( (OO1 == 0) && (OO2 == 0) && (OO3 == 0) && (OO4 == 0) && (OO5 == 0) && (OO6 == 0) && (OO7 == 0) &&
(OO8 == 0) && (OO9 == 0) && (OO10 == 0) && (OO11 == 0) && (OO12 == 0) && (OO13 == 0) && (OO14 == 0)
&& (OO15 == 0) && (OO16 == 0) && (OO17 == 0) && (OO18 == 0) && (OO19 == 0) && (OO20 == 0) && (OO21
== 0) && (OO22 == 0) && (OO23 == 0) && (OO24 == 0) && (OO25 == 0) && (OO26 == 0) && (OO27 == 0) &&
(OO28 == 0) && (OO29 == 0) && (OO30 == 0) && (OO31 == 0) ) begin
A11 = low; A22 = low; A33 = low; A44 = low; A55 = low;
```


Verilog-AMS code for digital control

```
// VerilogA for Graduation_Project, Digital_Control, veriloga_N
`include "constants.vams"
`include "disciplines.vams"
module Digital_Control(A1,A2,A3,A4,A5,D1,D2,D3,D4,D5,D6,D7,D8);
input A1, A2, A3, A4, A5;
output D1, D2, D3, D4, D5, D6, D7, D8;
electrical A1, A2, A3, A4, A5, D1, D2, D3, D4, D5, D6, D7, D8;
parameter real tr = 1e-12;
parameter real tf = 1e-12;
parameter real td = 1e-12;
parameter real high = 1.2;
parameter real low = 0;
real AA1, AA2, AA3, AA4, AA5;
real CC1, CC2, CC3, CC4, CC5, CC6, CC7, CC8;
analog begin
AA1 = (V(A1) > 0) ? 1.2 : 0;
AA2 = (V(A2) > 0) ? 1.2 : 0;
AA3 = (V(A3) > 0) ? 1.2 : 0;
AA4 = (V(A4) > 0) ? 1.2 : 0;
AA5 = (V(A5) > 0) ? 1.2 : 0;
//3.12m
//56
if( (AA1 == 0) && (AA2 == 0) && (AA3 == 0) && (AA4 == 0) && (AA5 == 0) ) begin
CC1 = high; CC2 = low; CC3 = low; CC4 = low; CC5 = low;CC6 = high;CC7 = low;CC8 = low;
end
//6.2m
//54
else if( (AA1 == 1.2) && (AA2 == 0) && (AA3 == 0) && (AA4 == 0) && (AA5 == 0) ) begin
CC1 = high; CC2 = low; CC3 = low; CC4 = high; CC5 = low;CC6 = low;CC7 = low;CC8 = low;
end
//9.37m
//52
else if( (AA1 == 0) && (AA2 == 1.2) && (AA3 == 0) && (AA4 == 0) && (AA5 == 0) ) begin
CC1 = high; CC2 = low; CC3 = low; CC4 = low; CC5 = low;CC6 = low;CC7 = low;CC8 = low;
end
//12.4m
//50
else if( (AA1 == 1.2) && (AA2 == 1.2) && (AA3 == 0) && (AA4 == 0) && (AA5 == 0) ) begin
CC1 = low; CC2 = low; CC3 = high; CC4 = low; CC5 = low;CC6 = high;CC7 = low;CC8 = low;
end
//15.62m
//48
else if( (AA1 == 0) && (AA2 == 0) && (AA3 == 1.2) && (AA4 == 0) && (AA5 == 0) ) begin
CC1 = low; CC2 = low; CC3 = high; CC4 = low; CC5 = high;CC6 = low;CC7 = low;CC8 = low;
end
//18.7m
//46
```

