EECC 730 VLSI Design Project Report
4x4 Signed/Unsigned Array Multiplier with BIST

Aniket Dilip Mhatre

and

Sajin George

Instructor – Dr. Amlan Ganguly

TA - Anirudh Easwar

Submission Date – February 28, 2012

Top Level Directory: /shared/eecc630/eecc630-axmsxg/project_workspace/final_core

Extra Credit – Universal 1 bit Multiplier Mode design(with 12 transistors)

& VHDL code to simulate universal 4x4 multipler with BIST

Rochester Institute of Technology

Department of Computer Engineering

Table of Contents

1. Abstract

The 4x4 universal multiplier with BIST was designed various blocks from different labs. The multiplier used mirrors full adders at different levels. A mode select line was introduced to set the sign bit for the operands. Build In Self Test circuit used was used to used to test the circuit. The BIST operates in the LFSR mode to generate a pseudo random sequence that is given as the inputs to the multiplier and the output of the multiplier is tested by the BIST. The multiplier can be even loaded the operands in parallel or serial shift in more.

2. Design Methodology and Theory

2.1 Mirror Adder

The design of the mirror adder is more compact and only use 24 transistors, this has been accomplished based on the observation that **Sum** can be factorized to reuse **Cout** as follows:

$$S= ABC+(A+B+C)Cout$$

$$Cout=(A.B)+(A \text{ xor } B)Cin$$

The design is completely symmetrical, ie. The pMOS and nMOS circuits are exactly identical. The Critical Path goes from C to Cout.The unit nMOS in the circuit has a resistance of R and capacitance C and unit pMOS has a resistance 2R and capacitance C. the Capacitance is proportional to the width and the resistance is inversely proportional to the width. The Width of the transistors are selected to have equal rise and fall-time. The variation of the resistance is due to change in mobility of Holes and Electrons.

2.2 Minimizing delay Transistor Sizing:

The following techniques were used to reduce the delay of the circuit.

· Feed the carry-in signal(C) to the inner inputs so that the internal capacitance is already discharged.(already done in the circuit diagram given)

· Make all transistors in the sum logic whose gate signals are connected to the carry-in and carry logic minimum size (1 unit e.g 4λ).This minimizes the branching effort on the critical path.

· The mirror adder has greater delay to compute **Sum** than **Cout**

2.3 D-FF

A D flip-flop is capable of storing the state of the input. It is a one bit memory element. The truth table of a D-FF is shown in Fig1. The circuit given in the lab is used to implement the D-FF for this lab. This circuit uses less number of transistors as compared to the Master Slave D-FF CMOS design. This circuit is suitable for high seed clocking as this circuit gets rid of the race condition. Figure 2 shows the circuit diagram of the D-FF. The D-FF is properly sized to obtain equal drive strength. The design is loaded with capacitor of 100fF. Based on the schematic the

layout is designed and is optimized for minimum area. Post-layout and pre-layout delays are recorded after simulation.

2.3 MUX

The mux is used for taking the output from the XNOR gate and the output is given to the inverter of the OR gate. The mux is basically used to select inputs-either for LFSR mode or for user inputs through scan_in

2.4 Multiplier Core

Multiplication is an operation in which you scale one number by the other. It is an operation which involves series of additions. Bit wise multiplication for binary numbers is carried out in a similar fashion, however repeated addition introduces delay. To reduce the propagation delay the multiplication operation is broken into smaller operations to produce partial products. Further addition of the partial products yields the final product. A simplified equation for the multiplication of unsigned numbers is shown below.

$$P = \left(\sum_{j=0}^{M-1} y_j 2^j \right) \left(\sum_{i=0}^{N-1} x_i 2^i \right) = \sum_{i=0}^{N-1} \sum_{j=0}^{M-1} x_i y_j 2^{i+j}$$

[Lab Reference Material]

Figure 1 gives an example of 5 x 5 bit multiplication achieved using partial product.

