

2006

Digital Channelized Wide Band Receiver Implemented with a Systolic Array of Multi-Rate FIR Filters

David M. Rodney
Wright State University

Follow this and additional works at: http://corescholar.libraries.wright.edu/etd_all



Part of the [Electrical and Computer Engineering Commons](#)

Repository Citation

Rodney, David M., "Digital Channelized Wide Band Receiver Implemented with a Systolic Array of Multi-Rate FIR Filters" (2006).
Browse all Theses and Dissertations. Paper 26.

This Thesis is brought to you for free and open access by the Theses and Dissertations at CORE Scholar. It has been accepted for inclusion in Browse all Theses and Dissertations by an authorized administrator of CORE Scholar. For more information, please contact corescholar@www.libraries.wright.edu.

DIGITAL CHANNELIZED WIDE BAND RECEIVER IMPLEMENTED
WITH A SYSTOLIC ARRAY OF MULTI-RATE FIR FILTERS

A thesis submitted in partial fulfillment
of the requirements for the degree of
Master of Science in Engineering

By

DAVID M. RODNEY
B.E., City College of New York, 2000

2006

Wright State University

WRIGHT STATE UNIVERSITY
SCHOOL OF GRADUATE STUDIES

June 9, 2006

I HEREBY RECOMMEND THAT THE THESIS PREPARED UNDER MY SUPERVISION BY David M. Rodney ENTITLED Digital Channelized Wide Band Receiver Implemented with a Systolic Array of Multi-Rate FIR Filters BE ACCEPTED IN PARTIAL FULFILLMRNT OF THE REQUIREMENTS FOR THE DEGREE OF Master of Science in Engineering.

Raymond Siferd, Ph.D.
Thesis Director

Fred Garber, Ph.D.
Department Chair

Committee on
Final Examination

Raymond Siferd, Ph.D.

Chien-In Henry Chen, Ph.D.

Marty Emmert, Ph.D.

Joseph F. Thomas, Jr., Ph.D.
Dean of the School of Graduate studies

ABSTRACT

David, M. Rodney, M.S.Egr., Department of Electrical Engineering, Wright State University, 2006. Digital Channelized Wide Band Receiver Implemented with a Systolic Array of Multi-Rate FIR Filters.

This thesis proposes the use of a Systolic Array of Multi-Rate FIR Filters to improve performance by eliminating the requirement of the FFT and De-Multiplexer associated with the conventional receiver while achieving the same functionality. The FFT is a major bottle neck for improving system performance for the conventional DCWBR because many complex multiplications and additions are required. The proposed new architecture is designed and evaluated in MATLAB to illustrate its viability. Two approaches for improved channel arbitration are accessed in MATLAB, namely, channel bin's rms comparison and parallelism of the Systolic Array Multi-Rate FIR Filters.

The FIR filters (high-pass & low-pass) were successfully designed with Cadence tools using 0.13um technology and are fully functional at clock frequencies up to 1.8 GHz. The limitation of computing resources/verification tools prevented the simulation of the entire array of Multi-Rate filters as proposed. Nevertheless, a two tier tree of Multi-Rate FIR filters demonstrated channelization in cadence (simulations which can be completed within the constraints of computing facilities) and is consistent with those of MATLAB; thus, proving the viability of using the Systolic Array of Multi-Rate FIR Filters as a potential architecture for improving performance of DCWBR.

Table of Contents

1	Introduction	1
1.1	Motivation	1
1.2	Thesis Objective	1
1.3	Thesis Organization	2
2	Background	3
2.1.0	Number Representation	3
2.1.1	Unsigned Integer	3
2.1.2	Signed Magnitude Number	4
2.1.3	Two's Compliment	4
2.1.4	Conversion of Decimal Fractions to Binary	6
2.2.0	Analog-to-Digital Conversion	6
2.2.1	Sampling	7
2.2.2	Quantization of sinusoidal signals	8
2.3.0	Digital Channelized Wide Band Receiver	9
3	Finite Impulse Response (FIR) Filters	10
3.1.0	FIR Filters	10
3.1.1	Filter Design Coefficient Generation	13
4	Digital channelized Wide Band Receivers (DCWBR)	15
4.1.0	Conventional DCWBR Architecture	15
4.1.1	Proposed Systolic Array DCWBR Architecture	17
4.1.2	MATLAB Performance Assessment of the systolic Array DCWBR Architecture	20
4.1.3	Systolic Array DCWBR Channel Arbitration Using Bin's rms Magnitude	25
4.1.4	Example of Incorrect channel Arbitration Using Bin's rms Comparison	29

4.1.5	Parallel systolic Array DCWBR for Increased Channel Arbitration.....	31
5	Circuit Designs of basic Building Blocks.....	34
5.1.0	D Flip-Flop (DFF).....	34
5.1.1	The Half Adder.....	36
5.1.2	The Exclusive OR Gate (XOR).....	38
5.1.3	The Multiplexer (MUX).....	40
5.1.4	The AND Gate.....	42
5.1.5	The Full Adder (.).....	43
5.1.6	The 7-bit Adder.....	46
5.1.7	The 7-bit 2's Compliment.....	49
5.1.8	6bit Multiplier.....	53
5.1.9	FIR Filter Design.....	57
5.2.0	The Clock Tree Design.....	59
6	FIR Filter Validation.....	63
6.1.0	FIR Filter Validation for a 100MHz Sinusoidal Input.....	63
6.1.1	FIR Filter Validation for a 300MHz Sinusoidal Input.....	67
6.1.2	FIR Filter Validation for a 450MHz Sinusoidal Input.....	70
6.1.3	FIR Filter Validation for a 600MHz Sinusoidal Input.....	74
6.2.0	Customization of Multipliers to Reduce Power Consumption.....	78
6.2.1	Revalidation of the FIR Filters for a 600MHz Input Sinusoidal.....	81
6.2.2	Revalidation of the FIR Filters for a 450MHz Input Sinusoidal.....	83
6.2.3	Revalidation of the FIR Filters for a 300MHz Input Sinusoidal.....	85
7	Multi-Rate FIR Filter Operations.....	87
	Two Tier FIR Filter Validation.....	87

8	Conclusion and Future Work.....	96
8.1	Conclusion on the work presented.....	96
8.2	Limitations.....	96
8.3	Suggestions for future Work.....	97

Appendices

A	MATLAB and SIMULINK Designs.....	99
A1.1.0	MATLAB Program for Generating FIR Filter Coefficients.....	99
A1.1.1	Simulink Design for MATLAB Simulation.....	100
A1.1.2	MATLAB Program for Channel Arbitration Using Bin's rms Comparison.....	103
B	Additional Cadence Simulations.....	104
B1.1.0	The Full-Adder.....	104
B1.1.1	Filter Veification.....	107

References.....	111
------------------------	------------

List of Figures

Figure	Page
2.1 Spectra of a signal sampled at (a) Nyquist rate (b) above Nyquist rate.....	7
2.2 Sampling and quantization of a sinusoidal signal.....	8
2.3 Main blocks of a wide-band digital receiver.....	9
3.1 Direct-form realization of FIR system.....	10
3.2 FIR filter, taking advantage of symmetry to reduce multiplication.....	11
3.3 Magnitude characteristics of physically realizable filters.....	12
3.4 Frequency response of low-pass and high-pass FIR filter.....	13
4.1 Conventional DCWBR Architecture.....	15
4.2 Channelized receiver magnitude frequency response.....	16
4.3 Proposed Systolic Array DCWBR Architecture.....	17
4.4a Frequency magnitude response of first tier systolic array of filters.....	18
4.4b Frequency magnitude response of second tier systolic array of filters.....	18
4.4c Frequency magnitude response of third tier systolic array of filters.....	19
4.5 Expected frequency magnitude response for figure A3.....	20
4.6 Frequency response to single sinusoidal input across frequency spectrum.....	22
4.7 Simultaneous detection of two input frequencies for systolic array DCWBR.....	23
4.8 Systolic array detection of a 500MHz pulse input.....	24
4.9 Systolic array detection of a 500 and 700MHz pulsed input.....	24
4.10 Frequency bin arbitration using rms comparison for single input frequency...	27
4.11 Single pulsed input frequency arbitrated using bin's rms comparison.....	27
4.12 Channel frequency bin arbitration of two inputs using bin's rms comparison	28
4.13 Systolic array DCWBR spectrum for a weak and strong input signals.....	29
4.14 Incorrect bin arbitration using rms comparison for systolic array DCWBR...	30
4.15 DCWBR with three Systolic Array Multi-Rate FIR Filter Banks.....	31
4.16 Reliable channel arbitration using parallelism of systolic array DCWBR.....	32

5.1	The DFF design.....	34
5.1a	Setup time requirement of the DFF.....	35
5.1b	Hold time requirement of the DFF.....	35
5.2	The half adder.....	36
5.2a	Half adder verification.....	37
5.3	The XOR gate design.....	38
5.3a	XOR gate verification.....	39
5.4	The 2:1 multiplexer design.....	40
5.4a	Verification of the multiplexer.....	41
5.5	Static CMOS AND gate.....	42
5.5a	Verification of AND gate.....	43
5.6	The Designed full adder.....	44
5.6a	Verification of the full adder.....	45
5.7	The 7bit adder.....	46
5.7a	I & II A and B input vectors respectively.....	47
5.7b	Correct output waveform for 7-bit adder for A & B vectors.....	48
5.8	7-bit 2's compliment design.....	49
5.8a	Random input vectors to validate the 2's compliment circuit.....	50
5.8b	Correct output waveform from 7bit 2's compliment circuit.....	51
5.8c	The zero pass circuit.....	52
5.9	6-bit 2-stage pipelined Wallace tree multiplier.....	54
5.9a	The 4-bit merging adder of the 6-bit multiplier.....	54
5.9b	Clocked A inputs used for validating the multiplier; B inputs are all 1s.....	55
5.9c	Correct simulation results of the 6-bit Multiplier.....	56
5.10	Design of -0 sign bit converted to positive.....	57
5.10a	15 tap FIR filter design.....	58
5.11	The FIR filter design with the clock node labeled.....	59
5.12a	Clock nodes showing same phase and very little skew.....	60
5.12b	Clock nodes showing same phase and very little skew.....	61
5.12c	Clock nodes showing same phase and very little skew.....	61
5.12d	Clock nodes showing same phase and very little skew.....	62

6.1a	Digitized 100MHz sinusoidal input.....	64
6.1b	MATLAB low-pass FIR filter response to 100MHz sinusoidal input.....	64
6.1c	MATLAB high-pass FIR filter response to 100MHz sinusoidal input.....	64
6.1d	Cadence low-pass FIR filter response to 100MHz sinusoidal input.....	65
6.1e	Cadence high-pass FIR filter response to 100MHz sinusoidal input.....	66
6.2a	Digitized 300MHz sinusoidal.....	67
6.2b	MATLAB low-pass FIR filter response to 100MHz sinusoidal input.....	67
6.2c	MATLAB high-pass FIR filter response to 300MHz sinusoidal input.....	68
6.2d	Cadence low-pass FIR filter response to 300MHz sinusoidal input.....	68
6.2e	Cadence high-pass FIR filter response to 300MHz sinusoidal input.....	69
6.3a	Digitized 450MHz sinusoidal.....	71
6.3b	MATLAB low-pass FIR filter response to 450MHz sinusoidal input.....	71
6.3c	MATLAB high-pass FIR filter response to 450MHz sinusoidal input.....	71
6.3d	Cadence low-pass FIR filter response to 450MHz sinusoidal input.....	72
6.3e	Cadence high-pass FIR filter response to 300MHz sinusoidal input.....	73
6.4a	Digitized 600MHz sinusoidal.....	74
6.4b	MATLAB high-pass FIR filter response to 600MHz sinusoidal input.....	74
6.4c	MATLAB low-pass FIR filter response to 600MHz sinusoidal input.....	75
6.4d	Cadence low-pass FIR filter response to 600MHz sinusoidal input.....	75
6.4e	Cadence high-pass FIR filter response to 600MHz sinusoidal input.....	76
6.5	Optimized multiplier for coefficient ± 0.000010	77
6.6	Optimized multiplier for coefficient ± 0.000101	78
6.7	Optimized multiplier for coefficient ± 0.010011	79
6.8	Optimized multiplier for coefficient ± 0.100000	79
6.9a	High-pass FIR filter response to 600MHz input using new multipliers.....	80
6.9b	Low-pass FIR filter response to 600MHz input using new multipliers.....	81
6.10a	High-pass FIR filter response to 450MHz input using new multipliers.....	82
6.10b	Low-pass FIR filter response to 450MHz input using new multipliers.....	83
6.11a	Low-pass FIR filter response to 300MHz input using new multipliers.....	84
6.11b	High-pass FIR filter response to 300MHz input using new multipliers.....	85
7.1	Two tier tree for cadence multi-rate operation.....	87
7.2	Two tier tree frequency response.....	88

7.3a	Channel 1 output for 600MHz input to 2 tier multi-rate FIR filter.....	89
7.3b	Channel 2 output for 600MHz input to 2 tier multi-rate FIR filter.....	89
7.3c	Channel 3 output for 600MHz input to 2 tier multi-rate FIR filter.....	90
7.3d	Channel 4 output for 600MHz input to 2 tier multi-rate FIR filter.....	90
7.4a	MATLAB channel 1 response to two tier filter system 600M.....	91
7.4b	MATLAB channel 2 response to two tier filter system 600M.....	91
7.4c	MATLAB channel 3 response to two tier filter system 600M.....	91
7.4d	MATLAB channel 4 response to two tier filter system 600M.....	91
7.5a	Channel 1 output for 100MHz input to 2 tier multi-rate FIR filter.....	92
7.5b	Channel 2 output for 100MHz input to 2 tier multi-rate FIR filter.....	93
7.5c	Channel 3 output for 100MHz input to 2 tier multi-rate FIR filter.....	93
7.5d	Channel 4 output for 100MHz input to 2 tier multi-rate FIR filter.....	94
7.6a	MATLAB channel #1 response to two tier filter system 100M.....	94
7.6b	MATLAB channel #2 response to two tier filter system 100M.....	94
7.6c	MATLAB channel #3 response to two tier filter system 100M.....	95
7.6d	MATLAB channel #4 response to two tier filter system 100M.....	95
8.1	The pulse latch timing conception.....	98
A1	Simulink model for generation digitized samples.....	100
A2	Simulink model for simulating MATLAB high-pass and low-pass FIR filters.....	100
A3	Simulink model for Multi-rate FIR filter operation.....	103
B1	The designed full-adder.....	104
B2	Verification of the full-adder under no load condition.....	105
B3	The full adder self loaded with it's A input.....	106
B4	Verification of the self loaded full-adder.....	106
B5	Simulation of the high-pass FIR filter with 100MHz sinusoid at 1.9GH.....	107
B6	Simulation of the low-pass FIR filter with 100MHz sinusoid at 1.9GHz.....	108
B7	Simulation of the high-pass FIR filter with 100MHz sinusoid at 1.5GHz....	109
B8	Simulation of the low-pass FIR filter with 100MHz sinusoid at 1.5GHz.....	110

List of Tables

Table	Page
2.1 Signed binary numbers.....	5
3.1 FIR filter coefficients encoded in binary.....	14
5.1 Characteristics of the DFF.....	36
5.2 Characteristics of the half adder.....	38
5.3 Characteristics of XOR gate.....	39
5.4 Characteristics of the multiplexer.....	40
5.5 Characteristics of AND gate.....	42
5.6 Characteristics of the full adder.....	45
5.7 Input vectors and expected sum for 7-bit adder.....	48
5.8 7-bit 2's compliment validation summary.....	51
5.9 6-bit Wallace tree multiplier validation summary.....	53
5.10 Clock nodes' rise and fall times and relative skew to 6REG_check1 node....	60
6.1 Digitized values of a 100MHz sinusoidal.....	63
6.1a Characterization for the designed FIR filter for 100MHz input.....	66
6.2a Digitized values for 300MHz sinusoidal.....	67
6.2b Characterization for the designed FIR filter for 300MHz input.....	69
6.3a Digitized values for 300MHz sinusoidal.....	70
6.3b Characterization for the designed FIR filter for 450MHz input.....	72
6.4a Digitized values for 300MHz sinusoidal.....	74
6.4b Characterization for the designed FIR filter for 600MHz input.....	76
6.5 Characterization of the FIR filter to 600MHz using new multipliers.....	81
6.6 Characterization of the FIR filter to 450MHz using new multipliers.....	83
6.7 Characterization of the FIR filter to 300MHz using new multipliers.....	85

Acknowledgments

I would like to thank my advisor, Dr. Raymond Siferd, for his guidance throughout this thesis. His technical insight and wisdom had guided me in successfully completing this thesis. Most importantly, this thesis was only possible because of the funding Dr. Siferd and the RAPCEval program afforded me.

I would like to express my gratitude and thanks to Dr. Chen and Dr. Emmert for partaking on my defense committee and taking the time to read through my material.

I cannot thank the following people enough for their technical and non-technical discussion over the past year in the VLSI research laboratory. In particular, Mike Myers, Saiyu Ren, Cyprian Sajabi, Mingzhen Wang, and Kumar Yelamarthi. I sincerely thank Kumar for his help in formatting my thesis.

I would like to thank the many friends I have met here on campus who always show genuine interests in my work and also, for the many discussion throughout my stay here at Wright State University.

I cannot thank enough my family in New York City who relentlessly supports me and all that I attempt to accomplish. Special Thanks to Stephanie Smith for all your encouragement and support despite being in Africa.

1. Introduction

1.1 Motivation:

The frequency processor is a major block of a digital channelized wide-band receiver (DCWBR) and consists of a demultiplexer (DEMUX), filter bank, and fast fourier transform (FFT). The required filter length and number of points of the FFT are variables affecting performance and hardware requirements. Much effort has been expended to realizing high performance DCWBR but the FFT remains a bottle neck for achieving this goal. A modest 16 point FFT requires 40 complex multiplies and 90 additions. To achieve a high performance DCWBR this thesis proposes the use of a Systolic Array of Multi-Rate FIR Filters to accomplish channelization of the input signals without the need for the DEMUX and FFT.

1.2 Thesis Objective:

The main objectives of this thesis are summarized as follow:

- Present a review of binary number representation.
- Present a brief overview of the analog-to-digital conversion process.
- Present a brief overview of the Digital Channelized Wide Band Receiver (DCWBR).
- An overview of the design of FIR filters.
- Present and investigate a new architecture for the DCWBR.
- Present an application of the FIR filter in a Systolic Array of Multi-Rate filter bank to perform signal detection/channelization and illustrating reliable arbitration despite significant cross talk between adjacent filters.
- Investigate the design of fast adders, the critical building block of the FIR filter, that operates at high frequencies.
- Suggest circuit design alternatives that may improve the performance of the FIR filter.

1.3 Thesis Organization:

Chapter 2 reviews basic binary number representations, the analog-to-digital conversion process, and the conventional digital Channelized Wide Band Receiver.

Chapter 3 presents an overview of FIR filter design. The design of the FIR filter used through this project is presented along with some noted characteristic of the filter.

Chapter 4 discusses the digital channelized wide band receiver (DCWBR) architecture and the proposed new architecture using the systolic array of multi-rate FIR filters.

Chapter 5 presents the design and verification of basic CMOS designed blocks using cadence tools to implement the FIR filters.

Chapter 6 shows the validation of the FIR filters (high-pass and low-pass) for various frequencies along with the corresponding MATLAB verification.

Chapter 7 the cadence validation of a two tier multi-rate FIR filter system, a prelude for the systolic array of multi-rate FIR filters for channelization.

Chapter 8 is the conclusion and future work suggestion.

Appendix A shows MATLAB code and simulink block design for MATLAB simulation.

Appendix B presents some additional cadence simulation showing the output transmission pass having little drive strength and justification of the frequency of operation.

2. BACKGROUND

2.1.0 Number Representations:

DSP algorithm requires repeated arithmetic manipulation of numbers, thus it is advantageous to have efficient number representation and fast data-path circuits. In general, numbers are represented in hardware systems as either fixed-point or floating-point. Consider the simple operation $10.0/3 = 3.33$. It is trivial knowledge that resulting answer when multiplied by the divisor should result in the numerator (for this discussion, let's refer to this operation as '*recheck*'). However, when '*recheck*' is performed in a floating point system the resulting answer is 9.99 which is not exactly 10.0. Even worse when '*recheck*' is executed in fixed-point system the resulting answer is 9!!! This is a simple illustration of induced errors in arithmetic operation. In hardware systems, two discrete signal values are interpreted for information and thus lend itself to the binary number system. Each discrete or binary value which is either a "0" or "1" is called a bit. The number of bits used to encode numbers is limited by cost factors such as area, desired accuracy, and speed of operation. In general it may be assumed that fixed-point representation have higher speed and lower cost while floating-point representation have higher dynamic range thus, eliminating the need for scaling in DSP system. For the purpose of this thesis, the filter coefficients which are floating point numbers are encoded as fixed point numbers as it is assumed appropriate downstream hardware have floating-point manipulation ability.

2.1.1 Unsigned Integer:

An N-bit unsigned binary number, X, have a range of $[0, 2^N - 1]$ and its representation is given as $X = \sum_{n=0}^{N-1} x_n 2^n$ Eq. 2.1

where x_n is the n^{th} binary digit of X (i.e., $x_n \in [0,1]$). The digit x_0 is called the least significant bit (LSB) with a relative weight of unity and x_{N-1} is the most significant bit (MSB) with relative weight of 2^{N-1} .

2.1.2 Signed Magnitude Number:

Signed magnitude numbers have separate representation for the sign and magnitude. In general, The MSB digit represents the sign and the remaining $N-1$ bits represent the magnitude. The range of this representation is $[-2^{N-1}, 2^{N-1}]$ and the

$$\text{representation here becomes: } X = \begin{cases} \sum_{n=0}^{N-1} x_n 2^n & X \geq 0 \\ - \sum_{n=0}^{N-1} x_n 2^n & X < 0 \end{cases} \quad \text{Eq. 2.2}$$

2.1.3 Two's Compliment:

Two's complement number representation is the most popular number representation in use in digital systems. The ease of adding several signed number as long as the final sum is in the N -bit range is the reason for this number system popularity. An N -bit two's complement number range is $[-2^{N-1}, 2^{N-1}-1]$ and its representation is:

$$X = \begin{cases} \sum_{n=0}^{N-1} x_n 2^n & X \geq 0 \\ \sum_{n=0}^{N-1} x_n 2^n - (2^N) & X < 0 \end{cases} \quad \text{Eq. 2.3}$$

The MSB must be a 1 for the $X < 0$ case in Eq. 2.3. The number representations in this thesis are two's complement. It is possible to have overflow during the addition of two's complement numbers (for example, when the addition of 2 positive numbers yields a negative sum) but no overflow protection or exception circuitry will be provided as it is assumed that all input are small enough to avoid this scenario. Table 2.1 below shows the

binary representation of the decimal number -8 through +7 in sign magnitude and two's compliment.

Decimal	Signed 2's Complement	Signed Magnitude
+7	0111	0111
+6	0110	0110
+5	0101	0101
+4	0100	0100
+3	0011	0011
+2	0010	0010
+1	0001	0001
+0	0000	0000
-0	—	1000
-1	1111	1001
-2	1110	1010
-3	1101	1011
-4	1100	1100
-5	1011	1101
-6	1010	1110
-7	1001	1111
-8	1000	—

Table 2.1 Signed Binary Numbers [1].

While Table 2.1 may not be very exciting it is important to note a couple differences between the two number system. First, it can be easily noted that positive number representation is identical between the two systems; however, for negative numbers, the representation does differ and while negative 0 is not legal in two's compliment system it is valid for sign magnitude system. This observation may seem inconsequential to the human mind but may make a dramatic difference in hardware computation!! In addition, this +0 -0 differences causes the negative range (most negative number) to be slightly greater for 2's compliment number system versus that of sign magnitude.

2.1.4 Conversion of decimal fractions to Binary [2]:

To convert a decimal fraction say, 0.6875, to binary requires multiplication by 2 to an integer and fraction. This process is repeated until the fractional part equals 0 or until there enough digits to give sufficient accuracy. This process is illustrated with the following example converting 0.6875 to binary.

$0.6875 * 2 = 1.3750$	Integer = 1	MSB
$0.3750 * 2 = 0.7500$	0	
$0.7500 * 2 = 1.5000$	1	
$0.5000 * 2 = 1.0000$	1	LSB

Thus, $(0.6875)_{10} = (0.1011)_2$

2.2.0 Analog-to-digital conversion [3]:

Most signals of practical interest, such as speech, biological signal, radar signals, and various communications signals such as audio and video signals are analog. To process analog signal by digital means, it is first necessary to convert them to digital form which is a sequence of numbers with finite precision. This procedure is called analog-to-digital (A/D) conversion. The front-end hardware to the filters of this thesis is in fact an A/D converter. Conceptually, A/D may be modeled as the following three-step process:

1. *Sampling*: The conversion of a continuous-time signal into discrete-time signal obtained by taking samples of the continuous-time signal at discrete-time instants. Thus, if $x_a(t)$ is the input signal, the output is $x_a(nT) \equiv x(n)$ where T is called the sampling interval.
2. *Quantization*: the conversion of a discrete-time continuous-value signal into a discrete-time discrete-value (digital) signal. The difference between the unquantized sample $x(n)$ and the quantized output $x_q(n)$ is called the quantization error.
3. *Coding*: Each discrete value $x_q(n)$ is represented by a N -bit binary sequence.

2.2.1 Sampling [4]:

A signal whose spectrum is bandlimited to B Hz can be reconstructed from its samples taken uniformly at a rate $F_s > 2B$ samples per second. Thus the minimum sampling frequency is $F_s = 2B$ Hz and is called the Nyquist rate. As illustrated in figure 2.1 (a) below, sampling at the Nyquist rate requires an ideal filter to recover $f(t)$ from $\bar{f}(t)$, thus the practical solution is to sample at a rate higher than the required Nyquist rate (i.e. $F_s > 2B$) which produces a finite band gap between successive cycles as illustrated in figure 2.1 (b) which permits the use of a filter with appropriate roll off to recover $f(t)$.

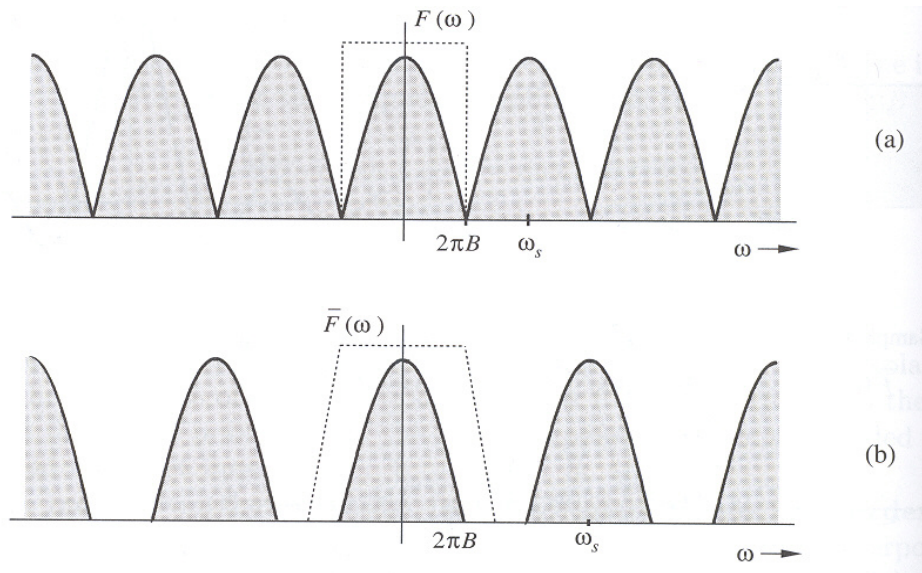


Figure 2.1 Spectra of a signal sampled at (a) the Nyquist rate (b) above the Nyquist rate.

2.2.2 Quantization of sinusoidal signals [5]:

Quantization errors or quantization noise, $e_q(n)$, are introduced when the continuous-value signal is represented by a finite set of discrete value levels and is define as the difference between the quantized value and the actual sample value. Therefore,

$$e_q(n) = x_q(n) - x(n) \quad \text{Eq. 2.4}$$

Figure 2.2 illustrates the sampling and quantization of an analog sinusoidal signal $x_a(t) = A\cos\Omega_0 t$. From the original analog signal $x_a(t)$ discrete-time, discrete-amplitude signal $x_q(nT)$ is obtained after quantization.

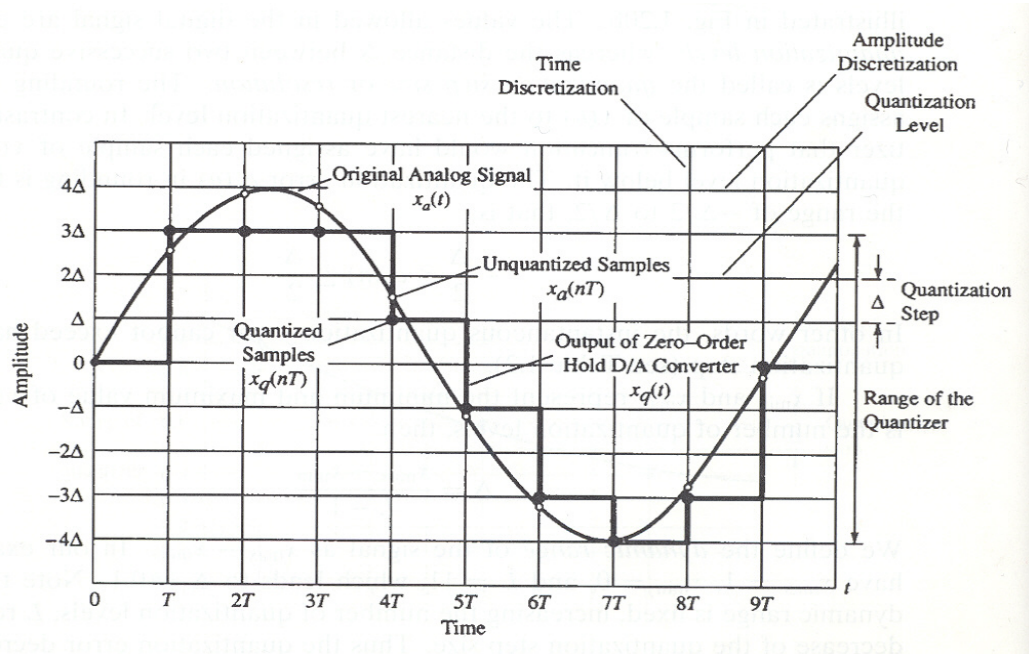


Figure 2.2 Sampling and quantization of a sinusoidal signal.

The analysis from [4] shows the signal-to-quantization noise ratio (SQNR) is

$$\text{SQNR(dB)} = 1.76 + 6.02b \quad \text{Eq. 2.5}$$

where b is the number of bits used for the binary representation of each quantized value. Since this thesis uses six effective bit, The signal-to-noise ration (SNR) is expected to be 37.88dB.

2.3.0 Digital Channelized Wide Band Receiver [6]:

Digital receivers take some radio frequency (RF) signals and digitize it using an ADC for further processing. A wide-band receiver covers approximately 1GHz of instantaneous bandwidth and may be used to intercept simultaneous radar pulses. The type of receivers that can process simultaneous signals are 1) channelized, 2) Bragg cell, and 3) compressive. Channelized and Bragg cell receivers are similar and both have parallel outputs. Channelized receivers use filters to separate the signals while Bragg cell receivers use optical techniques to separate signals. Compressive receivers have series outputs and use dispersive delay lines to separate signals. The received signals may then be converted into video signals by using crystal video detectors and then digitized for further signal processing. Channelization may be accomplished more easily using digital circuitry because of better control of filter shape.

Figure 2.3 below shows the main block of a digital wide-band receiver. The RF front end includes the digitizer and may be selected in the second aliasing zone of the ADC to avoid the second harmonic generated in the first aliasing zone [6]. The frequency processor is a bank of digital filters which separates input signals according to their frequency. This thesis focuses on the design implementation of the digital filters to channelize received signals. The parameter encoder converts the input signals into a pulse descriptor word (PDW) and may contain information such as frequency, amplitude, pulse width, and time and angle of arrival. Experience has shown that the parameter encoder block is the most difficult to implement.

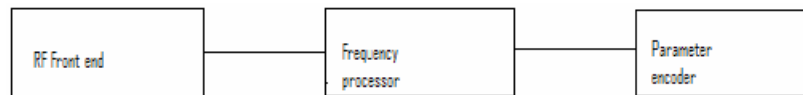


Figure 2.3 Main blocks of a wide-band digital receiver.

3. FINITE IMPULSE RESPONSE (FIR) FILTERS

3.1.0 FIR Filters [7]:

A FIR filter of length M with input $x(n)$ and output $y(n)$ is described by the difference equation $y(n) = b_0x(n) + b_1x(n-1) + \dots + b_{M-1}x(n-M+1)$

$$= \sum_{k=0}^{M-1} b_k x(n-k) \quad \text{Eq. 3.1}$$

where $\{b_k\}$ is the set of filter coefficients. Alternatively, the output sequence may be express as the convolution of the unit sample response $h(n)$ of the system with the input

$$\text{signal. Thus, } y(n) = \sum_{k=0}^{M-1} h(k)x(n-k) \quad \text{Eq. 3.2}$$

The filter can also be characterize by its system function

$$H(z) = \sum_{k=0}^{M-1} h(k)z^{-k} \quad \text{Eq. 3.3}$$

The above equations express the multiplication of a delay value by some coefficient. The direct form realization of this FIR system is shown in figure 3.1 below.

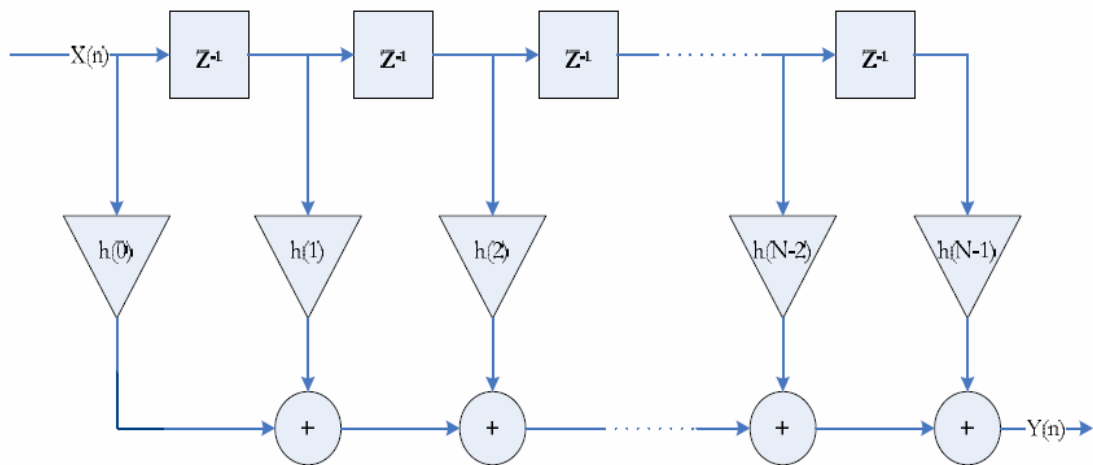


Figure 3.1 Direct-form realization of FIR system.

The above realization requires $M-1$ memory elements to store the previous $M-1$ inputs and has a complexity of M multiplications and $M-1$ additions; thus the design of FIR systems requires a huge amount of hardware and hence large power consumption. FIR filters have linear phase characteristics within the pass-band and satisfies the symmetry or asymmetry condition

$$h(n) = \pm h(M-1-n)$$

This symmetry reduces the multiplication from M to $\frac{M}{2}$ for M even and to $\frac{M-1}{2}$ for M odd; thus reducing the require amount of hardware. In effect, since two different samples are multiplied by the same coefficient, instead of multiplying them separately, they can be first added together and then multiplied by their common coefficient. This methodology requires extra summing circuits but since adders uses much less hardware than multipliers ones saves significantly on area and power. This new realization is shown in figure 3.2 below.

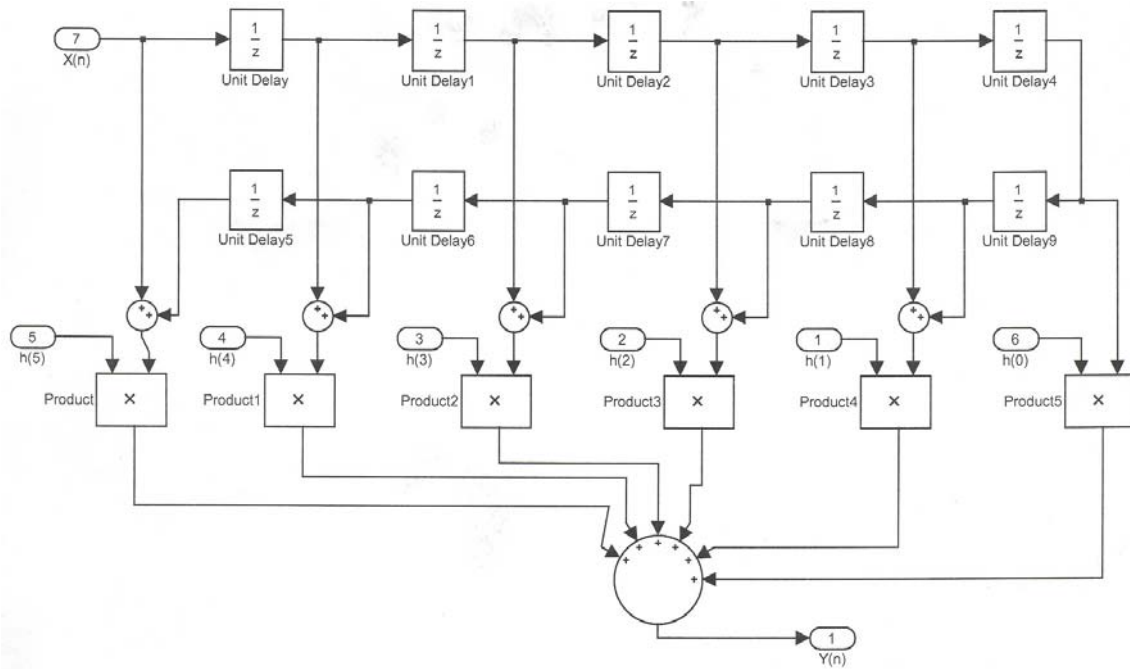


Figure 3.2 FIR filter, taking advantage of symmetry to reduce multiplication.

The frequency response of selective FIR filters is non-ideal in nature; therefore FIR systems are causal. Thus, (a) the frequency response $H(\omega)$ cannot be zero except at a

finite set of points in frequency; (b) the magnitude $|H(\omega)|$ cannot be constant in any finite range of frequencies and the transition from passband to stopband cannot be infinitely sharp (this is a consequence of the Gibbs phenomenon, which results from the truncation of $h(n)$ to achieve causality). As illustrated in figure 3.3 below a small amount of ripple is tolerable in both the passband and stopband. The transition of the frequency from passband to stopband defines the transition band of the filter. As illustrated in figure 3.3, ω_p defines the edge of the passband while ω_s defines the beginning of the stopband and hence the width of the transition band is $\omega_s - \omega_p$. The width of the passband is called the bandwidth of the filter which is ω_p in this illustration. A filter may be designed given (1) the maximum tolerable passband ripple, (2) the maximum tolerable stopband ripple, (3) the passband frequency, ω_p , and (4) the stopband frequency ω_s . Based on these specification we may select the parameters $\{a_k\}$ and $\{b_k\}$ in the frequency response characteristics. The degree to which $H(\omega)$ approximates the specifications depends in part on the criterion used in the selection of the filter coefficients $\{a_k\}$ and $\{b_k\}$ as well as the numbers (M,N) of coefficients.

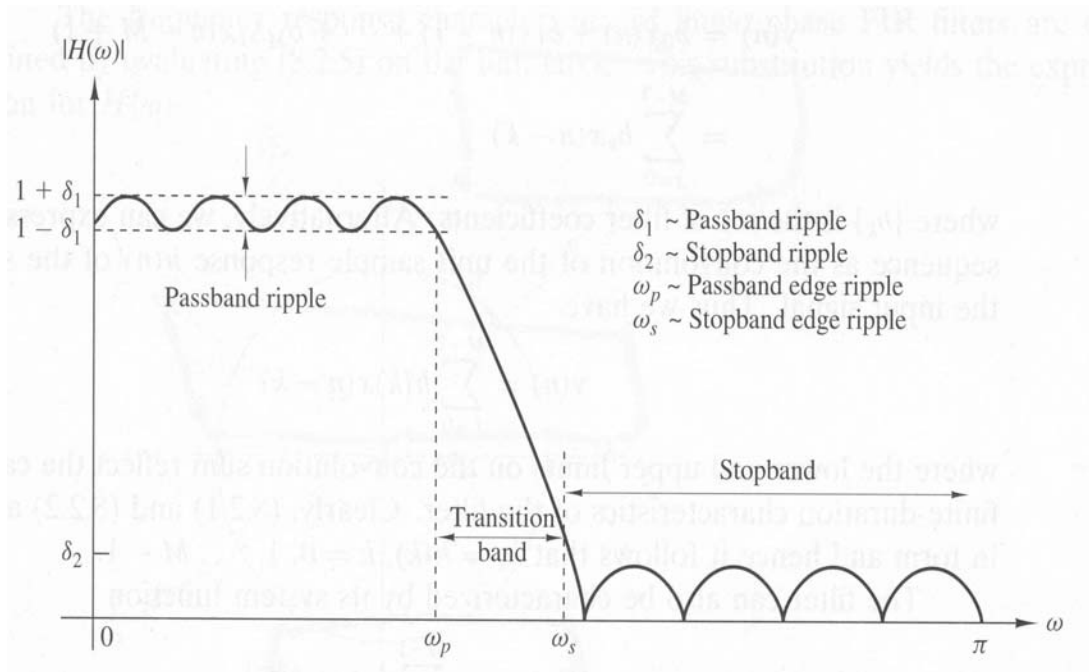


Figure 3.3 Magnitude characteristics of physically realizable filters.

3.1.1 Filter Design Coefficient Generation:

The program for generating coefficients of the FIR filter using the Parks-McClellan algorithm in MATLAB 7.0.4 is shown in A1.1.0 of appendix A, and the coefficients are shown below along with the frequency response characteristic of the filter as shown in figure 3.4 below:

Low Pass FIR: [0.0000; -0.0175; 0.0000; +0.0384; 0.0000; -0.0892; 0.0000; +0.3124
0.5000; +0.3124; 0.0000; -0.0892; 0.0000; +0.0384; 0.0000; -0.0175]

High Pass FIR: [0.0000; +0.0175; 0.0000; -0.0384; 0.0000; +0.0892; 0.0000; -0.3124
0.5000; -0.3124; 0.0000; +0.0892; 0.0000; -0.0384; 0.0000; +0.0175]

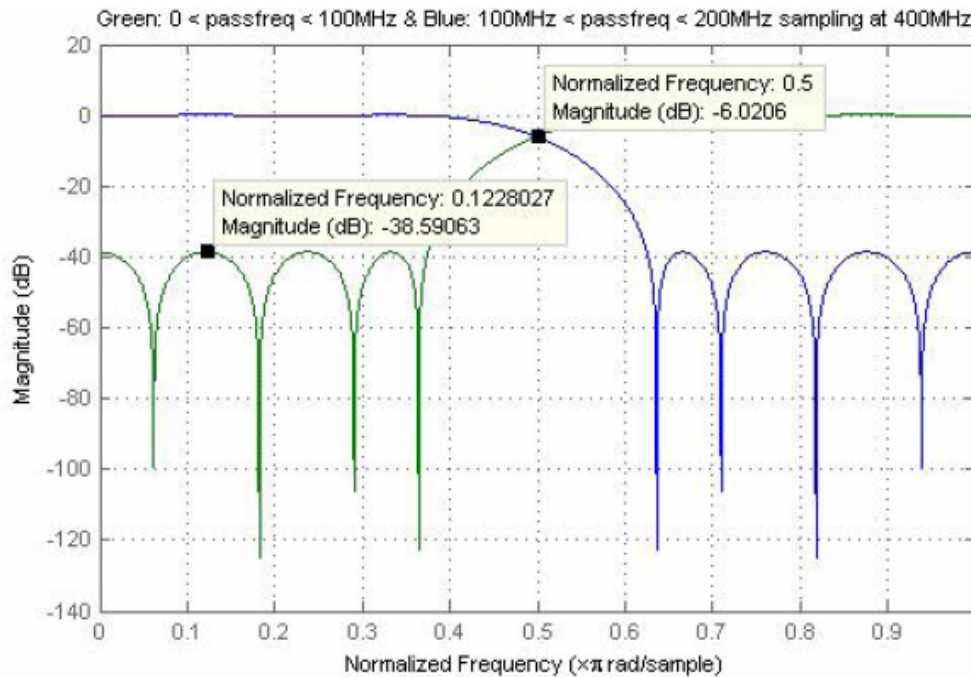


Figure 3.4 Frequency response of low-pass and high-pass FIR filter.

Some observations worth noting are:

- (a) Every alternate coefficient of the filter is 0 which implies a reduction of hardware as multiplication by 0 always yield 0.
- (b) The filter coefficients exhibit symmetry which can be exploited to reduce the number of multiplications required by adding first and multiplying by the common coefficient as illustrated in figure 3.2.
- (c) The filter coefficients only differ in the sign magnitude between the low-pass and high-pass filters; thus, easing the hardware implementation.
- (d) Very little ripple exists in the pass-band of the filters and the separation between passband and stopband is 38.6dB complying with Eq. 2.5.
- (e) The cross-over point (boundary frequency separating high-pass and low-pass) is attenuated by 6.02dB as illustrated in figure 3.4

The coefficients encoded in binary sign magnitude using the method illustrated in section 2.1.4 are shown in table 3.1 below.

Filter Tap #	Coefficient in floating point	Coefficient in binary
0,2,4,6,10,12,14	+0.0000	0.000000
1,15	± 0.0175	0.000001,1.000001
3,13	± 0.0384	0.000010,1.000010
5,11	± 0.0892	0.000101,1.000101
7,9	± 0.3124	0.010011,1.010011
8	+0.5000	0.100000

Table 3.1 FIR filter coefficients encoded in binary

4. Digital Channelized Wide Band Receivers (DCWBR)

4.1.0 Conventional DCWBR Architecture:

The architecture of a conventional DCWBR is illustrated in figure 4.1 below. Compared to figure 2.1, the RF analog down converter and ADC makes up the RF front end, the frequency processor is composed of a 1:16 de-multiplexer, filter bank, and 16 point FFT, and the parameter encoder here is the same as that shown in figure 2.1.