```

else if( (AA1 == 1.2) && (AA2 == 0) && (AA3 == 1.2) && (AA4 == 0) && (AA5 == 0) ) begin
CC1 = low; CC2 = low; CC3 = high; CC4 = low; CC5 = high;CC6 = low;CC7 = low;CC8 = low;
end
//21.87m
//44
else if( (AA1 == 0) && (AA2 == 1.2) && (AA3 == 1.2) && (AA4 == 0) && (AA5 == 0) ) begin
CC1 = low; CC2 = low; CC3 = high; CC4 = high; CC5 = low;CC6 = low;CC7 = low;CC8 = low;
end
//24.9m
//42
else if( (AA1 == 1.2) && (AA2 == 1.2) && (AA3 == 1.2) && (AA4 == 0) && (AA5 == 0) ) begin
CC1 = low; CC2 = low; CC3 = high; CC4 = high; CC5 = low;CC6 = low;CC7 = low;CC8 = low;
end
//28.12m
//40
else if( (AA1 == 0) && (AA2 == 0) && (AA3 == 0) && (AA4 == 1.2) && (AA5 == 0) ) begin
CC1 = low; CC2 = low; CC3 = high; CC4 = low; CC5 = low;CC6 = low;CC7 = low;CC8 = low;
end
//31.2m
//38
else if( (AA1 == 1.2) && (AA2 == 0) && (AA3 == 0) && (AA4 == 1.2) && (AA5 == 0) ) begin
CC1 = low; CC2 = low; CC3 = high; CC4 = low; CC5 = low;CC6 = low;CC7 = low;CC8 = low;
end
//34.37m
//36
else if( (AA1 == 0) && (AA2 == 1.2) && (AA3 == 0) && (AA4 == 1.2) && (AA5 == 0) ) begin
CC1 = low; CC2 = high; CC3 = low; CC4 = low; CC5 = low;CC6 = low;CC7 = low;CC8 = high;
end
//37.4m
//34
else if( (AA1 == 1.2) && (AA2 == 1.2) && (AA3 == 0) && (AA4 == 1.2) && (AA5 == 0) ) begin
CC1 = low; CC2 = high; CC3 = low; CC4 = low; CC5 = low;CC6 = low;CC7 = high;CC8 = low;
end
//40.62m
//32
else if( (AA1 == 0) && (AA2 == 0) && (AA3 == 1.2) && (AA4 == 1.2) && (AA5 == 0) ) begin
CC1 = low; CC2 = high; CC3 = low; CC4 = low; CC5 = low;CC6 = low;CC7 = high;CC8 = low;
end
//43.7m
//30
else if( (AA1 == 1.2) && (AA2 == 0) && (AA3 == 1.2) && (AA4 == 1.2) && (AA5 == 0) ) begin
CC1 = low; CC2 = high; CC3 = low; CC4 = low; CC5 = low;CC6 = low;CC7 = high;CC8 = low;
end
//46.87m
//28
else if( (AA1 == 0) && (AA2 == 1.2) && (AA3 == 1.2) && (AA4 == 1.2) && (AA5 == 0) ) begin
CC1 = low; CC2 = high; CC3 = low; CC4 = low; CC5 = low;CC6 = high;CC7 = low;CC8 = low;
end
//49.9m
//26

```