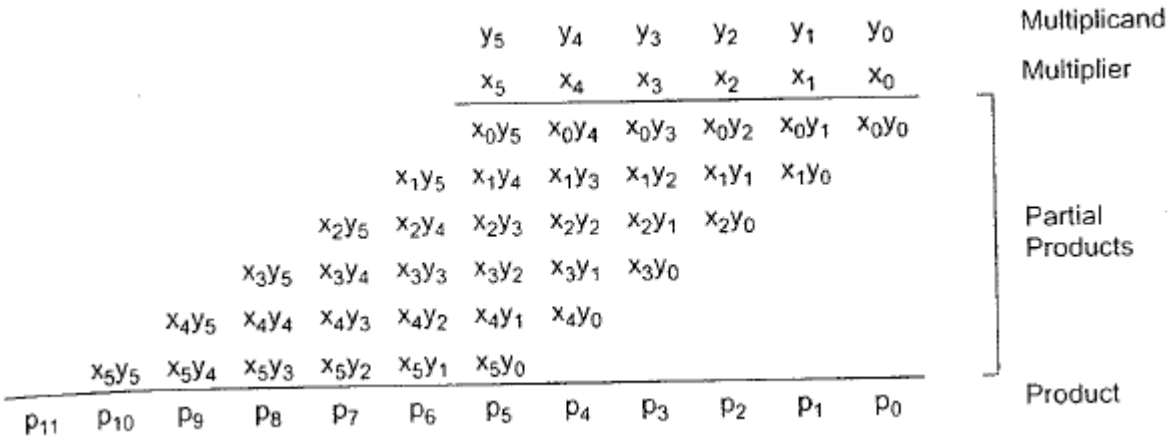


Figure 1. 5 x 5 bit multiplication using partial products. [Lab Reference Material]

In order to achieve multiplication of signed as well as unsigned number two's complement multiplication is performed. Two's compliment multiplication takes into consideration the negative weights of the numbers. The equation for a two's complement N x M is given as:

$$P = \left(-y_{M-1}2^{M-1} + \sum_{j=0}^{M-2} y_j 2^j \right)\left(-x_{n-1}2^{N-1} + \sum_{i=0}^{N-2} x_i 2^i \right)$$

$$= \sum_{i=0}^{N-2}\sum_{j=0}^{M-2} x_i y_j 2^{i+j} + x_{N-1}y_{M-1}2^{M+N-2} - \left(\sum_{i=0}^{N-2} x_i y_{M-1} 2^{i+M-1} + \sum_{j=0}^{M-2} x_{N-1} y_j 2^{j+N-1} \right)$$

[Lab Reference Material]

The multiplier speed depends on the number of partial products to be added. The modified Baugh-Wooley multiplier algorithm is used to reduce the number of partial products. The simplified technique to achieve two's complement 5 x 5 bit multiplication is shown below.

						y_5	y_4	y_3	y_2	y_1	y_0
						x_5	x_4	x_3	x_2	x_1	x_0
					1	$\overline{x_5y_0}$	x_0y_4	x_0y_3	x_0y_2	x_0y_1	x_0y_0
					$\overline{x_5y_1}$	x_1y_4	x_1y_3	x_1y_2	x_1y_1	x_1y_0	
				$\overline{x_5y_2}$	x_2y_4	x_2y_3	x_2y_2	x_2y_1	x_2y_0		
			$\overline{x_5y_3}$	x_3y_4	x_3y_3	x_3y_2	x_3y_1	x_3y_0			
		$\overline{x_5y_4}$	x_4y_4	x_4y_3	x_4y_2	x_4y_1	x_4y_0				
1	x_5y_5	$\overline{x_4y_5}$	$\overline{x_3y_5}$	$\overline{x_2y_5}$	$\overline{x_1y_5}$	$\overline{x_0y_5}$					
p_{11}	p_{10}	p_9	p_8	p_7	p_6	p_5	p_4	p_3	p_2	p_1	

[Lab Reference Material]

In order to achieve multiplication a mechanism is required to invert the input to certain partial products. This is done using the mode circuit. The block diagram for the modified Baugh-Wooley multiplier is figure 2. The shaded blocks represent the mode controlled blocks.