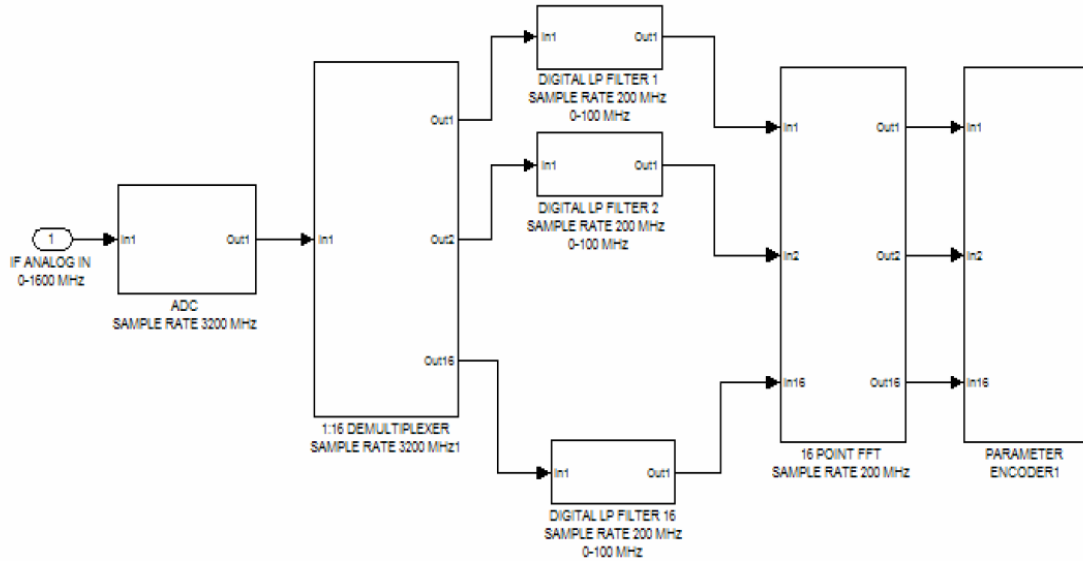


Figure 4.1 Conventional DCWBR Architecture.

For this receiver, the RF analog input is first down converted to 0 to 1.6GHz and an ADC digitizes the signal with a sampling rate of 3.2GHz. This digitized signal is the input to the 1:16 de-multiplexer (DEMUX) which is also clocked at 3.2GHz and produces 16 output decimated by a factor of 16. The outputs from the DEMUX are then used as input to 16 low pass filters clocked at 200MHz. These outputs provide the inputs to the hardware implementation of the 16 point FFT which is performed at the same clock rate of the low pass filters. Essentially, the FFT resolve the frequency spectrum of the input signals into 16 bins. The magnitude versus frequency response of the FFT output is illustrated in figure 4.2 below. As illustrated in figure 4.2, there is overlap between channel bins. The amount of overlap and roll off transition region is a design trade off. A

fast transition region for the bins permits less overlap at the cost of increasing the required length of the digital low pass filters and FFT.

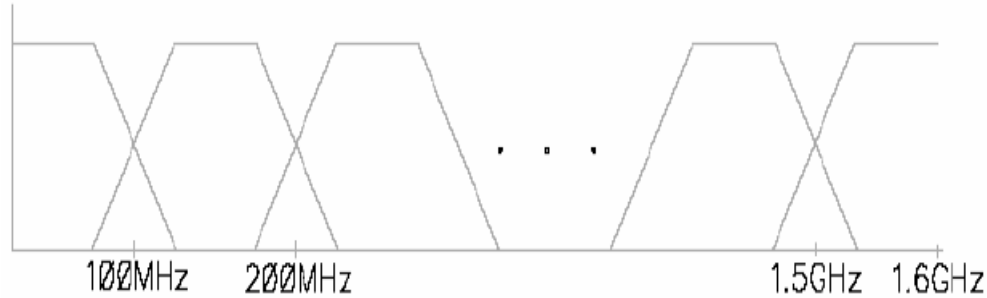


Figure 4.2 channelized receiver magnitude frequency response.

For the example illustrated above, each bin has a pass band of 50MHz and a transition bandwidth of 50MHz resulting in bins having center frequencies separated by 100MHz. Signals in the overlaps transition region thus appear in two frequency bins with typically different magnitude. This fact makes channel arbitration difficult between adjacent channel bins to determine the true frequency bins. Additionally, two signals with relatively close frequencies may be difficult to resolve. The channel arbitration task becomes more difficult for pulsed inputs because the leading and trailing edges of the pulse contain a great deal of broadband energy which spills into both adjacent and non-adjacent channels [7]. This phenomenon is known as the “rabbit-ear effect” because of the out-of-channel, time-domain output responses have a peak on the leading and trailing edges of the pulse due to the impulse response of the filters. These effects then require the channel arbiter to determine which channel the signal truly resides and must also reject the out-of-channel response. Techniques such as amplitude comparison of adjacent channels and techniques that detect the presence of the rabbit-ear effect have been employed to perform channel arbitration. These approaches use only the amplitude of filter bank outputs and have inherent limitations [8]. Parameters such as signal center frequency, amplitude, pulsewidth, and time-of-arrival of signals in the channel bins are estimated and encoded by the parameter encoder. For this example DCWBR the time to update the receiver outputs with instantaneous signal change is the time to fill the 16 tap low pass filters plus the time to compute the FFT, thus, $16 \times 5\text{ns} + 5\text{ns}$ for a total of 85ns.

4.1.1 Proposed Systolic Array DCWBR Architecture:

The new DCWBR architecture based on a Systolic Array of Multi-Rate FIR Filters is illustrated in figure 4.3 below.

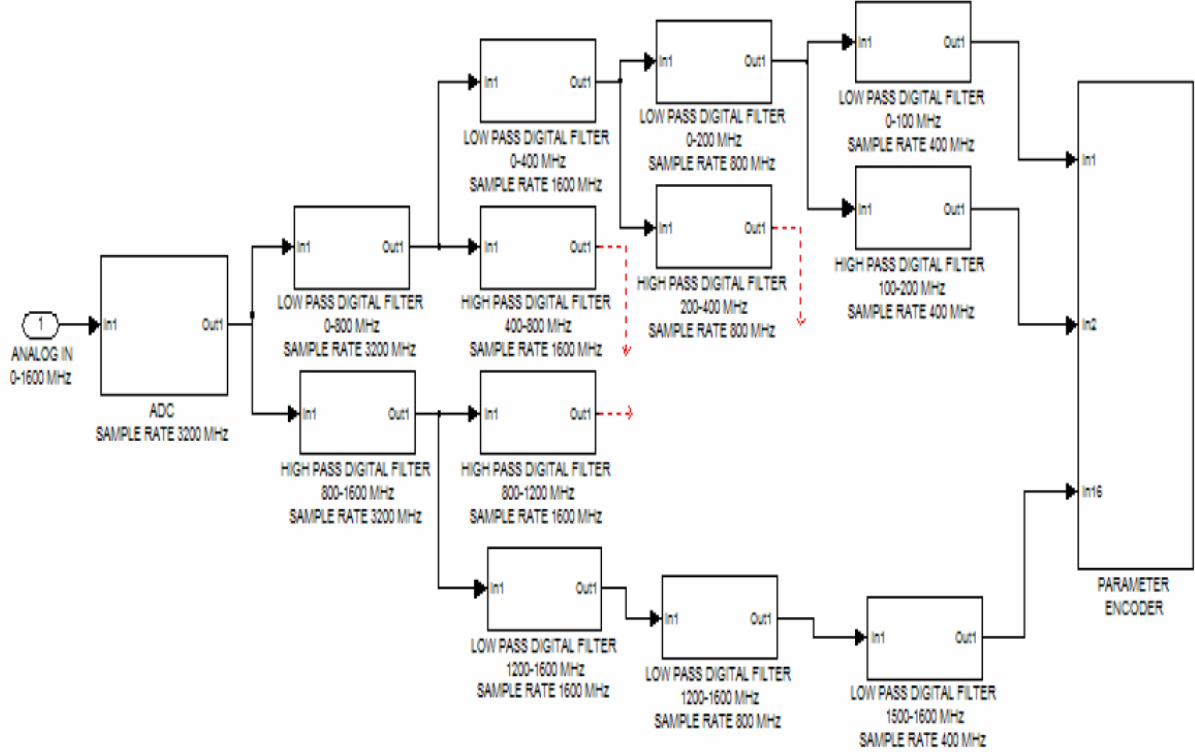


Figure 4.3 Proposed Systolic Array DCWBR Architecture.

This implementation has 15 identical low pass and 15 identical high pass filters with the coefficients from section 3.1.1. This identical use of filter elements makes the filter bank a systolic array, that is, a pipelined array of identical elements with clocked data flowing through the structure. The first tier of filters (two filters – one high pass and one low pass filter) are clocked at the same rate of the ADC at 3.2GHz. The low pass filter has a pass-band from 0 to 600MHz, a transition region from 600 to 1000MHz (400MHz transition bandwidth) and a stop-band from 1000 to 1600MHz. Conversely, the high pass filter has a stop band from 0 to 600MHz, a transition region from 600 to 1000MHz and pass-band from 1000 to 1600MHz. The low pass filter of the first tier resolve frequencies from 0 to 800MHz while the high pass filter resolve frequencies from 800 to 1600MHz. Thus the

first tier creates 800MHz bins. The magnitude frequency response of this first tier is shown in figure 4.4a below.

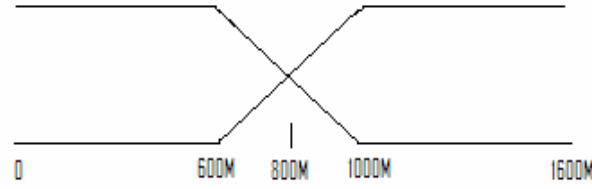


Figure 4.4a Frequency magnitude response of first tier systolic array of filters.

The second tier consists of 4 filter (identical filters as used in the first tier), two high pass and 2 low pass filters sampled at 1600MHz. This tier resolves the input frequency into 400MHz bins. The low pass filter has a pass-band from 0 to 300MHz, transition region from 300 to 500MHz and stop-band greater than 500MHz. The high pass filter has stop-band from 0 to 300MHz, transition frequency from 300 to 500MHz and pass-band frequency from 500 to 700MHz. The frequency magnitude response of the filters in the second tier is shown in figure 4.4b below.

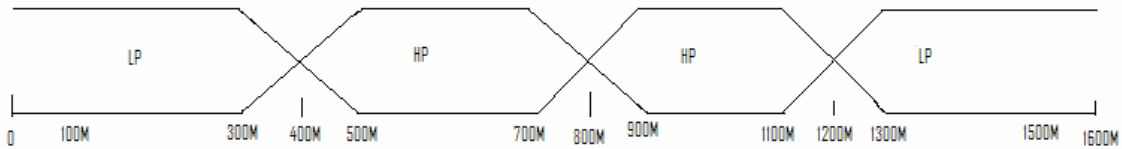


Figure 4.4b Frequency magnitude response of second tier systolic array of filters.

Again, since the sampling frequency of the second tier is half that of the first tier, the coefficients are identical. The third tier has four low pass and four high pass filters and are clocked at 800MHz and resolve input signals into a 200MHz bins with transition region of 100MHz. The coefficients of each high pass and low pass filters are identical to those in tiers 1 and 2. The frequency magnitude response of the third tier of filters are shown in figure 4.4c below which clearly illustrates frequency bins of 200MHz with 100MHz transition regions. All frequencies are in MHz in figure 4.4c.

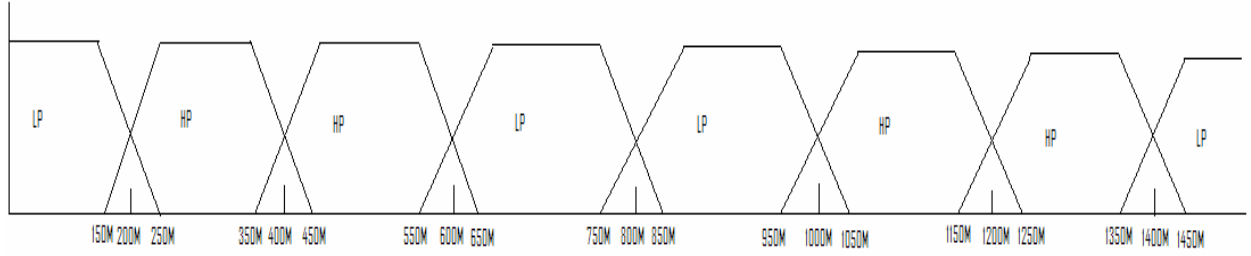


Figure 4.4c Frequency magnitude response of third tier systolic array of filters.

Logically, the fourth tier consists of 8 low pass filters and 8 high pass filters clocked at 400MHz and resolve input signals into 100MHz bins with 50MHz transition region. Tier 4 frequency magnitude response is therefore the same as that shown in figure 4.2. Thus, the new systolic array DCWBR can resolve input signals into channel bins identical to the conventional DCWBR whilst eliminating the need or the DEMUX and FFT which the bottle neck for increasing performance of the conventional DCWBR. By eliminating the need of the FFT the systolic array DCWBR may present opportunities for increased performance of the DCWBR.

4.1.2 MATLAB performance assessment of the systolic array DCWBR architecture:

The CMOS design implementation of the FIR filters were unable to operate at 3.2GHz as outlined above. The Maximum frequency of operations obtained for the designed FIR filters using the 0.13 μ m process is 1.8GHz. For this reason, the MATLAB simulations will be “scaled back” to operate at this frequency. The model built using the aid of SIMULINK for MATLAB simulation is shown in figure A3 of appendix A. The expected frequency magnitude response of figure A3 of appendix A is shown in figure 4.5 below.

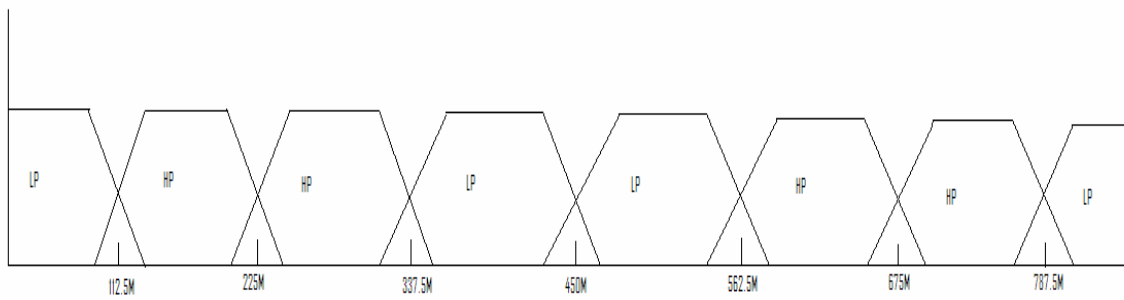
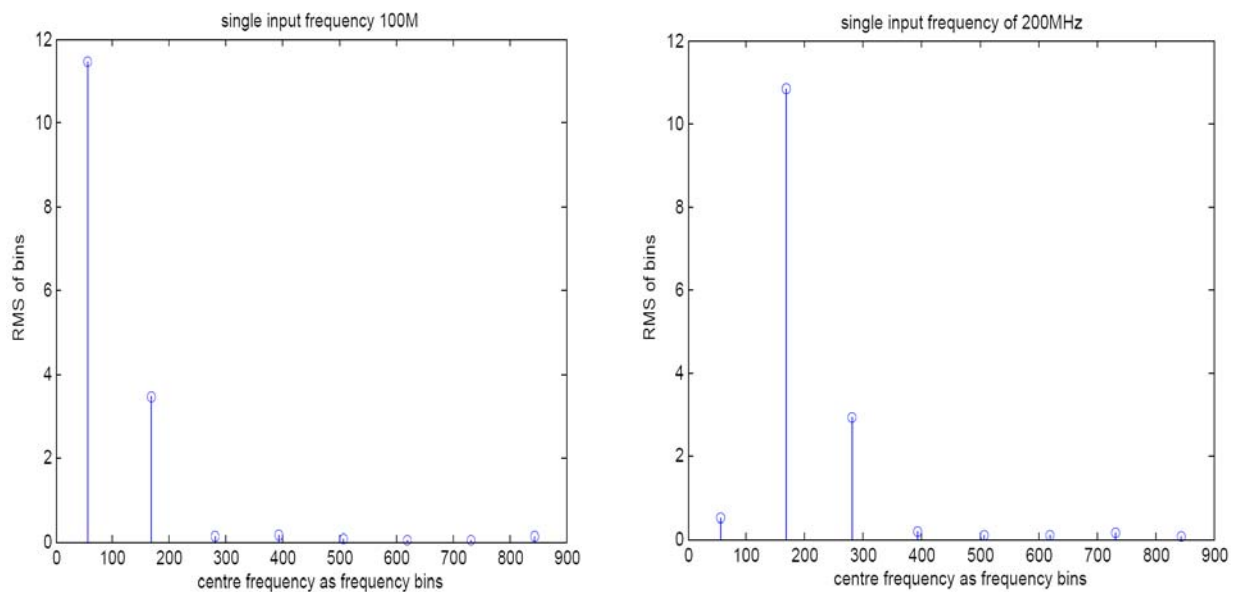
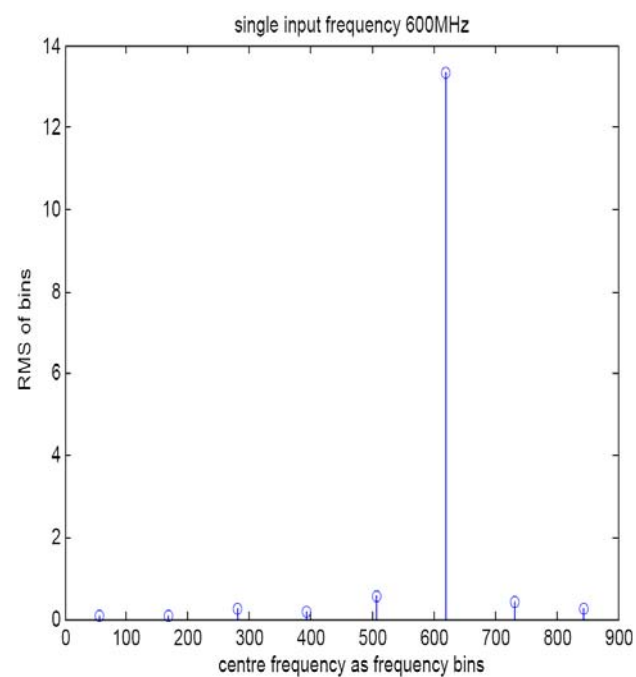
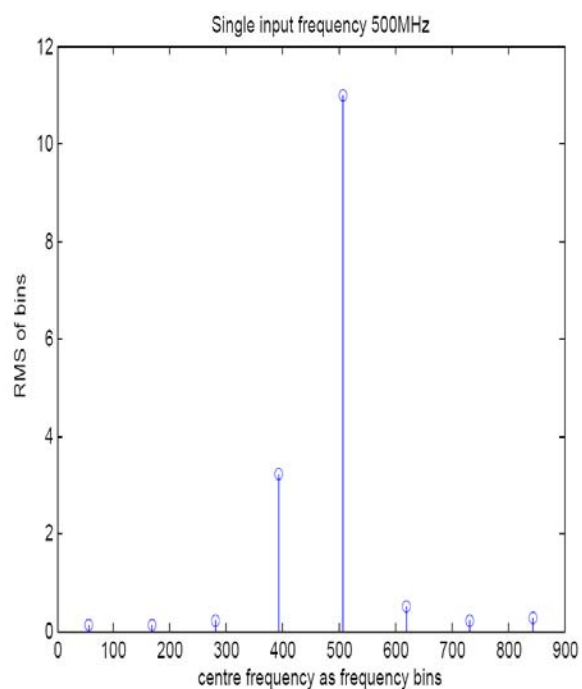
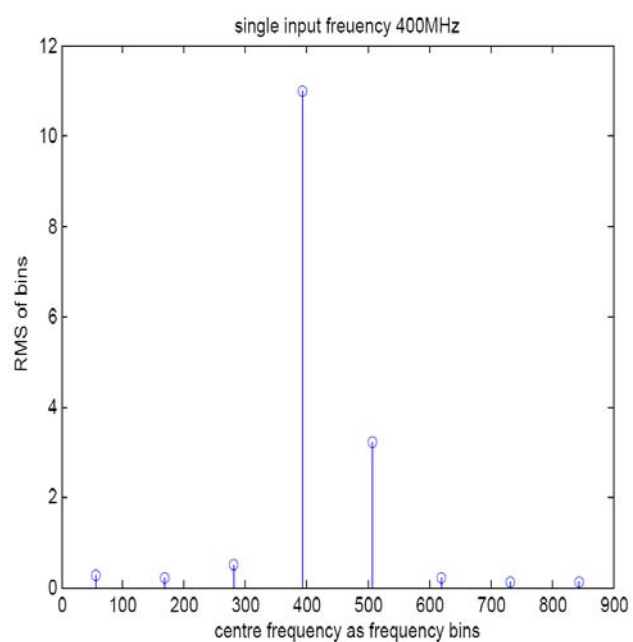
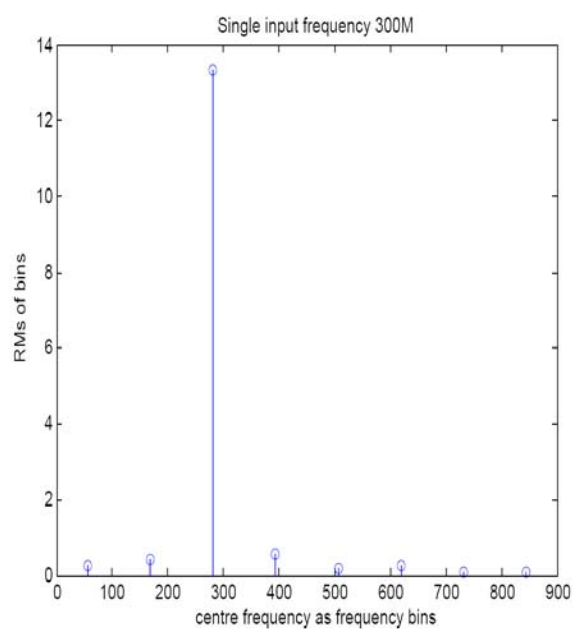


Figure 4.5 Expected frequency magnitude response for figure A3.

The following figures in figure 4.6 shows the RMS versus frequency bin response for signals of 100 through 800MHz in increments of 100MHz.





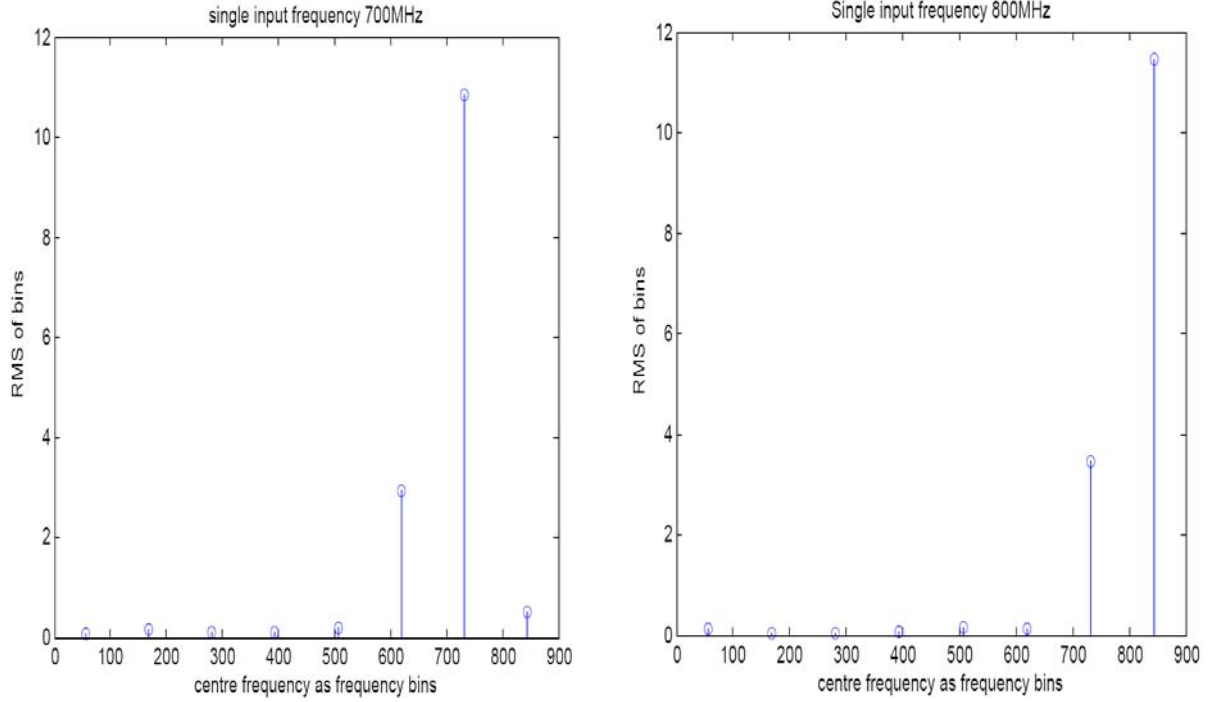


Figure 4.6 Frequency responses to single sinusoidal input across frequency spectrum.

The frequency bins are 0 – 112.5MHz, 112.5 – 225MHz, 225 – 337.5MHz, 337.5 – 450MHz, 450MHz – 562.5MHz, 562.5 – 675MHz, 675 – 787.5MHz, 787.5 – 900MHz and the respective centre frequency demarcations are 56.25, 168.75, 281.25, 393.75, 506.25, 618.75, 731.25, and 843.75MHz. All of the above plots illustrate two fundamental properties of the systolic array DCWBR architecture:

- 1) The systolic array of FIR filters correctly allocates input frequencies to the correct frequency bin thus making it a viable solution for frequency intercept problem the DCWBR seeks to solve.
- 2) The systolic array of FIR filters suffers from cross talk. As illustrated above, most of the signal appears to be in two frequency bands and thus the system needs to be able to resolve the signal into one frequency band.

As state before, DCWB may be required to detect multiple inputs of varying signal strength simultaneously. Thus for two signal inputs with relatively close frequencies, channel arbitration may be difficult. Figure 4.7 below illustrates the simultaneous detection of two signals appear to be in 4 frequency bins instead of the expected two.

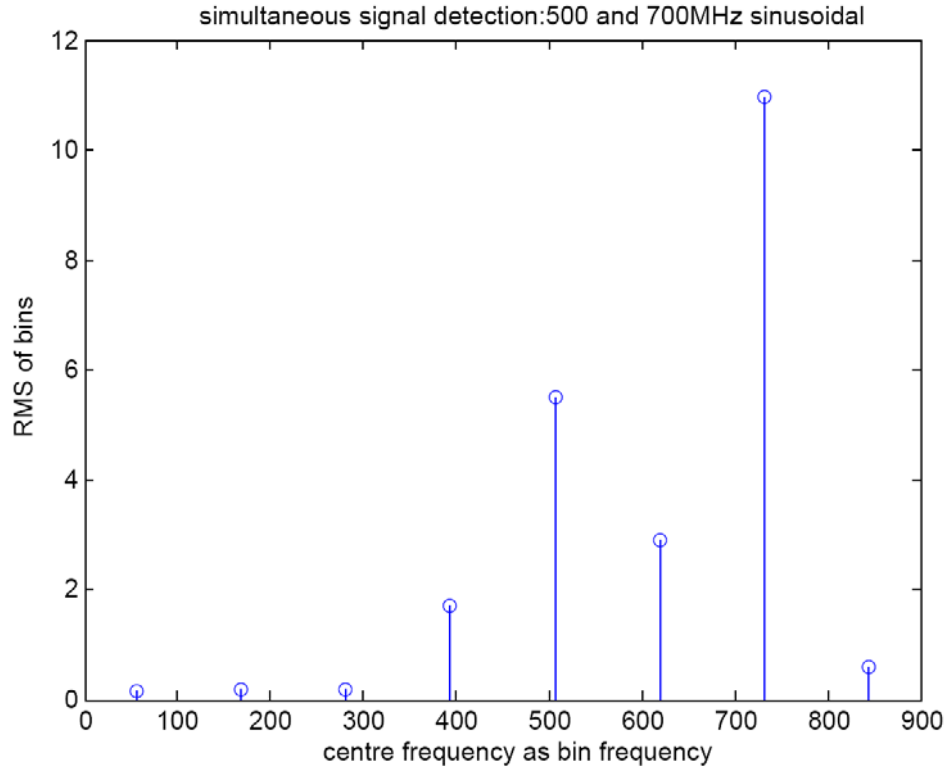


Figure 4.7 Simultaneous detection of two input frequencies for systolic array DCWBR.

Also stated earlier was the increased difficulty in detecting pulse inputs because signal energies may be spilled into non-adjacent channel bins. Figure 4.8 below shows the systolic array response to a 500MHz input clearly illustrating that significant signal energies are spilt into several bands. As mentioned before techniques to suppress these “rabbit-ear effect” may be use to aid the system. The 500MHz input signal appears to be possible in 4 frequency bins versus the expected 1 frequency bin. Figure 4.9 shows the frequency response to a 500 and 700MHz pulse inputs. Again, the issue of channel arbitration is apparent.

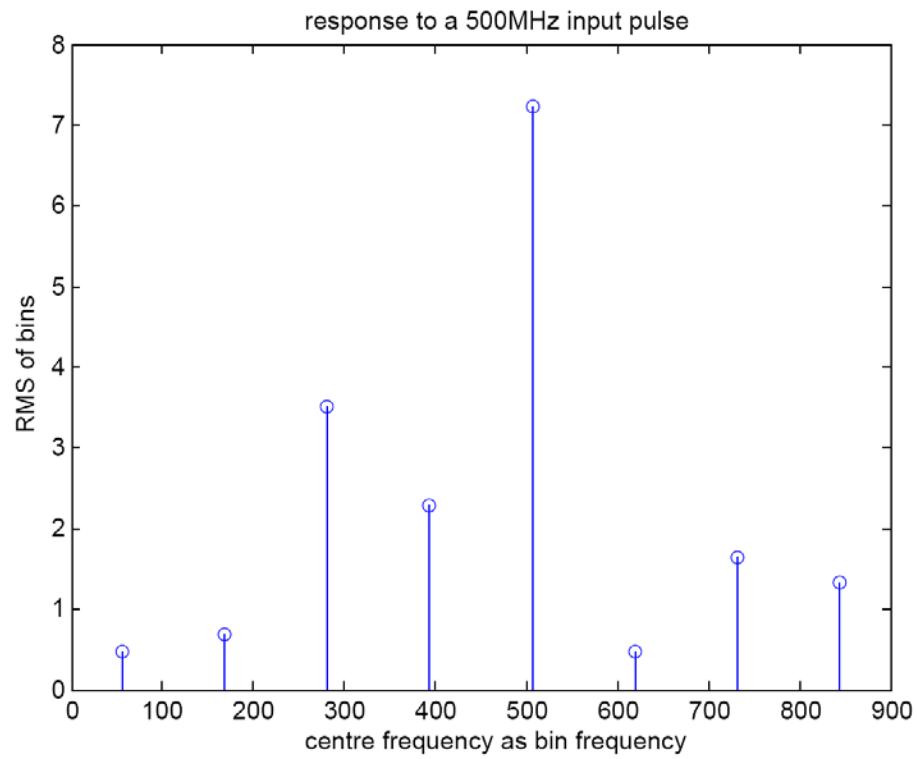


Figure 4.8 Systolic array detection of a 500MHz pulse input.

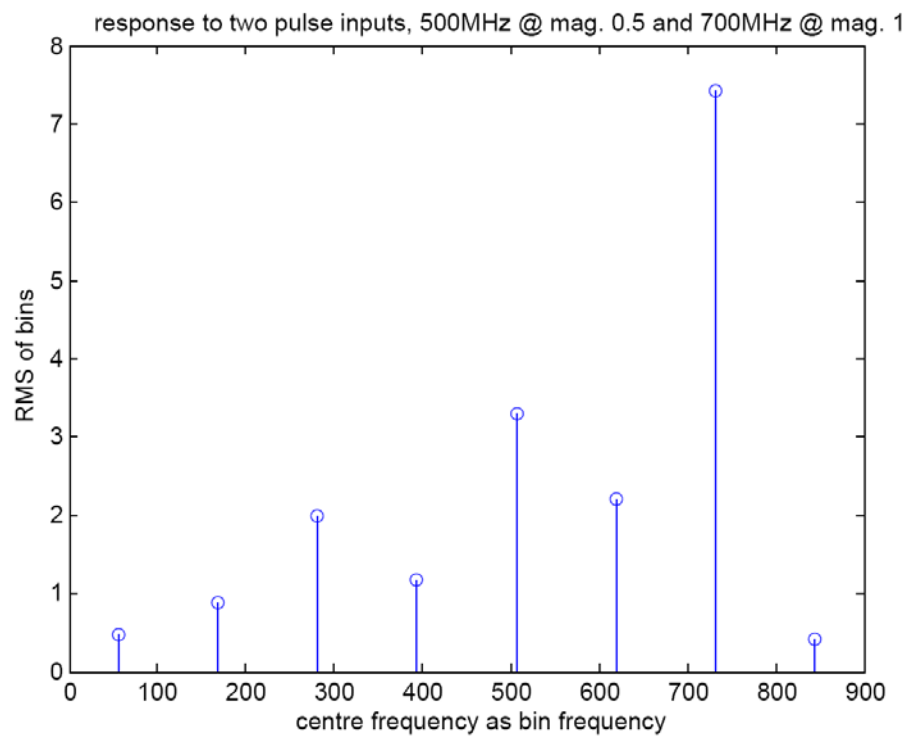
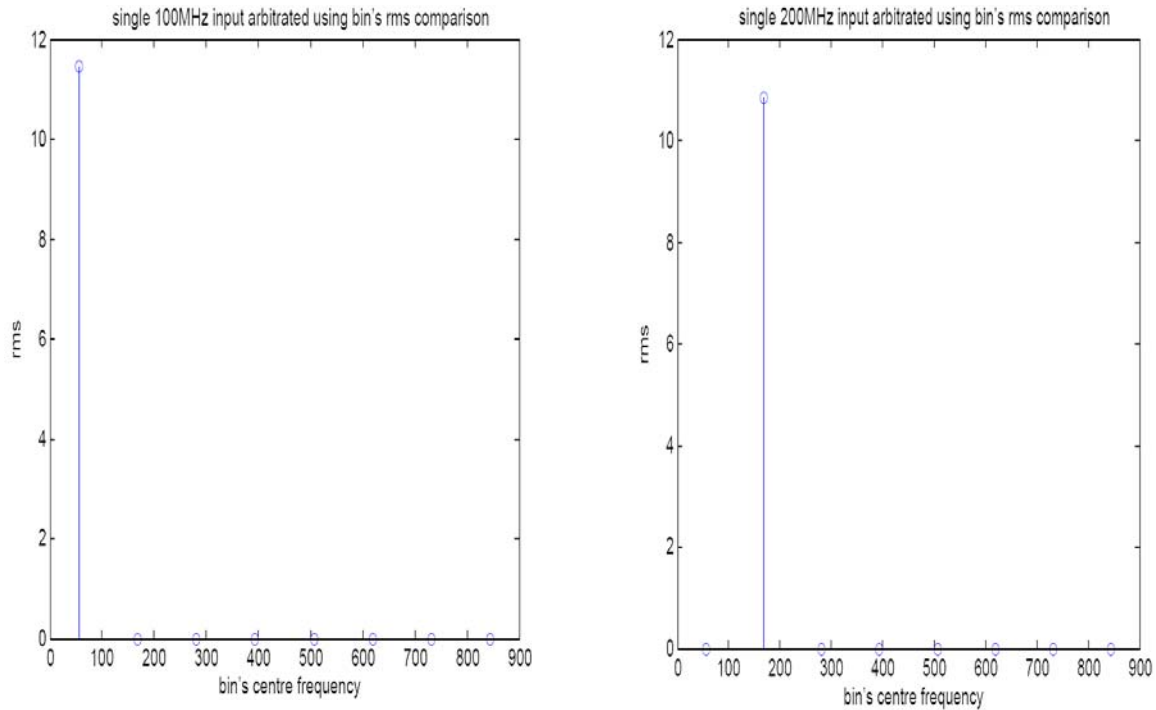
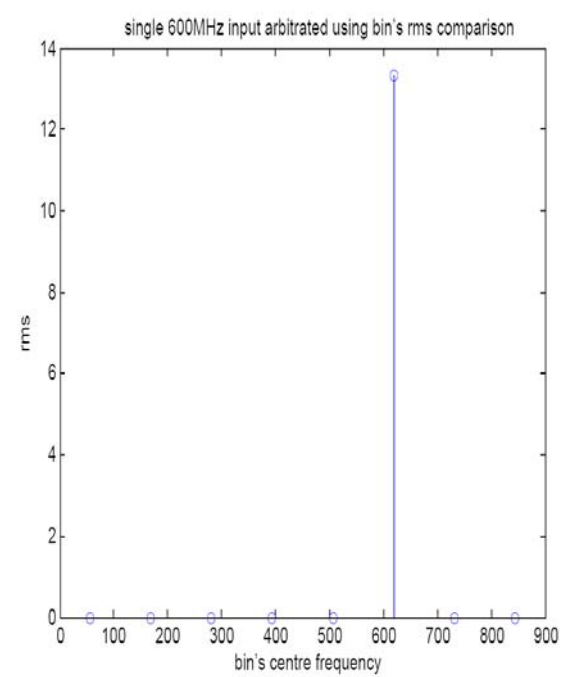
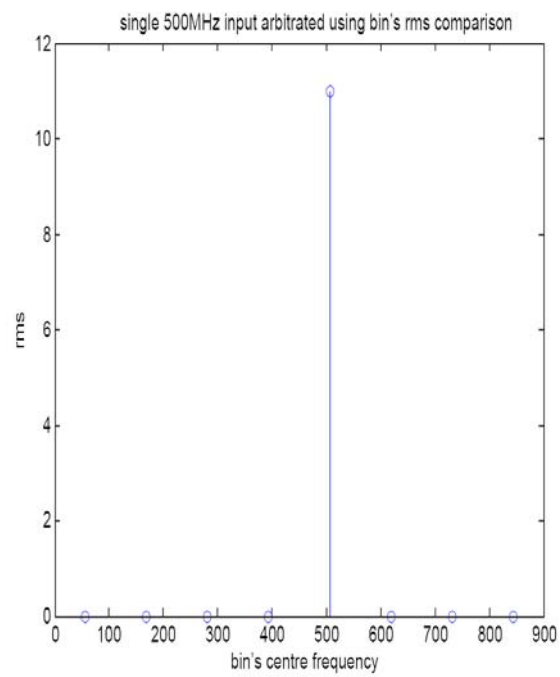
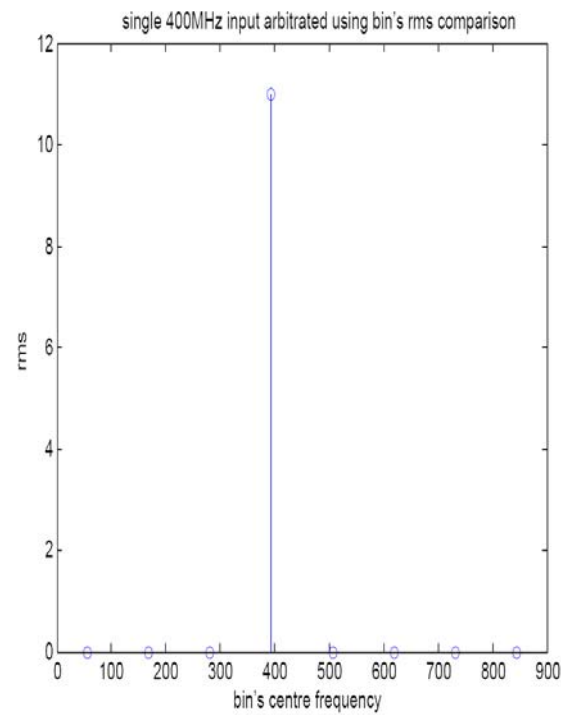
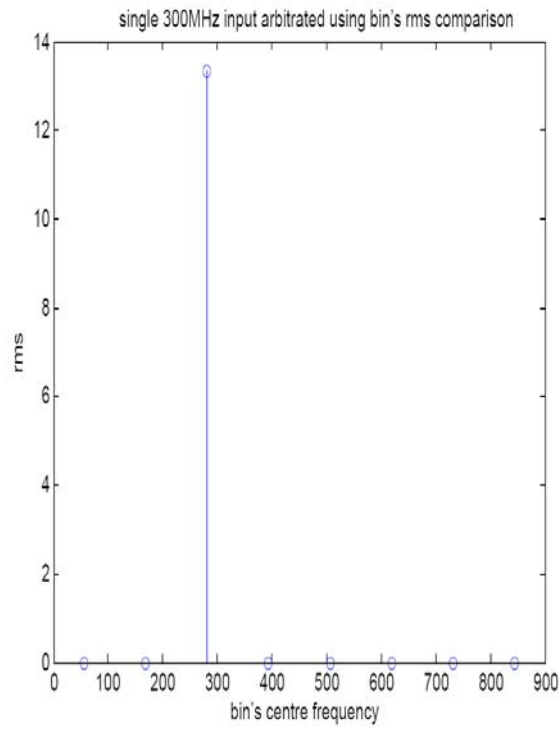


Figure 4.9 Systolic array detection of a 500 and 700MHz pulsed input.

4.1.3 Systolic Array DCWBR Channel Arbitration Using Bin's rms Magnitude:

It can be observed from the figures in the previous section that the signal residing in its correct frequency bin has the largest rms value compared to cross talk signal appearing in other bins. As mentioned before this approach does have inherent limitation [8]. The MATLAB algorithm for channel arbitration using the latter approach is shown in section A.1.1.2 of appendix A. The frequency response using the same input frequencies from figure 4.8 are shown in figure 4.10 below using the new algorithm for channel arbitration. Clearly, this algorithm will correctly resolve all single input frequency to its correct channel bin. The algorithm used here is an order N search algorithm; while this may be acceptable for this 8 channel example, the search time may be too long for say a 16 channel DCWBR. Thus other more efficient search algorithm such as a binary tree search may be employed.





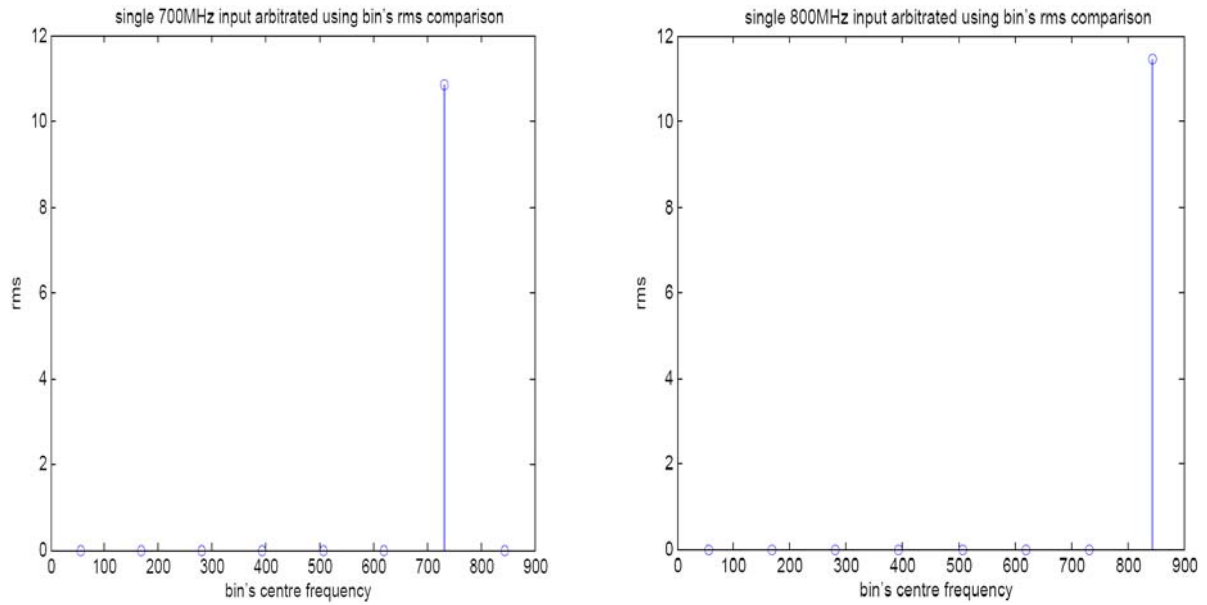


Figure 4.10 Frequency bin arbitration using rms comparison for a single input frequency.

As illustrated in figure 4.11 below, bin's rms comparison for single input pulsed frequency also works very well.

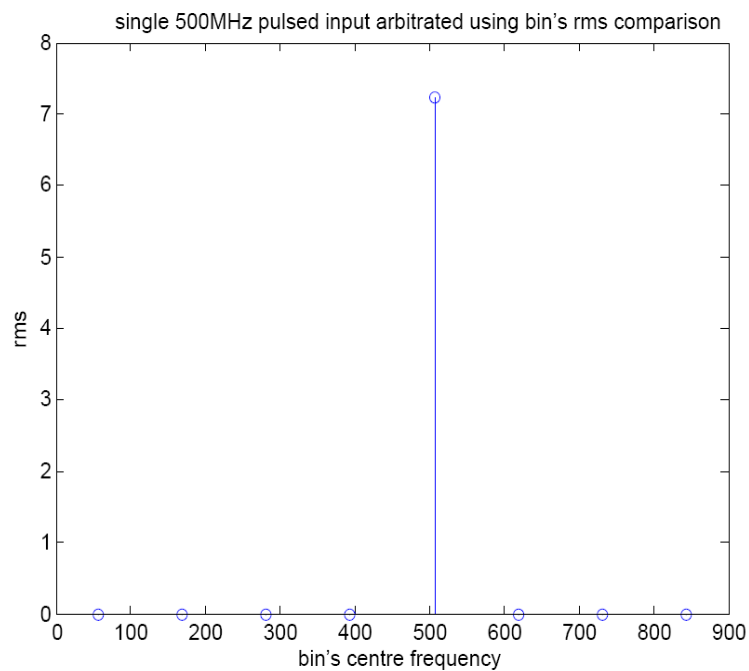


Figure 4.11 Single pulsed input frequency arbitrated using bin's rms comparison.

The bin's rms comparison for single input frequency bin arbitration may now be modified to resolve multiple input signals for simultaneous detection. This of course assumes that the signals with the strongest rms magnitudes will be appropriated as detected signal while those of weak magnitude will be rejected as incorrect signal from channel cross talk. This technique frequency bin resolution technique has the inherent limitation of possibly rejecting legitimate weak input frequencies that happens to have the same or smaller rms magnitude when compared to a cross talk signal of much stronger magnitude. A MATLAB example will shortly illustrate this short coming. The algorithm as presented in appendix A section A.1.1.2 is modified to resolve two input frequencies and its results is illustrated in figure 4.12 below for the same input frequencies for figures 4.7 and 4.9. In figure 4.12, it is definitively clear which channel bin the input frequencies reside in while figure 4.7 and 4.9 there appear to be signals in 4 frequency bins respectively (assuming rms magnitude of 1 or less is considered to be noise).