```

else if( (AA1 == 1.2) && (AA2 == 1.2) && (AA3 == 1.2) && (AA4 == 1.2) && (AA5 == 0) )
begin
CC1 = low; CC2 = high; CC3 = low; CC4 = low; CC5 = low;CC6 = high;CC7 = low;CC8 = low;
end
//53.12m
//24
else if( (AA1 == 0) && (AA2 == 0) && (AA3 == 0) && (AA4 == 0) && (AA5 == 1.2) ) begin
CC1 = low; CC2 = high; CC3 = low; CC4 = low; CC5 = low;CC6 = high;CC7 = low;CC8 = low;
end
//56.2m
//22
else if( (AA1 == 1.2) && (AA2 == 0) && (AA3 == 0) && (AA4 == 0) && (AA5 == 1.2) ) begin
CC1 = low; CC2 = high; CC3 = low; CC4 = low; CC5 = high;CC6 = low;CC7 = low;CC8 = low;
end
//59.3m
//20
else if( (AA1 == 0) && (AA2 == 1.2) && (AA3 == 0) && (AA4 == 0) && (AA5 == 1.2) ) begin
CC1 = low; CC2 = high; CC3 = low; CC4 = low; CC5 = high;CC6 = low;CC7 = low;CC8 = low;
end
//62.4m
//18
else if( (AA1 == 1.2) && (AA2 == 1.2) && (AA3 == 0) && (AA4 == 0) && (AA5 == 1.2) ) begin
CC1 = low; CC2 = high; CC3 = low; CC4 = low; CC5 = high;CC6 = low;CC7 = low;CC8 = low;
end
//65.62m
//16
else if( (AA1 == 0) && (AA2 == 0) && (AA3 == 1.2) && (AA4 == 0) && (AA5 == 1.2) ) begin
CC1 = low; CC2 = high; CC3 = low; CC4 = low; CC5 = high;CC6 = low;CC7 = low;CC8 = low;
end
//68.7m
//14
else if( (AA1 == 1.2) && (AA2 == 0) && (AA3 == 1.2) && (AA4 == 0) && (AA5 == 1.2) ) begin
CC1 = low; CC2 = high; CC3 = low; CC4 = low; CC5 = high;CC6 = low;CC7 = low;CC8 = low;
end
//71.87m
//12
else if( (AA1 == 0) && (AA2 == 1.2) && (AA3 == 1.2) && (AA4 == 0) && (AA5 == 1.2) ) begin
CC1 = low; CC2 = high; CC3 = low; CC4 = high; CC5 = low;CC6 = low;CC7 = low;CC8 = low;
end
//74.9m
//10
else if( (AA1 == 1.2) && (AA2 == 1.2) && (AA3 == 1.2) && (AA4 == 0) && (AA5 == 1.2) )
begin
CC1 = low; CC2 = high; CC3 = low; CC4 = high; CC5 = low;CC6 = low;CC7 = low;CC8 = low;
end
//78.12m
//8
else if( (AA1 == 0) && (AA2 == 0) && (AA3 == 0) && (AA4 == 1.2) && (AA5 == 1.2) ) begin
CC1 = low; CC2 = high; CC3 = low; CC4 = high; CC5 = low;CC6 = low;CC7 = low;CC8 = low;
end
end

```

```

//81.2m
//6
else if( (AA1 == 1.2) && (AA2 == 0) && (AA3 == 0) && (AA4 == 1.2) && (AA5 == 1.2) ) begin
CC1 = low; CC2 = high; CC3 = low; CC4 = high; CC5 = low;CC6 = low;CC7 = low;CC8 = low;
end
//84.37m
//4
else if( (AA1 == 0) && (AA2 == 1.2) && (AA3 == 0) && (AA4 == 1.2) && (AA5 == 1.2) ) begin
CC1 = low; CC2 = high; CC3 = low; CC4 = high; CC5 = low;CC6 = low;CC7 = low;CC8 = low;
end
//87.4m
//2
else if( (AA1 == 1.2) && (AA2 == 1.2) && (AA3 == 0) && (AA4 == 1.2) && (AA5 == 1.2) )
begin
CC1 = low; CC2 = high; CC3 = low; CC4 = high; CC5 = low;CC6 = low;CC7 = low;CC8 = low;
end
//90.62m
//0
else if( (AA1 == 0) && (AA2 == 0) && (AA3 == 1.2) && (AA4 == 1.2) && (AA5 == 1.2) ) begin
CC1 = low; CC2 = high; CC3 = low; CC4 = high; CC5 = low;CC6 = low;CC7 = low;CC8 = low;
end
//93.7m
//0
else if( (AA1 == 1.2) && (AA2 == 0) && (AA3 == 1.2) && (AA4 == 1.2) && (AA5 == 1.2) )
begin
CC1 = low; CC2 = high; CC3 = low; CC4 = high; CC5 = low;CC6 = low;CC7 = low;CC8 = low;
end
//96.87m
//0
else if( (AA1 == 0) && (AA2 == 1.2) && (AA3 == 1.2) && (AA4 == 1.2) && (AA5 == 1.2) )
begin
CC1 = low; CC2 = high; CC3 = low; CC4 = low; CC5 = low;CC6 = low;CC7 = low;CC8 = low;
end
//99.9m
//0
else if( (AA1 == 1.2) && (AA2 == 1.2) && (AA3 == 1.2) && (AA4 == 1.2) && (AA5 == 1.2) )
begin
CC1 = low; CC2 = high; CC3 = low; CC4 = low; CC5 = low;CC6 = low;CC7 = low;CC8 = low;
end
V(D1) <+ transition (CC1,td,tr,tf);
V(D2) <+ transition (CC2,td,tr,tf);
V(D3) <+ transition (CC3,td,tr,tf);
V(D4) <+ transition (CC4,td,tr,tf);
V(D5) <+ transition (CC5,td,tr,tf);
V(D6) <+ transition (CC6,td,tr,tf);
V(D7) <+ transition (CC7,td,tr,tf);
V(D8) <+ transition (CC8,td,tr,tf);
end
endmodule

```

ADC Verilog – A