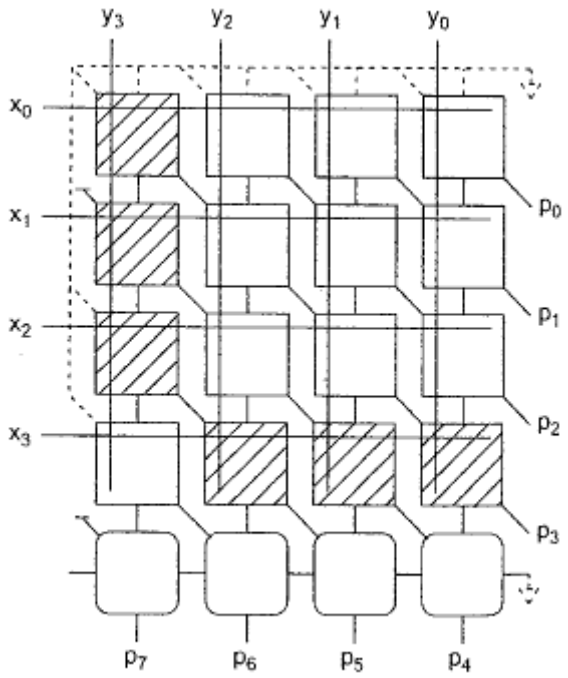


Figure 2. Modified Baugh-Wooley multiplier block diagram. [Lab Reference Material]

The multiplier design implemented in the project uses components implemented in the previous labs. The multiplier design was implemented in blocks. The block diagram of the multiplier core is shown below.