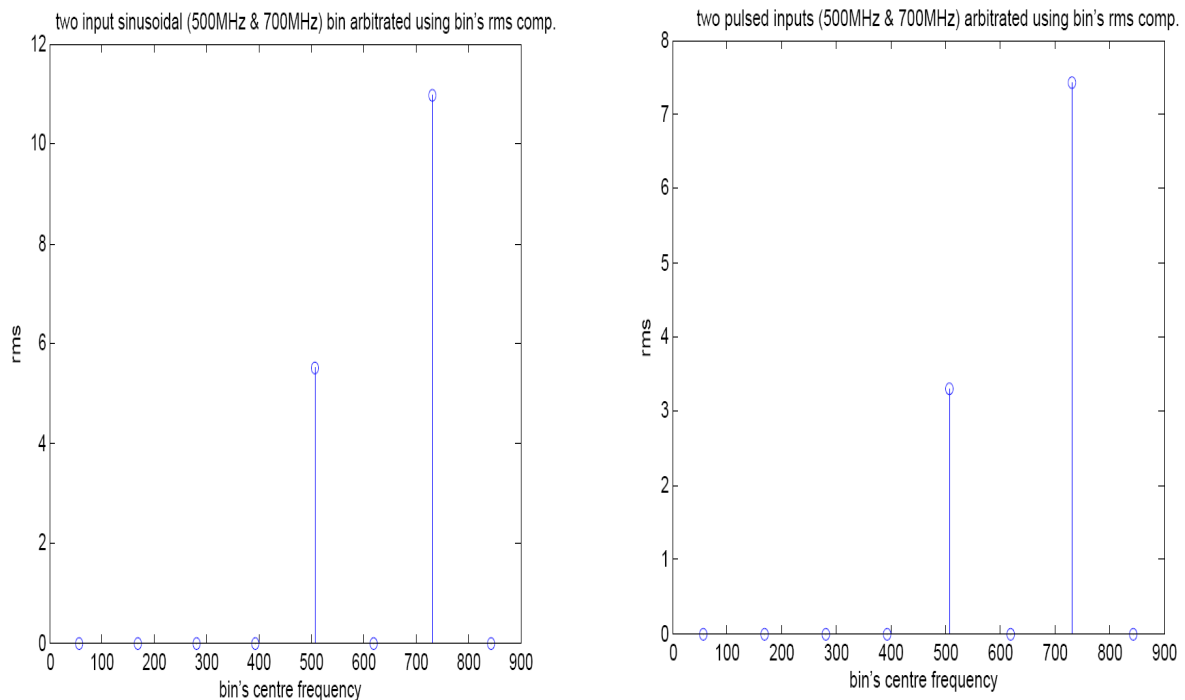


Figure 4.12 Channel frequency bin arbitration of two inputs using bin's rms comparison.

4.1.4 Example of incorrect channel arbitration using bin's rms comparison:

As mentioned before, channel arbitration using bin's rms magnitude comparison is vulnerable to cases when a strong input signal spills energy into adjacent channel band that is stronger than the rms magnitude of another signal. The spectrum of such a scenario is shown in figure 4.13 for a 700MHz input of magnitude 1.02 and 500MHz input of magnitude 0.26. Clearly illustrated in the figure is the cross talk signal from the 731.25MHz (700MHz input) bin signal appearing in the 618.75MHz bin with rms magnitude slightly larger than that of the 506.25MHz (500MHz) bin signal. This scenario causes the rms magnitude comparison approach to break down and incorrectly arbitrate the two input signal into the 731.25MHz and 618.75MHz bins as illustrated in figure 4.14. Obviously, the signal in the 618.75MHz bin is actually the 700MHz signal and hence this approach discards the legitimate weaker 500MHz. There is not much one can do about this rms magnitude comparison limitation.

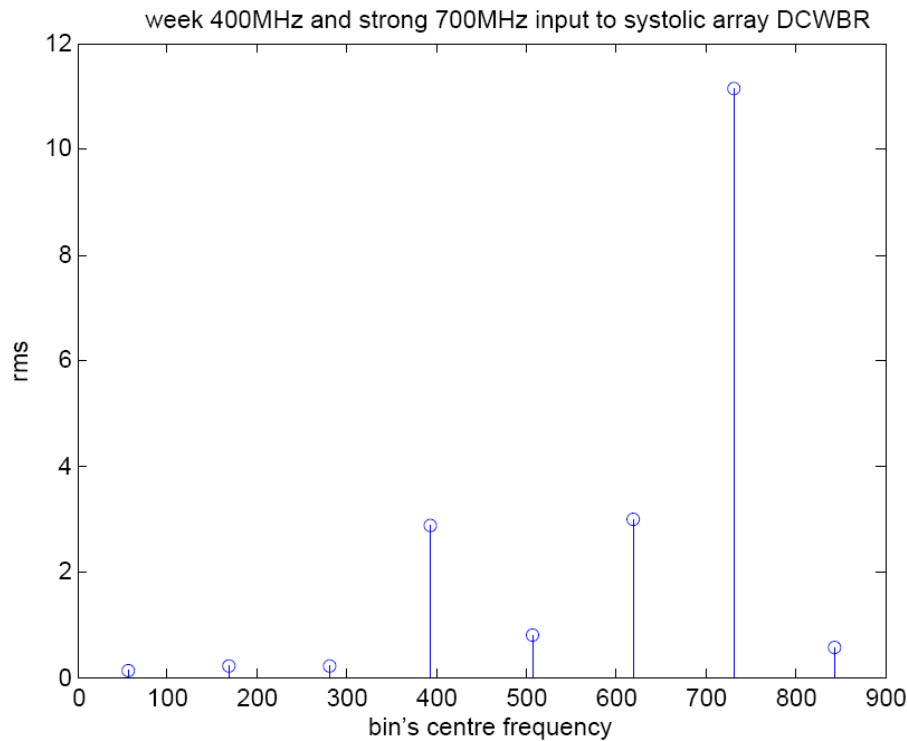


Figure 4.13 Systolic array DCWBR spectrum for a weak and strong input signals.

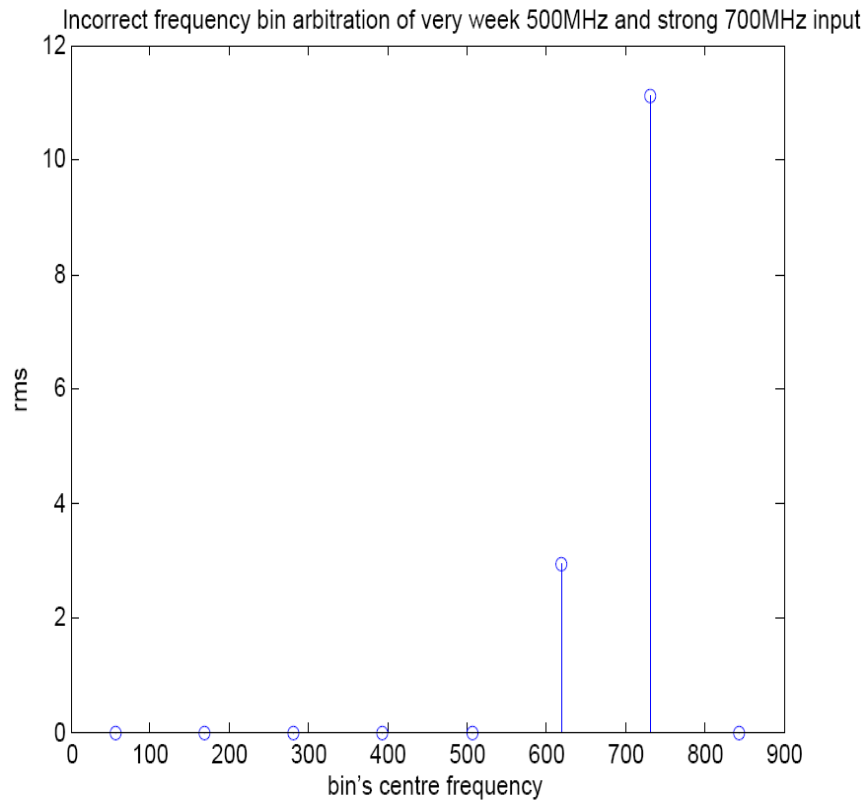


Figure 4.14 Incorrect bin arbitration using rms comparison for systolic array DCWBR.

4.1.5 Parallel Systolic Array DCWBR for increased channel arbitration:

Another method to increase frequency resolution and arbitration performance is to invoke parallel operation of the Systolic Array DCWBR. An example of this system is shown in figure 4.15 below. The receiver in figure 4.15 has three identical (including filter coefficients) Systolic Array Multi-Rate Filter banks.

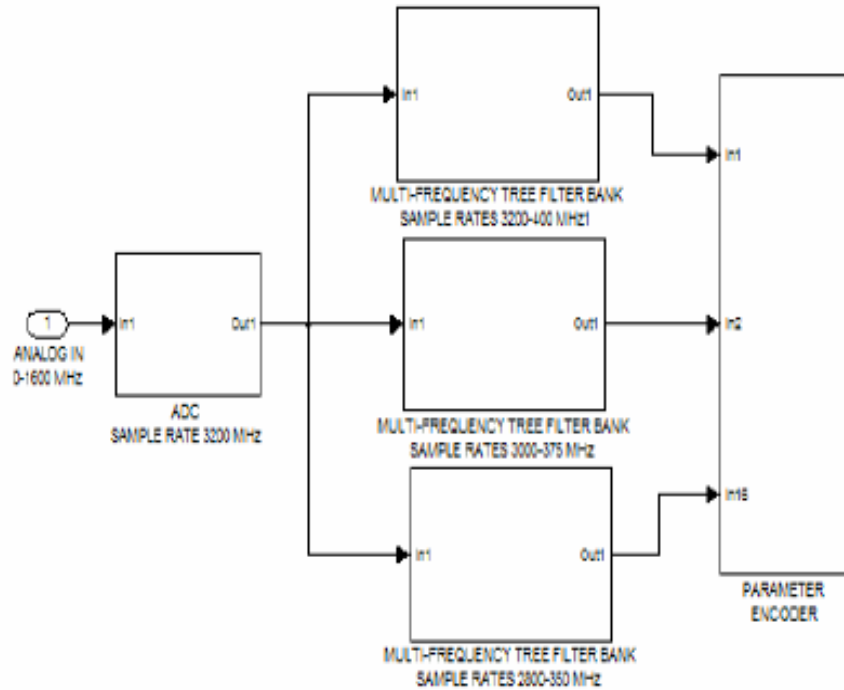


Figure 4.15 DCWBR with three Systolic Array Multi-Rate FIR Filter Banks.

In this example, the first bank operates with the first two filters at 3.2GHz and steps down to tier four with 16 filters operating at 400MHz. The second filter bank operates with the first two filters at 3.0GHz and its tier four of 16 filters operating at 375MHz. Finally, the third filter bank operates with the first two filters at 2.8GHz and its tier four of 16 filters operating at 350MHz. Consequently, the three filter banks each have 16 frequency bins with different spacing. The first filter bank has 16 bins of 0 – 100MHz, 100 – 200MHz, ..., 1500 – 1600MHz. The second filter bank also has 16 bins of 0 – 93.75MHz, 93.75 – 187.5MHz, ..., 1406.25 – 1500MHz. the thirds tier has 16 bins of 0 – 87.5MHz, 87.5 – 175MHz, ..., 1212.5 – 1400MHz. The outputs from all the 48 bins are collected and the

rms value of each bin is computed. The combined output of the parallel systolic Array DCWBR from 0 – 87.5MHz would be the rms sum of the first bin from each bank. These combinations of collecting three outputs to formulate one bin level for each of the 16 channel bins results in increased resolution and arbitration compared to a single systolic Array DCWBR.

4.1.6 MATLAB Performance of Parallel Systolic Array DCWBR Architecture:

Figure 4.14 illustrate an example of incorrect channel arbitration using the rms comparison approach. We now illustrate in figure 4.16 below correct channel arbitration (for identical inputs used in figure 4.14 example) using the systolic array methodology described above.

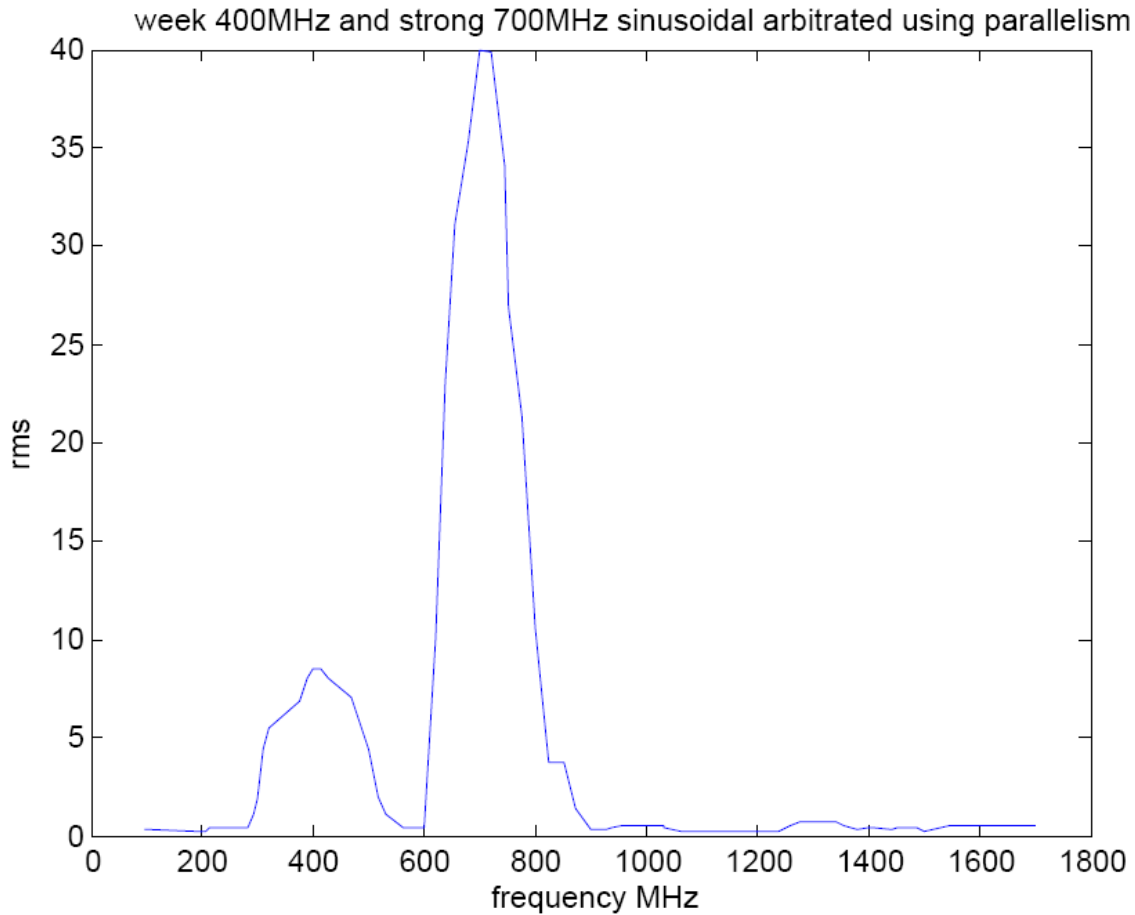


Figure 4.16 Reliable channel arbitration using parallelism of systolic array DCWBR.

Using parallelism resulted in better channel arbitration and hence better signal parameter estimation. However, this increased resolution comes at the cost of increased power dissipation and area. Using the rms comparison approach will result in less area and power dissipation but may be less reliable and potentially slower.

5. Circuit Designs of Basic Building Blocks

5.1.0 D Flip-Flop (DFF):

The DFF is the basic storage element used in the registers to perform pipelining. The DFF design is shown in figure 5.1a below while figures 5.1b and 5.1c shows the setup and hold time verification using a 1.9GHz clock. The latter figures clearly shows that the setup and hold times occurs around the triggering edge of the DFF. The setup and hold times for a DFF operating at a much slower speed occurs around the sampling clock edge of the DFF. Table 5.1 below shows the characteristics of this DFF design. The setup time is 84.25ps while the hold time is 16.1ps.

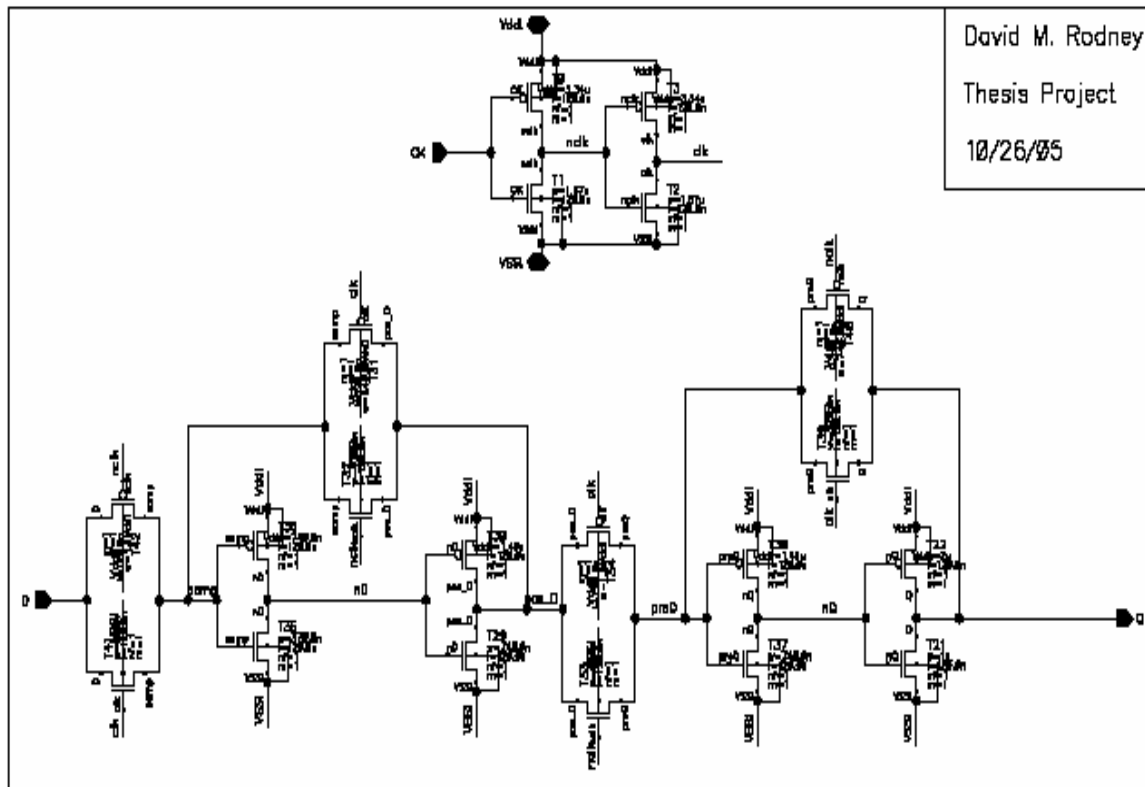


Figure 5.1 The DFF design.

Sizes:

PMOS of clock buffer: 3.34 μ m

NMOS of clock buffer: 1.67 μ m

PMOS of pass & hold gates: 1.8 μ m & 740nm

NMOS of pass and hold gates: 900nm

PMOS of inverters 1,2, 3: 0.740 & 1.49 μ m

NMOS of inverters 1,2, 3: 0.740 & 1.49 μ m

PMOS of output inverter: 2 μ m

NMOS of output inverter: 1 μ m

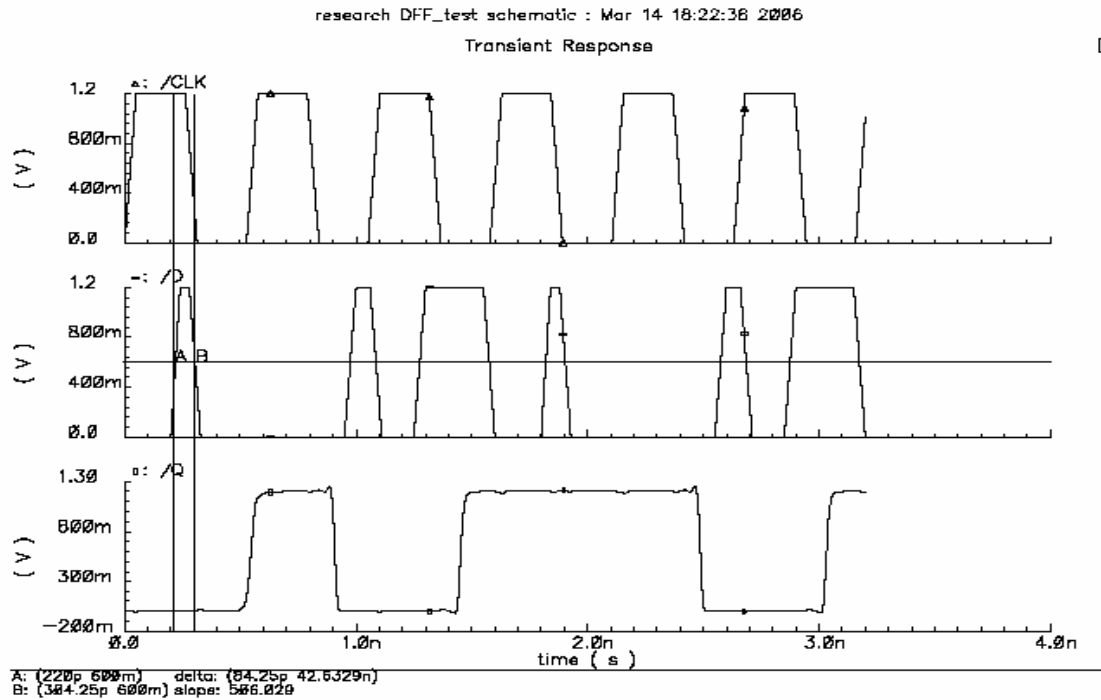


Figure 5.1a Setup time requirement of the DFF.

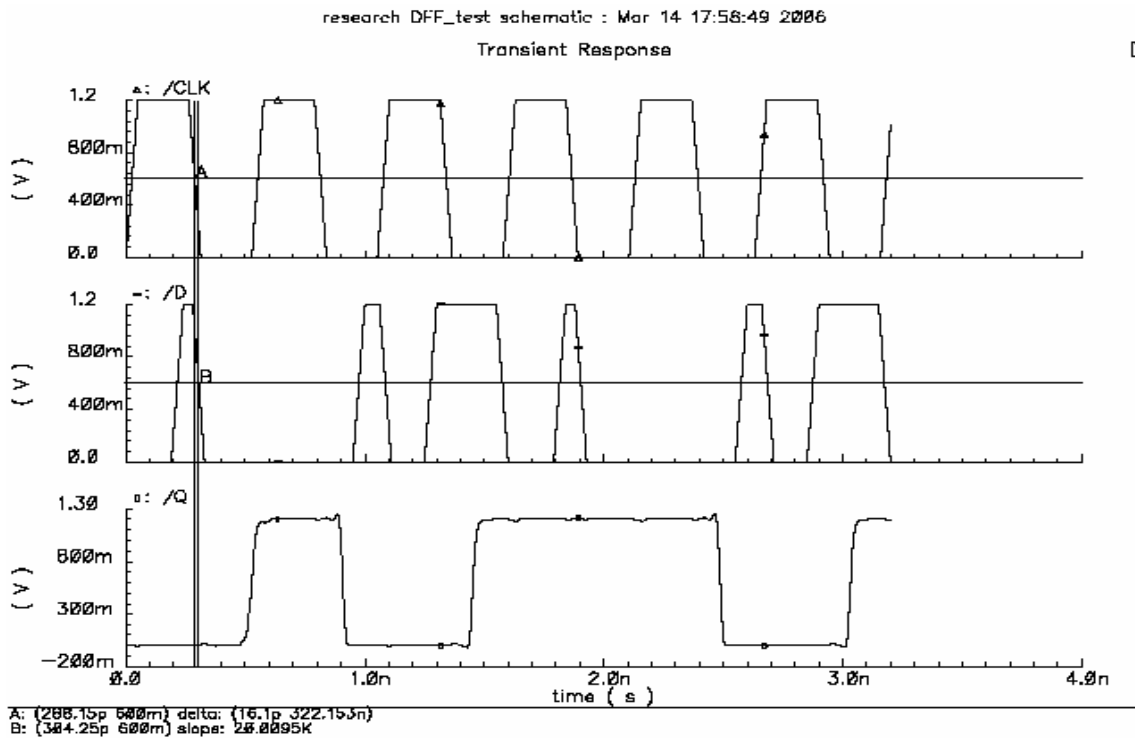


Figure 5.1b Hold time requirement of the DFF.

Propagation delay shown below is measured from the triggering edge of the DFF which is the falling edge of the clock. The power dissipation of the DFF is 141.29 μ W.

	Rise time	Fall time	Propagation delay, rise	Propagation delay, fall
Q	42.48ps	21.25ps	111.768ps	94.11ps

Table 5.1 Characteristics of the DFF.

5.1.1 The Half Adder:

A half adder with inputs A & B have outs S, sum and C, carry –out are describe by equations 5.1 and 5.2 respectively.

$$S = A \oplus B \quad \text{Eq. 5.1}$$

$$C = A \cdot B \quad \text{Eq. 5.2}$$

Figure 5.2 below shows the design of the half adder, in addition, figures 5.2a shows the verification of the half with the clocked input at 1.9GHz.

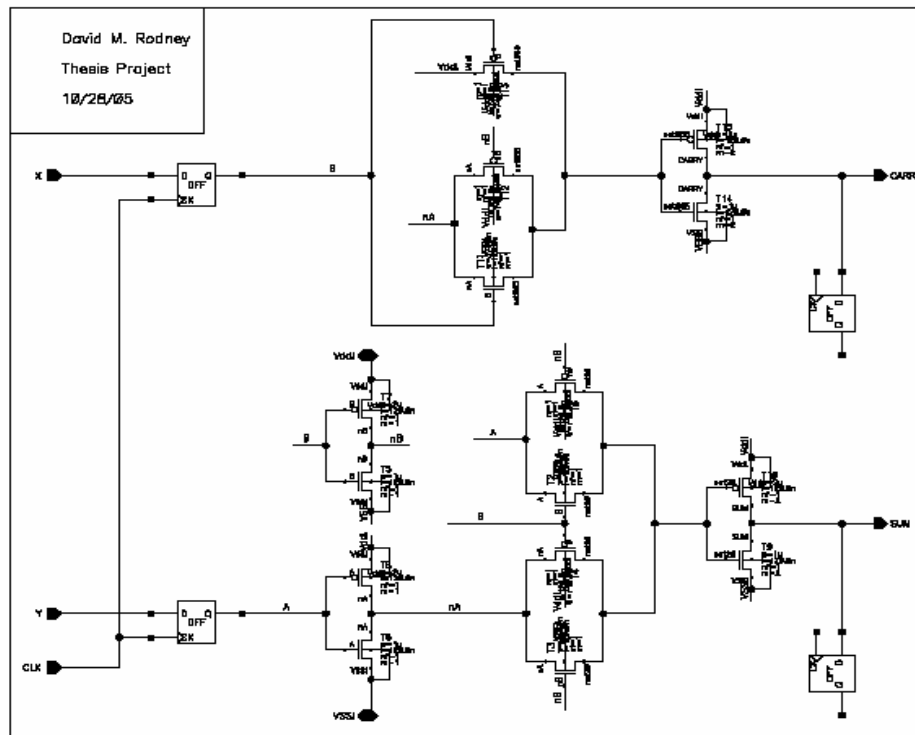


Figure 5.2 The half adder

Sizes: All PMOS are 740nm and all NMOS are 370nm.

Verification/validation clocking will be 1.9GHz throughout this thesis as the critical path component can only operate at this speed. The D flip-flop inputs are used to provide realistic rise and fall times of the inputs for verification purposes while also providing an output load. Table 1 below shows the characteristics of the half adder. The power dissipation reported in table 1 includes the power consumed by the two clocked D flip-flop as illustrated above. Naturally, the power consumption of the D flip-flops must be excluded to determine the half adder's power consumption. The half adder above is designed with 15 transistors.

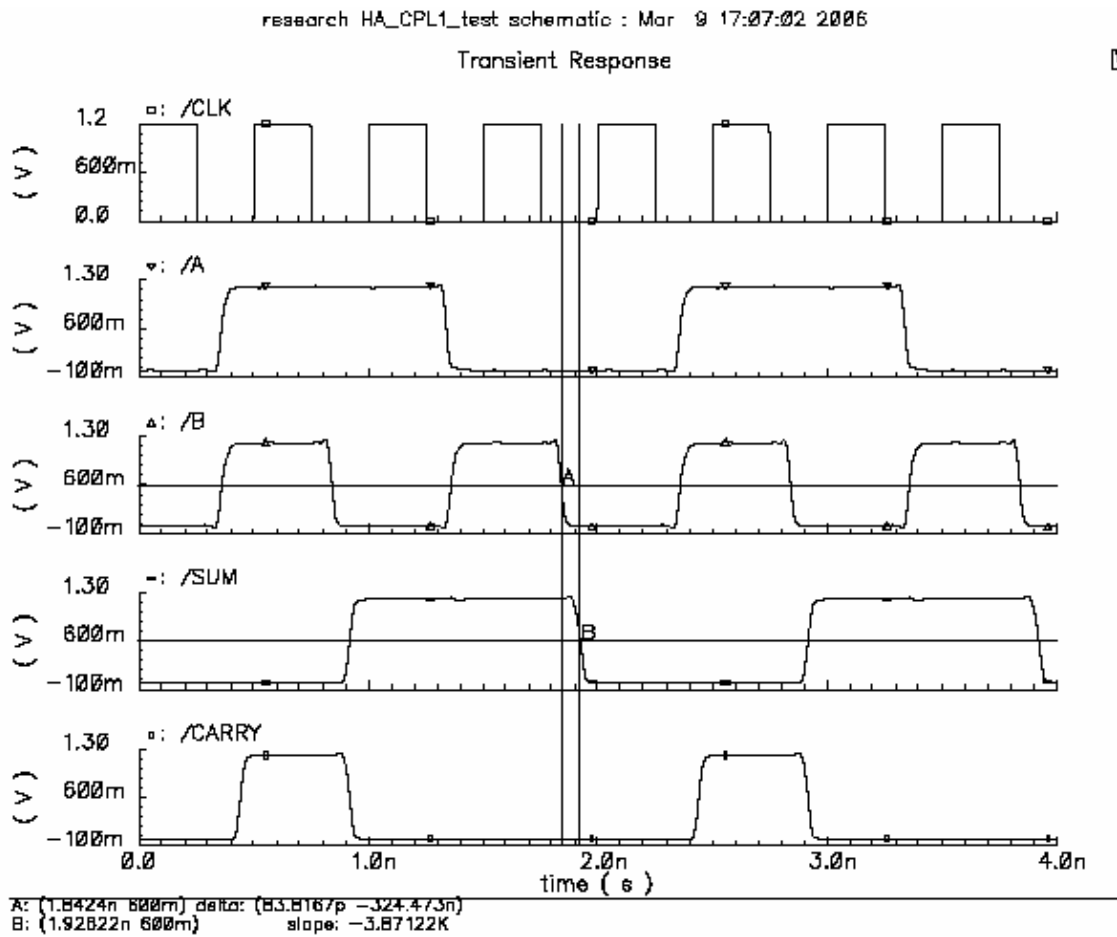


Figure 5.2a Half adder verification.

	Rise time	fall time	Propagation delay, rise	Propagation delay, fall
SUM	36.95ps	42.51ps	76.2659ps	83.82ps
CARRY	46.45ps	34.64ps	81.04ps	76.44ps

Table 5.2 Characteristics of the half adder.

The power consumption of the above configuration is $335.46\mu\text{W}$ and hence the half adder's power consumption is $335.46\mu\text{W} - 2*141.29\mu\text{W} = 42.88\mu\text{W}$.

5.1.2 The Exclusive OR Gate (XOR) $\Rightarrow \oplus$:

XOR is described by equation 5.3 for two inputs A & B.

$$\text{XOR} = A\bar{B} + \bar{A}B \quad \text{Eq. 5.3}$$

Figure 5.3 below shows the design of the XOR gate and the corresponding verification is shown in figure 5.3a. The design topology is chosen because it uses 2-4 less transistors than static CMOS and consumes less power. Characteristics of the XOR gate is shown in table 5.3 below.

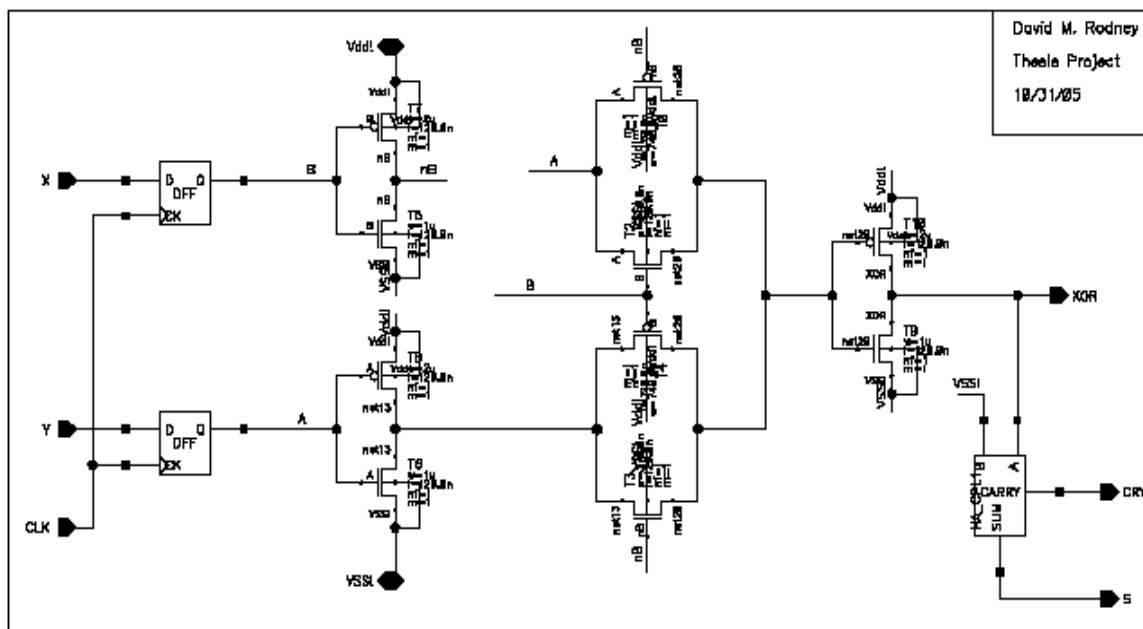


Figure 5.3 The XOR gate design.

Sizes:

PMOS for all inverter #1: 1.2 μ m
 PMOS for all pass gates are: 740nm
 PMOS for all other inverters: 740nm

NMOS for all inverters are: 370nm
 NMOS for all pass gates are: 370nm

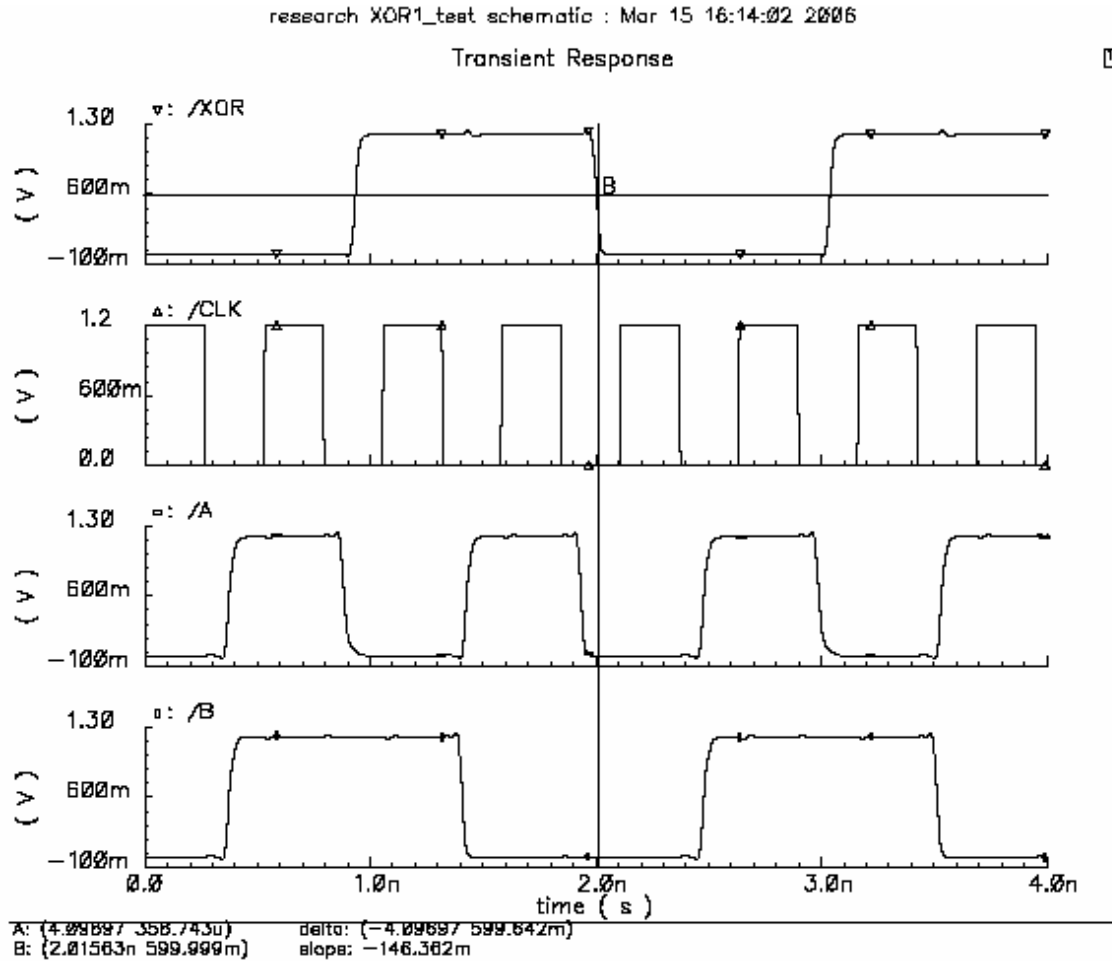


Figure 5.3a XOR gate verification.

	Rise time	fall time	Propagation delay, rise	Propagation delay, fall
XOR	42.61ps	38.85ps	67.53ps	69.18ps

Table 5.3 Characteristics of XOR gate.

Power consumption of the XOR gate is measured with all the above DFFs removed and the A & B inputs have 50ps rise and fall times. This configuration yields a power consumption of 48.52 μ W.

5.1.3 The Multiplexer (MUX):

The MUX is commonly refer to as steering logic thus its ability to select one input from a given sets of inputs and is define by equation 5.4 for a two input MUX.

$$\text{MUX_OUT} = A \cdot S + B \cdot \bar{S} \quad \text{Eq. 5.4}$$

The 2:1 multiplexer design is shown in figure 5.4 below and the corresponding verification is shown in figure 5.4a. The multiplexer is primarily used in the select adder design. The multiplexer's inputs are simulated with inputs having rise and fall times of 50ps and is illustrated in figure 5.4a below; in addition, the DFF is used as the multiplexer's load in this verification. Table 5.4 summarizes the properties of the multiplexer.

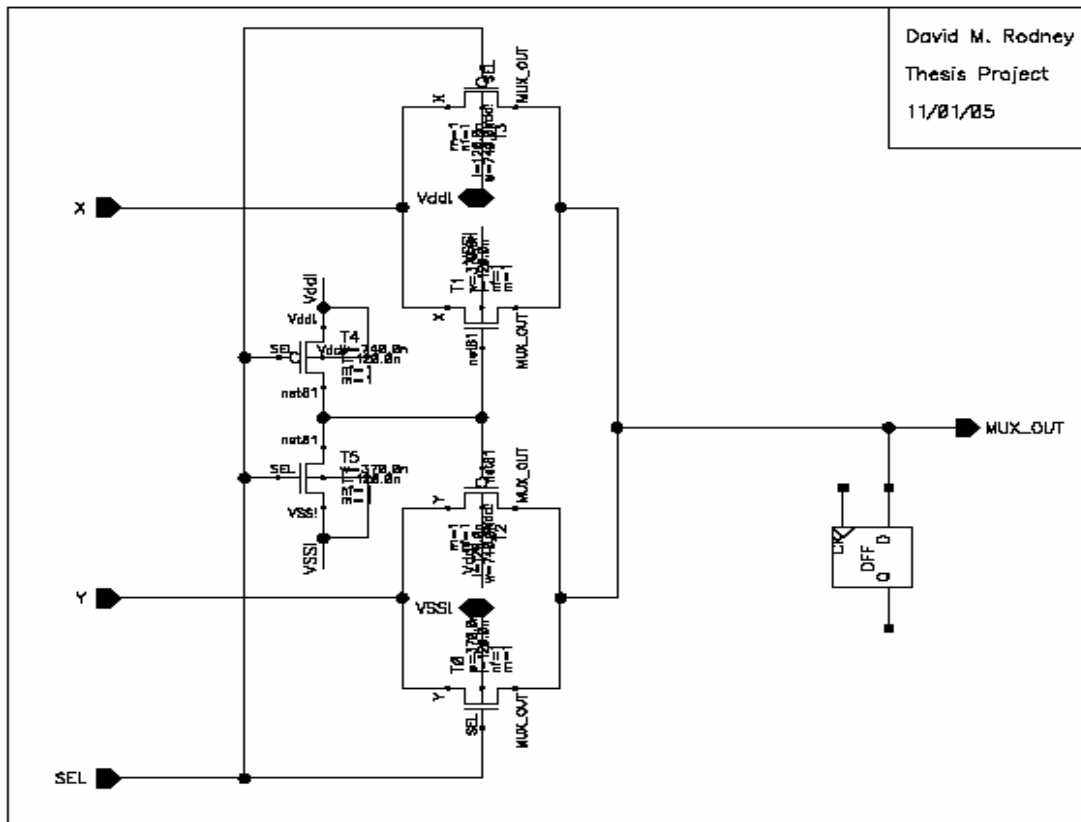


Figure 5.4 The 2:1 multiplexer design.

Sizes:

All PMOS sizes are 740nm.

All NMOS sizes are 370nm.

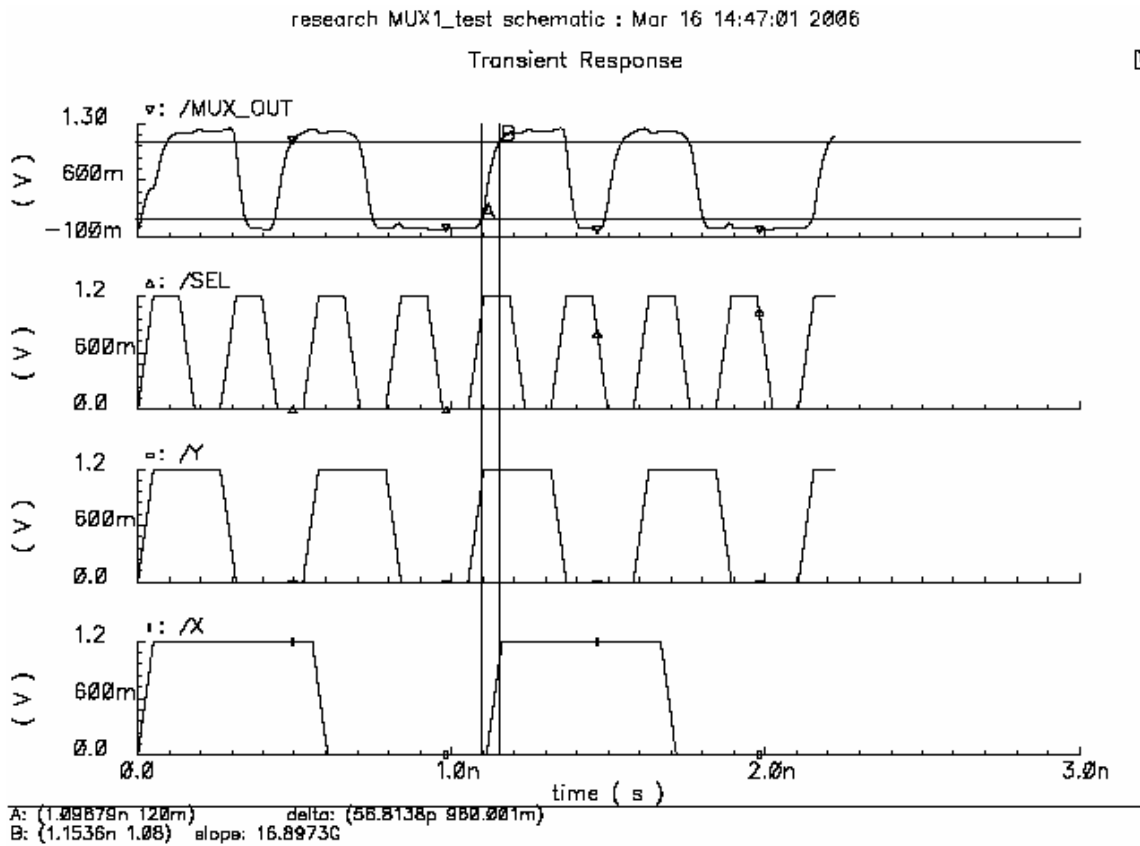


Figure 5.4a Verification of the multiplexer.

	Rise time	Fall time	Propagation delay, rise	Propagation delay, fall
MUX	56.52ps	39.18ps	39.8ps	35.33ps

Table 5.4 Characteristics of the multiplexer.

The power dissipation of the multiplexer with the above inputs and the DFF load removed is 25.92 μ W.

5.1.4 The AND Gate (·):

Figure 5.5 shows the design of the AND gate (under test) using static CMOS topology and figure 5.5a shows the verification of the AND gate. The AND gate will be primarily used for producing the product terms of the multiplier. Table 5.5 shows some basic properties of the designed AND gate. As is the common practice through this thesis, the inputs to the AND gate are driven by D flip-flops and the output is loaded with a full adder. This is clearly illustrated in figure 5.5 below.

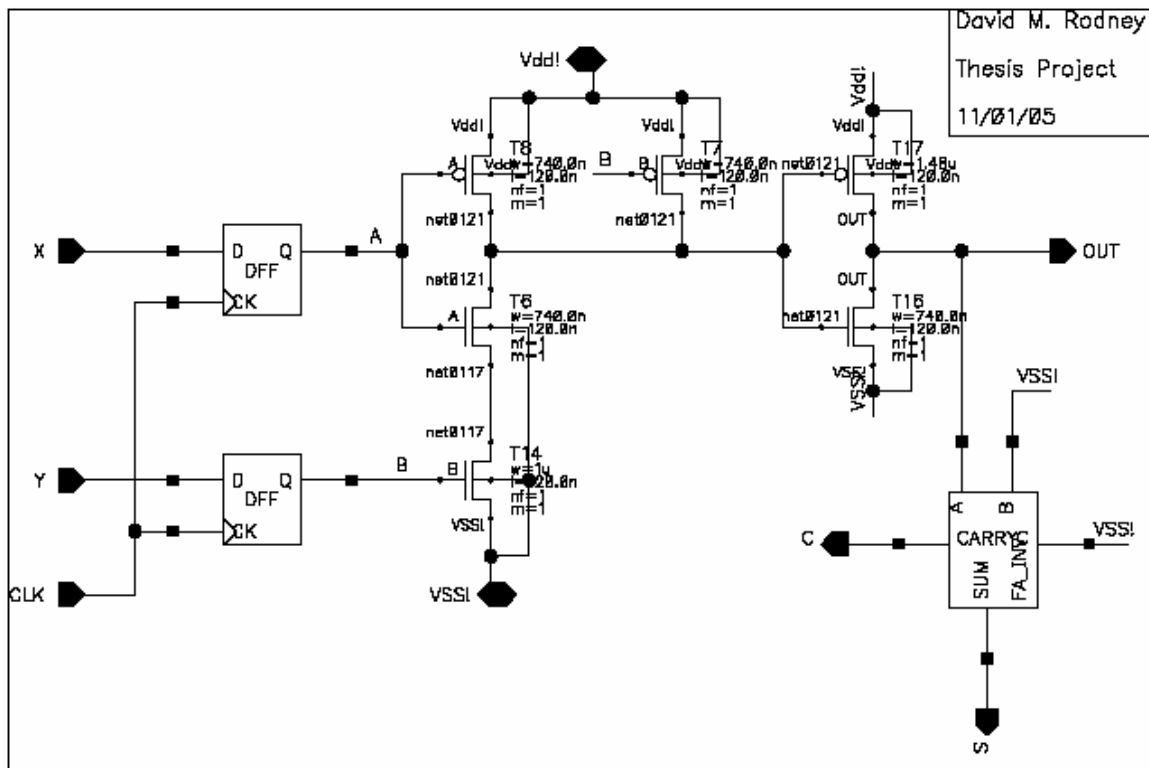


Figure 5.5 Static CMOS AND gate

	Rise time	fall time	Propagation delay, rise	Propagation delay, fall
AND	92.31ps	43.82ps	58.56ps	66.45ps

Table 5.5 Characteristics of AND gate.