```
// VerilogA for try, sarlogic8bit, veriloga

`include "constants.vams"

`include "disciplines.vams"

module

sarlogic8bit(comp,clk,clks,s1,s2,sp1,sp2,sp3,sp4,sp5,sp6,sp7,sp8,sn1,sn2,sn3,sn4,sn5,sn6,sn7,sn8
,eoc,b0,b1,b2,b3,b4,
b5,b6,b7);

input comp,clk,clks;

output
s1,s2,sp1,sp2,sp3,sp4,sp5,sp6,sp7,sp8,sn1,sn2,sn3,sn4,sn5,sn6,sn7,sn8,eoc,b0,b1,b2,b3,b4,b5,b6,
b7;

electrical

comp,clk,clks,s1,s2,sp1,sp2,sp3,sp4,sp5,sp6,sp7,sp8,sn1,sn2,sn3,sn4,sn5,sn6,sn7,sn8,eoc,b0,b1,b
2,b3,b4,b5,b6,b7;

parameter real Vlow = 0 ;

parameter real Vhigh = 1.2 ;

integer counter,bb0,bb1,bb2,bb3,bb4,bb5,bb6,bb7;

real
ss1,ss2,spp1,spp2,spp3,spp4,spp5,spp6,spp7,spp8,snn1,snn2,snn3,snn4,snn5,snn6,snn7,snn8,eoc
;

analog

begin

@(initial_step) begin

counter=0;

ss1=1.2; ss2=1.2; spp1=1.2; spp2=1.2; spp3=1.2; spp4=1.2; spp5=1.2; spp6=1.2; spp7=1.2;
spp8=1.2; snn1=1.2;

snn2=1.2; snn3=1.2; snn4=1.2; snn5=1.2; snn6=1.2; snn7=1.2; snn8=1.2; eoc=0;

end
```

```

@(cross(V(clk)-1,+1)) begin
counter=counter+1;
if (counter==1) begin
ss1=0; ss2=0;
end
if (counter==2) begin
if (V(comp)>Vhigh/2) begin
spp1=0; bb7=1.2;
end
else if (V(comp)<Vhigh/2) begin
snn1=0; bb7=0;
End
end
if (counter==3) begin
if (V(comp)>Vhigh/2) begin
spp2=0; bb6=1.2;
end else if (V(comp)<Vhigh/2) begin
snn2=0; bb6=0;
End
end
if (counter==4) begin
if (V(comp)>Vhigh/2) begin
spp3=0; bb5=1.2;
end else if (V(comp)<Vhigh/2) begin
snn3=0; bb5=0;
End
end
end

```