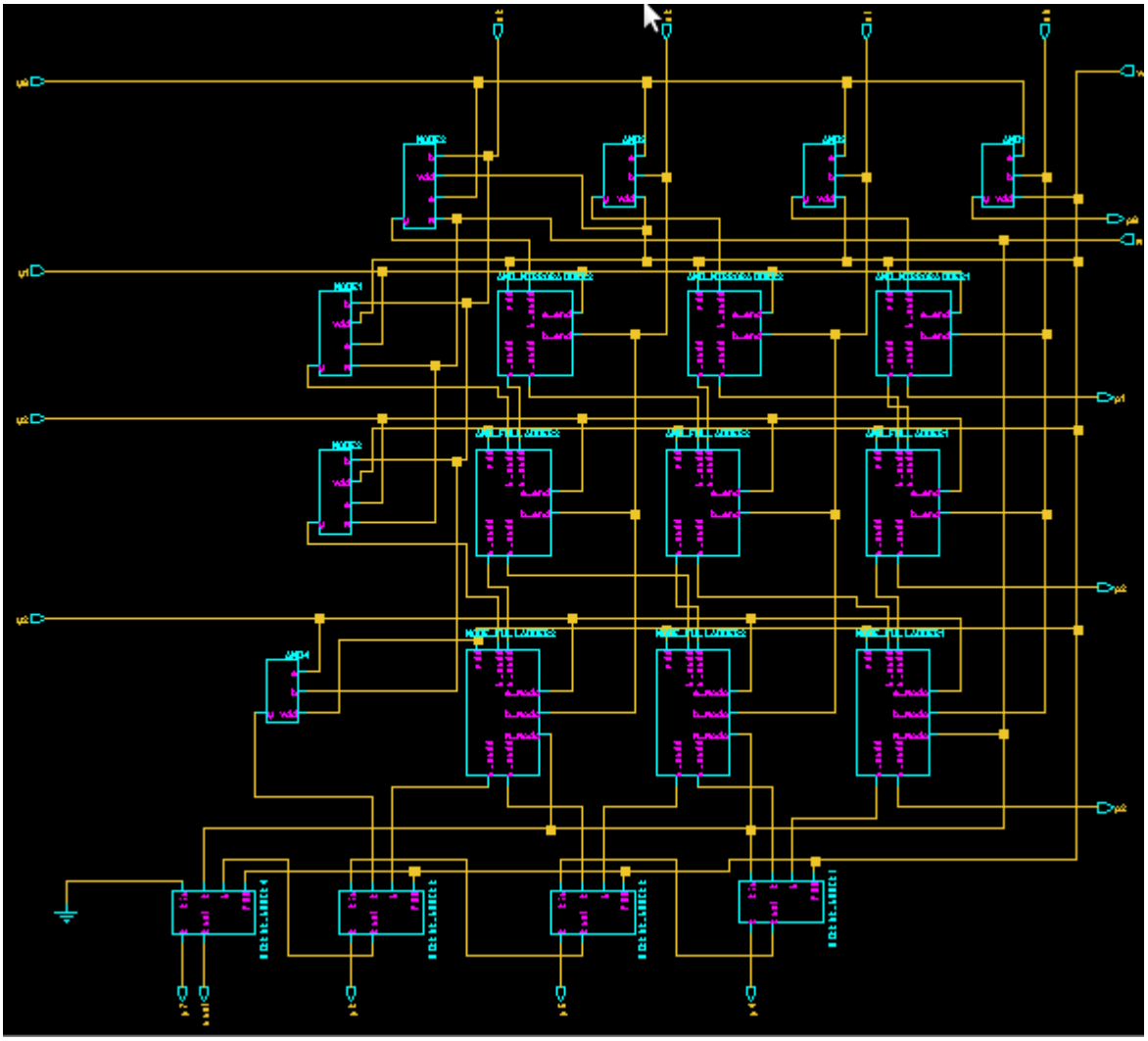


Figure 3. Block diagram of Multiplier

The block used were and_mirroradder (half adder and the AND gate) , and_fulladder (full adder and AND gate), mode, mirror adder, mode_fulladder (mode and fulladder) and AND gates. Full adder and half adder were implemented using full mirror adder. In order to implement half adder the cin of the full adder is connected to ground.

MODE DESIGN (1 bit Universal Multiplier) :

The mode block was designed for extra credits and it was implemented using a different approach. The circuit of the mode block is shown in figure 4.

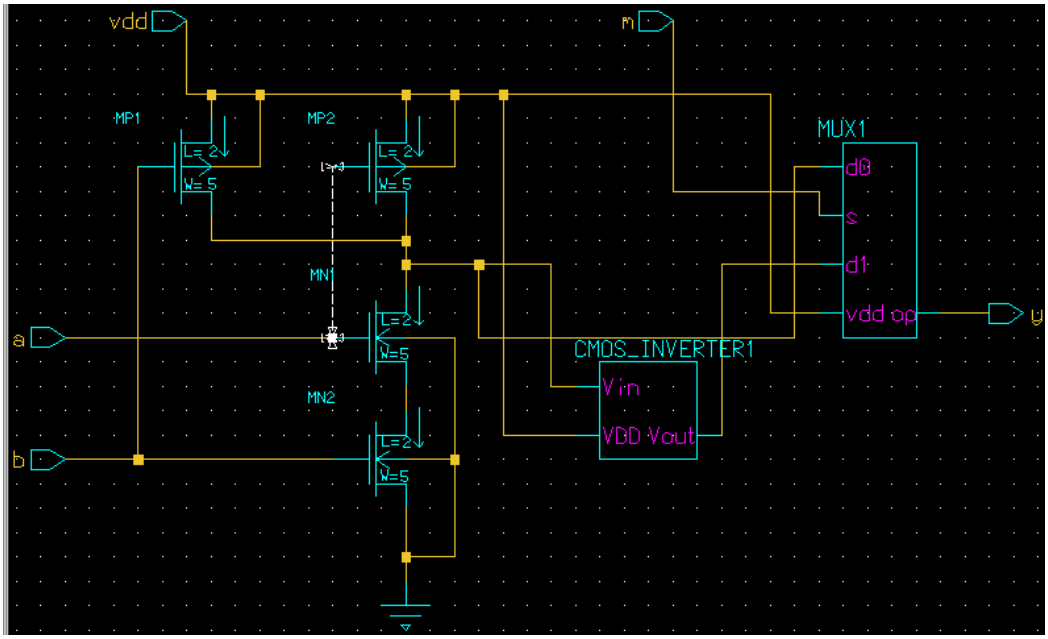


Figure. 4 Circuit Diagram of Mode Block

The logic behind the implementation is given as follows : A and B are given as inputs to the NAND gate to produce the output as /(A.B). The output of the NAND is given to an inverter to obtain (A.B). Now a MUX is used to select the inputs i.e. (A.B) and /(A.B) based on the select which is mode.

The challenge in the design was to implement the block in not more than 12 transistors. NAND uses 4 transistors, the inverter uses 2 inverter and the transmission gate MUX uses 6 transistors. Thus the block was implemented using 12 transistors. The mode block functionality is shown in the table below.