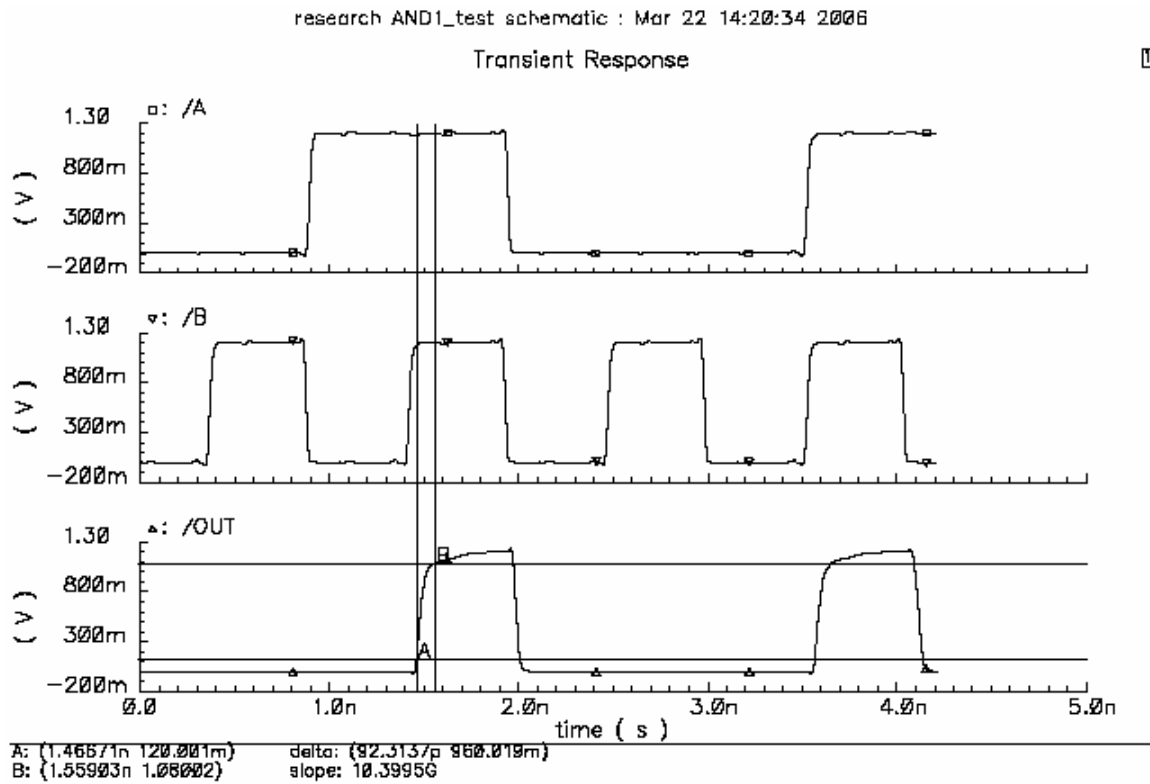


Figure 5.5a Verification of AND gate.

The power consumption of the AND gate with inputs of rise and fall times of 50ps and the DFF and output load removed is 10.04 μ W.

5.1.5 The Full-Adder (FA):

All digital signal processing application extensively use arithmetic operations such as addition, subtraction, and multiplication. The FA is the basic building block of these modules and therefore affects fundamental figure of merit such as speed of operation, area, and power dissipation. For this reason the novel 16-transistor CMOS 1-bit full-adder [5] is chosen. As noted in appendix B1.1.0 the FA is design with buffers on

the outputs to provide drive strength. This design is shown in figure 5.6 and the corresponding verification is shown in figure 5.6a. FA properties are shown in table 5.6. The commonly known equations for the SUM and CARRY outputs of the full-adder are shown in equations 5.5 and 5.6 respectively.

$$\begin{aligned} \text{SUM} &= ABC + A\bar{B}\bar{C} + \bar{A}B\bar{C} + \bar{A}\bar{B}C \\ &= (A \oplus B) \oplus C \end{aligned} \quad \text{Eq. 5.5}$$

$$\begin{aligned} \text{CARRY} &= AB + AC + BC \\ &= AB + C \cdot (A \oplus B) \end{aligned} \quad \text{Eq. 5.6}$$

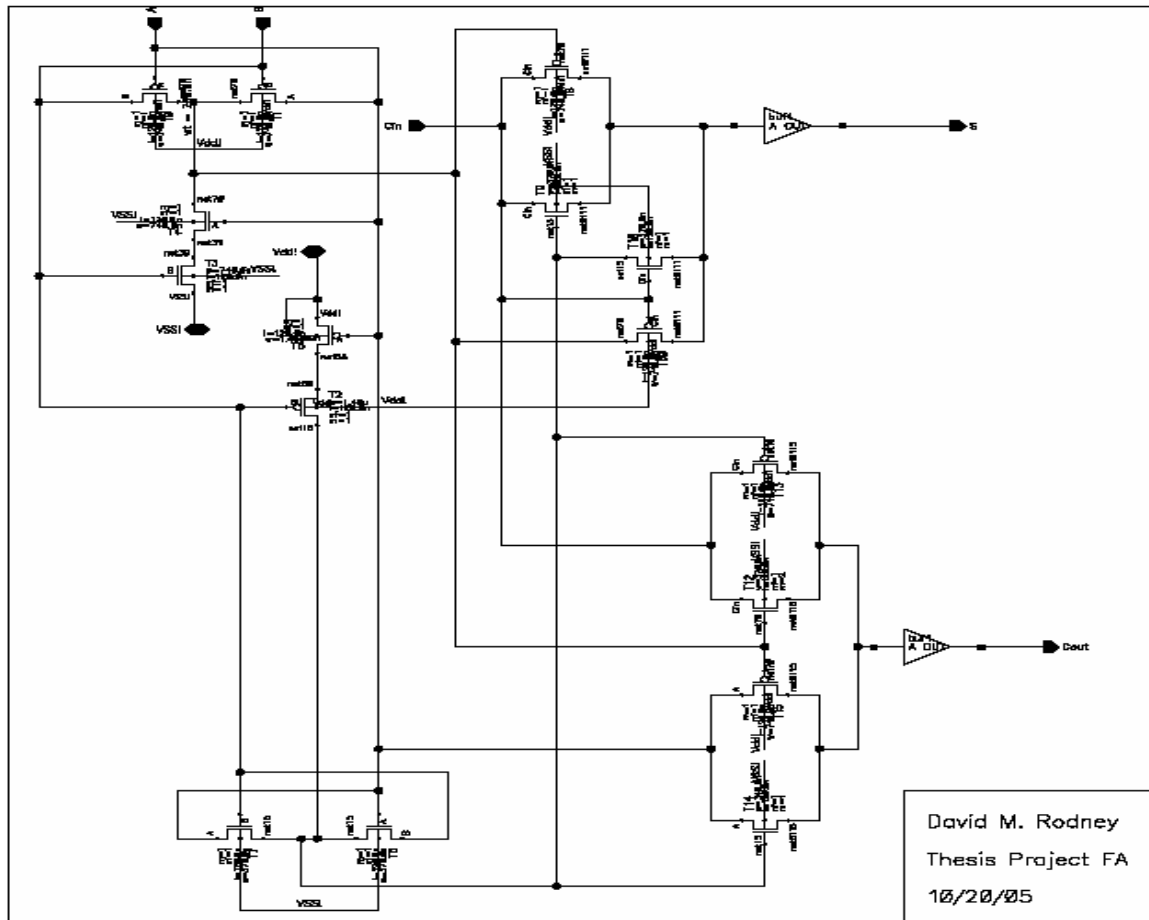


Figure 5.6 The designed full-adder.

Sizes:

All PMOS are 740nm except for the two vertical series PMOS which are 1.49μm each.

All NMOS are 370nm except for the two vertical series NMOS which are 740nm each.

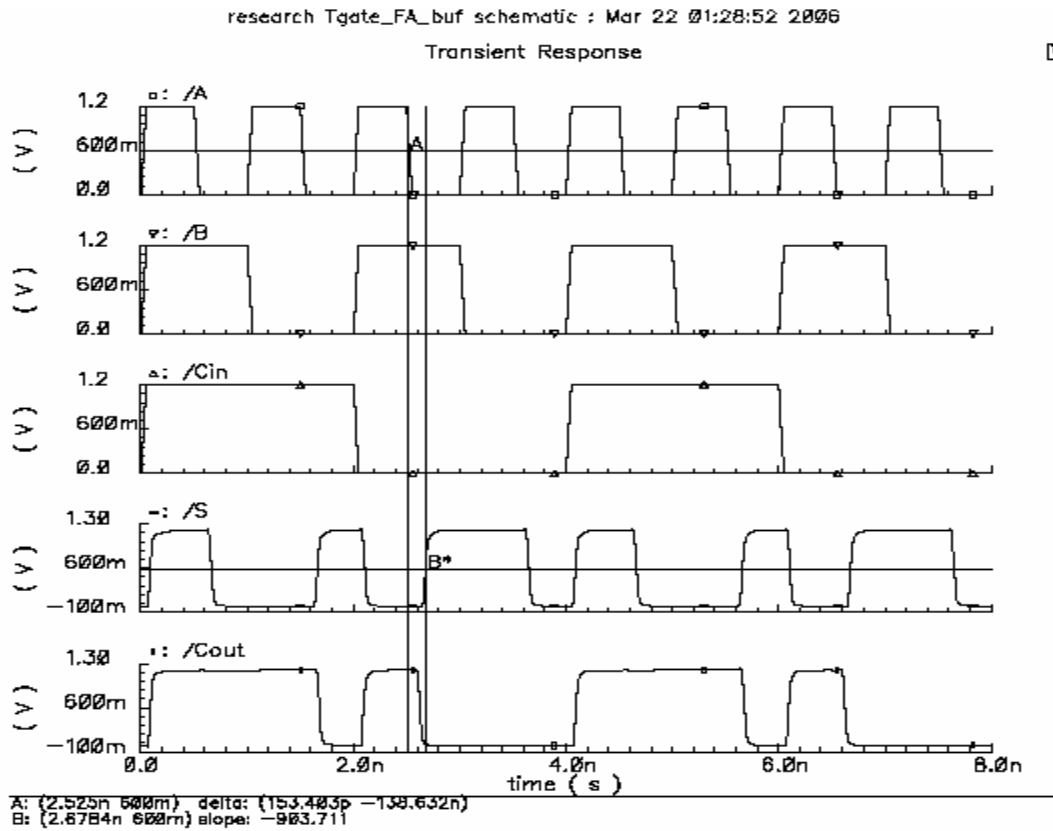


Figure 5.6a Verification of the full-adder.

	Rise time	Fall time	Prop. Delay, rise	Prop. Delay, fall
SUM, ave.	51.07ps	42.95ps	116.93ps	115.86ps
CARRY, ave.	48.9ps	42.65ps	67.06ps	127.68ps
SUM (worst)	54.08ps	43.49ps	153.32ps	137.97ps
CARRY (worst)	50.16ps	42.69ps	69.99ps	155.04ps

Table 5.6 Characteristics of the full-adder.

The power consumption of the FA with the above input stimuli under no load condition is 40.13 μ W.

5.1.6 The 7-bit adder:

Figure 5.7 below shows the configuration for the 7 bit 2s complement adder under test while figures 5.7b I & II shows the input combination used for validation and figure 5.7c shows the corresponding simulation output.

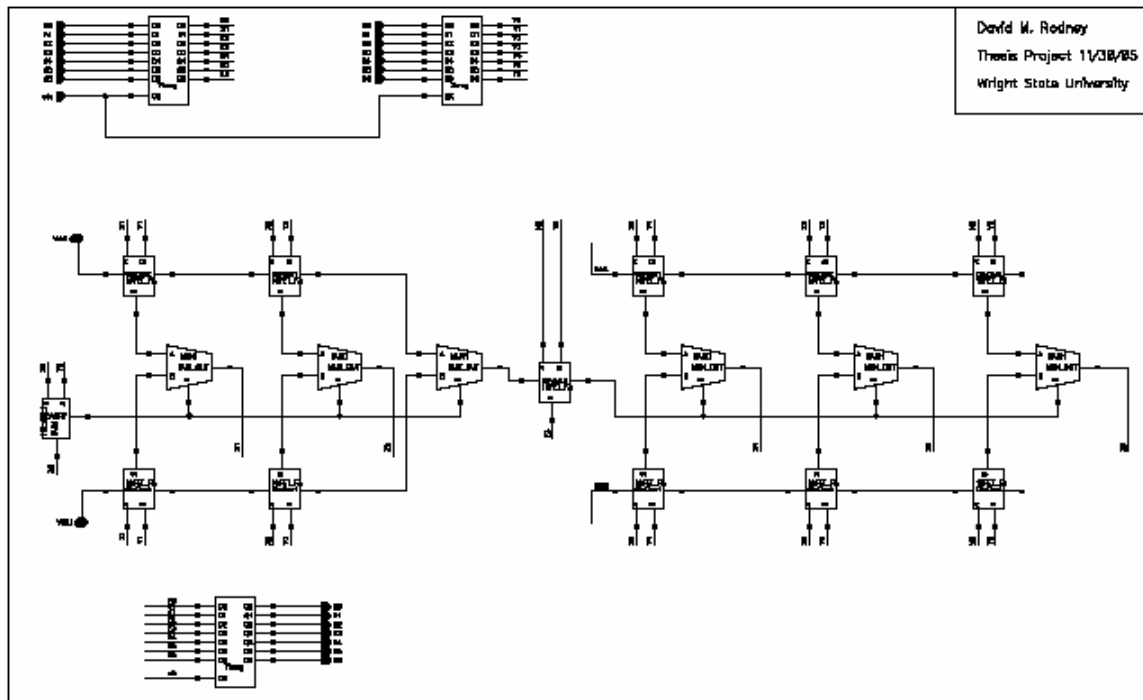


Figure 5.7 The 7bit adder.

The basic topology chosen above is a carry select configuration. The worst case delay for the adder shown above is expected to be the carry delay for the 2 half adders + sum delay of the full adder and 3 MUX delays. The 7-bit adder is functionally correct for clock operation of 1.9GHz as shown for the randomly selected vectors shown in figure 7b. Note that correct result is expected on the falling edge of clock after the second cycle. Table 5.7 below tabulates the input vectors and expected sums and indicates that the 7-bit adder produced the expected output. Figure 5.7c below shows the produced output using the above input vectors.

Figure 5.7b I and II shows the randomly selected vectors used for validation of the 7bit adder.

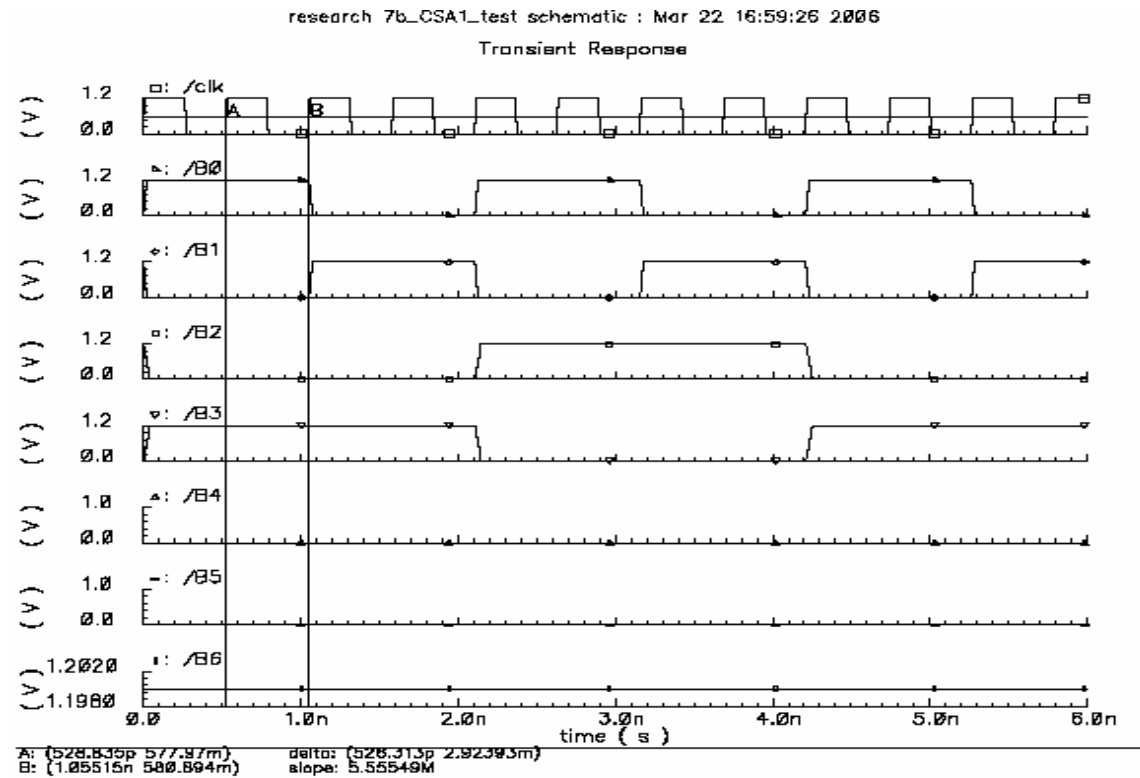
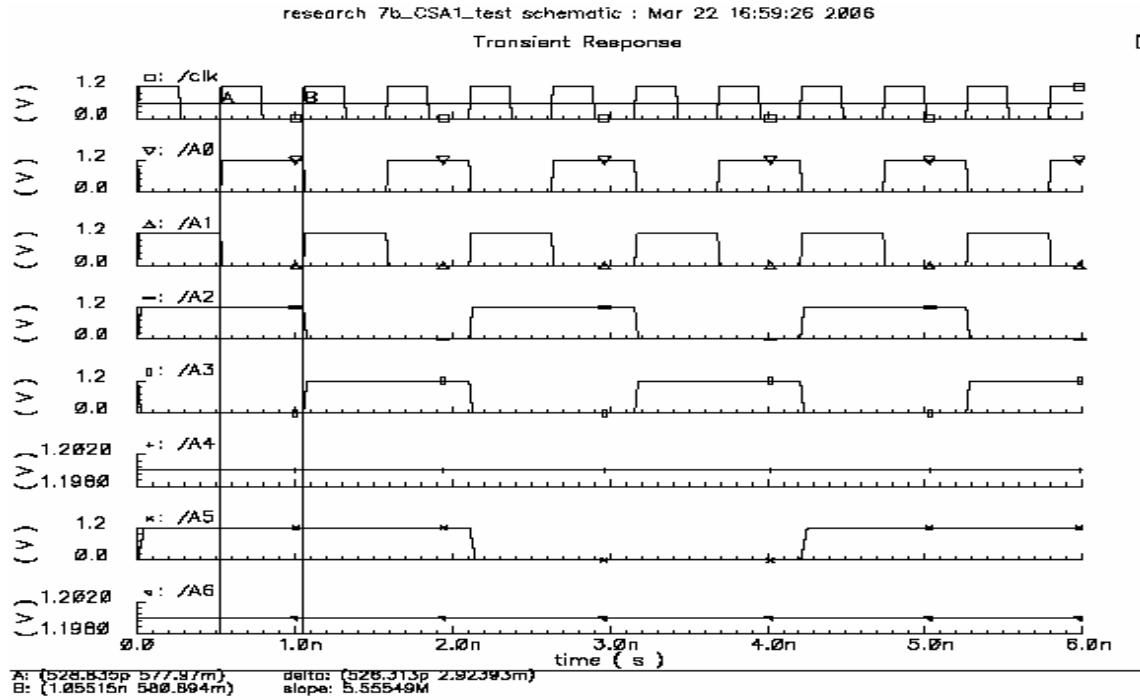


Figure 5.7a I & II A and B input vectors respectively.

Vector A	Vector B	SUM	7-bit adder's SUM
1110110	1001001	0111111	0111111
1110101	1001001	0111110	0111110
1111010	1001010	1000100	1000100
1111001	1001010	1000011	1000011
1010110	1000101	0011011	0011011
1010101	1000101	0011010	0011010
1011010	1000110	0100000	0100000
1011001	1000110	0011111	0011111

Table 5.7 Input vectors and expected sum for 7-bit adder.

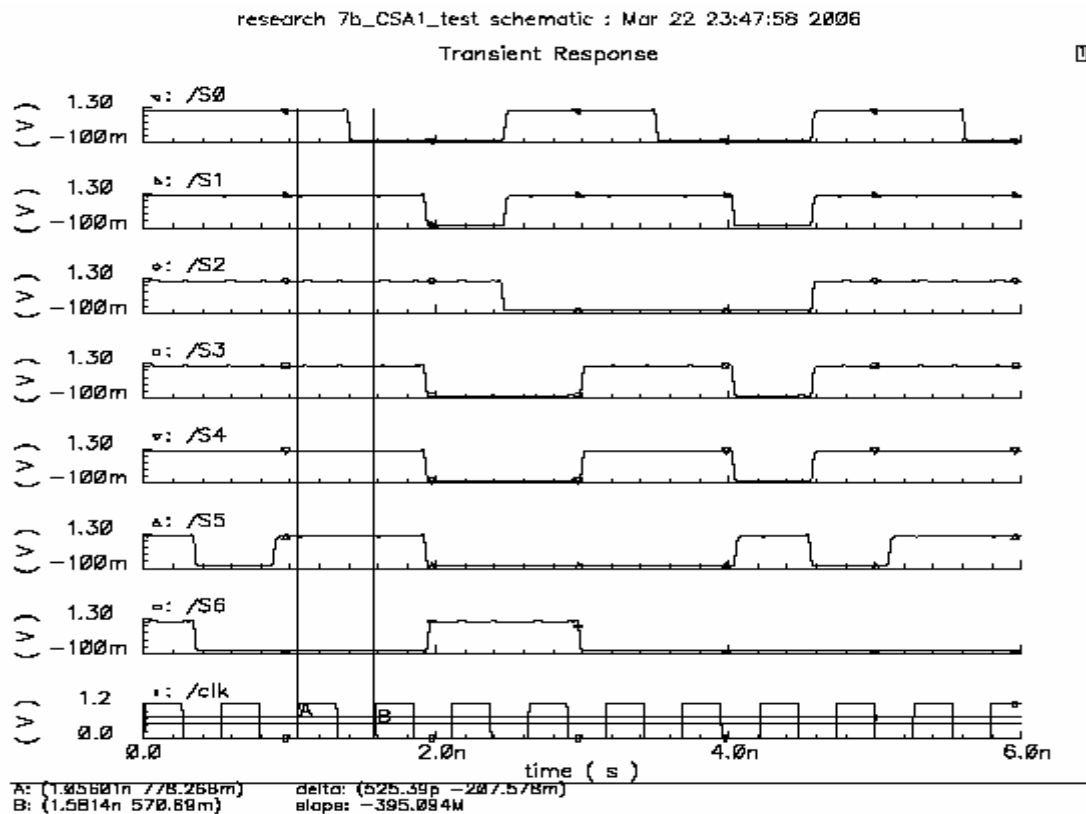


Figure 5.7b Correct output waveform for 7-bit adder for A & B vectors above.

The power dissipation of the 7bit adder with no load and input flip-flops using the above inputs with 50ps rise and fall times is 202.27 μ W.

5.1.7 The 7-bit 2's compliment:

For the purpose of avoiding 2's compliment multiplication, which involves 4 different cases because of the sign, signed magnitude multiplication is employed because less hardware is needed. Therefore, after 2's compliment addition, the number is converted to signed magnitude for multiplication by the appropriate filter coefficient. After this multiplication, 2's compliment operation is again performed to reconvert the multiplicative answer to 2's compliment. The circuit for 7-bit 2's compliment is shown in figure 7.8 below and the corresponding validation is shown in figure 7.8a & b which shows the validation input and resulted simulated output.

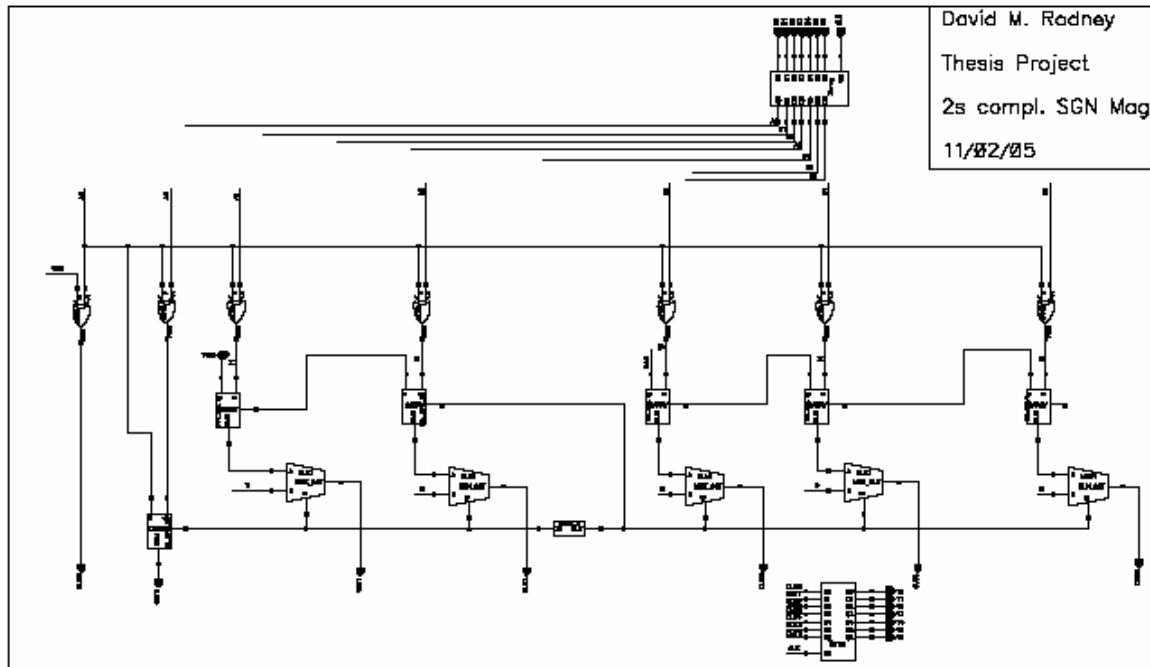


Figure 5.8 7-bit 2's compliment design.

The above schematic illustrates the classic 2's compliment operation of inverting all bits and adding 1 if the number is negative (MSB is 1). The MSB goes to all XOR gates. The left most XOR gate merely passes the MSB to the output unaltered as it is necessary to preserve the sign. If the MSB is 1 (implying a negative number) the output of all XOR gates produces the compliment of its respective input. Bit position 0 (the second left most) complimented input is added to the MSB resulting in the appropriate output. If the CARRY-OUT from this addition is a zero then the resulting complimented

output of the XOR gate is passed to the output via the MUX. A pass zero circuit, shown in figure 7.8c, also determines if the next upper 3 output bits merely gets the compliment of its respective input. When the CARRY-OUT is 1, the MUX chooses the half adder output of 1 added to the complimented respective input. If the MSB is 0, all inputs passes through the XOR gate unchanged. This condition results the CARRY-OUT the first half adder being 0 and hence passes the unchanged input to the output via the MUX. Figure 7.8a shows the inputs used for validating this circuit while figure 7.8b shows the resulting correct output Table 7.8 summarizes this validation exercise.

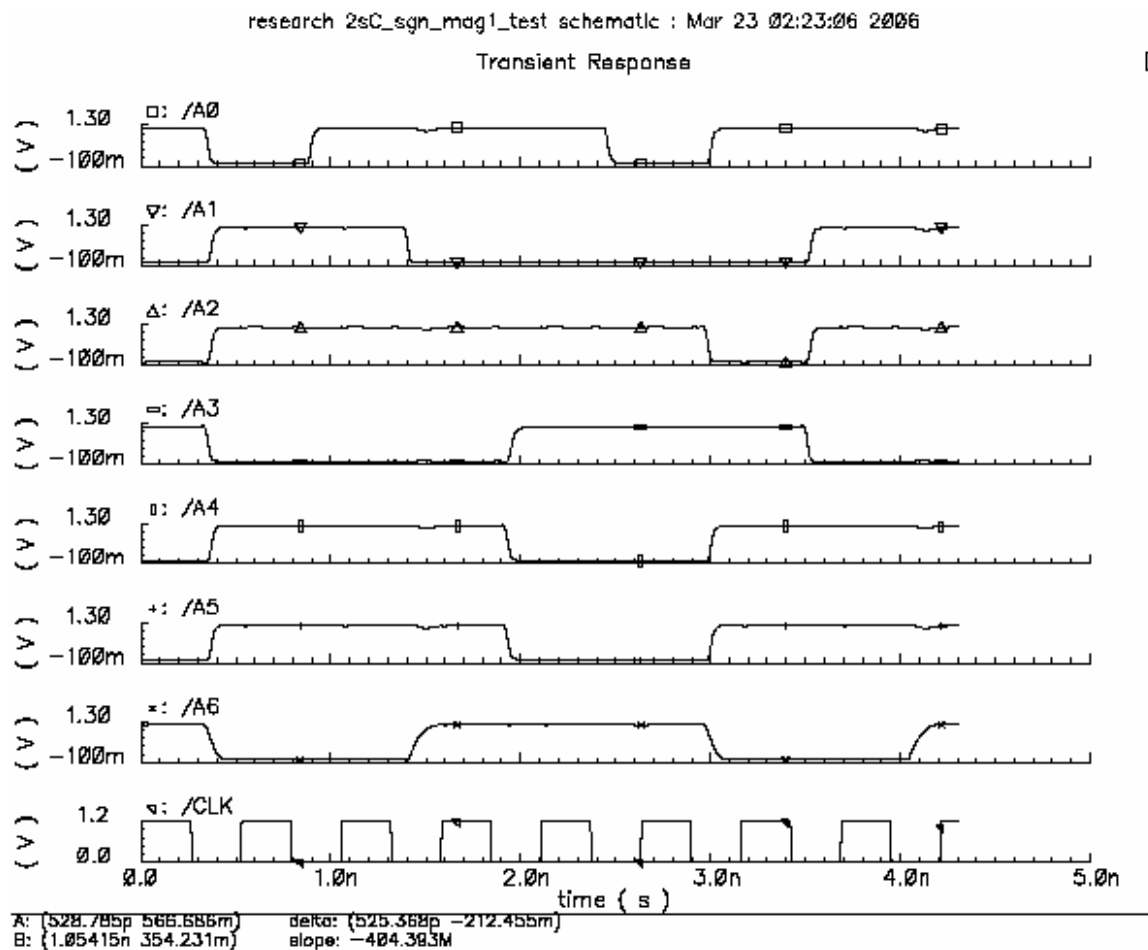
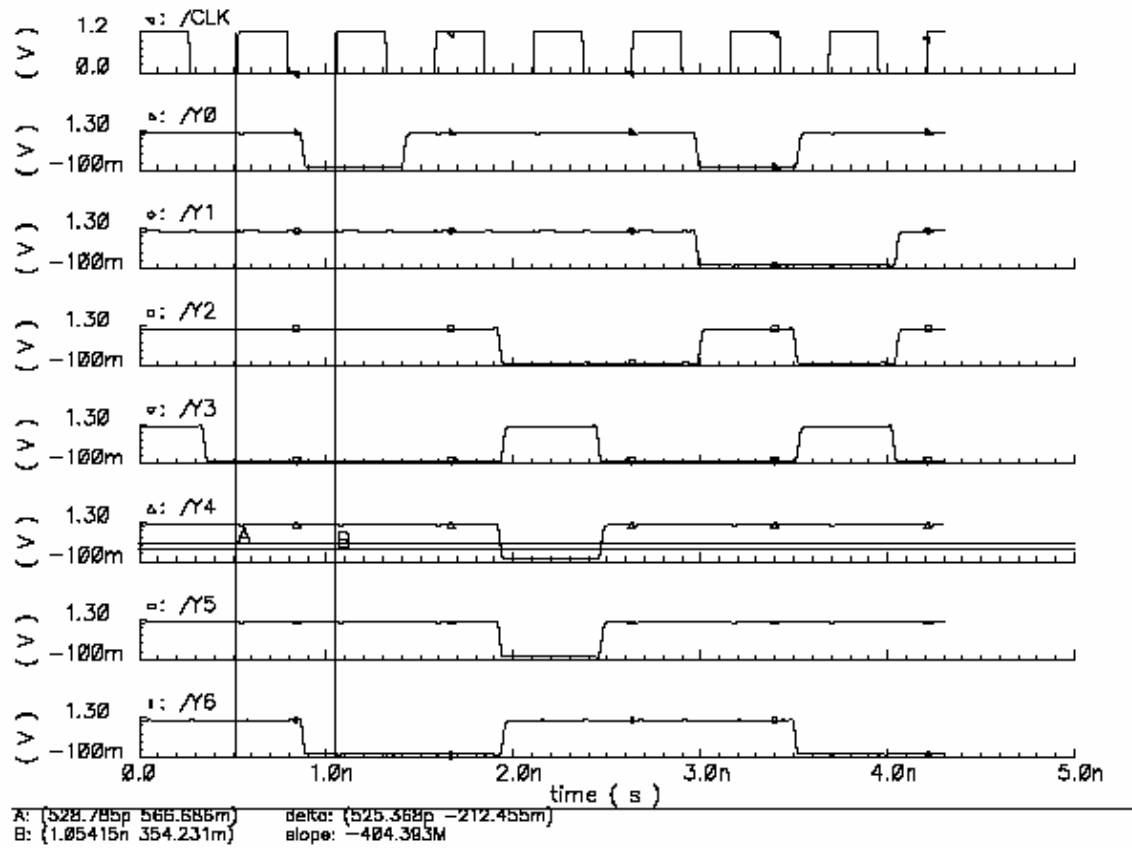


Figure 5.8a Random input vectors to validate the 2's complement circuit.

Transient Response



5.8b Correct output waveform from 7bit 2's complement circuit.

Vector A	Decimal Value	Sign Mag. Value	7-bit 2's compl. Output
0110110	+54	0110110	0110110
0110111	+55	0110111	0110111
1110101	-11	1001011	1001011
1001101	-51	1110011	1110011
1001100	-52	1110100	1110100
0111001	+57	0111001	0111001

Table 5.8 7-bit 2's complement validation summary.

The input flip flops shown in figure 7-2a above is clocked at 1.9GHz; also notice that half adders are used throughout this design. It is observed that the MSB goes to 7 XOR gates, this is a caused for concern as slow rise and fall time may cause timing failure or increase power dissipation. The rise & fall times of the MSB was measured and found to be 101.87ps and 139.68ps respectively. The power dissipation for the 7-bit 2's compliment without the input and output DFF and with all inputs having a 50ps rise and fall times was found to be 475.16 μ W.

The circuit below is a zero pass circuit used in the 7bit 2's compliment between output bit 2 and 3. When the input (carry out first adder) to the circuit is "1" the Out node is un-driven and thus the node is driven by carry out of the 3rd adder in the circuit. If carry out of the fist adder is a zero (input to pass zero circuit) then no carries will be generated and thus the other inputs are passed to the output. This resulted in a faster circuit for this case while not hurting the other case.

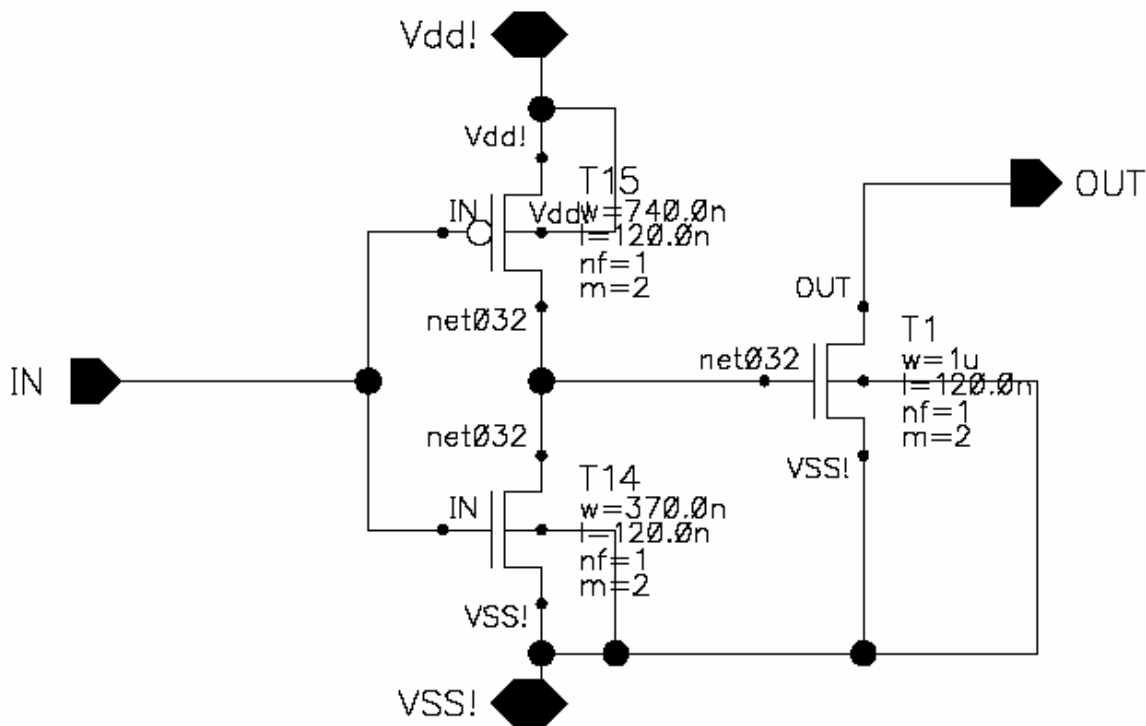


Figure 5.8c the pass zero circuit.

5.1.8 6bit multiplier:

The multiplier is another essential component to implement digital signal processing operations such as filtering as can be clearly infer from equation 1 where delay data samples are multiplied by the respective filter coefficient. Given that an N-bit multiplier produces 2N-bit results, this component may become the critical path component to attaining design speed goals. Naturally, the operating speed and latency/throughput of the multiplier may be influenced by the chosen multiplier architecture. A common multiplier architecture is the array multiplier where an n x n multiplier requires n(n-2) full adders, n half adders, and n^2 AND gates. The worst-case delay associated with this multiplier is $(2n+1)\tau_g$, where τ_g is the worst case adder delay [9]. Another multiplier architecture that seeks to take advantage of the 3:2 compression provided from the full adder is the Wallace tree multiplier. The delay of this multiplier is expected to be $\log_{1.5}N$ + delay of final merging adder, where N is the width of the multiplier [10]; this multiplier topology is therefore implemented. To achieve high speed of operation, a two stage pipeline is implemented as is illustrated in figure 5.8 below. Table 5.8 below tabulates the input vectors and the corresponding expected and obtained simulation outputs.

Input Vector, A	Natural number rep.	Expected output	Simulated output
110110	$63*54 = 3402$	110101001010	110101001010
110101	$63*53 = 3339$	110100001011	110100001011
111010	$63*58 = 3654$	111001000110	111001000110
111001	$63*57 = 3591$	111000000111	111000000111
100110	$63*38 = 2394$	100101011010	100101011010
100101	$63*37 = 2331$	100100011011	100100011011
101010	$64*42 = 2646$	101001010110	101001010110
101001	$63*41 = 2583$	101000010111	101000010111
010110	$63*22 = 1386$	010101101010	010101101010
010101	$63*21 = 1323$	010100101011	010100101011
011010	$63*26 = 1638$	011001100110	011001100110
011001	$63*25 = 1575$	011000100111	011000100111

Table 5.9 6-bit Wallace tree multiplier validation summary.

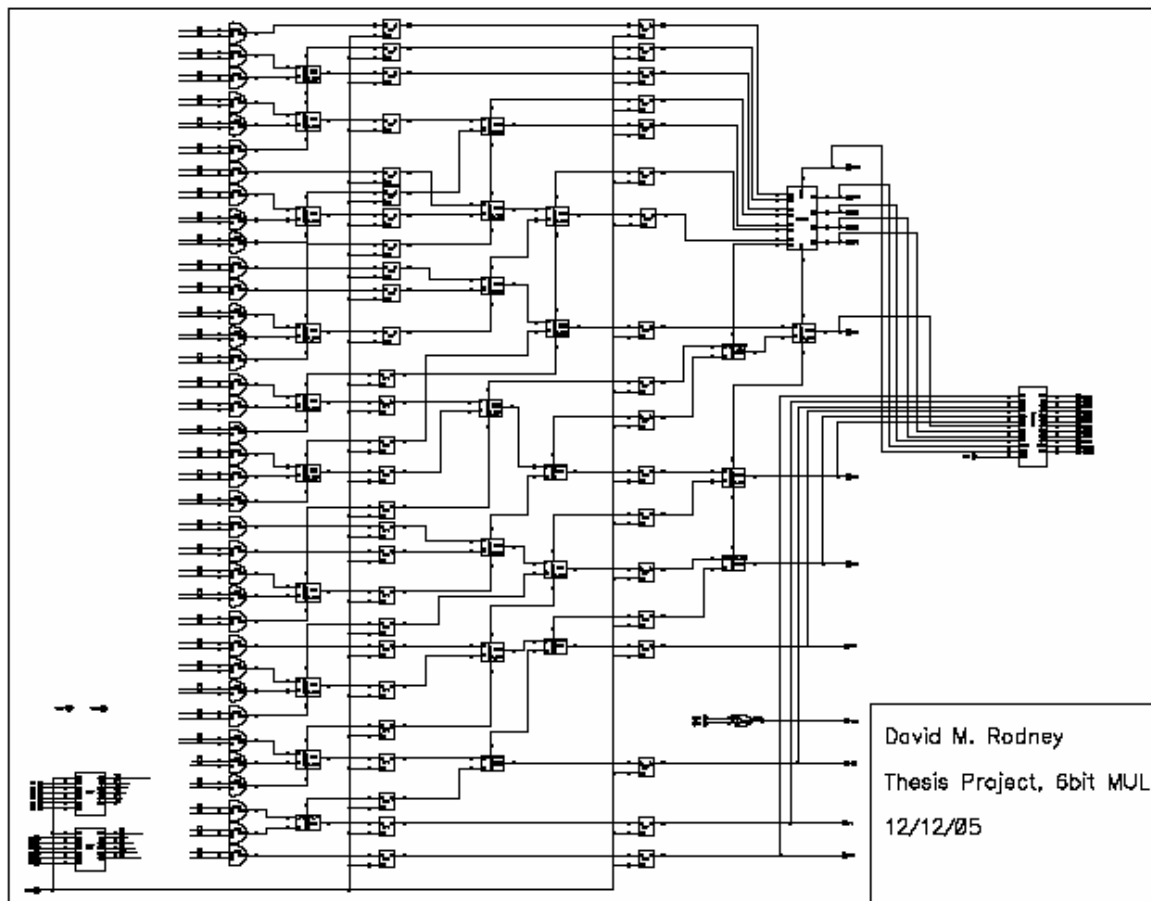


Figure 5.9 6-bit 2-stage pipelined Wallace tree multiplier.

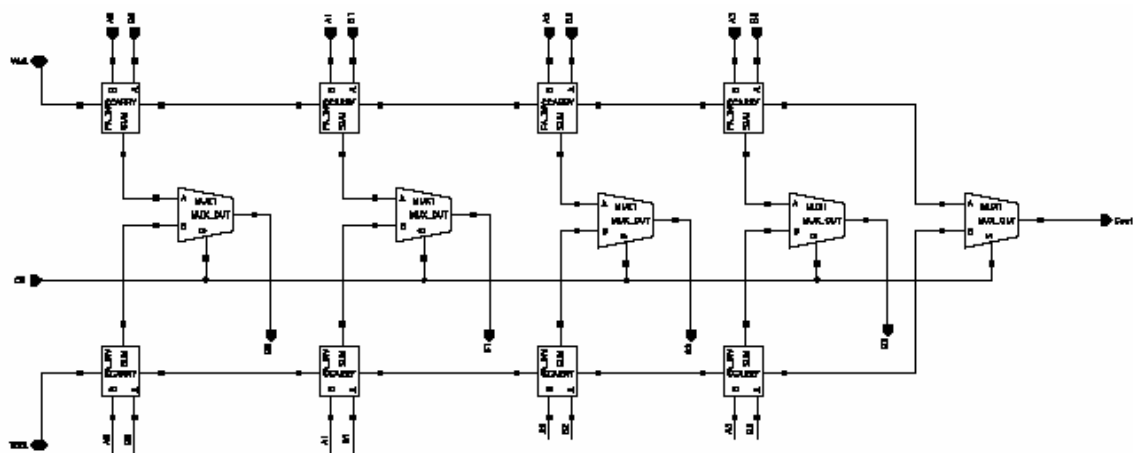


Figure 5.9a The 4-bit merging adder of the 6-bit multiplier.

For this validation exercise, the signed bit is turned off as it is a trivial case delayed by 2 clock cycles. The latency of the multiplier is 2 clock cycles as can be directly inferred from the number of piped stages. Figure 5.8b below shows the input waveform used for validation. The other input vector is all 1s and is therefore not shown here. The inputs are clocked at 1.9GHz.