```
if (counter==5) begin
if (V(comp)>Vhigh/2) begin
spp4=0; bb4=1.2;
end else if (V(comp)<Vhigh/2) begin
snn4=0; bb4=0;
End
end
if (counter==6) begin
if (V(comp)>Vhigh/2) begin
spp5=0; bb3=1.2;
end else if (V(comp)<Vhigh/2) begin
snn5=0; bb3=0;
End
end
if (counter==7) begin
if (V(comp)>Vhigh/2) begin
spp6=0; bb2=1.2;
end else if (V(comp)<Vhigh/2) begin
snn6=0; bb2=0;
End
end
if (counter==8) begin
if (V(comp)>Vhigh/2) begin
spp7=0; bb1=1.2;
end else if (V(comp)<Vhigh/2) begin
snn7=0; bb1=0;
End
```

```

end

if (counter==9) begin
if (V(comp)>Vhigh/2) begin
bb0=1.2;

end else if (V(comp)<Vhigh/2) begin
bb0=0;

end

counter=0; eoc=0;

ss1=1.2; ss2=1.2; spp1=1.2; spp2=1.2; spp3=1.2; spp4=1.2; spp5=1.2; spp6=1.2; spp7=1.2;
spp8=1.2; snn1=1.2;

snn2=1.2; snn3=1.2; snn4=1.2; snn5=1.2; snn6=1.2; snn7=1.2; snn8=1.2; eoc=1.2;

end

V(s1)<+transition(ss1); V(s2)<+transition(ss2); V(sp1)<+transition(spp1);
V(sp2)<+transition(spp2);

V(sp3)<+transition(spp3);

V(sp4)<+transition(spp4); V(sp5)<+transition(spp5); V(sp6)<+transition(spp6);
V(sp7)<+transition(spp7);

V(sn1)<+transition(snn1); V(sn2)<+transition(snn2); V(sn3)<+transition(snn3);
V(sn4)<+transition(snn4);

V(sn5)<+transition(snn5);

V(sn6)<+transition(snn6); V(sn7)<+transition(snn7);

V(eoc)<+transition(eoc); V(b0)<+transition(bb0); V(b1)<+transition(bb1);
V(b2)<+transition(bb2);

V(b3)<+transition(bb3);          V(b4)<+transition(bb4); V(b5)<+transition(bb5);
V(b6)<+transition(bb6);

V(b7)<+transition(bb7);

end

endmodule

```

IV. Appendix D: VHDL Codes

ADC VHDL code

```
library IEEE;

use IEEE.STD_LOGIC_1164.ALL;

use IEEE.STD_LOGIC_ARITH.ALL;

use IEEE.STD_LOGIC_UNSIGNED.ALL;

---- Uncomment the following library declaration if instantiating
---- any Xilinx primitives in this code.

--library UNISIM;

--use UNISIM.VComponents.all;

entity sar is

port( enable , clock , comp , reset : in std_logic;

      Svip , Svin , Sp1 , Sp2 , Sp3 , Sp4 , Sp5 , Sp6 , Sp7 , Sp8 :out std_logic;

      Sn1 , Sn2 , Sn3 , Sn4 , Sn5 , Sn6 , Sn7 , Sn8 :out std_logic;

      eoc : out std_logic);

end sar;

architecture Behavioral of sar is

type statetype is (state0 , state1 , state2 , state3 , state4 , state5 , state6 , state7 , state8);

signal state : statetype;

begin

process(clock,reset,enable)

begin

if reset = '1' then

state<= state0;

Svip<='1'; Svin<='1'; Sp1<='1'; Sp2<='1'; Sp3<='1'; Sp4<='1'; Sp5<='1'; Sp6<='1'; Sp7<='1';
Sp8<='1';
```