MODE	Output
0	A.B
1	/(A.B)

Table. MODE functionality

2.5 BILBO

Build in Logic Block Observer, is used to test the functionality of the multiplier design core. When chips are manufacture the chips have to be tested automatically for checking for errors. Hence there is a need for a testing circuit on all chips. The BILBO is a multipurpose chip which can be used to test the chip, generate random sequence, and load bits parally and test the signature.After loading the the bilbo with the required test vector we observe for the Golden Signature to test if the circuit function is proper.The BILBO block can work in different modes the LFSR mode, Shift Register Mode, Parallel shift in/ shift out mode and Serial Shift in/ Shift out mode. The diagram below show the basic schematic of a four bit bilbo [source: my courses lab handout]

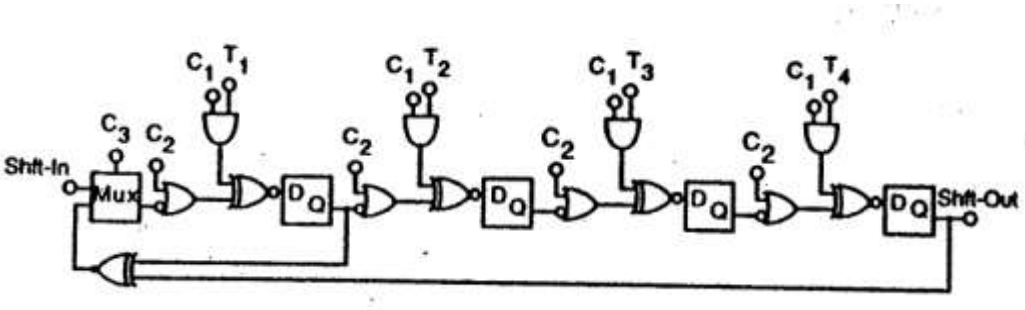


fig 2.2.1

the various modes of operation of the bilbo is controlled my the control line C1, C2, C3. table 2.2.1 show the various modes of operation.

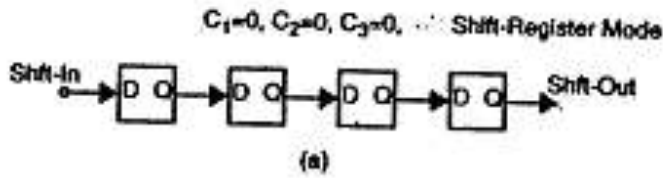
C1	C2	C3	Mode
0	0	0	Shift - register
0	0	1	LFSR MODE
0	1	X	Load “0”
1	0	1	Signature - Analysis
1	1	X	Parallel Mode
1	0	0	Not used

table 2.2.1

Modes of Operation[source lab-hand]

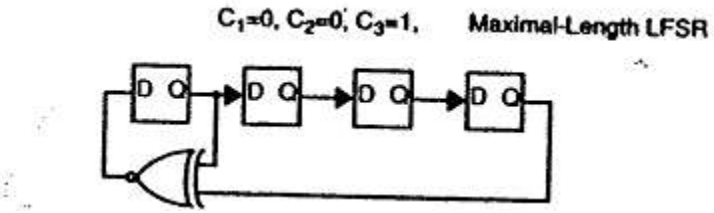
Shift- Register Mode:

In the shift registered mode is used to load the values into the D filpflop serially, this process of loading the bits will require 8 clock cycles. The data to be fed in is given at the SHIFTIN pin.



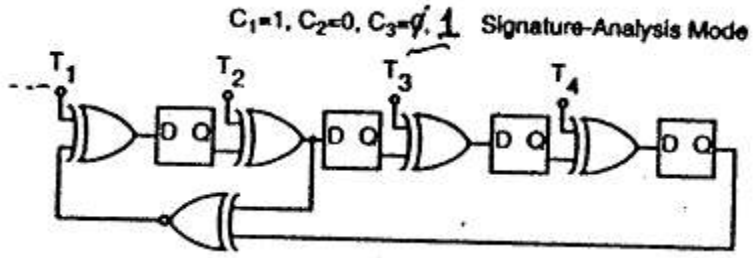
LFSR mode:

LFSR (Liner Feedback Shift Register) is used to create pseudo random sequence that will create a pseudo random input sequence to the testing core. The pseudo output of the LFSR can be seen in the (appendix) The initial value of the inside the LFSR is called as the seed value.