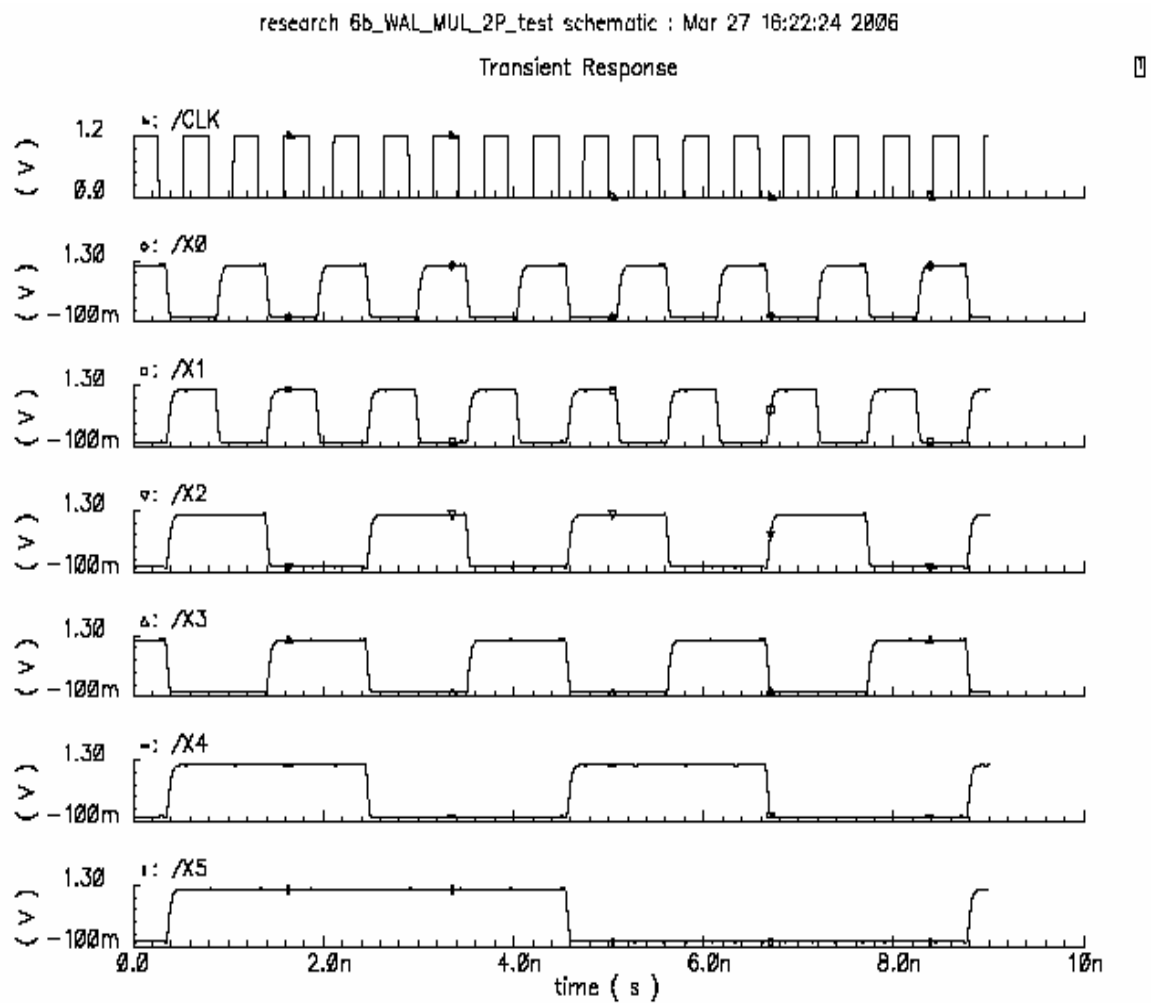


Figure 5.9b Clocked A inputs used for validating the multiplier; B inputs are all 1s.

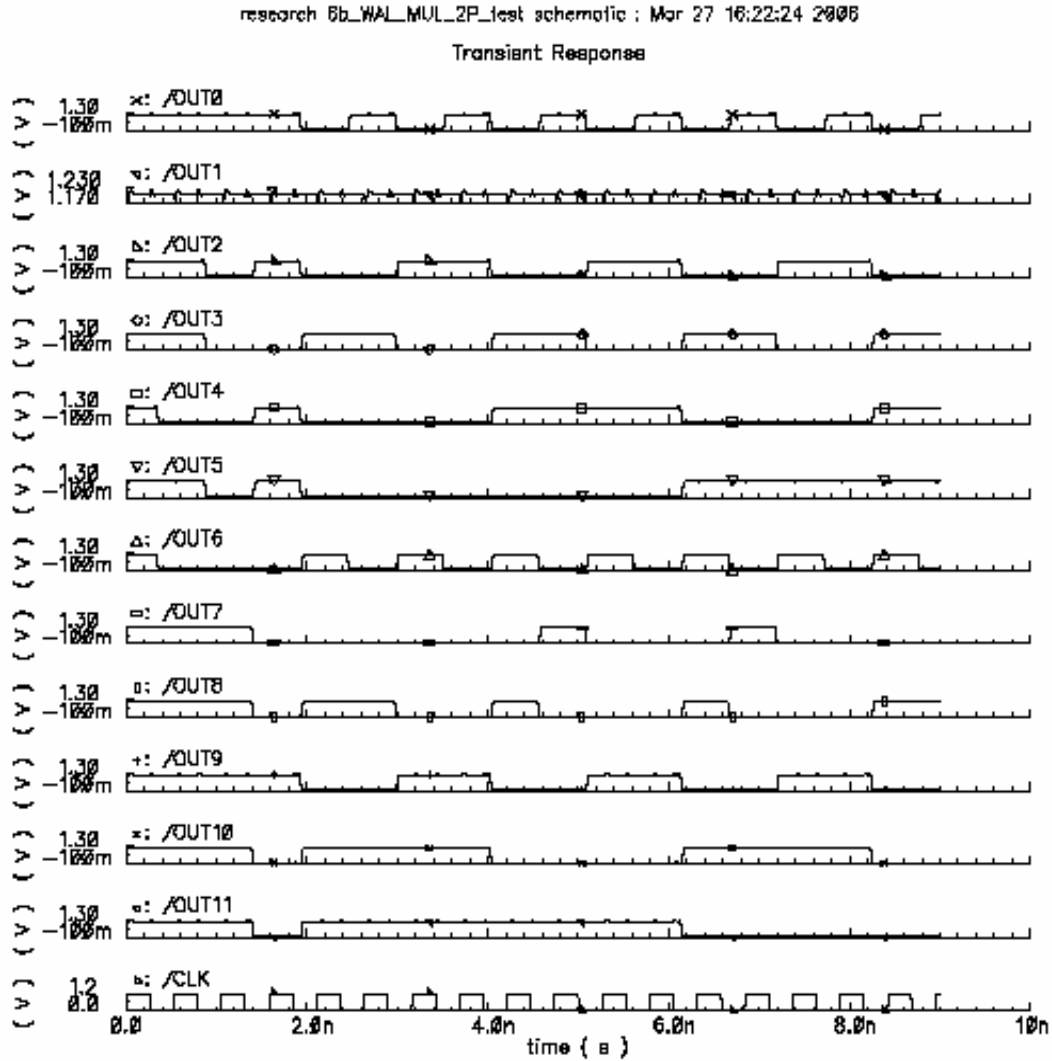


Figure 5.9c Correct simulation results of the 6-bit multiplier.

The final output result of the digital filter is a 7bit result. An attempt was made to maintain the above 13bit (including the sign bit) results through the merging adder network clocked at 1.9GHz; however, simulation shows 18% failure rate when trying to perform 13bit addition in 1 clock cycle at 1.9GHz. The decision was therefore made to truncate the output results of the multiplier. Thus, OUT[5:0] from the multiplier is discarded. Naturally, truncating the final output after the last summation addition would have led to more accurate results. However, to avoid pipelining the 13 bit adder, and hence adding more latency to the design; 7bit summation which perform consistently in one clock cycle 1.9GHz is preferred. The average power dissipation using the above test vectors is 7.63mW.

5.1.9 FIR Filter Design.

Significant difficulty was experienced producing valid outputs from either filter (high-pass & low-pass). The filters were first tested with inputs that happen to cause overflow problems. Studies and simulations were done and it was concluded that the upper and lower bound for the inputs is ± 31 . However, the filter still failed to operate correctly after limiting inputs to ± 31 . Further debugging proved that -0 zero was the culprit as when it got added in the summation tree causes a sign changed. Unfortunately, the multiplication by “2” coefficient causes this to frequently occur because truncation causes this output to be zero most of the time except when the other input to the multiplier is 62. At this point it was decided that a circuit that converts -0 to +0 was needed. This circuit is shown in figure 5.10 below; this circuit merely detects when all inputs are 0 and then ensure the sign bit is +. The addition of this macro comes at a penalty of one additional clock cycle for the filter. The FIR filter circuit design is presented in figure 5.10a below.

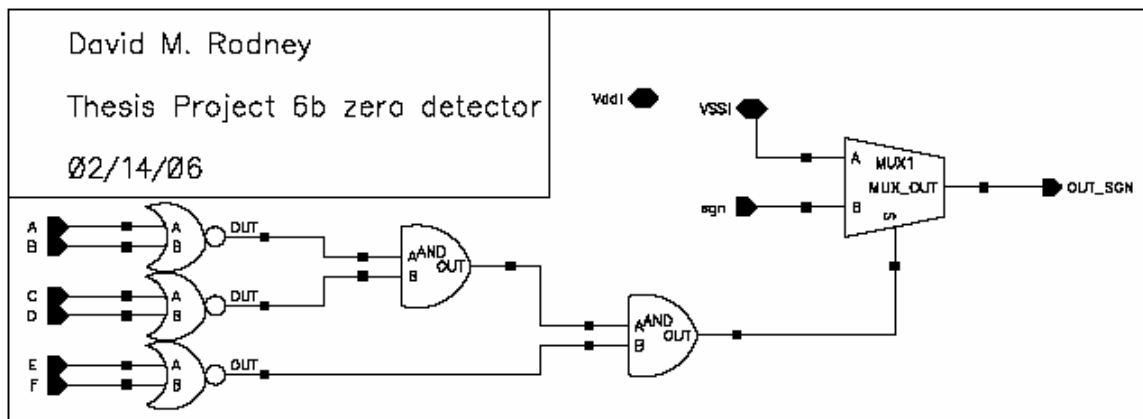


Figure 5.10 Design of -0 sign converted to positive.

The coefficient 0.0175 encoded to binary as 0.00001 when its [5:0] output is truncated (discarded) always result in zero output thus all those associated circuitry are eliminated except of course for the input delays which must be maintained. The filter uses 38 7bit register, 6 7bit adders, 8 sign/magnitude converters, 4 multipliers and 4 -0 sign converted to positive converter. The multiplier consists of two pipeline stages and uses a total of 47

DFF. The circuit can be clocked at 1.8GHz and takes 22 clock cycles to fill the filter pipeline. The filter consume between 62 and 69mW (depending on input frequency).

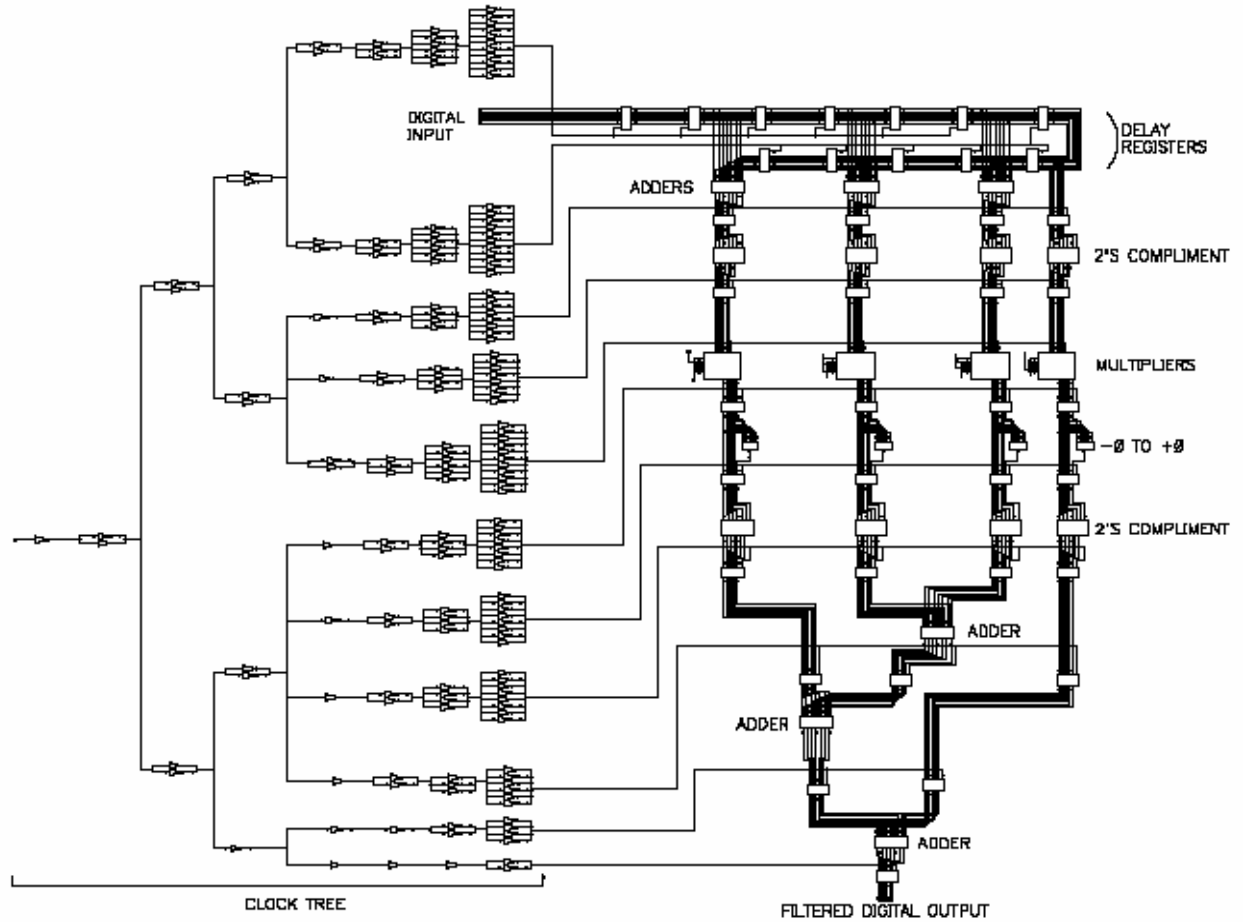


Figure 5.10a 15 tap FIR filter design.

5.2.0 The Clock tree Design.

The clock tree design is illustrated with the tree of inverters on the left side of figure 5.9a above. This image is also shown in figure 5.10 below with the clock node labeled.

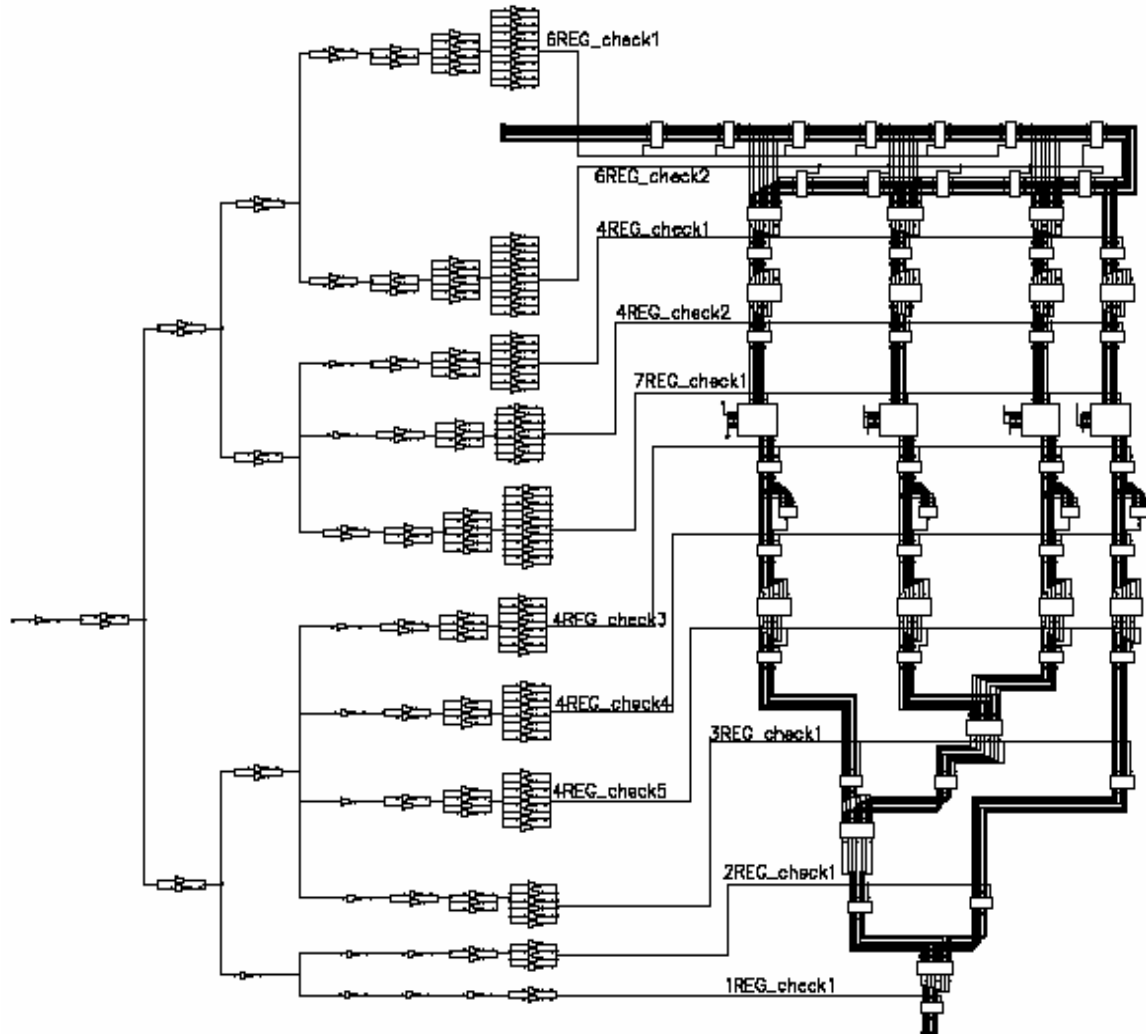


Figure 5.11 The FIR filter design with the clock node labeled.

The objective of the clock tree design is to ensure very little skew so that all the registers clocked their inputs at the same time. This is typically done by ensuring all the clocked nodes meet a particular rise and fall time target. The rise and fall time target for this design is to be less than 50ps. Table 5.9 below records the rise and fall times of each clocked node and also the differences of each clocked node relative to the first clocked

node “6REG_check1”. All 11 clock nodes curves are shown in figures 5.11a through 5.11d. Clearly illustrated in these figures are that each node have the same phase.

Clock Node	Rise Time, t_r	Fall Time, t_f	$\Delta t_r, \Delta t_f$
6REG_check1	49.264ps	49.346ps	—, —
6REG_check2	49.250ps	49.329ps	0.014, 0.017ps
4REG_check1	48.662ps	48.741ps	0.602, 0.605ps
4REG_check2	48.784ps	48.864ps	0.480, 0.482ps
7REG_check1	49.443ps	49.249ps	0.179, 0.097ps
4REG_check3	48.735ps	48.854ps	0.529, 0.492ps
4REG_check4	48.613ps	48.750ps	0.651, 0.596ps
4REG_check5	48.775ps	49.901ps	0.489, 0.555ps
3REG_check1	49.148ps	49.306ps	0.116, 0.040ps
2REG_check1	47.223ps	47.066ps	2.041, 2.28ps
1REG_check1	47.300ps	47.011ps	1.964, 2.335ps

Table 5.9 Clock nodes' rise and fall times and relative skew to 6REG_check1 node.

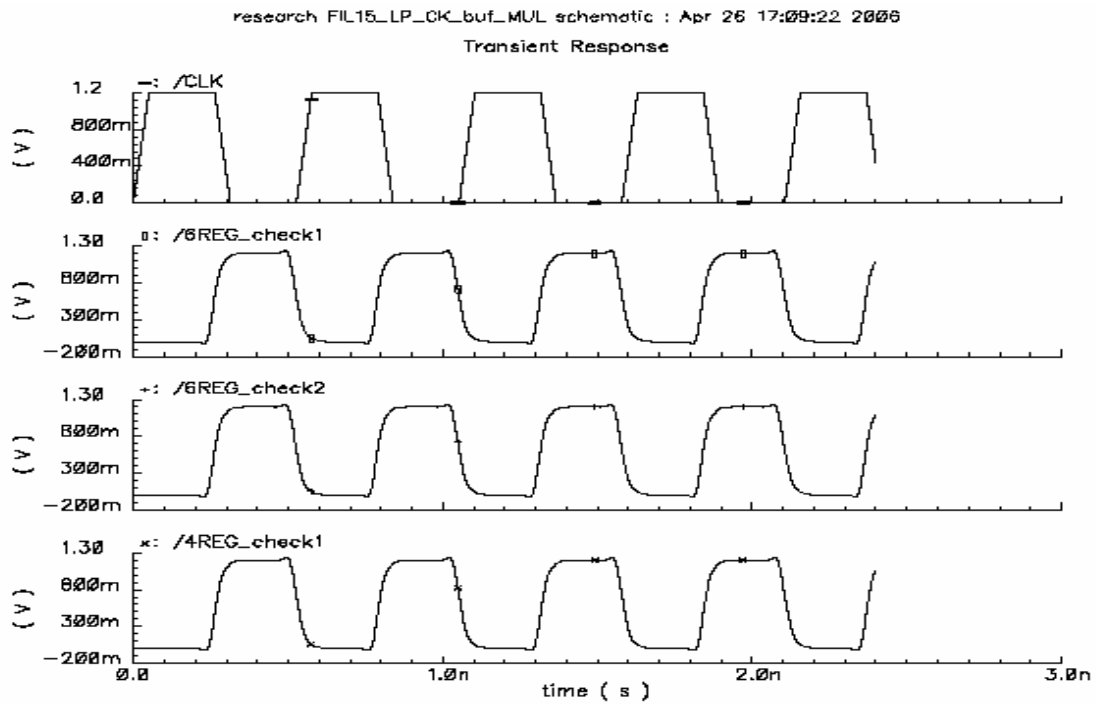


Figure 5.12a Clock nodes showing same phase and very little skew.

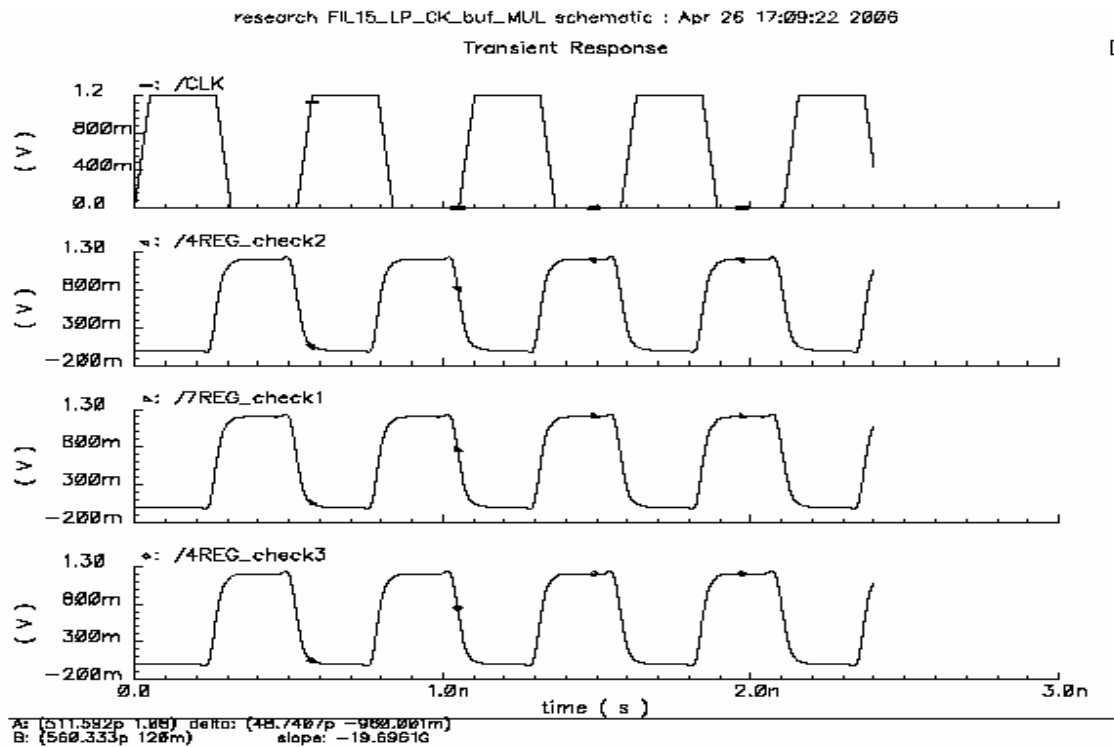


Figure 5.12b Clock nodes showing same phase and very little skew.

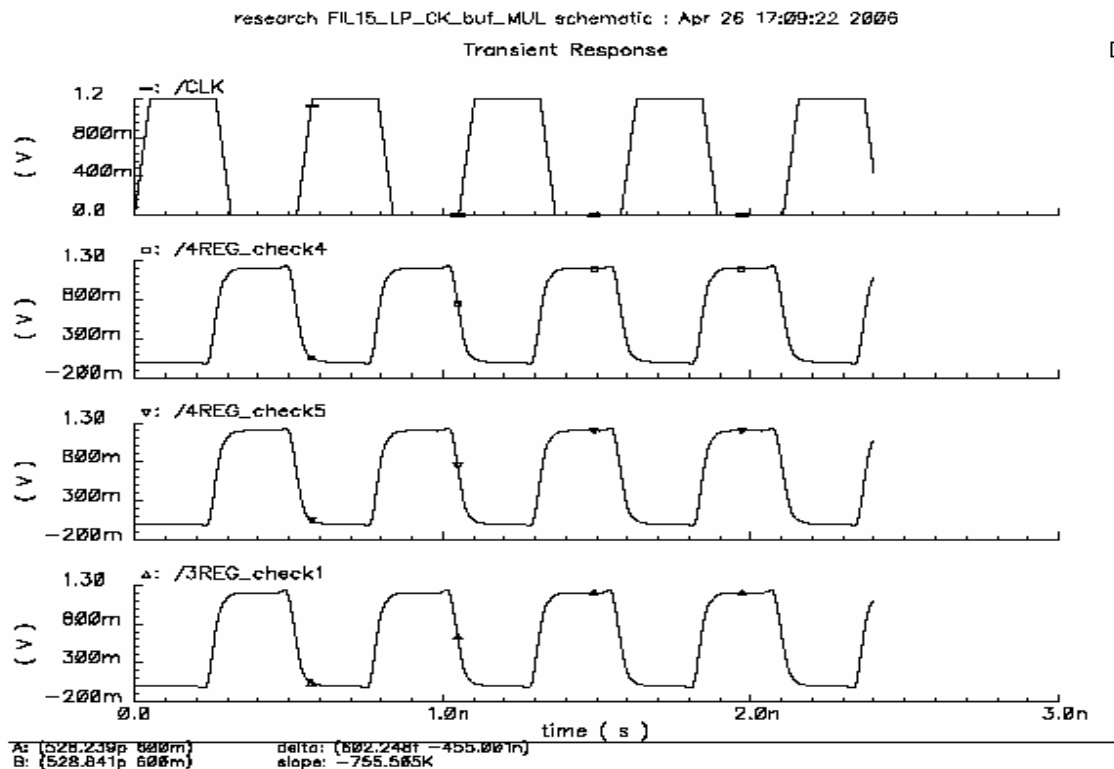


Figure 5.12c Clock nodes showing same phase and very little skew.

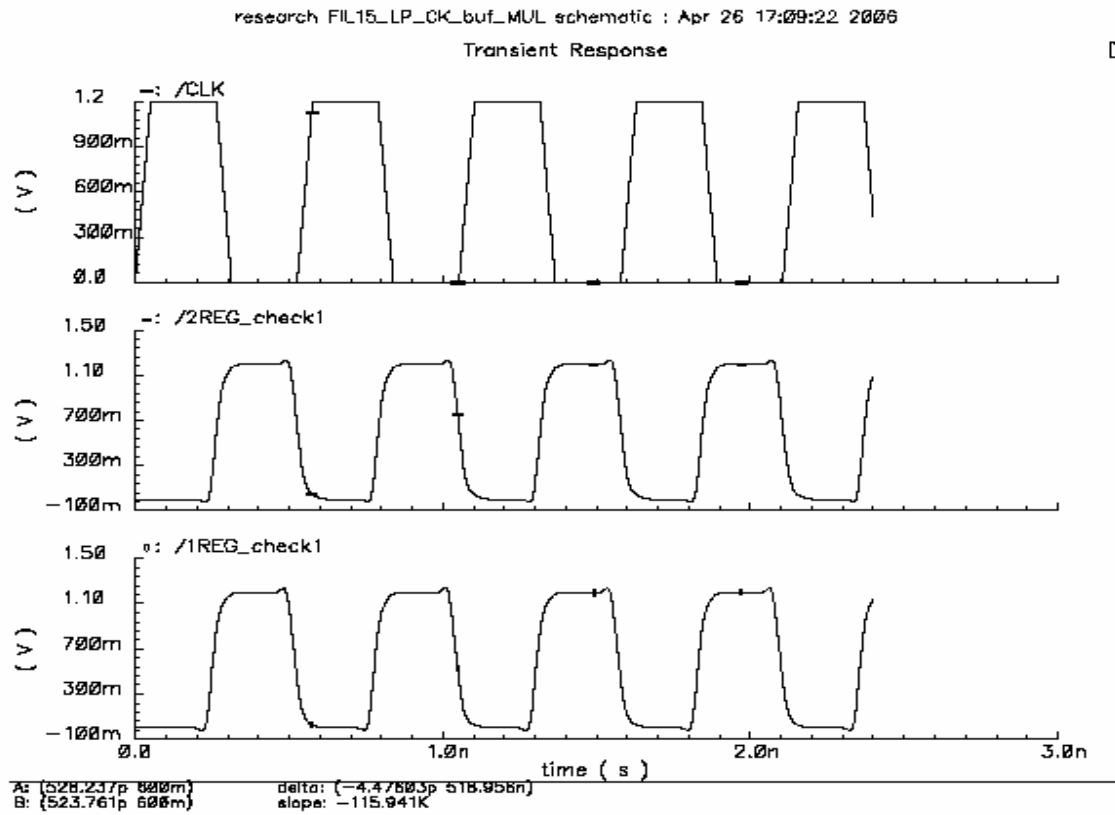


Figure 5.12d Clock nodes showing same phase and very little skew.

6. FIR Filter Validation

6.1.0 FIR Filter Validation for a 100Mhz sinusoidal input

The following figures shows assessment of the FIR filter at various frequencies in both low-pass and high-pass configuration clocked at 1.8GHz. Frequencies of 0 to 450MHz are within the passband of the low-pass FIR filter while these frequencies are rejected for the high-pass FIR filter. Similarly, frequencies of 450 to 900MHz are within the passband of the high-pass filter but are rejected by the low-pass filter. Obviously, frequencies above 900MHz will be rejected by both filters.

MATLAB values	Values for cadence	Binary representation (cadence)
0	0	0000000
10.656	11	0001011
19.859	20	0010100
26.641	27	0011011
30.516	31	0011111
30.516	31	0011111
26.641	27	0011011
19.859	20	0010100
10.656	11	0001011
0	0	0000000
-10.656	-11	1110101
-19.859	-20	1101100
-26.641	-27	1100101
-30.516	-31	1100001
-30.516	-31	1100001
-26.641	-27	1100101
-19.859	-20	1101100
-10.656	-11	1110101

Table 6.1 digitized values of a 100MHz sinusoidal.

Using the MATLAB simulink design shown in figure A1 of appendix A, table 6.1 above shows the digitized values of the 100MHz sinusoidal and figures 6.1b & c shows the corresponding MATLAB response. These MATLAB responses are obtained using the model illustrated in figure A2 of appendix A. This model output response will be correspondingly compared against that of cadence.

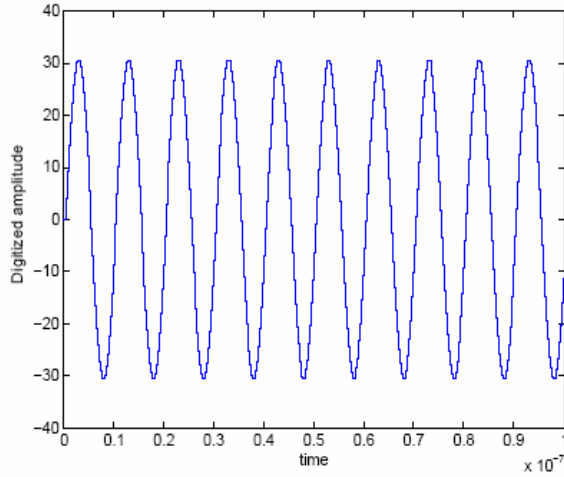


Figure 6.1a Digitized 100MHz sinusoidal input.

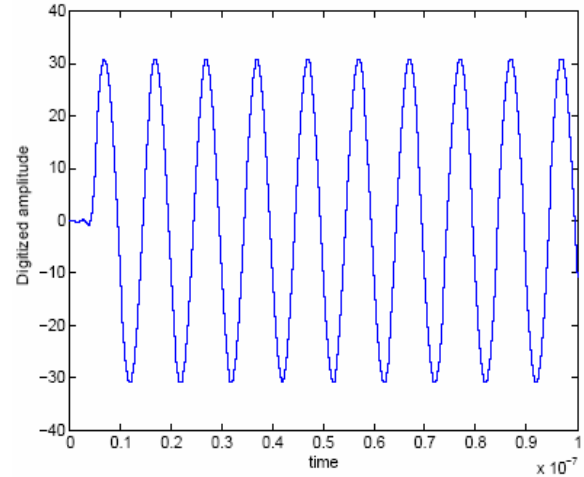


Figure 6.1b MATLAB low-pass FIR filter response to 100MHz sinusoidal input.

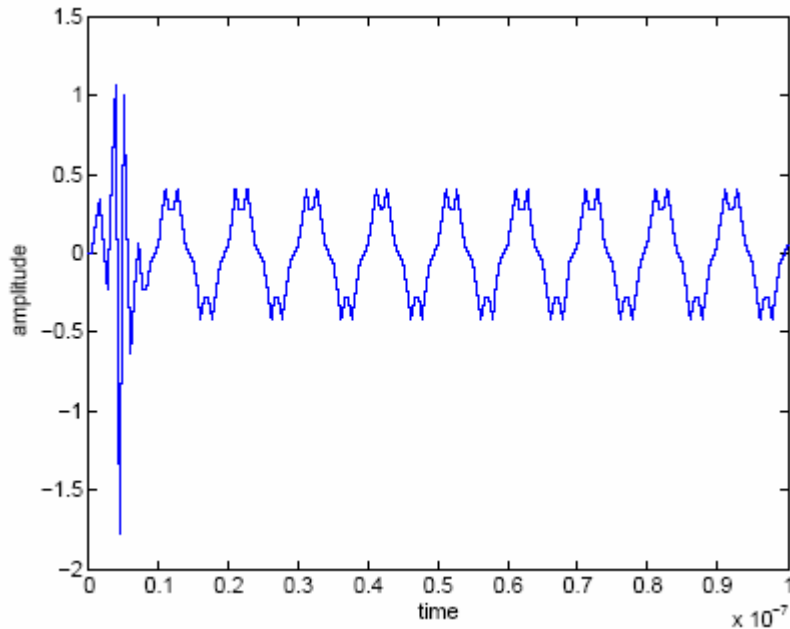
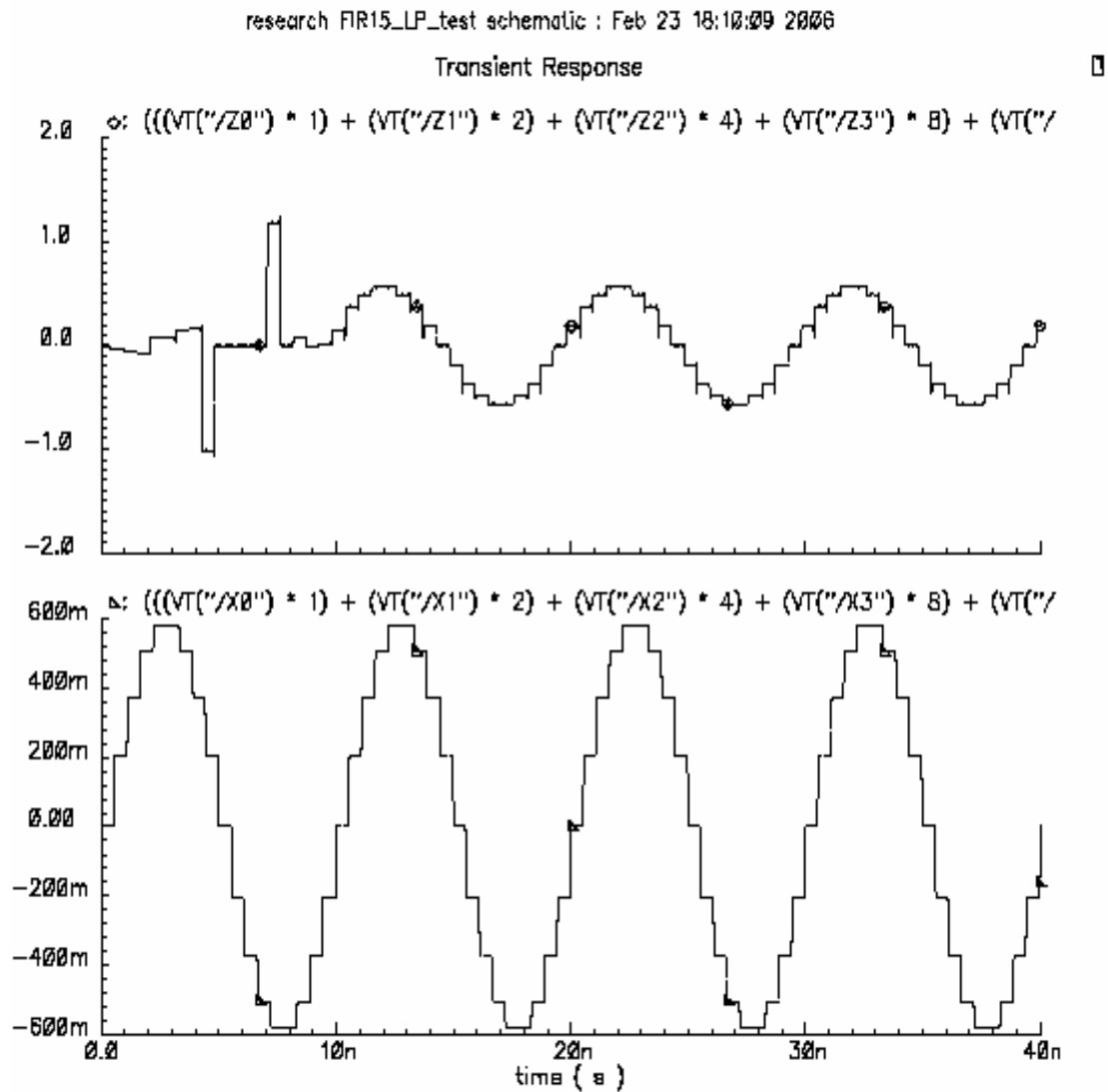


Figure 6.1c MATLAB high-pass FIR filter response to 100MHz sinusoidal input.

Figures 6.1b & c illustrates MATLAB correctly predicts the filter response to the 100MHz sinusoidal input. The corresponding response cadence response of the designed FIR filter is illustrated in figures 6.1d & e to the 100MHz sinusoidal input.



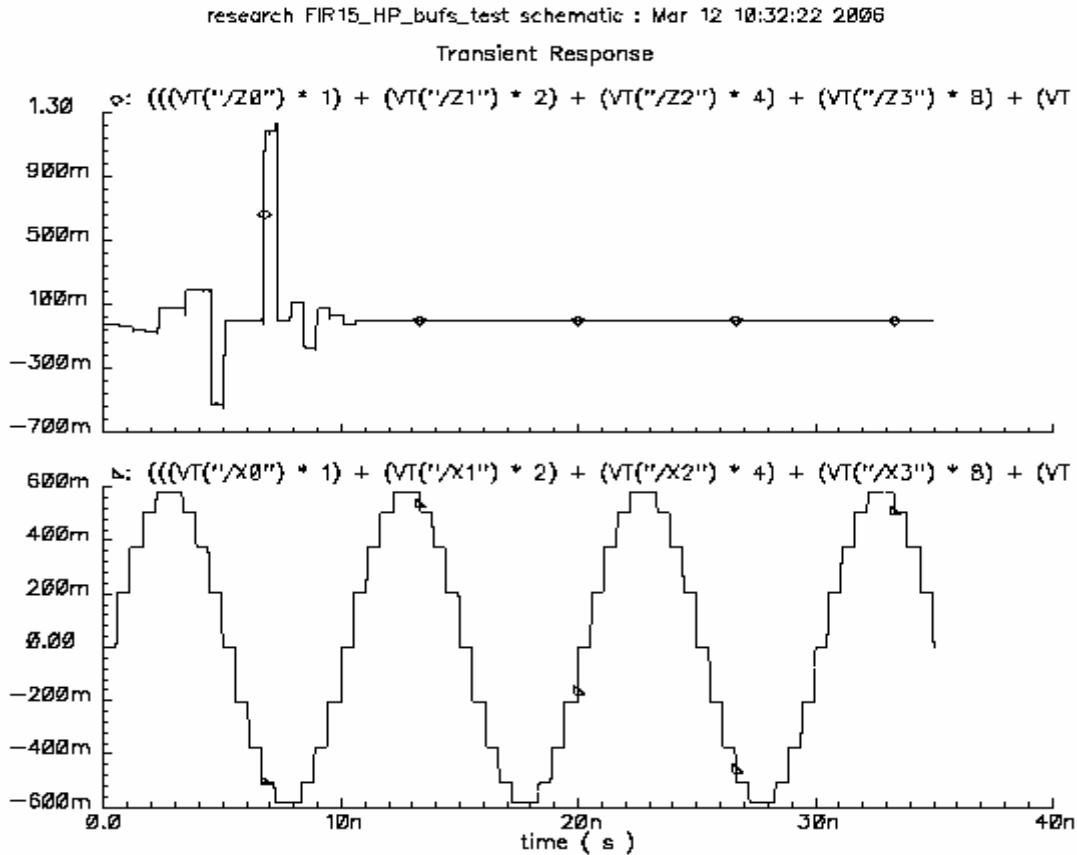


Figure 6.1e Cadence high-pass FIR filter response to 100MHz sinusoidal input.
Topmost curve is the OUTPUT and the bottom curve is the INPUT.

Figures 6.1d & e illustrates excellent response of the design FIR filter to 100MHz sinusoidal inputs conforming to MATLAB prediction. Table 6.1a below characterizes the cadence FIR response for the 100MHz sinusoidal input.

	Peaks/Range	Mid-point	Power con.	Periodicity
LP filter	$\pm 562.450\text{mV}$	$-39.7\mu\text{V}$	62.24mW	10.00ns
HP filter	—	$-12.89\mu\text{V}$	62.22mW	—
input	$\pm 581.25\text{mV}$	0	—	10.00ns

Table 6.1a Characterization for the designed FIR filter for 100MHz input.

Truncation results in loss of amplitude of the signal in the passband of the filter. The above shows a loss of 3.23% or 0.29dB of passband amplitude versus input amplitude.

6.1.1 FIR Filter Validation for a 300MHz sinusoidal input

The FIR filter will now be characterized for a 300MHz sinusoidal. Table 6.2a shows the digitized values for a 300MHz sinusoidal obtained from the simulink quantizer and figures 6.2a, b, & c are the corresponding MATLAB simulation.

MATLAB values	Values for cadence	Binary representation (cadence)
0	0	0000000
26.641	27	0011011
26.641	27	0011011
0	0	0000000
-26.641	-27	1100101
-26.641	-27	1100101

Table 6.2a Digitized values for 300MHz sinusoidal.

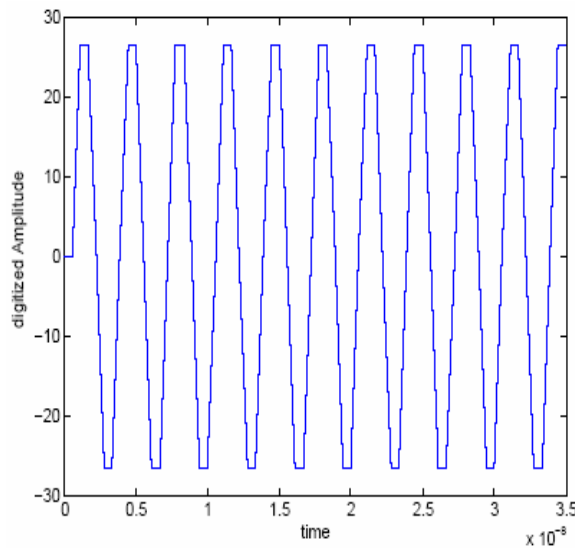


Figure 6.2a Digitized 300MHz sinusoidal.

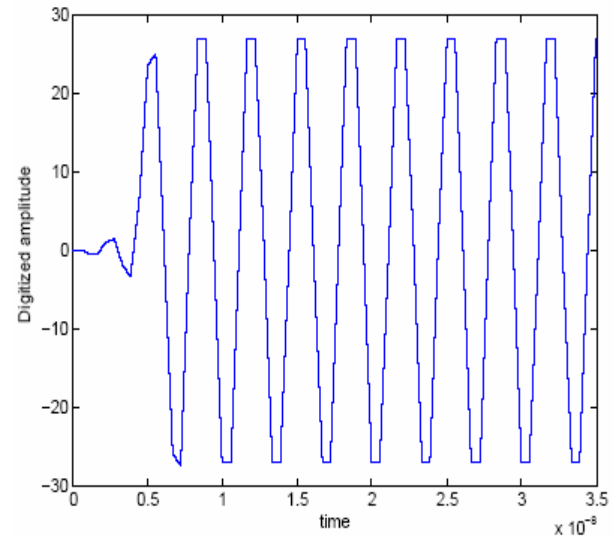


Figure 6.2b MATLAB low-pass FIR filter response to 300MHz sinusoidal.

Figure 6.2b above shows the 300Hz signal (well within the passband of the low-pass filter) passing through the FIR network un-attenuated. MATLAB illustrate in figure 6.2c the high-pass FIR filter rejects the 300MHz as expected (very attenuated).

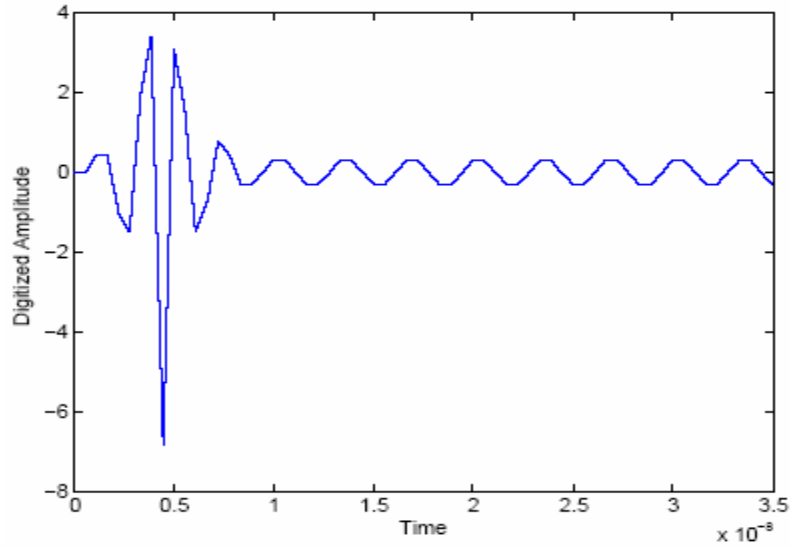


Figure 6.2c MATLAB high-pass FIR filter response to 300MHz sinusoidal input.