```

Sn1<='1'; Sn2<='1'; Sn3<='1'; Sn4<='1'; Sn5<='1'; Sn6<='1'; Sn7<='1'; Sn8<='1';
eoc<='0';
elseif(clock'event and clock='1' and enable='1') then
case state is
when state0=> state <= state1;
Svip<='0'; Svin<='0'; Sp1<='1'; Sp2<='1'; Sp3<='1'; Sp4<='1'; Sp5<='1'; Sp6<='1'; Sp7<='1';
Sp8<='1';
Sn1<='1'; Sn2<='1'; Sn3<='1'; Sn4<='1'; Sn5<='1'; Sn6<='1'; Sn7<='1'; Sn8<='1';
eoc<='0';
when state1=> state <= state2;
if comp='1' then
Sp1<='0';
elseif comp='0' then
Sn1<='0';
end if;
when state2=> state <= state3;
if comp='1' then
Sp2<='0';
elseif comp='0' then
Sn2<='0';
end if;
when state3=> state <= state4;
if comp='1' then
Sp3<='0';
elseif comp='0' then
Sn3<='0';
end if;

```

```
when state4=> state <= state5;
if comp='1' then
Sp4<='0';
elseif comp='0' then
Sn4<='0';
end if;
when state5=> state <= state6;
if comp='1' then
Sp5<='0';
elseif comp='0' then
Sn5<='0';
end if;
when state6=> state <= state7;
if comp='1' then
Sp6<='0';
elseif comp='0' then
Sn6<='0';
end if;
when state7=> state <= state8;
if comp='1' then
Sp7<='0';
elseif comp='0' then
Sn7<='0';
end if;
when state8=> state <= state0;
if comp='1' then
Sp8<='0';
```

```

elseif comp='0' then
Sn8<='0';
end if;
eoc<='1';
end case;
end if;
if enable = '0' then
Svip<='1'; Svin<='1'; Sp1<='1'; Sp2<='1'; Sp3<='1'; Sp4<='1'; Sp5<='1'; Sp6<='1'; Sp7<='1';
Sp8<='1';
Sn1<='1'; Sn2<='1'; Sn3<='1'; Sn4<='1'; Sn5<='1'; Sn6<='1'; Sn7<='1'; Sn8<='1';
eoc<='0';
end if;
end process;
end Behavioral;

```

VCO : Verilog A code for frequency calibrator

```

`include "constants.vams"
`include "disciplines.vams"
module VCO_Calibrator(Vdd,GND,Div,Ref,b);
input Div,Ref,Vdd,GND;
electrical Div,Ref,Vdd,GND;
output [0:3] b;
electrical [0:3] b;

real temp[0:3];
real Diff;
integer i,j,BitNo,CalEn;
parameter real Threshold=0.6;
parameter integer MaxCount=256;
analog begin
@(initial_step)
begin
i=0;
j=0;
temp[0]=V(GND);
temp[1]=V(GND);
temp[2]=V(GND);
temp[3]=V(Vdd);

```

```

Diff=1;
BitNo=4;
end
@(timer(70n))
CalEn=1;
@(cross(V(Ref)-Threshold,+1))
if(CalEn==1)
begin
if(i<MaxCount)
i=i+1;
end
@(cross(V(Div)-Threshold,+1))
if(CalEn==1)
begin
if(i<MaxCount)
begin
j=j+1;
end
end
if(CalEn==1)
begin
if(i>=MaxCount)
begin
Diff=(i>j)? V(GND):V(Vdd);
i=0;
j=0;
BitNo=BitNo-1;
end
end
if(CalEn==1)
begin
if(BitNo==3)
begin
temp[3]=(Diff>Threshold)? V(Vdd):V(GND);
temp[2]=V(Vdd);
end
else if(BitNo==2)
begin
temp[2]=(Diff>Threshold)? V(Vdd):V(GND);
temp[1]=V(Vdd);
end
else if(BitNo==1)
begin
temp[1]=(Diff>Threshold)? V(Vdd):V(GND);
temp[0]=V(Vdd);
end
else if(BitNo==0)
begin
temp[0]=(Diff>Threshold)? V(Vdd):V(GND);
CalEn=0;
BitNo=4;
i=0;

```

```
j=0;  
end  
end  
V(b[0]) <+ temp[0];  
V(b[1]) <+ temp[1];  
V(b[2]) <+ temp[2];  
V(b[3]) <+ temp[3];  
end  
endmodule
```