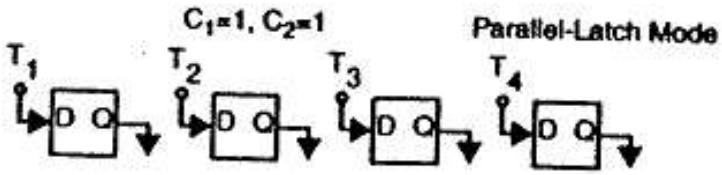
Signature Analysis

The signature analysis is used to test the output from the the testing core. The output of the multiplier is fed into the BILBO us int the T1-T8 , parallel input lines of the bilbo.



Parallel Mode

The Parallel mode is used to load values into the D flipflop in parallel.



3. Results/Data Analysis

3.1 Results

Multiplier:

	Vdd(V)	Temp(C)	Load	Rise Time	Fall Time	Tp,HL	Tp,LH
Worst Case(Pre-layout)	1.62	125	100	105.32 ps	102.99ps	2.516Ns	2.138Ns
Worst Case (post Layout)	1.62	125	100	1.58 ns	880.9 ps	2.6629	2.0107

Multiplier with BIST:

	Vdd(V)	Temp(C)	Load	Rise Time	Fall Time	Tp,HL	Tp,LH
Worst Case(Pre-layout)	1.62	125	100	971.87	223.91	39.979 ns	39.924
Worst Case (post Layout)	1.62	125	100	5.25 ns	1.14 ns	39.977	39.841

Functionality Check:

The functionality of the circuit was tested using the following test vectors and output verified.

Test Vector	A	B	Product (Unsigned)	Product (Signed)
1	0000	0000	00000000	00000000
2	1111	1111	11100001	00000001
3	0101	0101	00011001	00011001
4	1010	1010	01100100	00100100
5	0000	1111	00000000	00000000
6	1111	0000	00000000	00000000
7	0000	0000	00000000	00000000

3.2 Area of Layout

Multiplier core area: 1470 X 535 ⍀

BIST area: 1465 X 150 ⍀

Entire area (Core + BIST) : 1470 X 860 ⍀

Fraction of pad area used: 8.54 %

3.3 Number of NMOS and PMOS Transistors

Multiplier Core : 456 transistors

BILBO : 174 transistors

Entire core (BILBO+ Multiolier Core) : 630 transistors

3.4 Analysis of Result

From the result it is clearly visible that there is a one clock cycle delay in the output of the signature analyser because the load shift register shift for every clock, and since all the D flip flops are at the initial state of "0" it takes one more cycle to for outputs to change at the output.

The signature was verified by manually calculating the golden signature for the the test vectors, A VHDL program (appendix 2.3) was written to simulate the whole system and the output of the the VHDL code matched the output of the 4x4 multiplier with BIST.

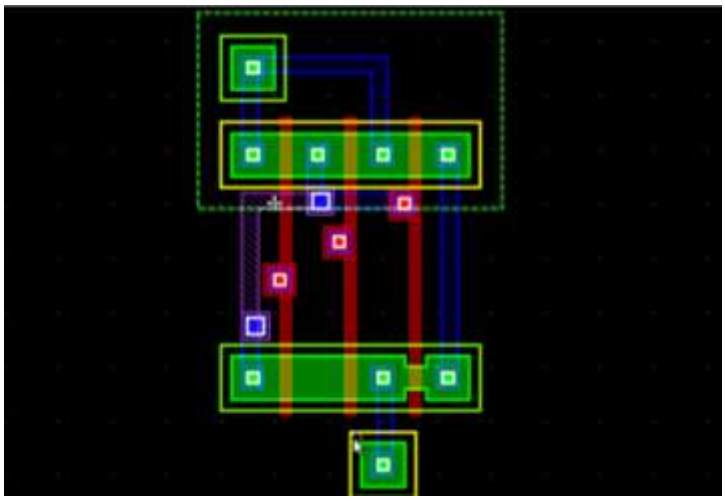
The output of the post layout Bilbo was different due to the additional parasitic capacitance. This further increases the delay caused by the D Flip Flop.