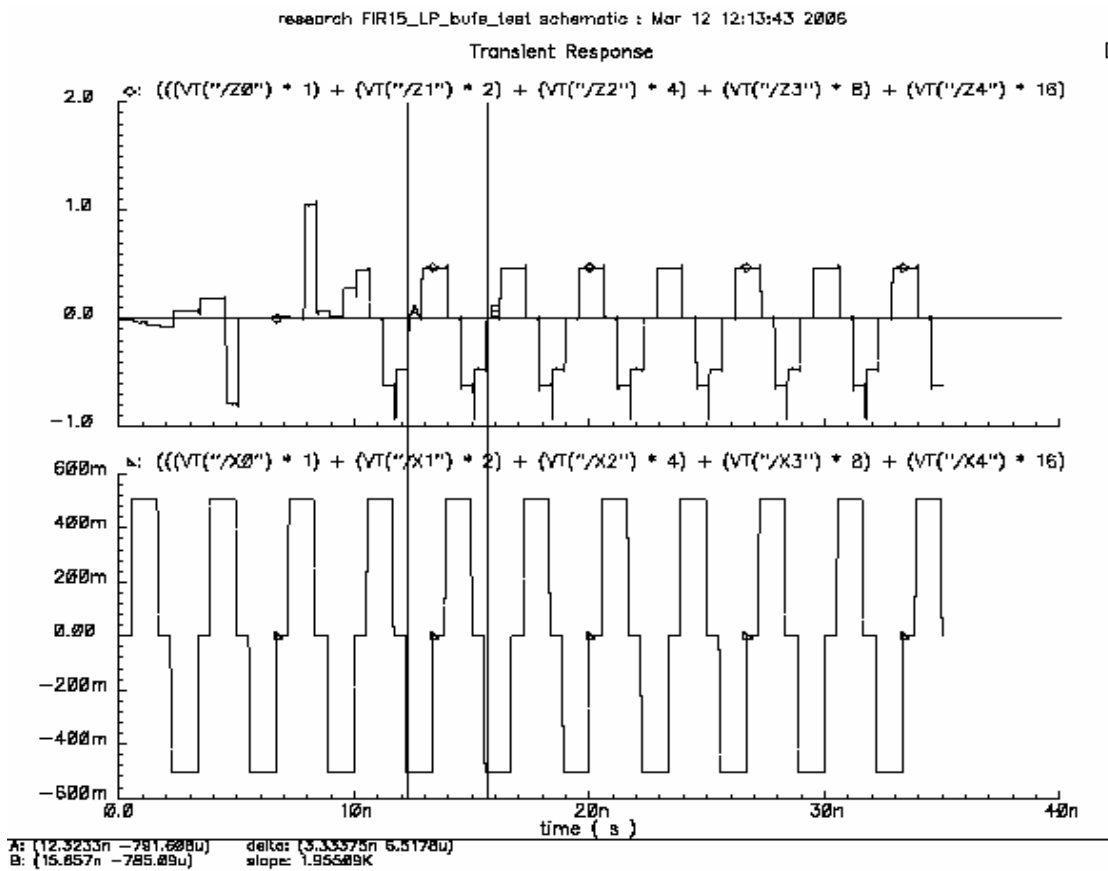


Figure 6.2d Cadence low-pass FIR filter response to 300MHz sinusoidal input.

Topmost curve is the OUTPUT and the bottom curve is the INPUT.

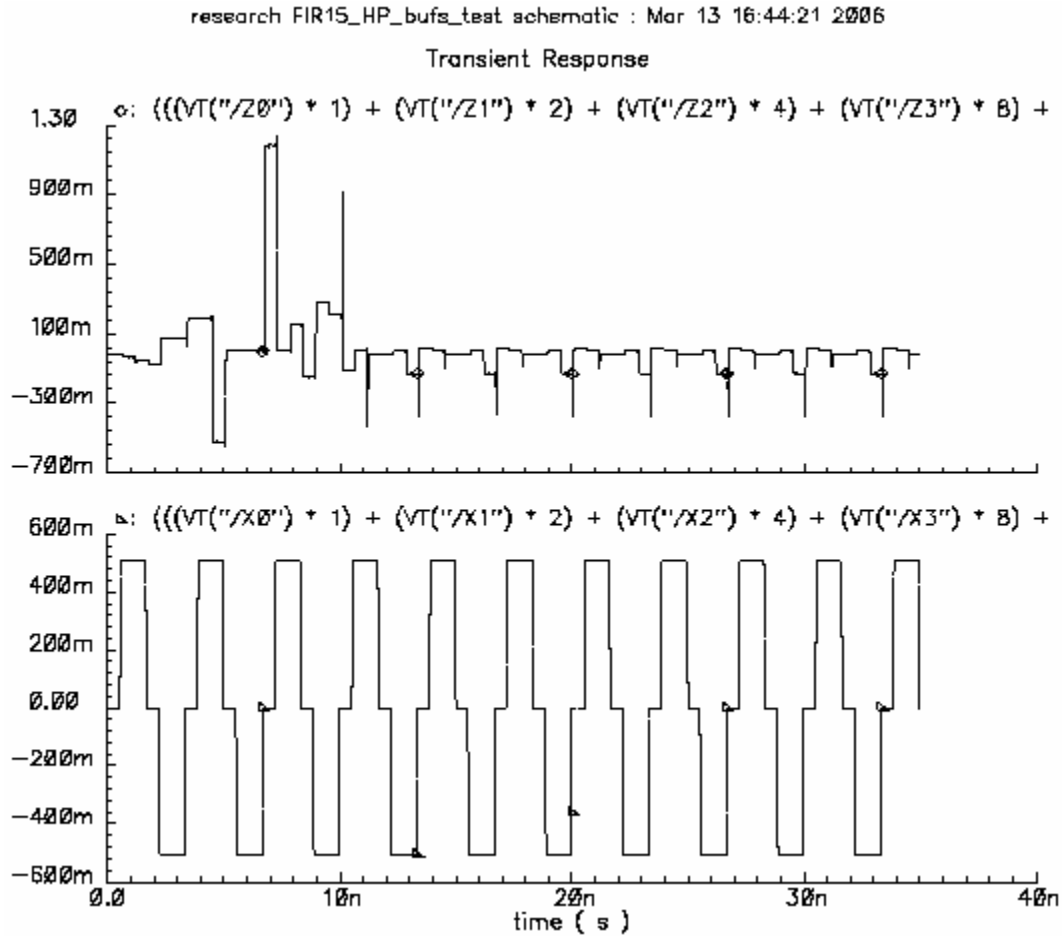


Figure 6.2e Cadence high-pass FIR filter response to 300MHz sinusoidal input.

Topmost curve is the OUTPUT and the bottom curve is the INPUT.

Again figures 5.11d & e illustrates good response of the design FIR filter to 300MHz sinusoidal inputs conforming closely to MATLAB prediction. Figure 5.11d does show some sort of error (droop) in its curve which may be quantization error. Table 5.10b below characterizes the cadence FIR response for the 300MHz sinusoidal input.

	Peaks/range	Mid-point	Power cons.	Periodicity
LP filter	+471.73 to -470.53mV	-27.48 μ V	61.88mW	3.33ns
HP filter	+37.44 to -112.63mV	-80.58 μ V	62.39mW	3.33ns
input	\pm 506.25mV	0	—	3.33ns

Table 6.2b Characterization for the designed FIR filter for 300MHz.

The MATLAB curve of figure 6.2b has peak values of $\pm 0.3117V$. This is a 38.64dB ($20\log(26.641/0.3117)$) attenuation compared to the input which is on target for stopband requirement. Correspondingly, the cadence simulation shows 22.62dB attenuation for the positive peak and 13.05dB attenuation for the negative peak for the 300MHz input sinusoidal into the high-pass filter. Thus the cadence implementation and MATLAB's prediction are not perfectly correlated and hence there is significant source of error. One may speculate that the obvious sources of error such as truncation may be the difference. In the passband (low-pass filter) the positive peak differ by 6.82% or 0.61dB versus the positive input peak; similarly, the negative peak differ by 7.06% or 0.64dB.

6.1.2 FIR Filter Validation for a 450MHz sinusoidal input

The FIR filter will now be observed for a 450MHz sinusoidal. Table 6.3a shows the digitized values for a 450MHz sinusoidal obtained from the simulink quantizer and figures 6.3a, b, & c are the corresponding MATLAB simulation.

MATLAB values	Values for cadence	Binary representation (cadence)
0	0	0000000
31	31	0011111
0	0	0000000
-31	-31	1100001

Table 6.3a Digitized values of a 450MHz sinusoidal.

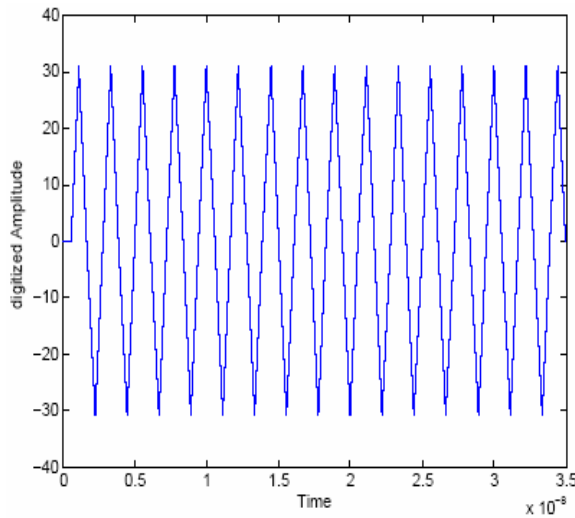


Figure 6.3a Digitized 450MHz sinusoidal.

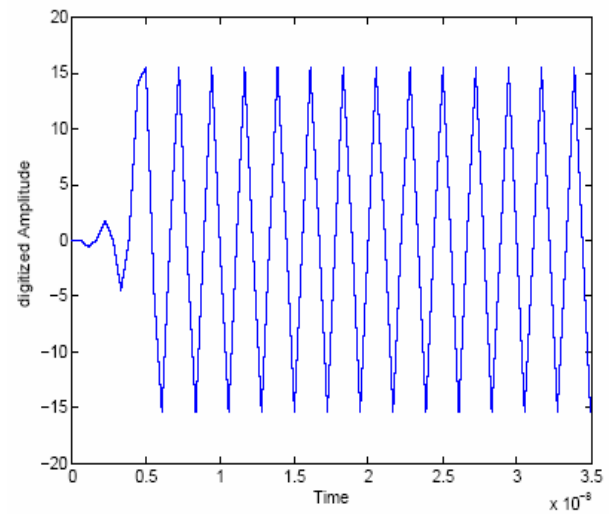


Figure 6.3b MATLAB low-pass FIR filter response to 450MHz sinusoidal input.

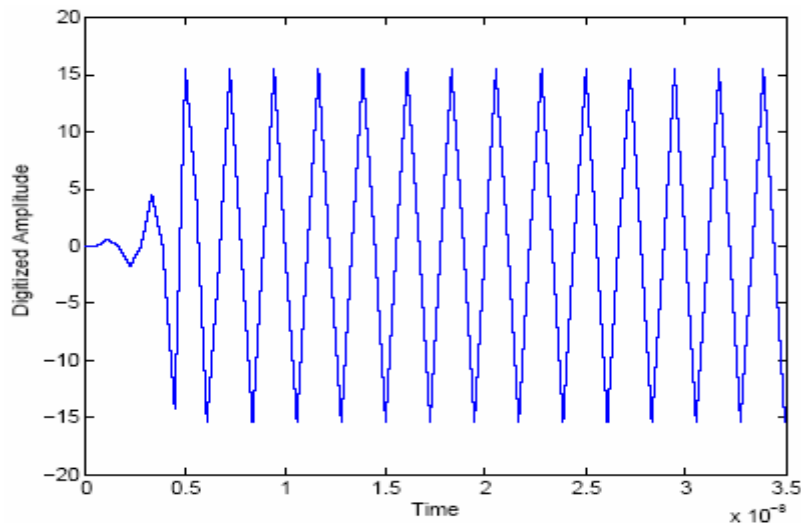


Figure 6.3c MATLAB high-pass FIR filter response to 450MHz sinusoidal input.

For this simulation, 450MHz is the exact boundary frequency that separates the high-pass and low-pass filter and is thus represented by the cross-over point of the high-pass and low-pass filter as illustrated in figure 3.4. Figures 6.3b & c clearly illustrates that the signal gets attenuated by half its amplitude which is 6.02dB by both filters. The

intension of this thesis is to demonstrate that filters can perform channel arbitration. Consider that channel X have a range of 300 – 400MHz and channel Y have a range of 400 – 500MHz; the latter simulations illustrate that a 450MHz signal will be present in both channels X and Y with the same signal strength while clearly one can see that the 450MHz signal should be slotted to channel Y. Resolving similar issues to show that the signal belongs to channel Y was demonstrated in section 4. Cadence simulation of the implemented filters for the 450MHz input closely conform the same behavior as MATLAB and are illustrated in figures 6.3d & e below. Table 5.11b shows the filters characterized at 450MHz.

	Peaks/Range	Mid-point	Power cons.	Periodicity
LP filter	+281.23 to -282.20mV	-4.51 μ V	60.94mW	2.22ns
HP filter	+282.90 to -281.19mV	-4.51 μ V	60.93mW	2.22ns
input	\pm 581.25mV	0	—	2.22ns

Table 6.3b Characterization for the designed FIR filter at 450MHz.

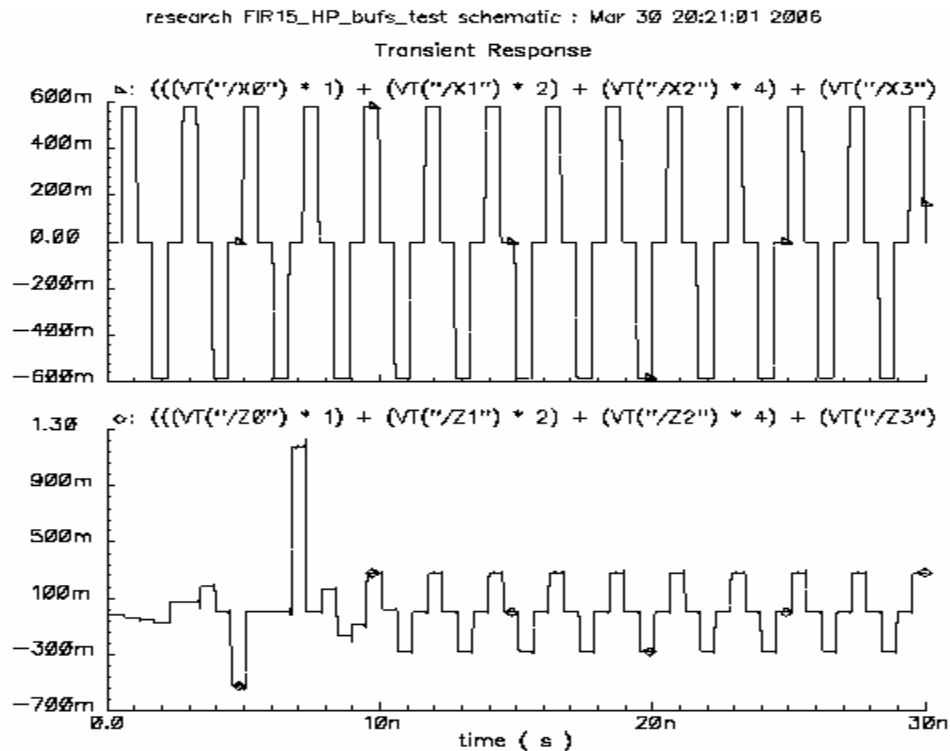


Figure 6.3d The designed high-pass FIR filter's response to a 450MHz sinusoidal input.

Topmost curve is the INPUT and the bottom curve is the OUTPUT.

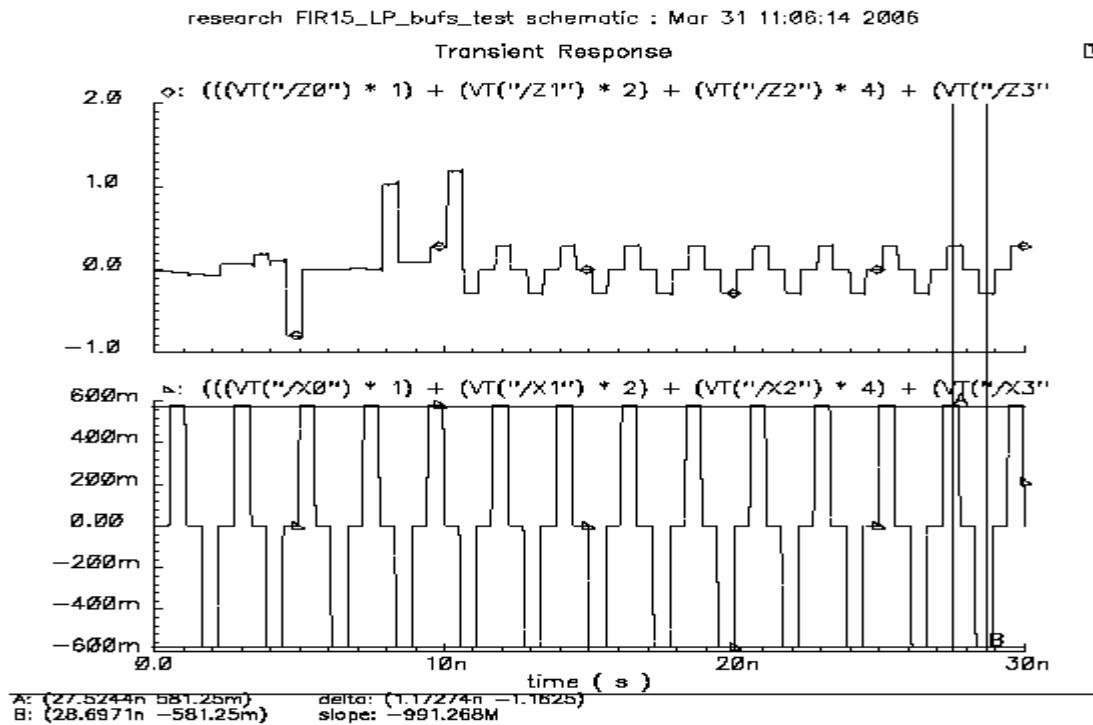


Figure 6.3e The designed low-pass FIR filter's response to a 450MHz sinusoidal input.

Topmost curve is the INPUT and the bottom curve is the OUTPUT.

From table 6.3b, the peak to peak voltage of the input signal is 1.0125V and that for the low-pass filter output is 0.56343 and hence the 450MHz sinusoid is attenuated by $20\log(1.0125/0.56343) = 5.091\text{dB}$. Similarly, the 450MHz sinusoid is attenuated by 5.081dB via the high-pass filter. The results are remarkably close to the MATLAB prediction of 6dB attenuation.

6.1.3 FIR Filter Validation for a 600MHz sinusoidal input

The next chosen frequency the filter is characterized at is 600MHz and is demonstrated in a similar manner by figures 6.4a through 6.4b. Table 6.4a shows the digitized 600MHz values.

MATLAB values	Values for cadence	Binary representation (cadence)
0	0	0000000
29.547	30	0011110
-25.188	-25	1100111

Table 6.4a digitized values of a 600MHz sinusoidal.

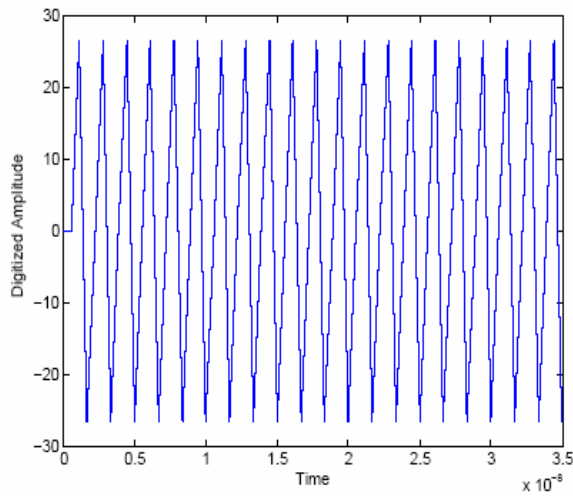


Figure 6.4a Digitized 600MHz sinusoidal.

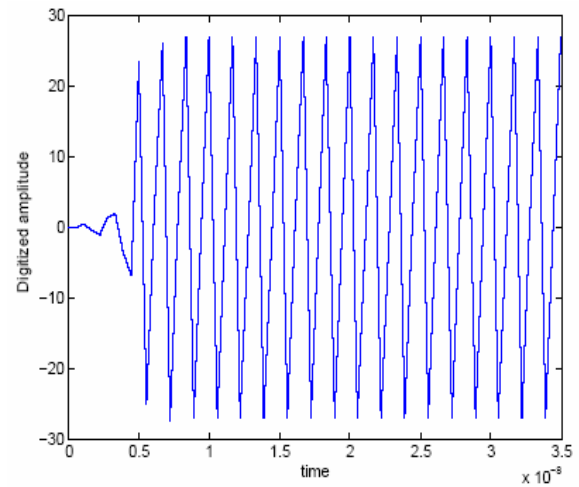


Figure 6.4b MATLAB high-pass FIR filter response to 600MHz sinusoidal input.

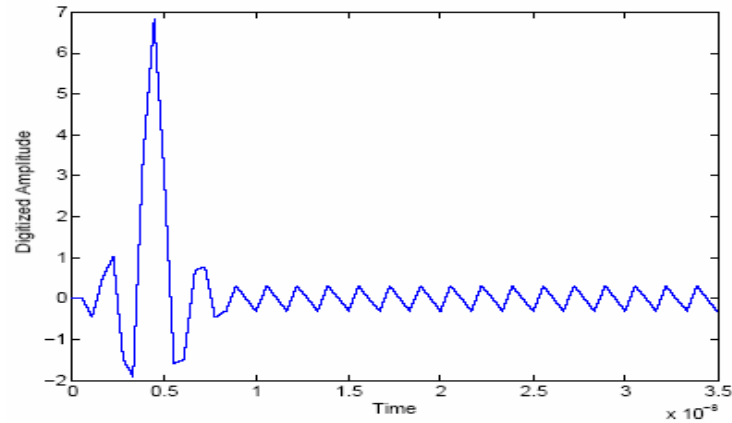


Figure 6.4c MATLAB low-pass FIR filter response to 600MHz sinusoidal input.

The latter MATLAB simulations correlates with the expected behavior that a 600MHz signal, namely, it is in the passband of the high-pass FIR filter and the well within the stopband of the low-pass FIR filter and where it is thus rejected. Figures 6.4d & e shows the cadence simulation of the implemented FIR filter also strongly correlates with the MATLAB simulations.

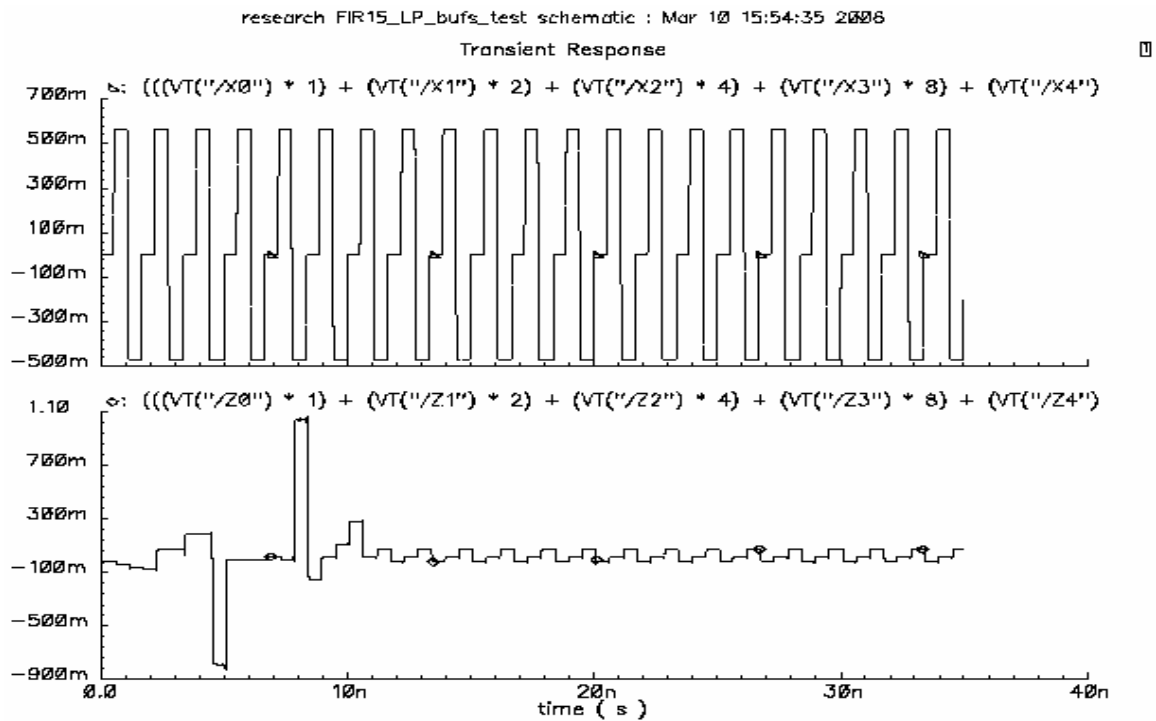
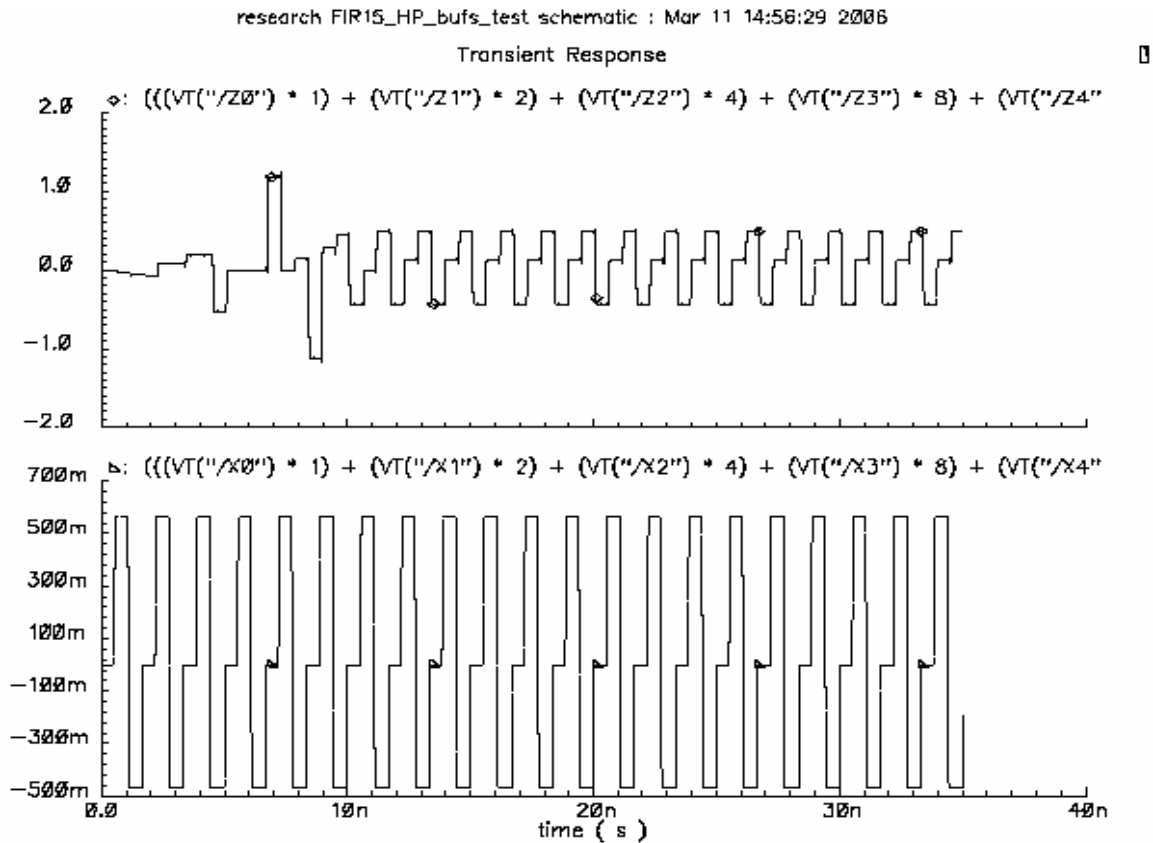


Figure 6.4d The designed low-pass FIR filter's response to a 600MHz sinusoidal input.

Topmost curve is the INPUT and the bottom curve is the OUTPUT.



6.2.0 Customization of Multipliers to Reduce Power Consumptions

Since the coefficients are now fixed, one may take advantage of the fact that zeros in coefficient word make no useful contribution to the final multiplier output. Thus removing nodes that yield product terms of zero (e.g. $X_1 \cdot Y_1$, if X_1 is '0', the resulting AND output is '0') and full-adders that cannot now produce a carry of '1' will reduce power dissipation and area. This optimization also reduced run-time for simulation as the number of simulation nodes may be reduced considerably. Figures 6.5 through 6.8 shows the resulting multipliers for coefficients $\pm.000010$, $\pm.000101$, $\pm.010011$ and $\pm.1000000$ respectively. This effort only reduces the power dissipation and will not yield an overall speed push as the critical path now shifted from the multipliers to the 2s compliment circuit and adders for which no clever rework was realized to date. Figure 6.5 and 6.8 are very simplistic because they are perfect exponent of base 2 and thus multiplication can be achieve by simple shifting.

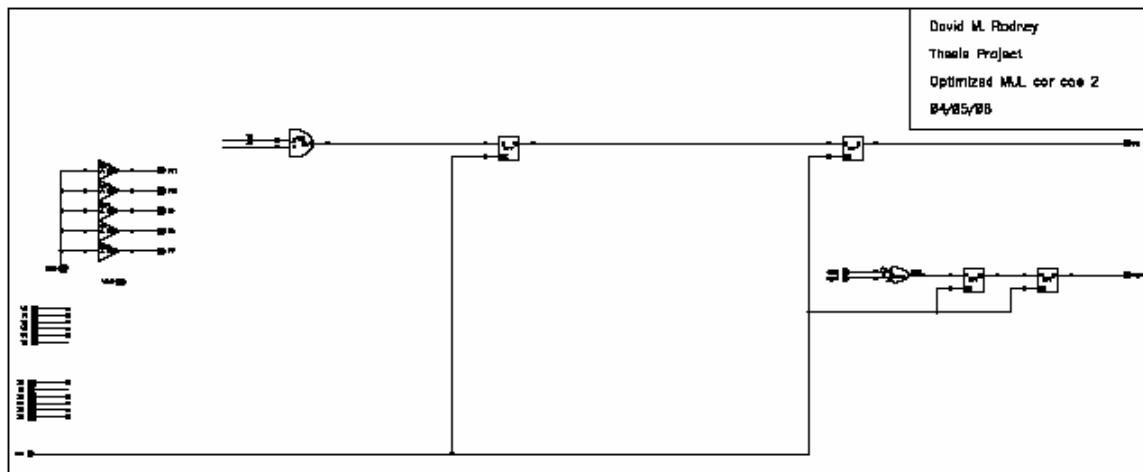


Figure 6.5 Optimized multiplier for coefficient $\pm.000010$.

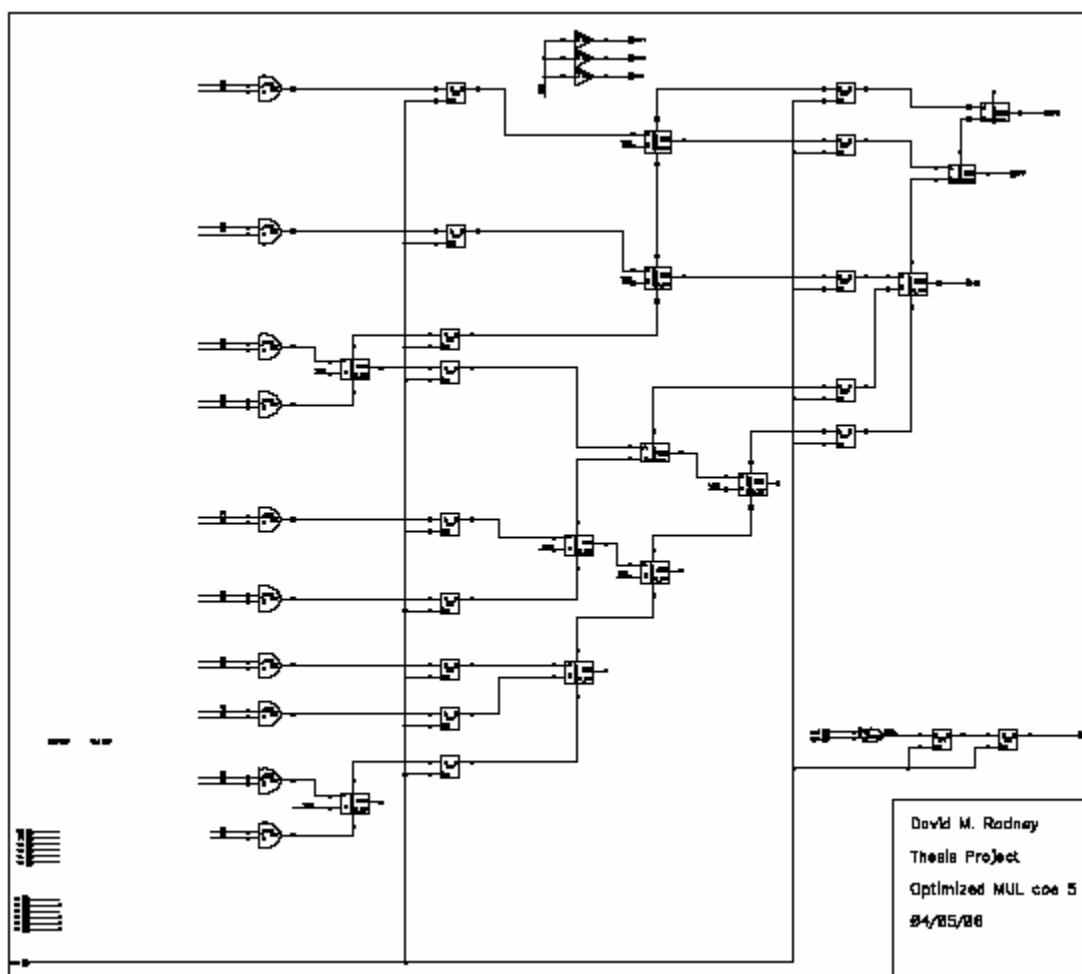


Figure 6.6 Optimized multiplier for coefficient $\pm.000101$.

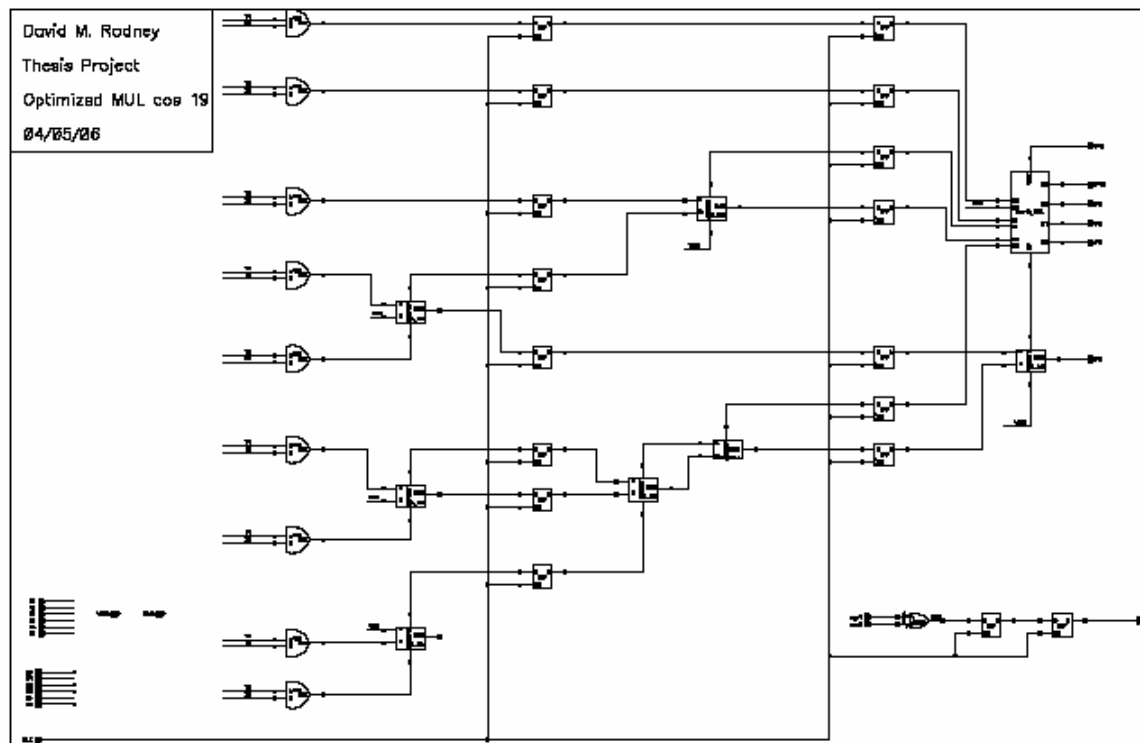


Figure 6.7 Optimized multiplier for coefficient $\pm.010011$ rounded to $\pm.010010$

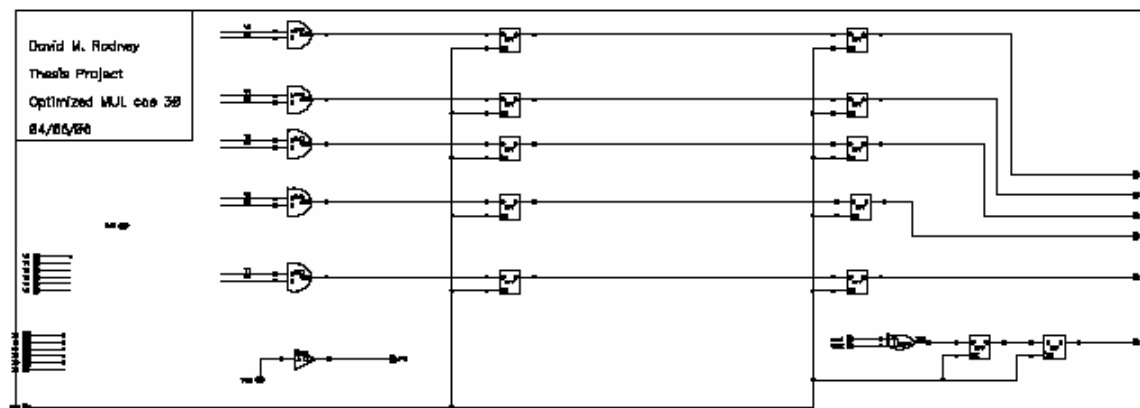


Figure 6.8 Optimized multiplier for coefficient $+.100000$.

The filters are revalidated with some chosen frequencies using the latter multipliers.

6.2.1 Revalidation of the FIR filters for a 600MHz input sinusoidal.

Figures 6.9a & b respectively shows the high-pass and low-pass response of the FIR filter using the new multipliers and table 6.5 shows the corresponding characterization.

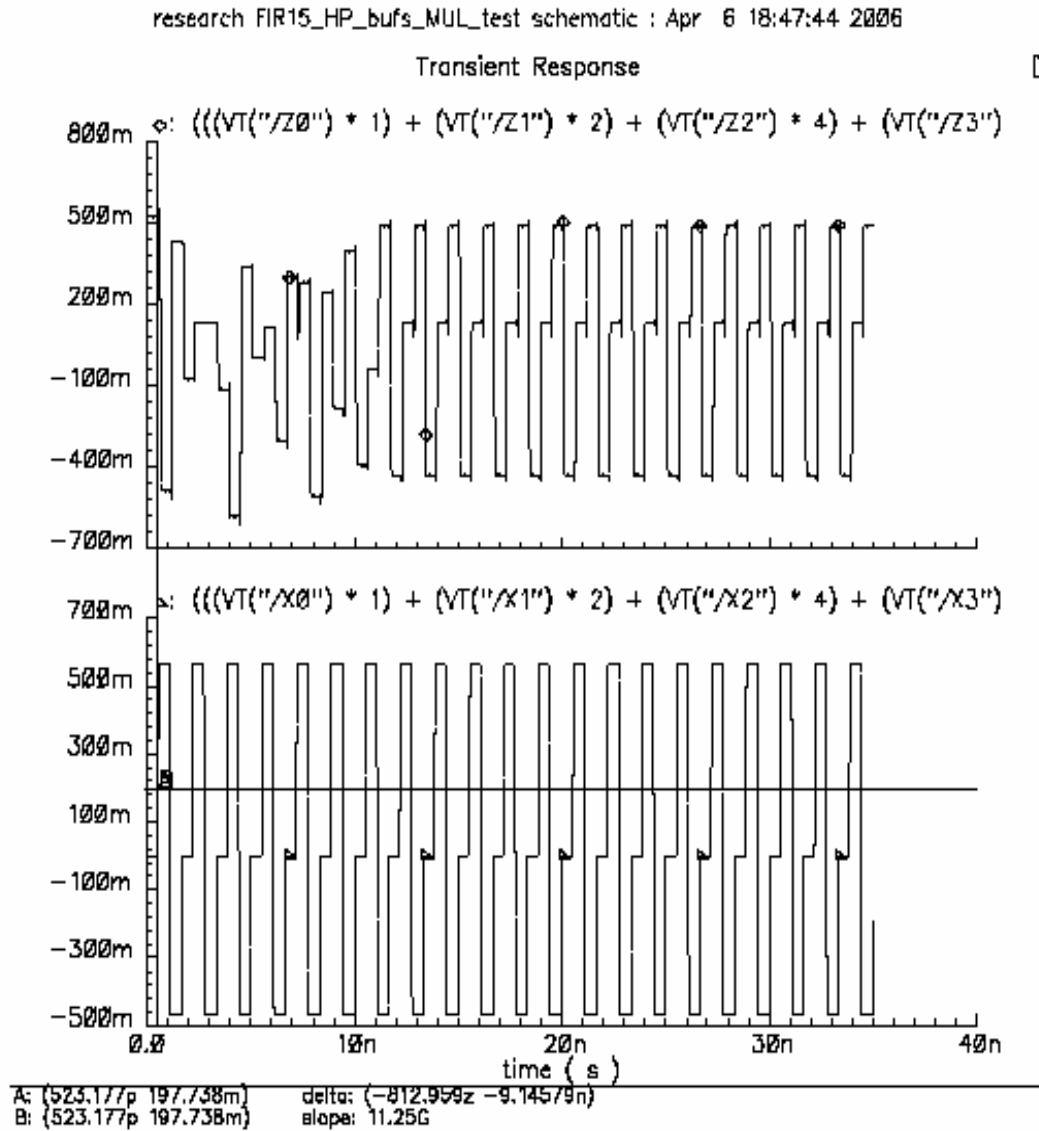


Figure 6.9a High-pass FIR filter response to 600MHz input using new multipliers.

Topmost curve is the OUTPUT and bottom curve is the INPUT.

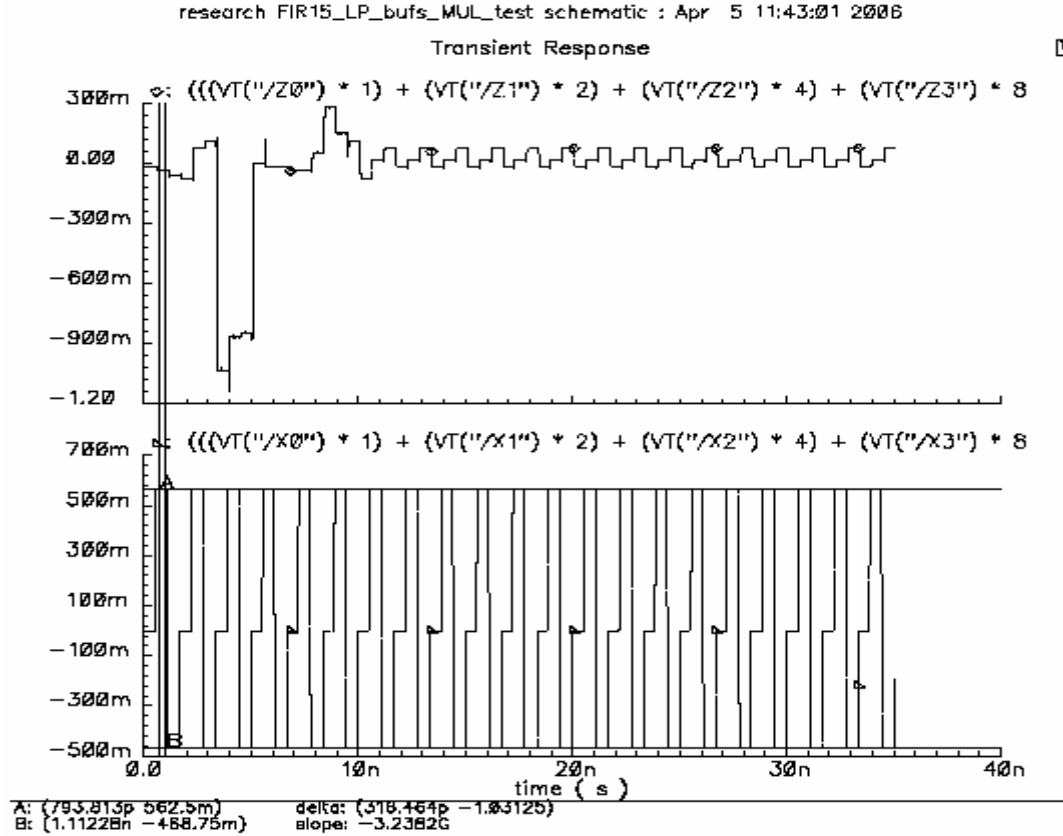


Figure 6.9b Low-pass FIR filter response to 600MHz input using new multipliers.
Topmost curve is the OUTPUT and bottom curve is the INPUT.

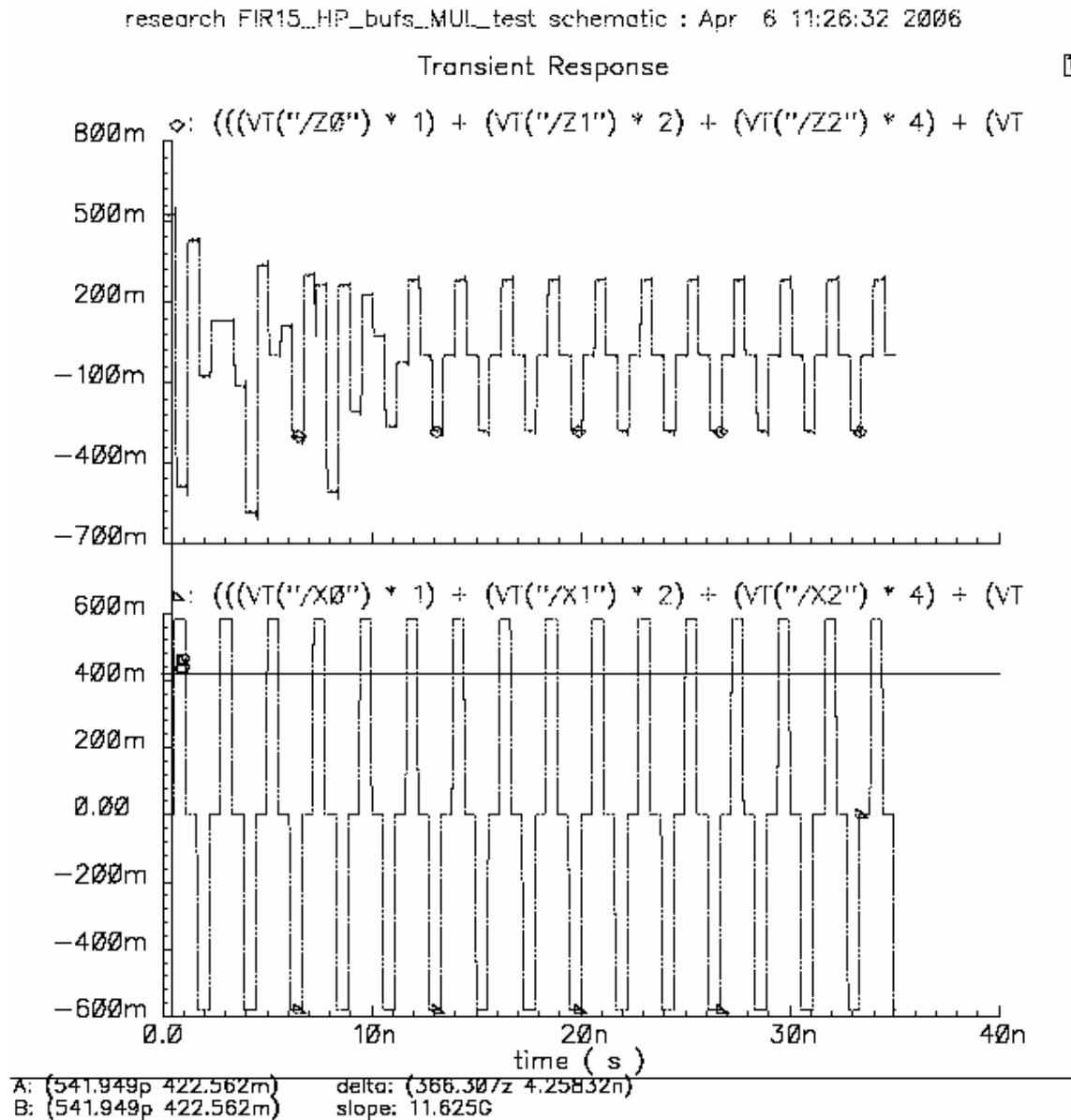
	Range	Mid-point	Power cons.	Periodicity
LP filter	+74.41 to -18.76mV	+18.69mV	55.29mW	1.67ns
HP filter	+487.49 to -431.23mV	+131.25mV	56.35mW	1.67ns
input	+562.5 to -468.75mV	0	—	1.67ns

Table 6.5 Characterization of the FIR filter to 600MHz using new multipliers.

The low-pass filter which will reject the 600MHz signal attenuates the positive and negative peaks by 16.33 and 27.23dB respectively. Rounding the coefficient ± 0.010011 to ± 0.010010 resulted in very little difference to the resulting filter operation, thus justifying this rounding. Comparing table 6.4b to table 6.6 shows the voltages differ by the 0.79 and 0.21% for the positive and negative peaks of the Low-pass filter while they differ by 0.06 and 0.74% for the high-pass filter. A power saving of 19.68% (13.68mW) is realized using the customized multipliers for this case.

6.2.2 Revalidation of the FIR filters for a 450MHz input sinusoidal.

Figures 6.10a & b respectively shows the high-pass and low-pass response of the FIR filter using the new multipliers and table 6.6 shows the corresponding characterization.



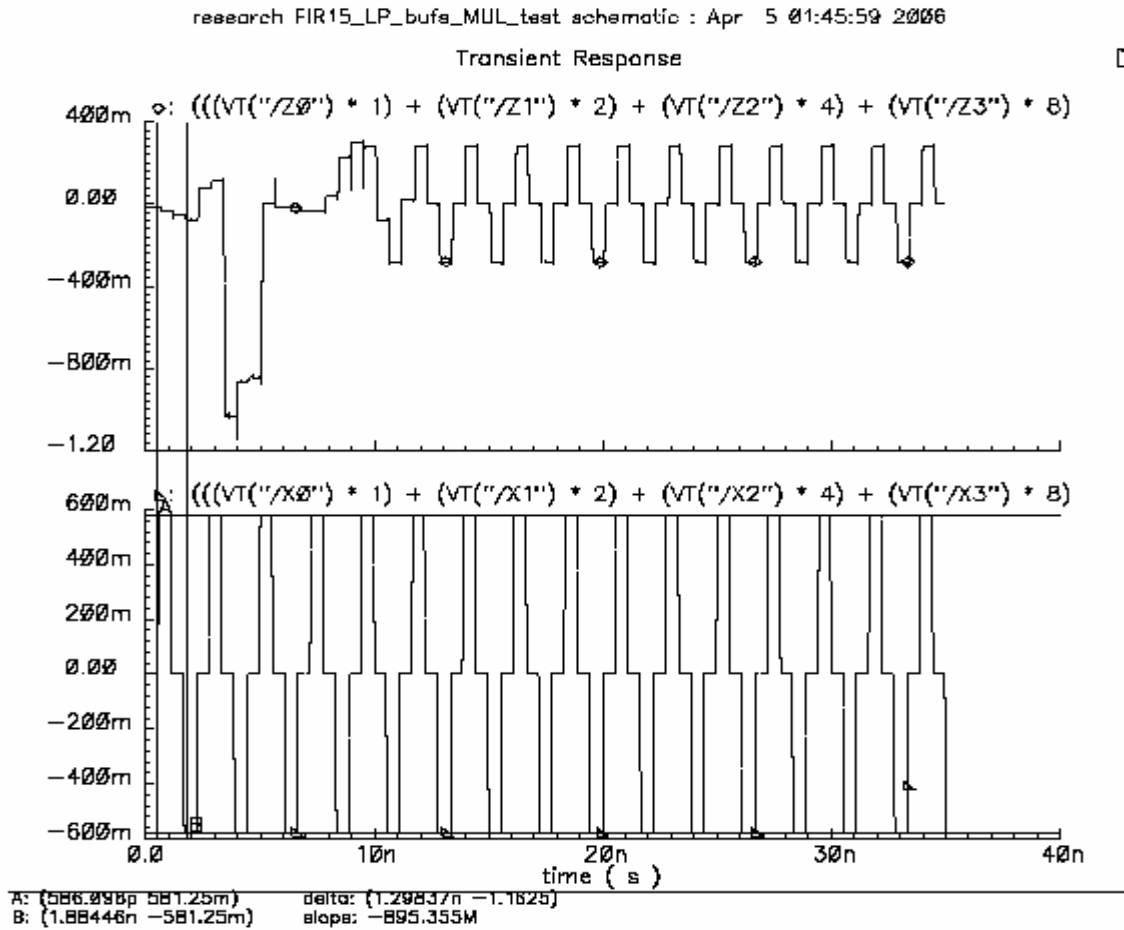


Figure 6.10b Low-pass FIR filter response to 450MHz input using new multipliers.

Topmost curve is the OUTPUT and bottom curve is the INPUT.

	Range	Mid-point	Power cons.	Periodicity
LP filter	+281.36 to -281.24mV	-39.14μV	49.32mW	2.22ns
HP filter	+281.24 to -281.24mV	-17.14μV	49.54mW	2.22ns
input	±581.25mV	0	—	2.22ns

Table 6.6 Characterization of the FIR filter to 450MHz using new multipliers.

Again the results are remarkably similar to those with the un-customized multipliers with the upside of the power savings using the customized multipliers. Comparing table 6.3b with table 6.6 above, for the low-pass filter the positive and negative peaks differ by 0.05 and 0.34% respectively. Similarly, the high-pass results

differ by 0.59 and 0.02%. The power savings with the customized multipliers is 18.88% (11.51mW).

6.2.3 Revalidation of the FIR filters for a 300MHz input sinusoidal.

Figures 6.11a & b respectively shows the high-pass and low-pass response of the FIR filter using the new multipliers for a 300MHz sinusoidal input and table 6.7 shows the corresponding characterization.

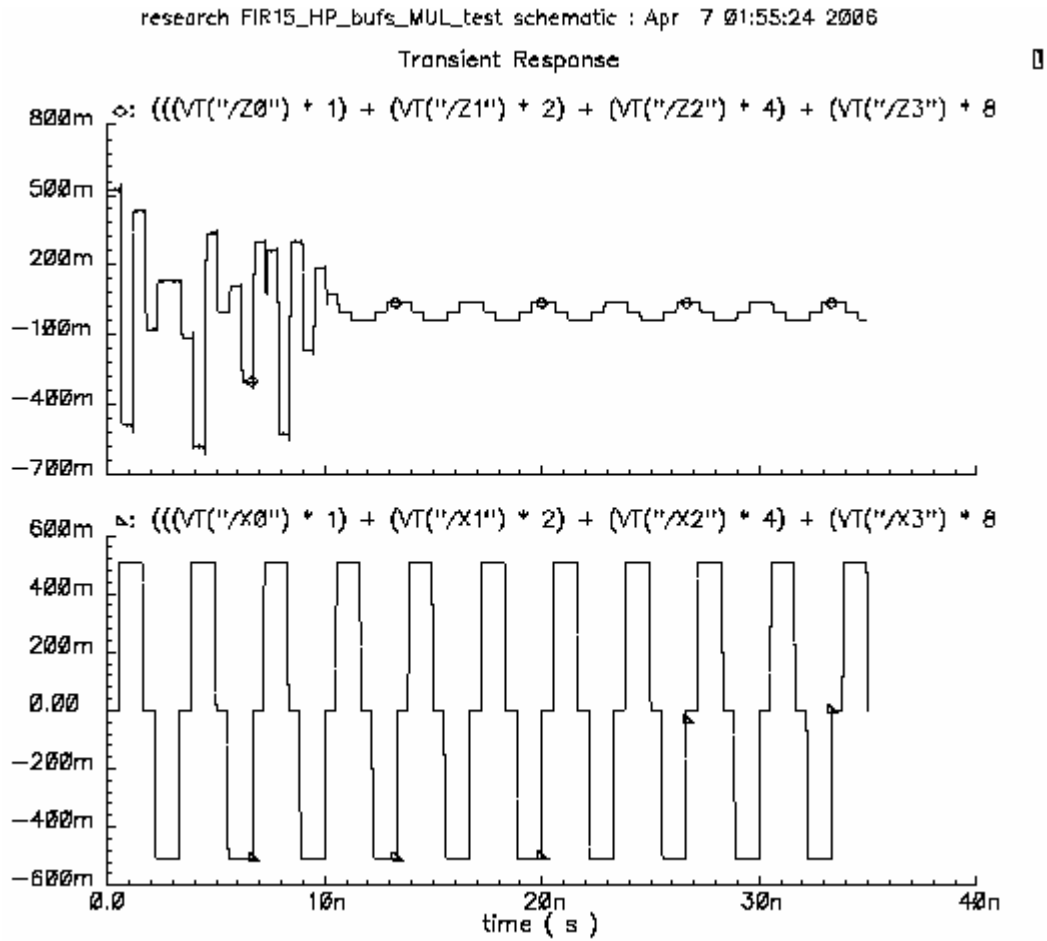
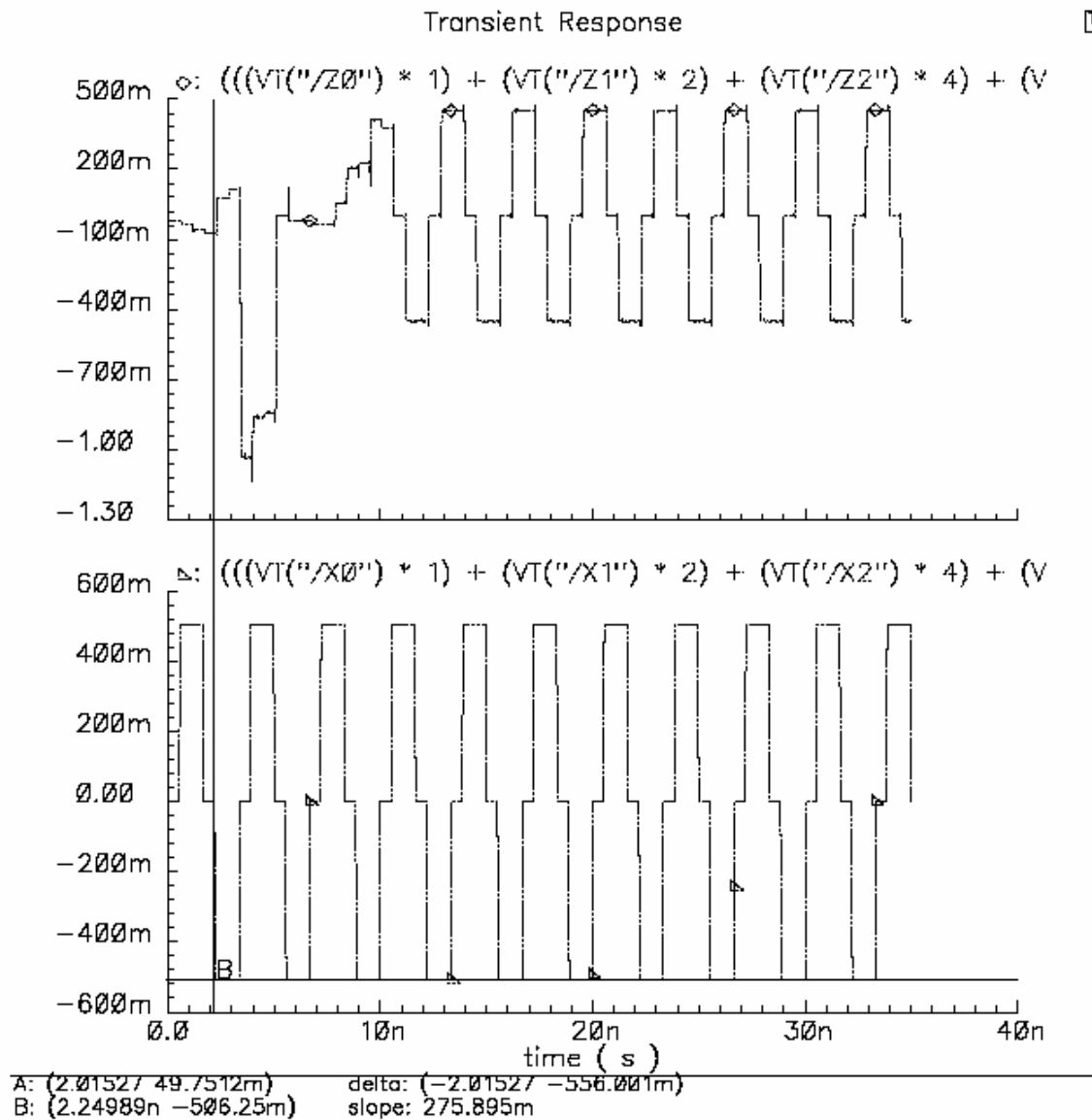


Figure 6.11a Low-pass FIR filter response to 300MHz input using new multipliers.



The result of table 6.7 shows considerable deviations from the previous two results comparisons. Comparing table 6.2b to table 6.7 shows that positive amplitude peak of the passband (LP filter) differ by 4.6% while that of the negative peak differ by 4.36%. Similarly, the stopband positive amplitude peak differs by 0.4% while the negative amplitude differ by large 66.71%!! This large difference may indicate that the original multiplier with coefficient ± 0.010011 operated on timing boundary and may have produced erroneous results explaining both the large negative amplitude drop in the stop band (HP filter) and also the pronounce negative droop of the passband results. Thus using coefficient ± 0.010010 resulted in not propagating a carry and hence being able to better meet the timing requirements. The results of table 6.7 is also much more symmetrical and does not exhibit any pronounce peaking of any sort. Table 6.7 shows that the passband exhibit a signal amplitude loss of 11.11% or 1.02dB on both positive and negative amplitudes. The stopband (HP filter) exhibits signal attenuation of 22.66 and 22.61dB on the positive and negative peaks respectively. Power saving of 19.46% was achieve using this customized multiplier.

7. Cadence Multi-Rate FIR Filter Operations

7.1.0 Two Tier FIR Filter Validation:

Because of simulation resource limitation, multi-rate operations of two frequencies (two bank level) will be demonstrated and thus successful operation necessarily imply that the filter tree can be extended to appropriate level for channel bandwidth. Figure 6.1 below shows the two tier tree structure.

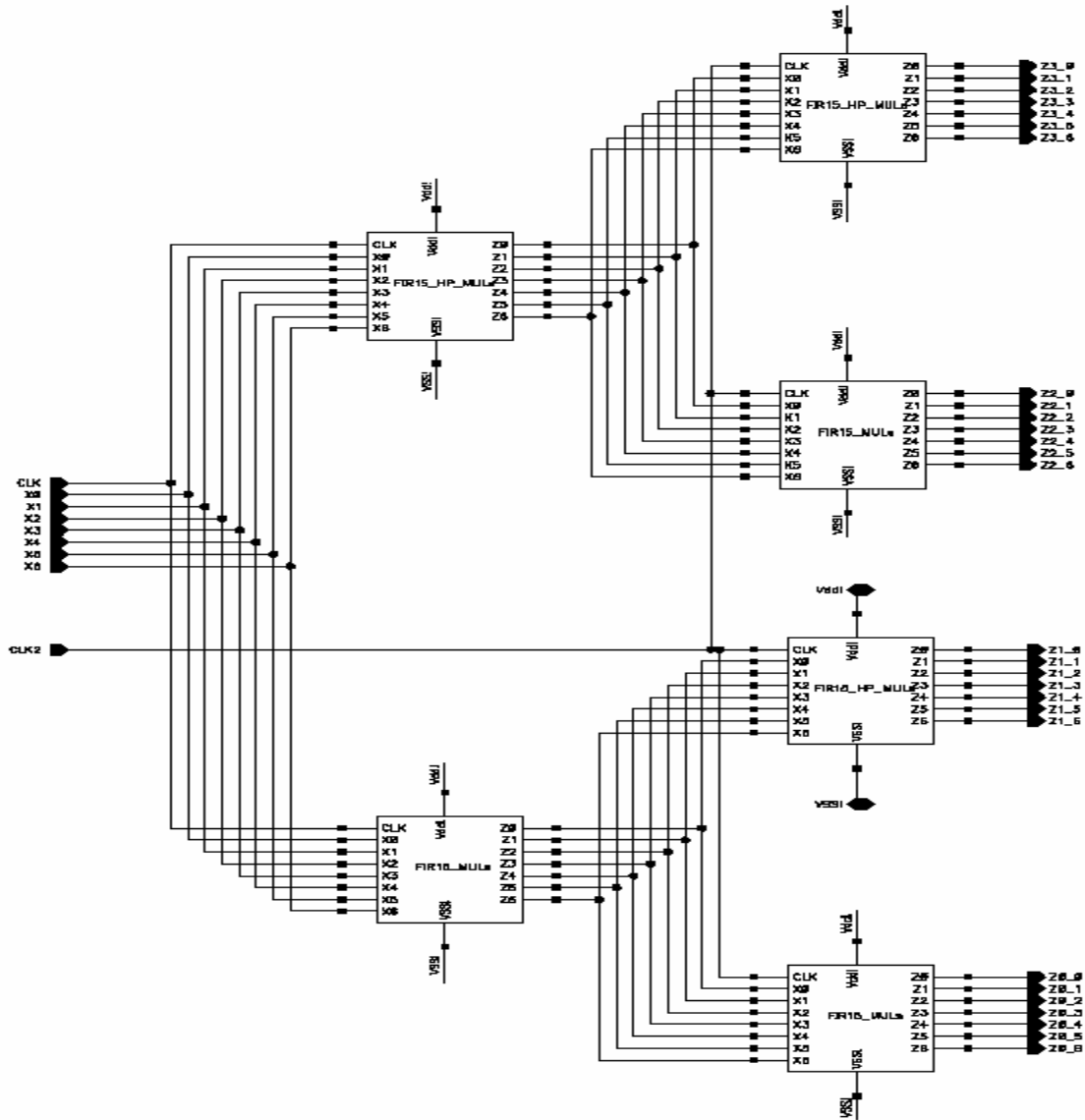


Figure 7.1 Two tier tree for cadence multi-rate operation.

Cadence simulation of the two tier tree shown above takes a week to complete! Thus simulation is painfully slow and renders extension of the tree to more tier impractical.

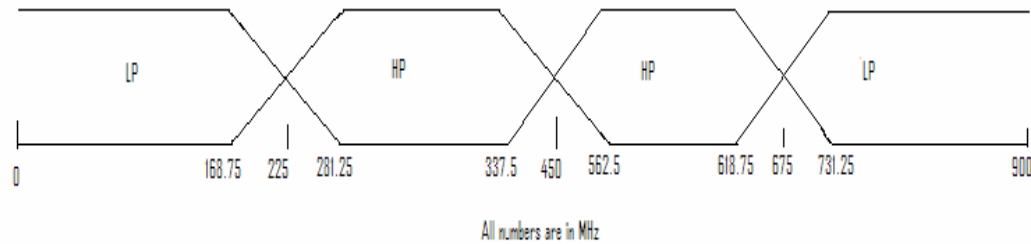


Figure 7.2 Two tier tree frequency response

In figure 7.1, the bottommost filter is the low-pass filter and represent channel 1 and arbitrate input frequencies into the 0 – 225MHz channel bins. The next filter is a high pass filter that resolves the input the 225 – 450MHz channel bin will be called channel 2. The third channel is supposed to be arbitrated with a high-pass filter and have frequency bin 450 – 675MHz as shown in figure 7.2, however, figure 7.1 have an error as its channel 3 is arbitrated with a low pass filter and will consequently resolve input frequencies into a 675 – 900MHz bin. Channel 4 in figure 7.1 is arbitrated with the high-pass filter and hence resolve signals into the 450 – 675 MHz bin. In short, comparing figure 7.1 to 7.2, channel 3 and 4 are reverse. The data collected with the erroneous arrangement of filters in figure 7.1 will still be presented because simulation for the corrected arrangement are unavailable because the computer network system broke on several occasion and thus prevented the completion of simulations. For comparisons, the filters in MATLAB will be arrange the same way as those in figure 7.1 to ensure proper comparisons. Figure 7.1 is first simulated with a 600MHz input and its results are illustrated with the following curves.

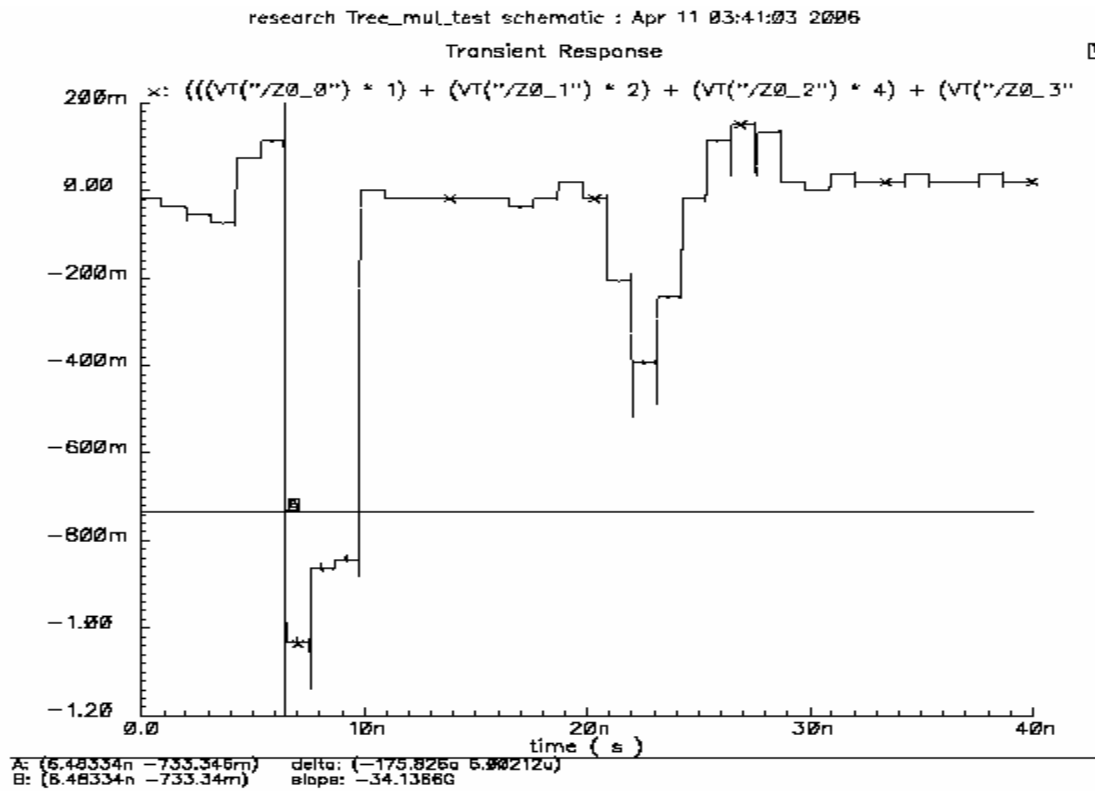


Figure 7.3a Channel 1 output for 600MHz input to 2-tier multi-rate FIR filters.

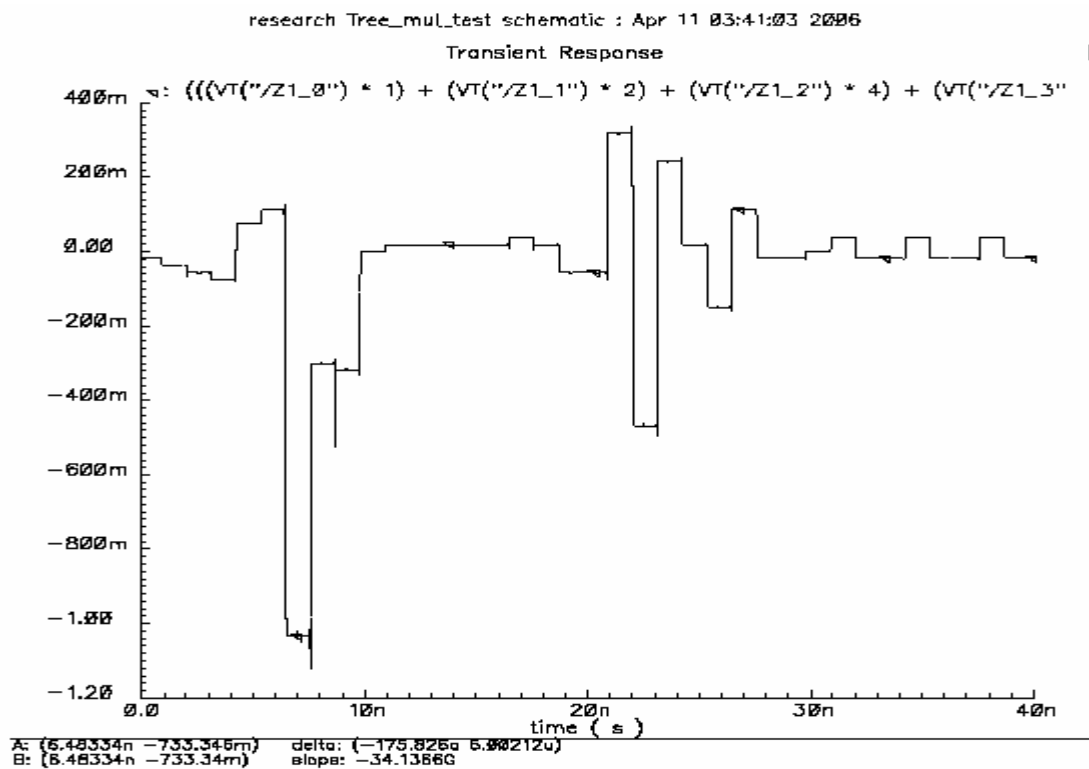
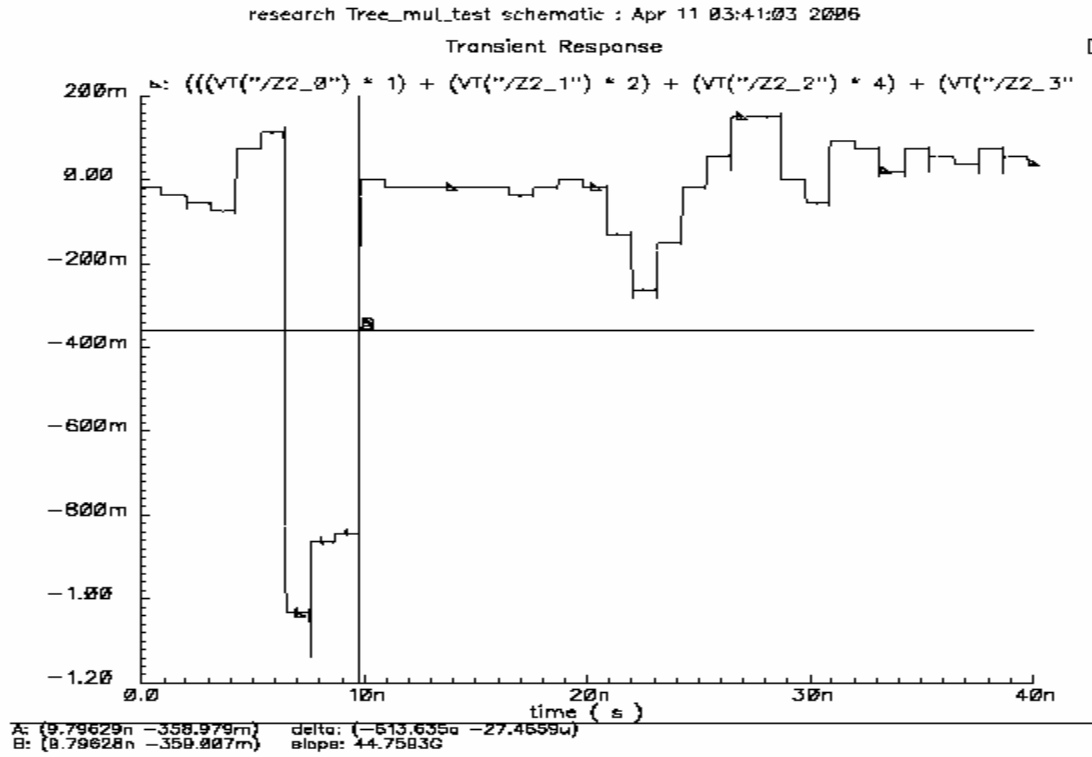


Figure 7.3b Channel 2 output for 600MHz input to 2-tier multi-rate FIR filters.



The latter curves clearly correctly resolve the 600MHz signal into channel 4 (as per figure 7.1 filter arrangement). Note that the second level of the two tier filter system is clocked at 900MHz and hence the Nyquist rate is 450MHz. Consequently, input signals of frequency greater than 450MHz will appear decimated by a factor of 2. Clearly shown in figure 7.3d is the detected 600MHz having a periodicity of 3.333ns or 300MHz. The peaks for the detected 600MHz signal is +412.508mV and -449.999mV. The measured total power dissipation is 276.58mW. Figures 7.4a through 7.4d shows the corresponding MATLAB simulation of figure 7.1 for the 600MHz input.

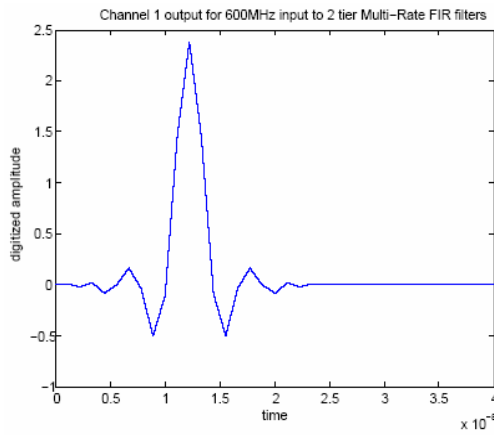


Figure 7.4a MATLAB channel #1
response to two tier filter system 600M.

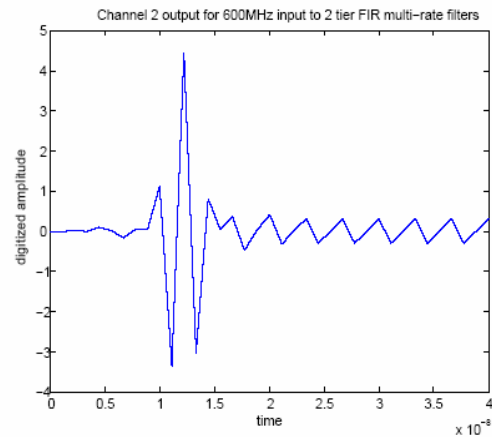


Figure 7.4b MATLAB channel #2
response to two tier filter system 600M.

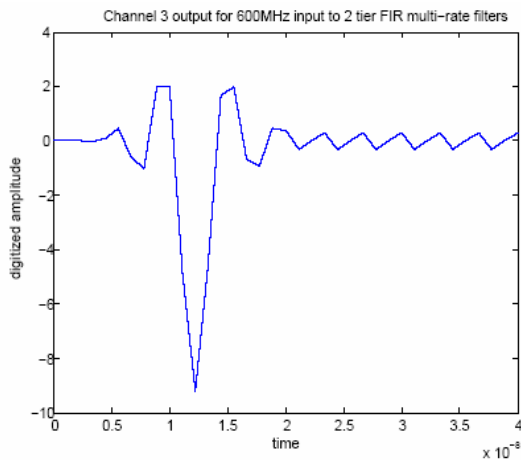


Figure 7.4c MATLAB channel #3
response to two tier filter system 600M.

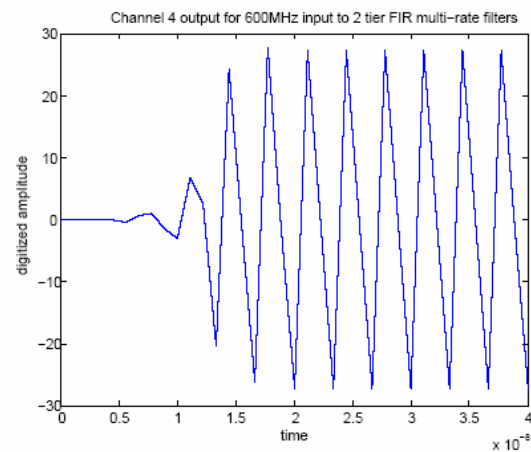


Figure 7.4d MATLAB channel #4
response to two tier filter system 600M.

The MATLAB responses in figures 7.4a through 7.4d is in perfect agreement with the cadence response.

The two tier system, figure 7.1, is now tested with a 100MHz sinusoidal input. The 100MHz signal will be expected to be resolve into channel 1 and no cross talk signal is expected as the 100MHz is well within the channel bin. The responses of the 4 channels are shown in figures 7.5a through d.

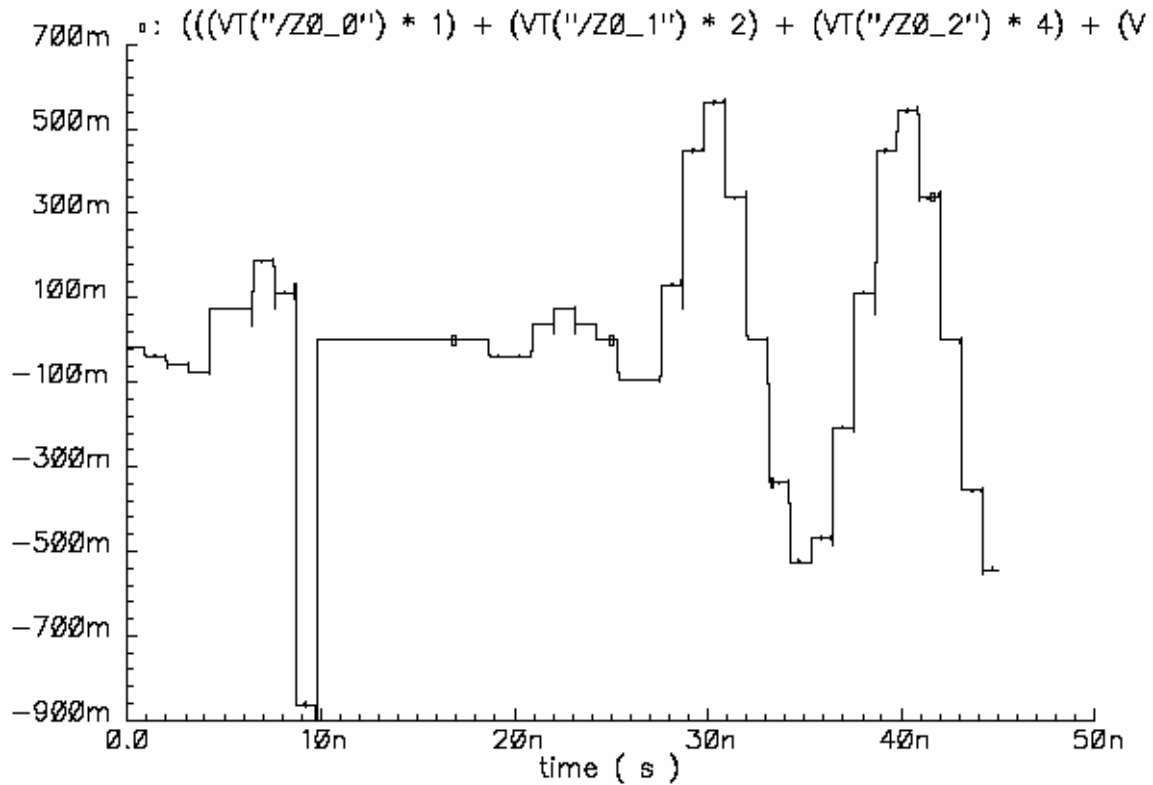


Figure 7.5a Channel 1 output for 100MHz input to 2-tier multi-rate FIR filters.

The detected 100MHz signal peaks is +562.509mV and -525mV. The periodicity is 10.00ns.

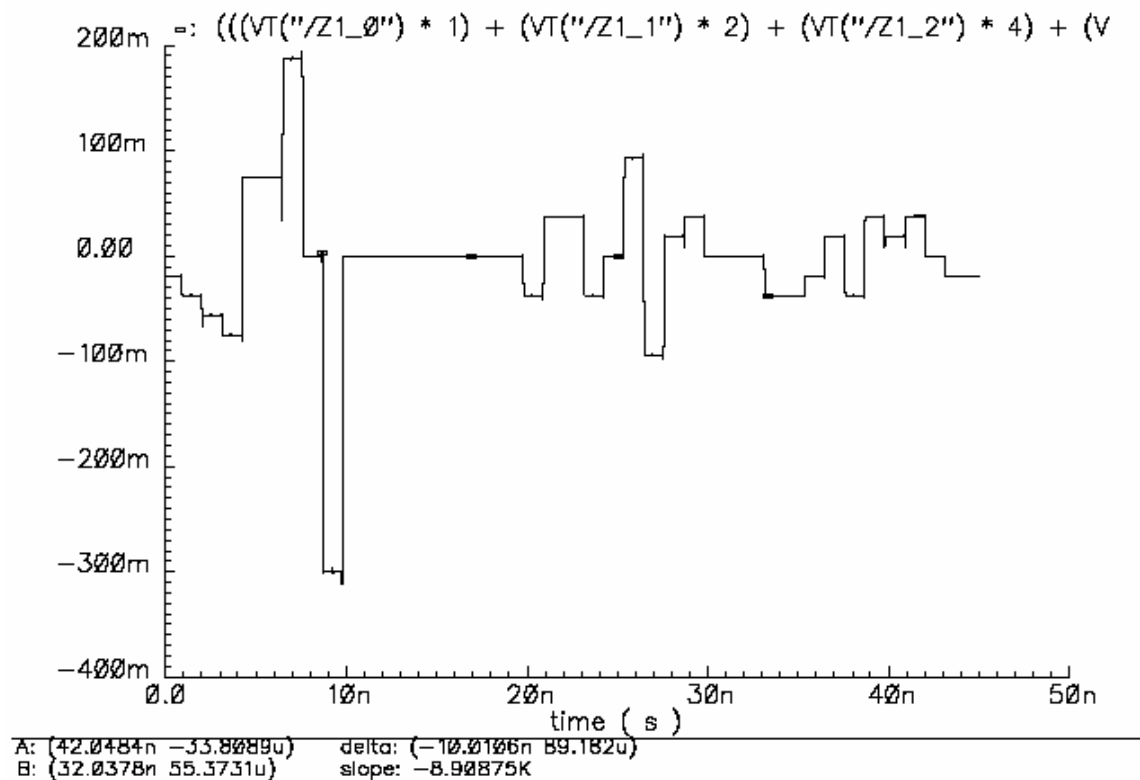
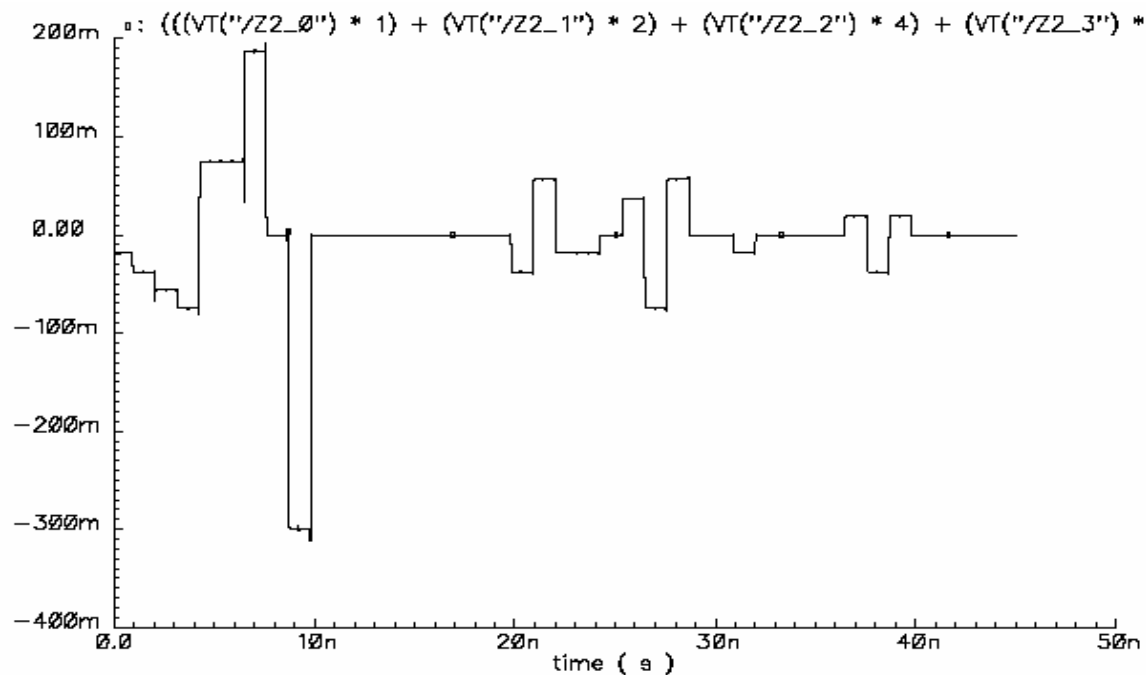


Figure 7.5b Channel 2 output for 100MHz input to 2-tier multi-rate FIR filters.



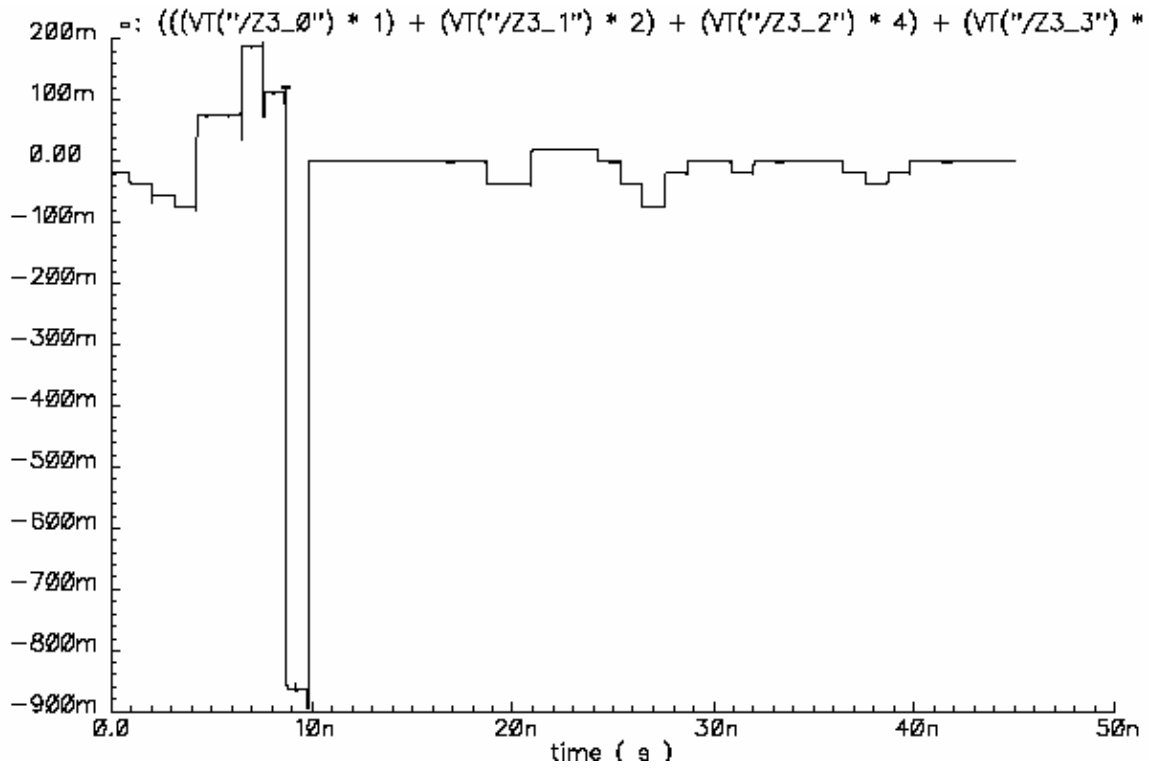


Figure 7.5d Channel 4 output for 100MHz input to 2-tier multi-rate FIR filters.

The corresponding MATLAB simulation for the latter cadence simulation are shown below.

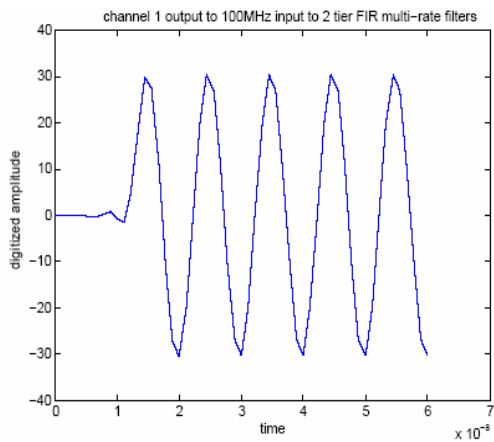


Figure 7.6a MATLAB channel #1 response to two tier filter system 100M.

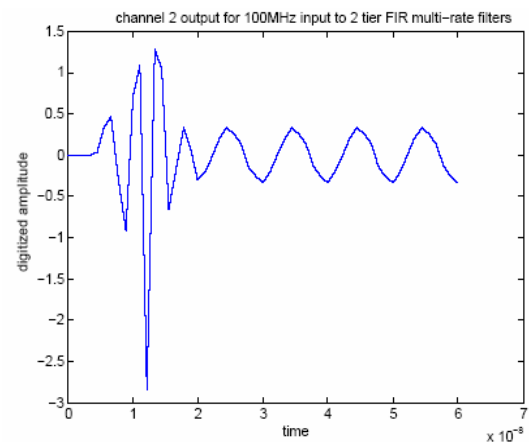


Figure 7.6b MATLAB channel #2 response to two tier filter system 100M.

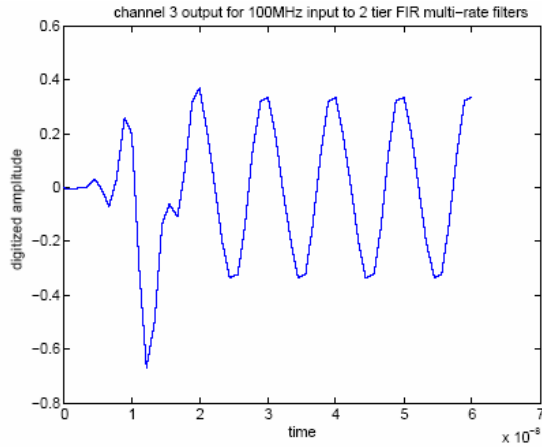


Figure 7.6c MATLAB channel #3
response to two tier filter system 100M

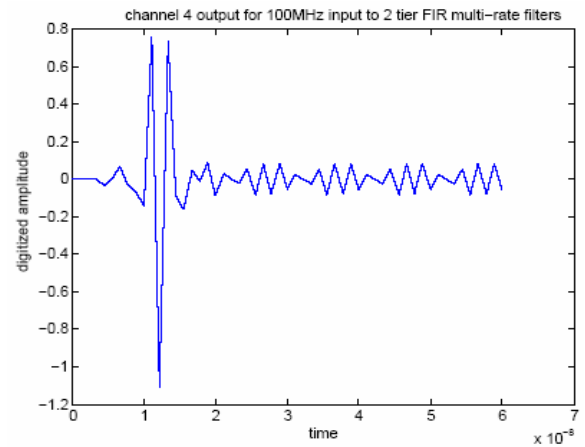


Figure 7.6d MATLAB channel #4
response to two tier filter system 100M

There is a strong correlation between the MATLAB and cadence simulations. One may therefore infer that the cadence CMOS design implementation does perform channel arbitration and hence using the systolic array of FIR filters is indeed a viable way to design a channelized receiver. For a three tier systolic array system there will be a total of 14 filters and power consumption will be 646mW. Logically, a 4 tier system will have 30 filters and will consume 1.39mW. This power consumption is comparable with the conventional receiver.

8. Conclusion and Future Work

This section is a reflection on the results of the work presented, describes some of the limitations encountered for the FIR filter implementation, and finally, suggests possible circuit design techniques to improve performance for design implementation.

8.1 Conclusion on the work presented:

The CMOS design implementation of the FIR filter for a Systolic Array of Multi-Rate filters to perform channel arbitration for a channelized receiver was demonstrated. It is fully functional at 1.8GHz, however, this does not meet initial speed goals expectation of 3.2GHz. Area and power consumption is large because of the use of parallelism (select adders), and pipelining to achieve high performance. Assuming the filter coefficients are fixed, customization of the multipliers by 1) discarding circuitry whose output are not used because of truncation, and 2) coefficients bits of zero in which the generate (AND) function does not propagate carries are also discarded. This customization resulted in a 19% reduction of power consumption.

8.2 Limitations:

Because this is a large design, and thus has very large amount of switching nodes, the Cadence spice based simulator is not very suited for verification and consequently resulted in extremely large simulation runs. This is why verification of the Systolic Array of FIR filters was limited to two-tier in Cadence. The used of adders in FIR filter design is pervasive. As a result, speed, area, and power consumption will be greatly dependant on the full adder design topology.

8.3 Suggestions for Future Work:

Pass transistor logic (PTL) families was once hyped to be the dominant logic family of the future as it was shown to be very fast and compact (less area) and consume very little power. Having buy into this paradigm, significant circuit failure was realized when attempts was made to design logic function in $0.13\mu\text{m}$ technology using PTL type styles. Weeks of failure forced the use of complimentary pass transistor (CPTL) design with very little or no performance gains. Through debugging and further reading it was realize that the threshold voltage, v_t , is now a much greater percentage (35% or more!!) of supply voltage for $0.13\mu\text{m}$ and smaller technologies. In light of this, prediction of the demise of static CMOS topology is untrue. The moral of the story here is DO NOT DESIGN with PTL type LOGIC unless there is a clearly demonstrated performance advantage in the $0.13\mu\text{m}$ or smaller technology!

Unfortunately, my inexperience did lead me to choose the fastest available working full adder from research papers. The issue of very little drive strength under output load condition was demonstrated in Appendix B. With the current experience I now have, I would not recommend the use of this adder. I now believe (not yet proven) that using a well optimize traditional Static CMOS full adder would have yielded better performance. Base on my experience gained during this thesis, to achieve the raw speed goals of 3.2GHz and greater, I would recommend the use of DYNAMIC CIRCUIT TOPOLOGY for the multi-bit adders. Particularly, using the Manchester carry chain design would have resulted in a very fast very compact design. Of course one does need to be a little more careful with dynamic circuits which are sensitive to charge leakage and noise issues.

As shown in section 5.1.0, preconceive beliefs of data being stable around the sampling edge of the DFF and thus data appearing on the Q output on the triggering edge have been place in serious doubts. In other words, data changes during sampling phase of the clock are transparent to the output, thus set up and hold times now occur on the triggering edge of the DFF. This realization makes more prudent to use a latch for sequencing the design as it will have less area and hence reduce the power dissipation. However, latches require the use of non-overlapping phase clocks to avoid race problems. Generation and distribution of non-overlapping is difficult. To avoid this, the use of

PULSE LATCHES is a workable solution. Provided the transparent pulse width of the latch is smaller than the delay through the combinational logic, data will propagate through the system once every clock cycle on each pulse and hence avoids the race issues encountered using sample two phase latch system. This concept is illustrated in figure 7.1 below. The reduce power consumption of using a pulse latch may help to compensate for the increase power dissipation of using dynamic logic. The pulse gator logic may be built into the latch and hence avoids distributing the pulse over a large area.

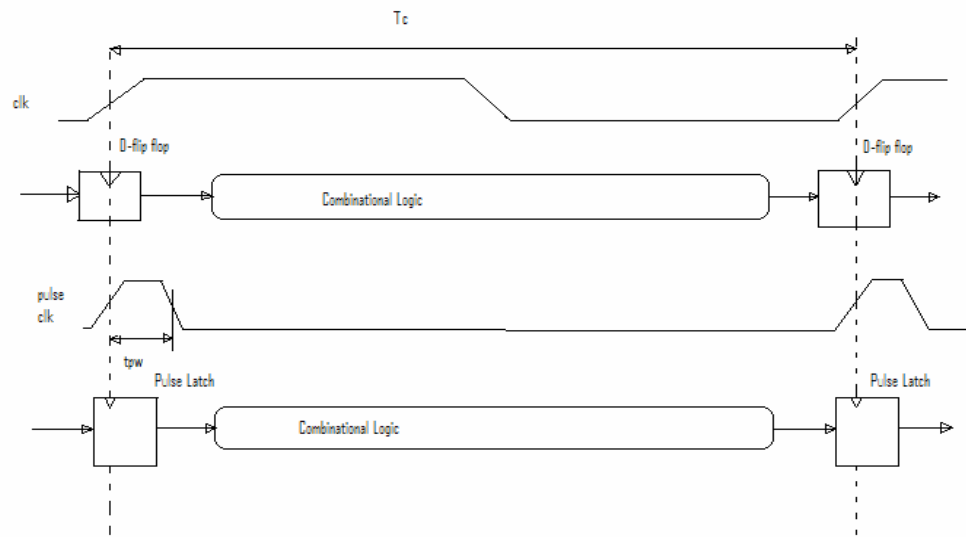


Figure 8.1 The pulse latch timing conception.

APPENDIX A

MATLAB and SIMULINK Designs

A1.1.0 MATLAB Program for Generating FIR Filter Coefficients.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% David M. Rodney                                     %
% Thesis Project                                     %
% Initially created by Dr R. Siferd                  %
% MATLAB program to generate FIR filter              %
% coefficients and frequency response                 %
% using the Parks-McClellan algorithm.               %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

clear all;

outName0 = 'coefficient_dec0.dat'; %FIR coefficients written to this file.
outName1 = 'coefficient_dec0.dat';

n = 16; % The # of filter TAPS

a1 = [1,1,0,0]; %low pass filter
a2 = [0,0,1,1]; %high pass filter

f1 = [0,0.375,0.625,1]; %75/225 125/225 fs = 400M.
bL1 = firpm(n,f1,a1);
bH1 = firpm(n,f1,a2);
fvtool (bL1,1,bH1,1);
title ('Green: 0 < passfreq < 100MHz & Blue: 100MHz < passfreq < 200MHz sampling at
400MHz');

fid0 = fopen ( outName0, 'w' );
fid1 = fopen ( outName1, 'w' );

for j=1:1:n
    fprintf (fid0, '\t%5.4f', bL1(j));
    fprintf (fid1, '\t%5.4f', bH1(j));
end;

fclose (fid0);
fclose (fid1);
```


A1.1.1 Simulink Design for MATLAB Simulation.

The model below is used for generating quantized samples to be used for both Cadence and MATLAB simulation. Note the gain is set to 31 as this is the maximum input value to not cause overflow in the cadence design. The frequency of the sinusoidal is set in the Sine Wave1 generator shown below and the corresponding digitized vales are collected in the simout workspace after simulating the model.

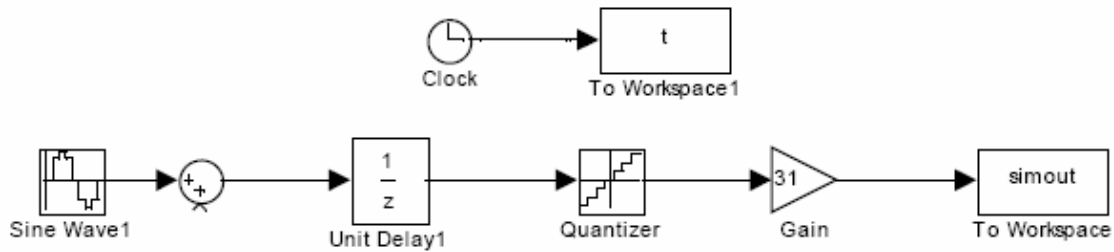


Figure A1 Simulink model for generation digitized samples.

The simulink model below is used for simulating the MATLAB response of the FIR filters operating at a sample frequency of 1.8GHz. In figure A2, simout plots the digitized input while simout1 and simout3 plots the low-pass and high-pass FIR filter response respectively.

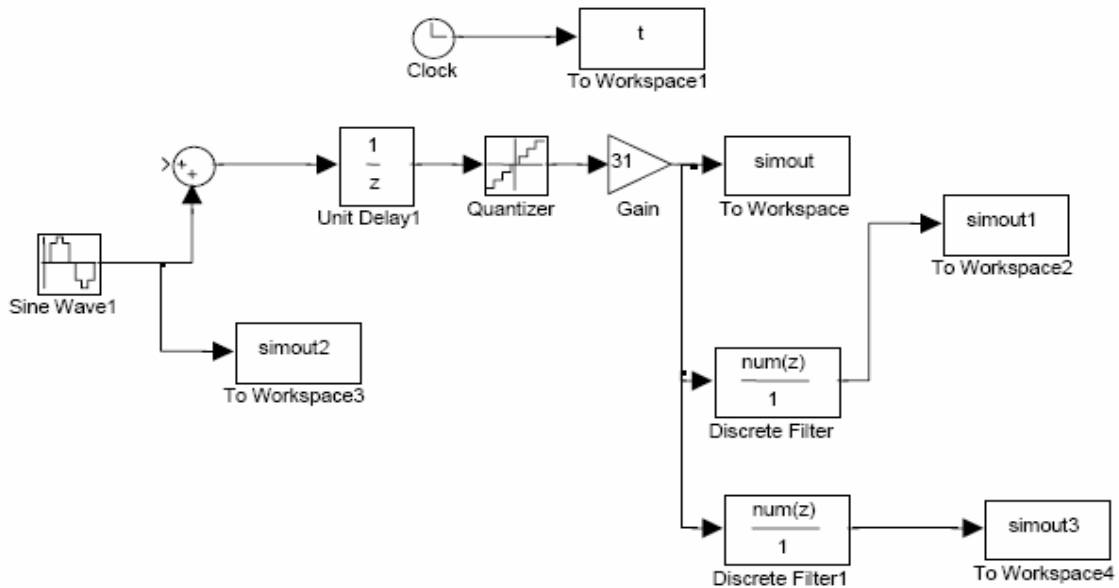


Figure A2 Simulink model for simulating MATLAB high-pass and low-pass FIR filters.

The simulink model for simulating multi-rate FIR filters is shown in figure A3 below. Again, the ADC and first tier of filters samples at 1.8GHz. The second and third tier of filters are sampled at 900MHz and 450MHz respectively.

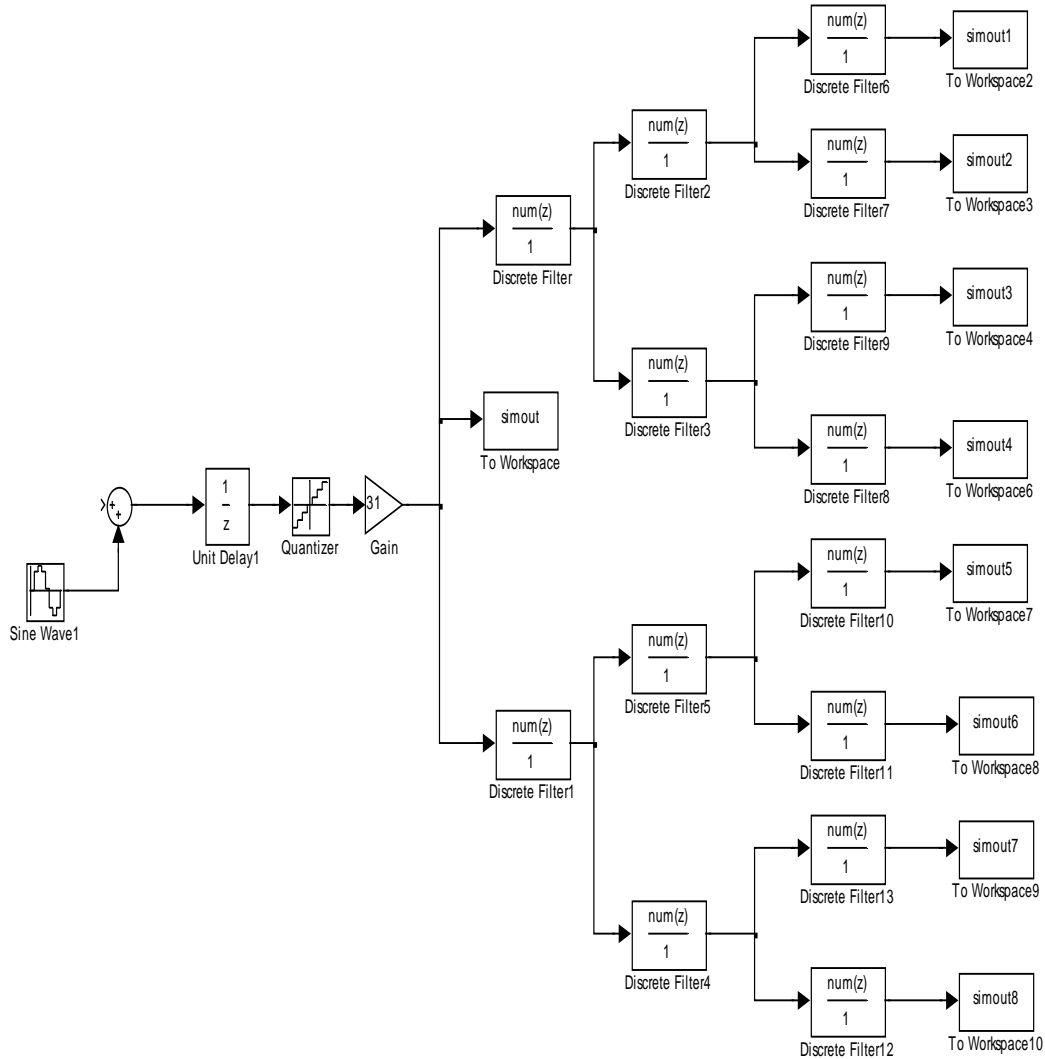


Figure A3 Simulink model for Multi-rate FIR filter operation.

Using the above model, the output of from the third tier of filter are collected into “simout” bins that represent channels 1 (topmost bin) through 8 (bottommost). Each of these bins’ channel bandwidth is 112.5MHz and spans a total range from 0 to ~900MHz. The MATLAB program for plotting the rms value of each channel as a function channel centre frequency is shown below.

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% David M. Rodney %
% Thesis Project %
% MATLAB program for plotting output rms values %
% as a function channel/bin centre frequency. %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

f=0;
a=0;
in=0;
in(1)=mean(abs(simout1));
in(2)=mean(abs(simout2));
in(3)=mean(abs(simout3));
in(4)=mean(abs(simout4));
in(5)=mean(abs(simout5));
in(6)=mean(abs(simout6));
in(7)=mean(abs(simout7));
in(8)=mean(abs(simout8));

f(1)=sqrt(in(1)*in(1));
f(2)=sqrt(in(2)*in(2));
f(3)=sqrt(in(3)*in(3));
f(4)=sqrt(in(4)*in(4));
f(5)=sqrt(in(5)*in(5));
f(6)=sqrt(in(6)*in(6));
f(7)=sqrt(in(7)*in(7));
f(8)=sqrt(in(8)*in(8));

for i=0:7;
    a(1+i)=56.25+i*112.5; % 112.5 channel bandwidth
end;

stem(a,f);

```

A.1.1.2 MATLAB Program for Channel Arbitration Using Bin's RMS Comparison

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```

% David M. Rodney %
% Thesis Project %
% MATLAB program for resolving a single input to its %
% appropriate channel using channel bin rms comparison. %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

f=0; a=0; d=0; in=0;

in(1)=mean(abs(simout1));
in(2)=mean(abs(simout2));
in(3)=mean(abs(simout3));
in(4)=mean(abs(simout4));
in(5)=mean(abs(simout5));
in(6)=mean(abs(simout6));
in(7)=mean(abs(simout7));
in(8)=mean(abs(simout8));
f(1)=sqrt(in(1)*in(1));
f(2)=sqrt(in(2)*in(2));
f(3)=sqrt(in(3)*in(3));
f(4)=sqrt(in(4)*in(4));
f(5)=sqrt(in(5)*in(5));
f(6)=sqrt(in(6)*in(6));
f(7)=sqrt(in(7)*in(7));
f(8)=sqrt(in(8)*in(8));

for i=0:7;
    a(1+i)=56.25+i*112.5; % 112.5 channel bandwidth
end;

temp = f(1); bin = 1;

for i=1:7;
    if (f(i+1) > temp)
        temp=f(i+1);
        bin = i+1;
    else %do nothing
    end;
end; %end for

%temp now contains the rms value of the channel and
%bin is the resolved channel of the signal

for i=1:8
    if (bin==i)
        d(i)=temp;
    else
        d(i)=0;
    end; %end if
end; %end for

stem(a,d);

```

APPENDIX B

Additional Cadence simulations:

B1.1.0 The Full-Adder

Figure B1 shows the design of the full-adder as illustrated in [5] and figure B2 shows the corresponding simulation with no output load.

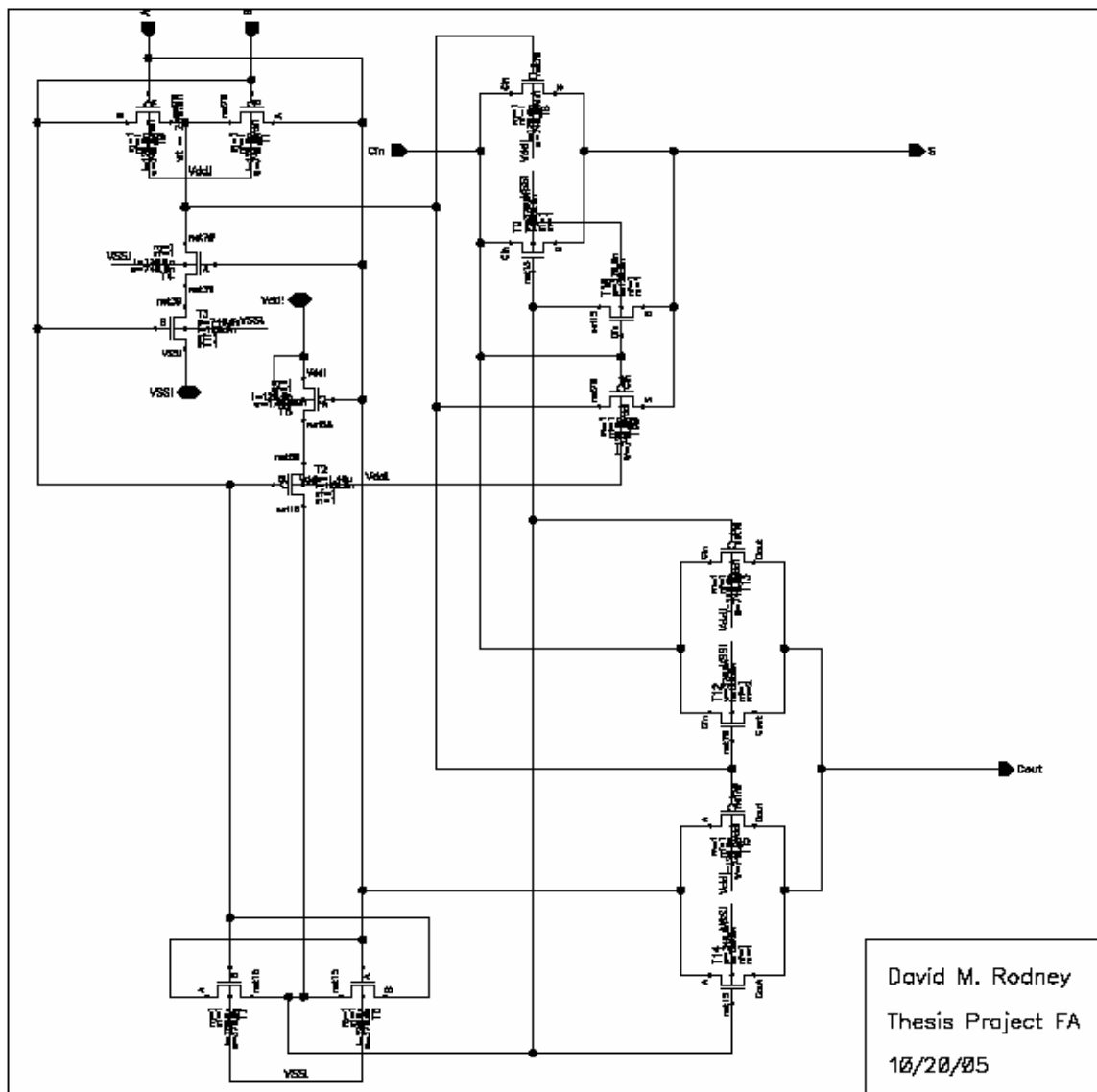


Figure B1 The designed full-adder.

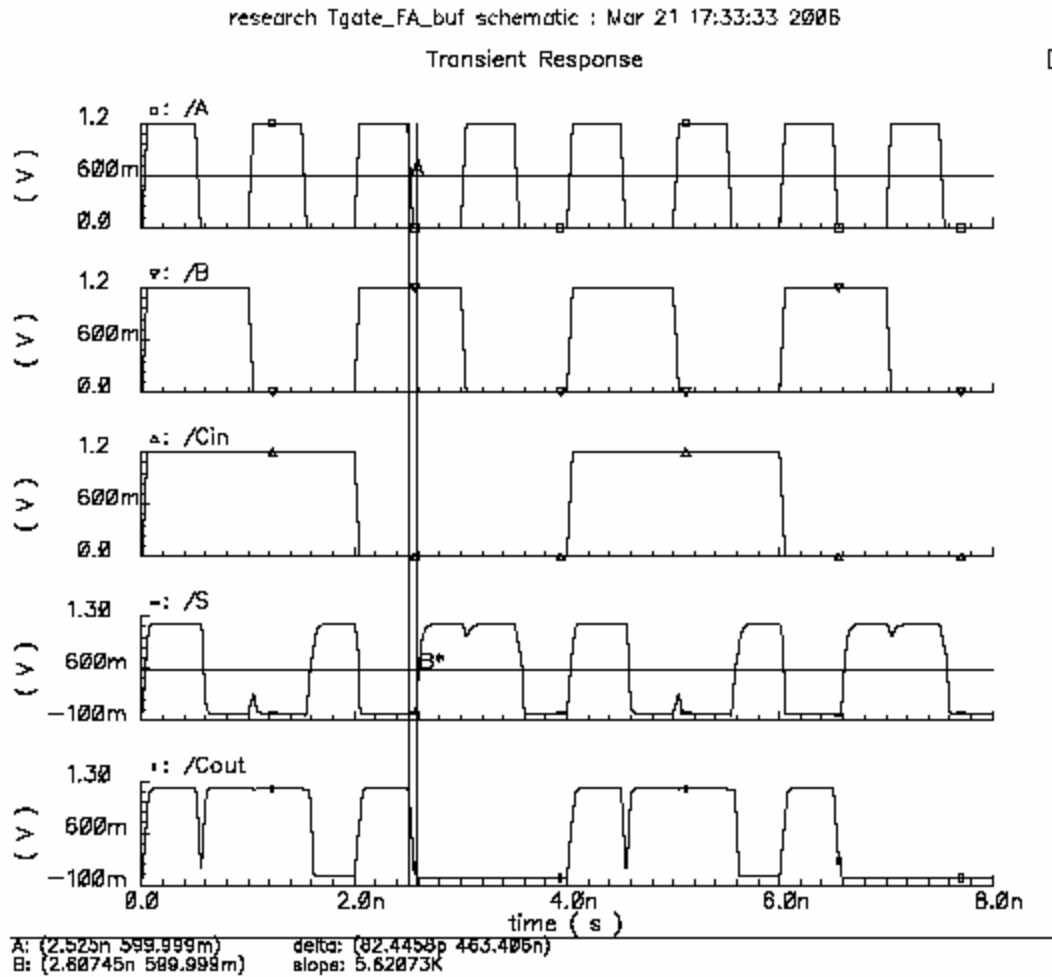


Figure B2 Verification of the full-adder under no load condition.

This simulation performed with no load on the outputs looks excellent with delays up to 82.45ps for the SUM output, S. Figure B3 shows the full adder self-loaded with it's A input and figure B4 shows the corresponding simulation. The simulation now shows delay increases to 164.12ps and some SUM output have large rise time and just barely manage to reach the VDD rail. The fastest input value is the pulsed A-input with a 1ns period. Clearly this design does not have strong drive strength and it may therefore be necessary to use an output buffer to provide appropriate drive strength for correct operation. However, for this combination of input the CARRY output appears acceptable.

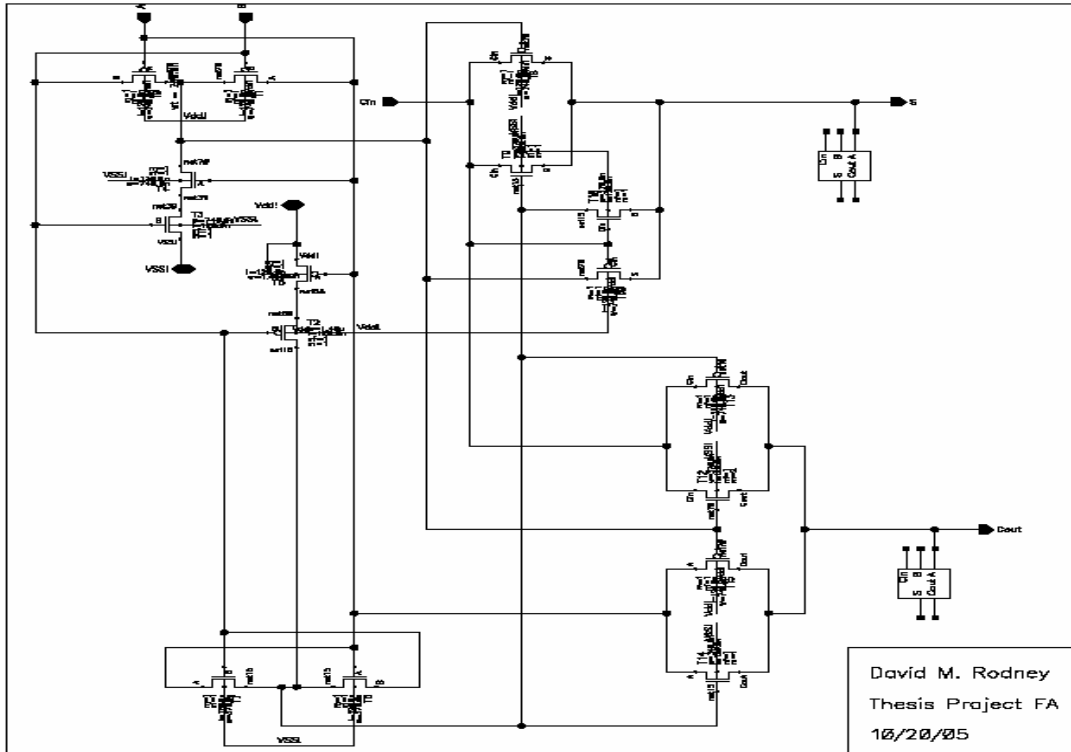


Figure B3 The full adder self loaded with it's A input.

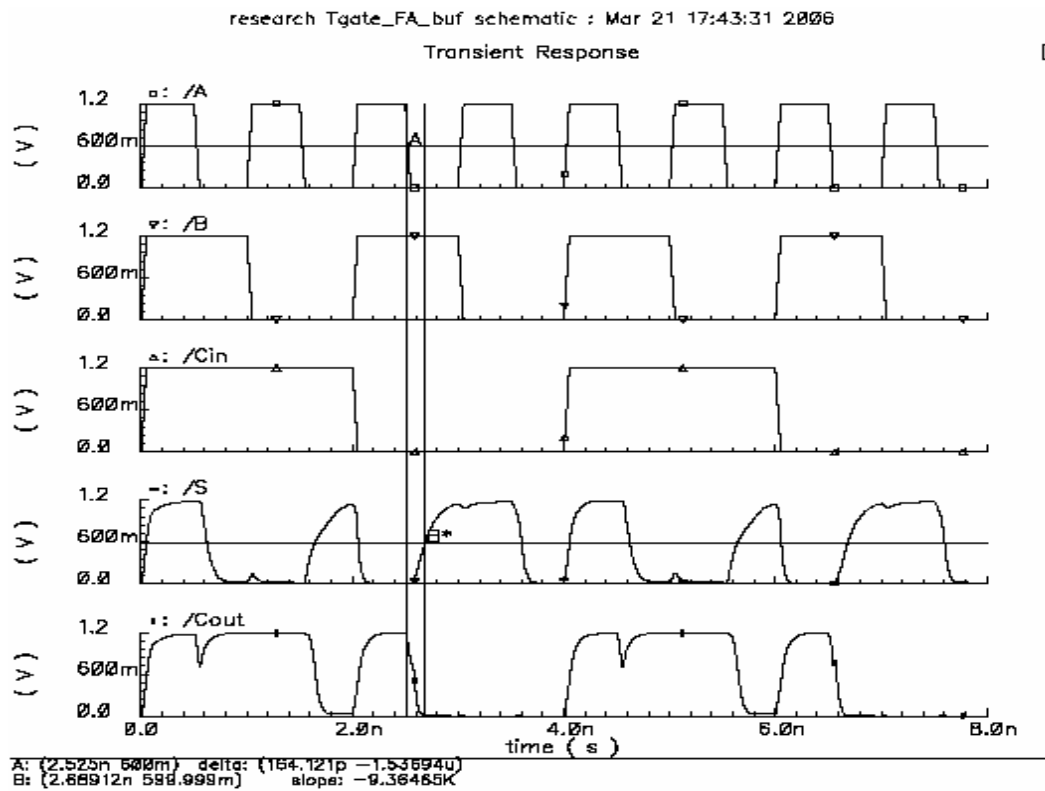


Figure B4 Verification of the self loaded full-adder.

B1.1.1 filter verification.

Sampled values from a digitized sinusoid are used to verify filtering operation of the filter. First, the high-pass (HP) filter is tested at 1.9GHz with a 100MHz input also sampled at 1.9GHz. This result is shown in figure B5 below. Clearly from figure B5, the filter is performing rejection operations as expected but the three pronounce spikes indicate a possible computational error or timing violation of one or more of the circuit components. The 100MHz signal is then tested on the low-pass (LP) filter and its result is shown in figure B6. In all the following figures, Xs are inputs and the Zs are the outputs.

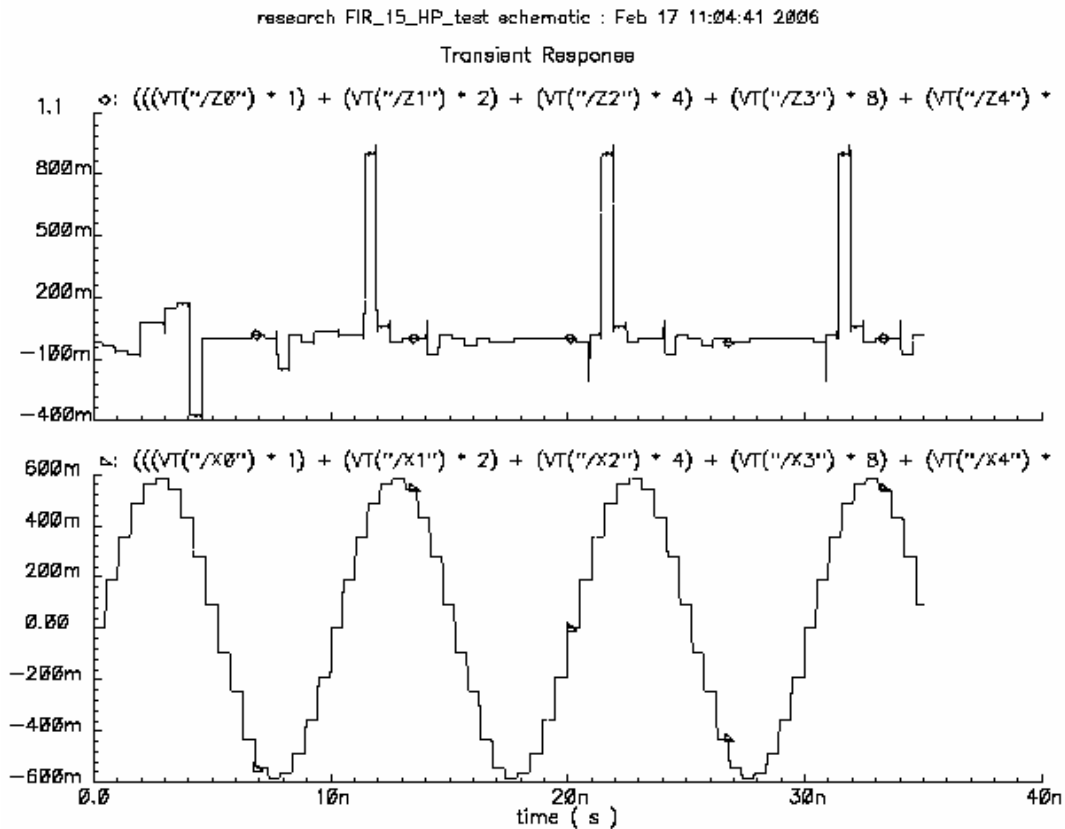


Figure B5 Simulation of the high-pass FIR filter with 100MHz sinusoid at 1.9GHz.

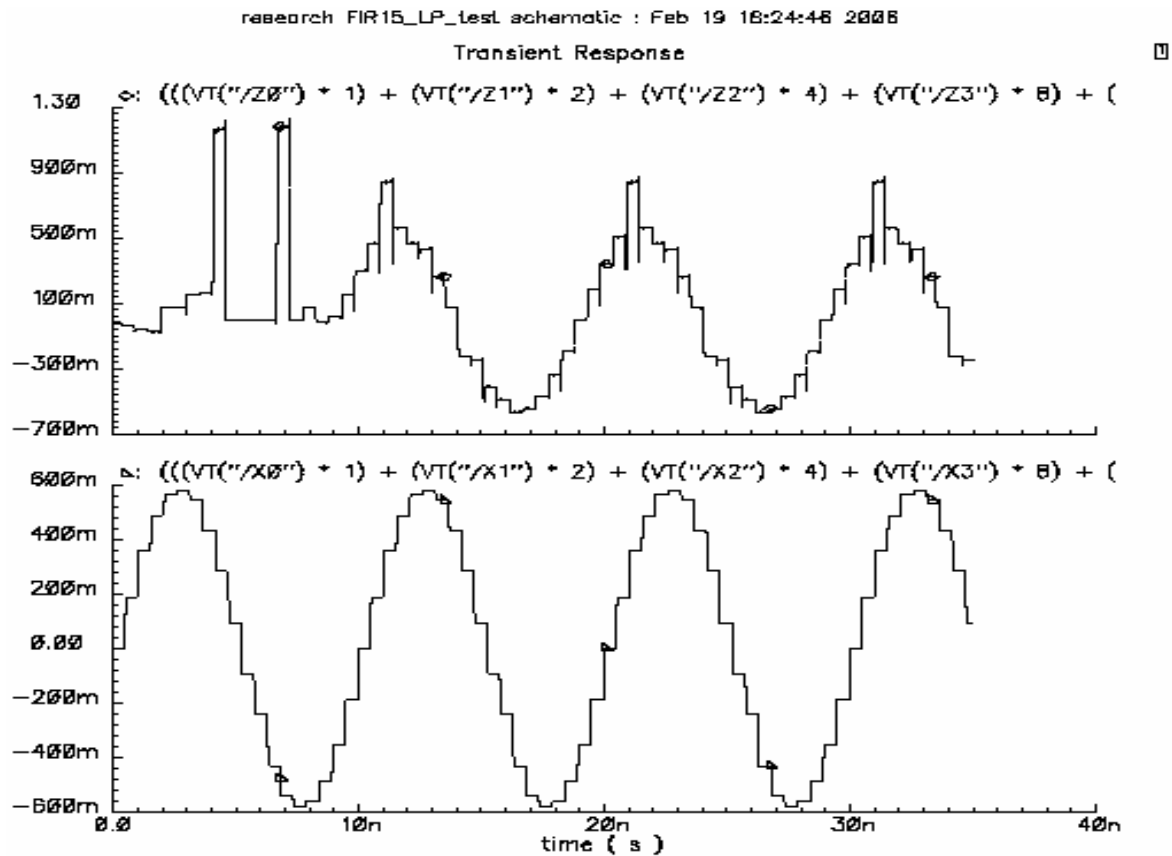


Figure B6 Simulation of the low-pass FIR filter with 100MHz sinusoid at 1.9GHz.

The 100MHz signal is well within the pass band of the low-pass filter and thus the same 100MHz signal is expected at the output of the low-pass filter. Figure B6 shows the low-pass filter passing the 100MHz signal but again the pronounce spikes indicates errors of some sort.

The simulation of the HP and LP filters are repeated at a much lower frequency to observe any differences with 1.9GHz operation. This frequency is chosen to be 1.5GHz and the respective results for HP and LP operations are shown in figures B7 and B8 respectively.

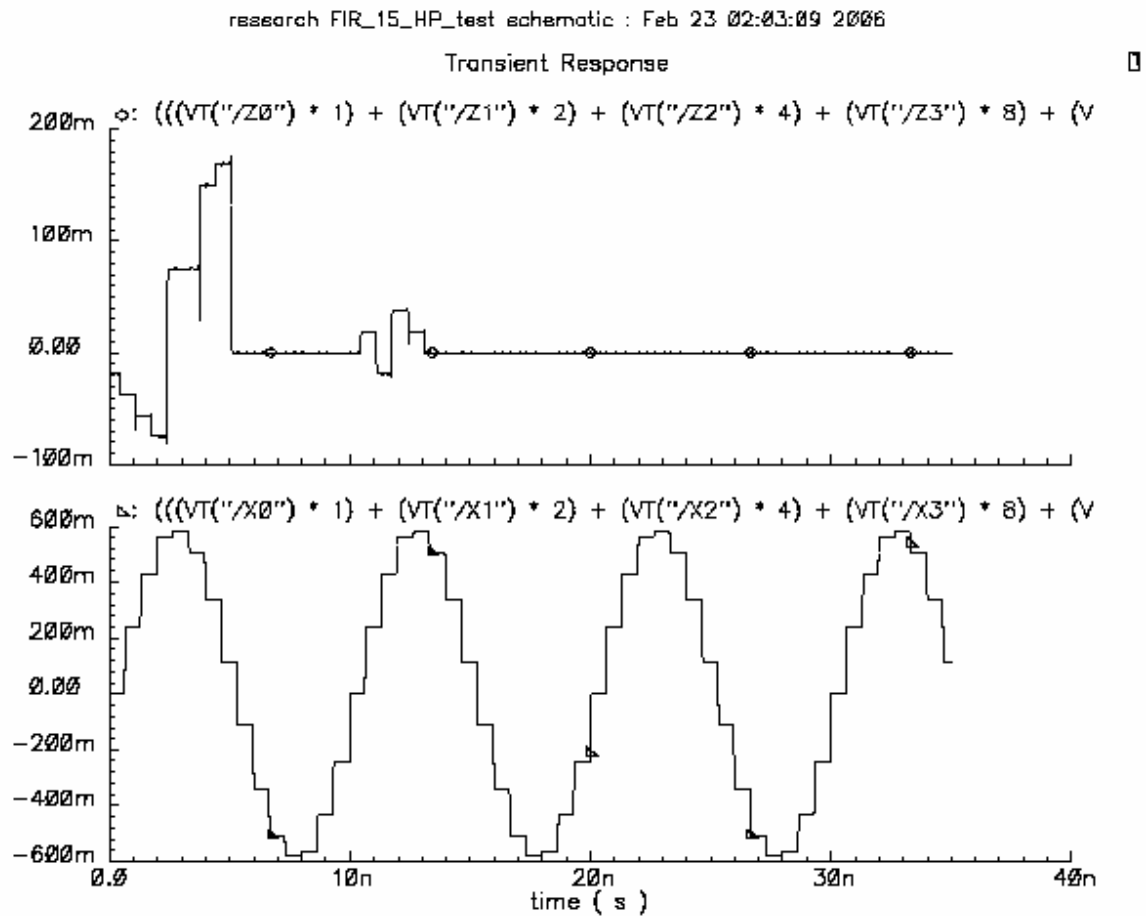
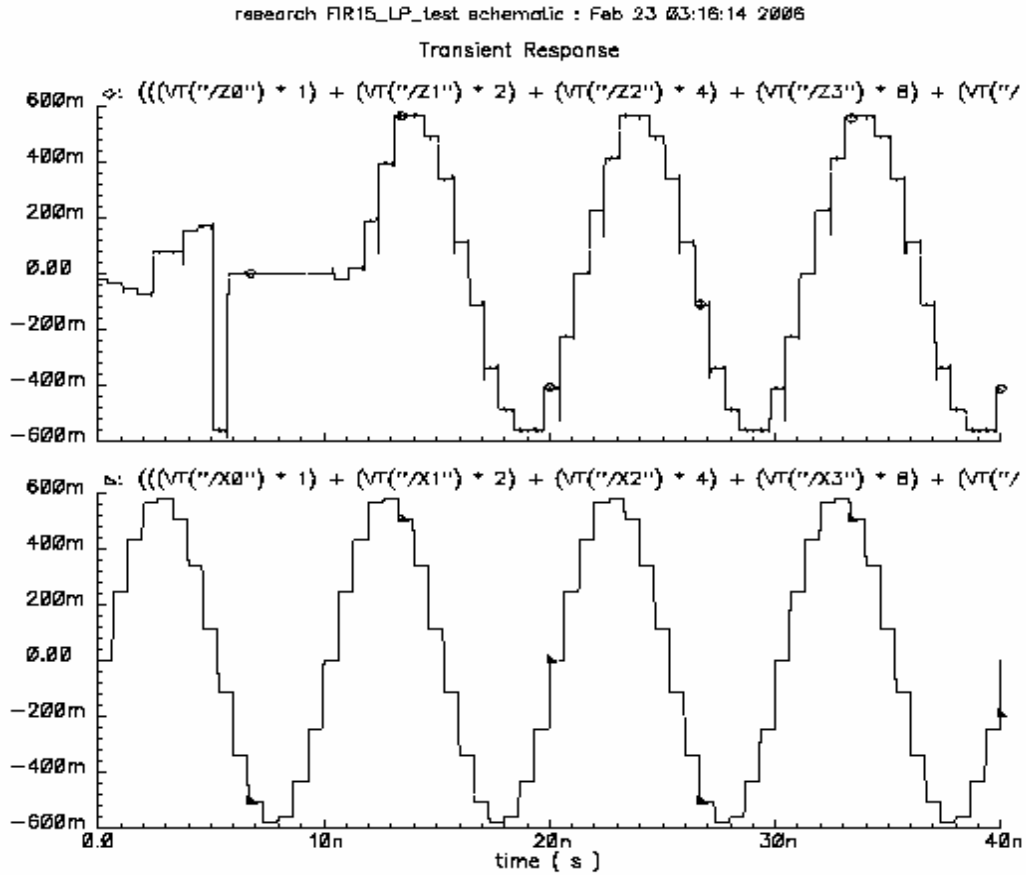


Figure B7 Simulation of the high-pass FIR filter with 100MHz sinusoid at 1.5GHz.

The latter simulation shows the high pass filter rejecting the low frequency 100MHz signal without pronounce peaking. This will now become the “gold standard” model for which a higher frequency of operation must match. Figure B8 shows the low pass filter passing the low frequency 100MHz signal as expected without any peaking distortion. Thus higher frequency of operations must show little distortion to be considered valid.



References:

[1], [2] M. Morris Mano and Charles R. Kime, Logic and Computer Design Fundamentals, p. 140 & 16 2nd edition Prentice Hall 2001.

[3], [7] John G. Proakis and Dimitris G. Manolakis, Digital Signal Processing Principles, Algorithms, and Applications, p. 21, 36, 620 3rd edition Prentice Hall 1996.

[4] B. P. Lathi, Signal Processing & Linear Systems, p. 319 & 326 Oxford University Press 1998.

[5] Ahmed M. Shams and Magdy A. Bayoumi, A Novel High-Performance CMOS 1-Bit Full-Adder Cell, p. 478 – 481 IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing, VOL. 47, NO. 5, May 2000.

[6] James B. Y. Tsui and James P. Stephens Sr, Digital Microwave Receiver Technology, p. 699 – 705 IEEE Transactions on Microwave Theory and Techniques, VOL. 50, NO. 3, March 2002.

[7] Daniel R. Zahirniak, David L. Sharpin, and Timothy W. Fields, A Hardware-Efficient, Multi-rate, Digital Channelized Receiver Architecture, p. 137 – 152 IEEE Transactions on Aerospace and Electronic Systems, VOL. 34, NO. 1 January 1998.

[8] James Tsui, Microwave Receivers with Electronic Warfare Applications, Wiley, New York 1986.

[9], [10] Neil Weste and Kamran Eshraghian, Principles of CMOS VLSI Design, p. 546 & 556 2nd edition Addison-Wesley 1993.