4. Conclusion

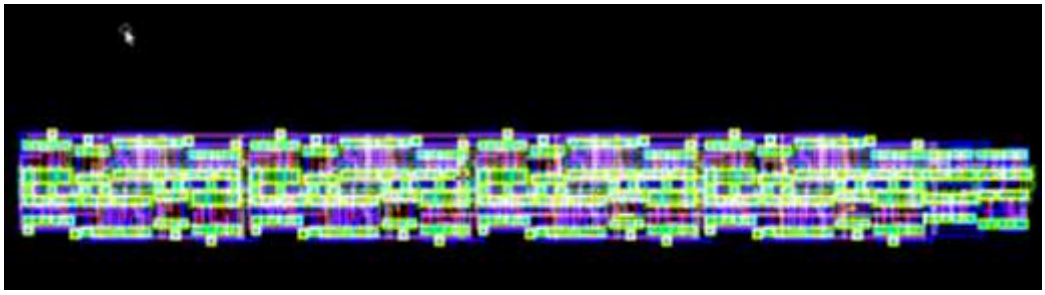
The 4x4 universal multiplier with , BIST was designed using a hierarchical design. The POST layout performance was lower compared to the PRE- LAYOUT due to the play of parasitic capacitance. there is increase of 10% in the delay, which shows the efficiency of the design. The golden signature for the test vector was verified and a the obtained output was verified againts a VHDL simulation of the whole circuit.

5. Appendix

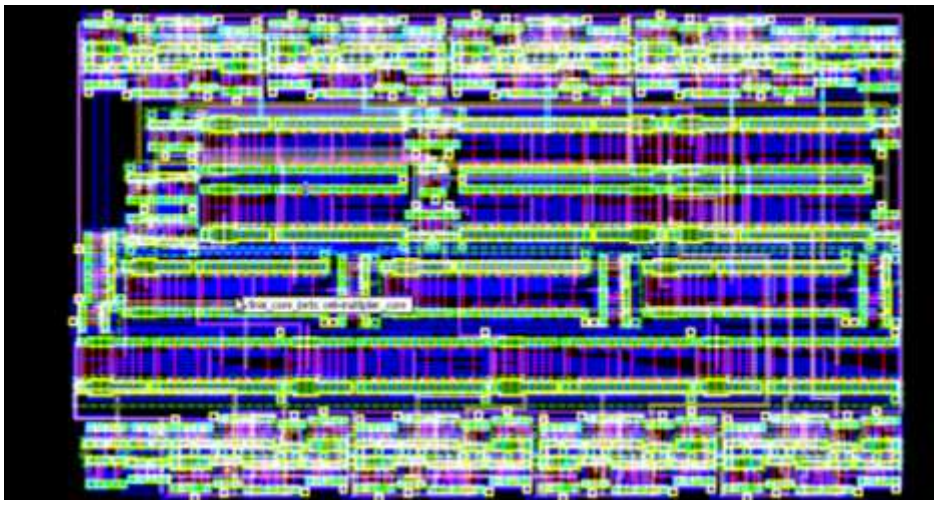
Layout:



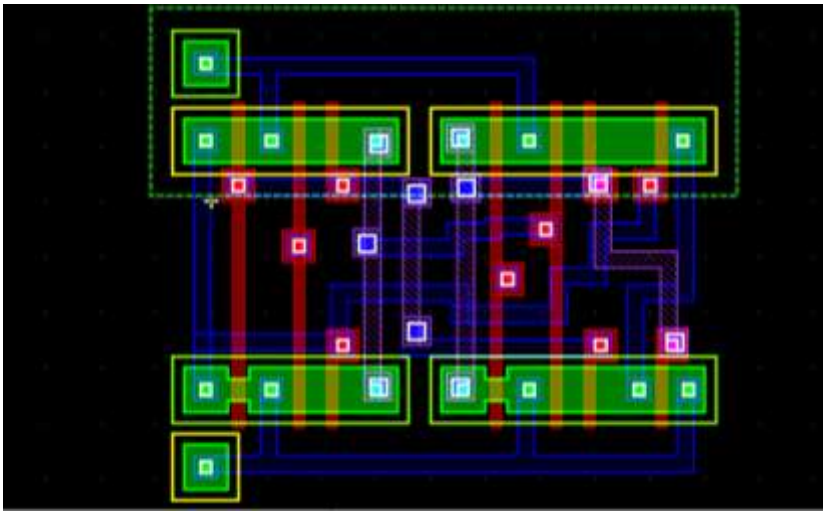
and



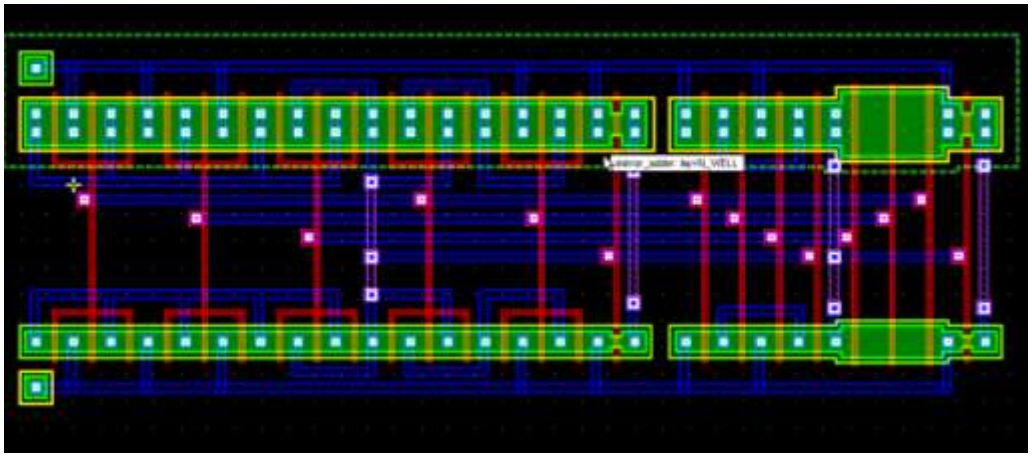
Bilbo



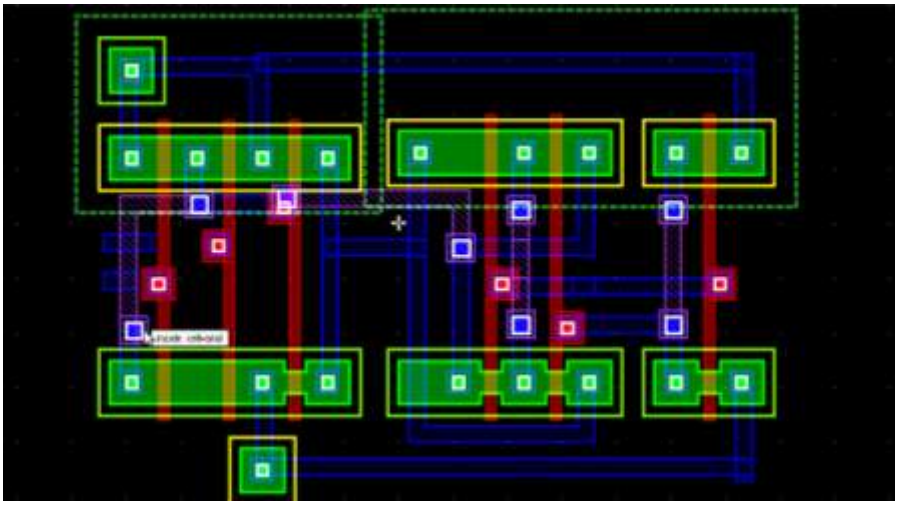
Core



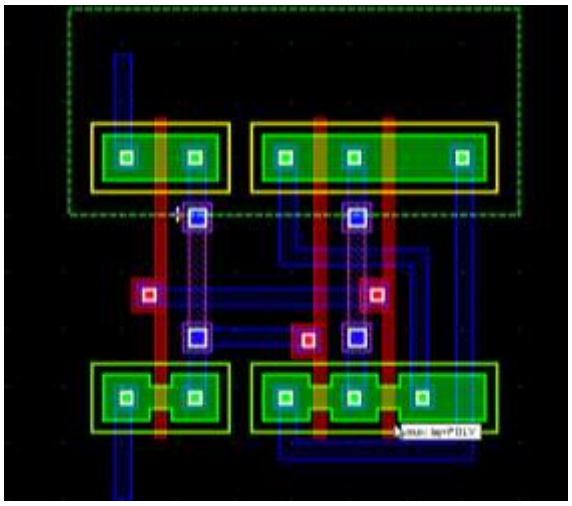
Dff



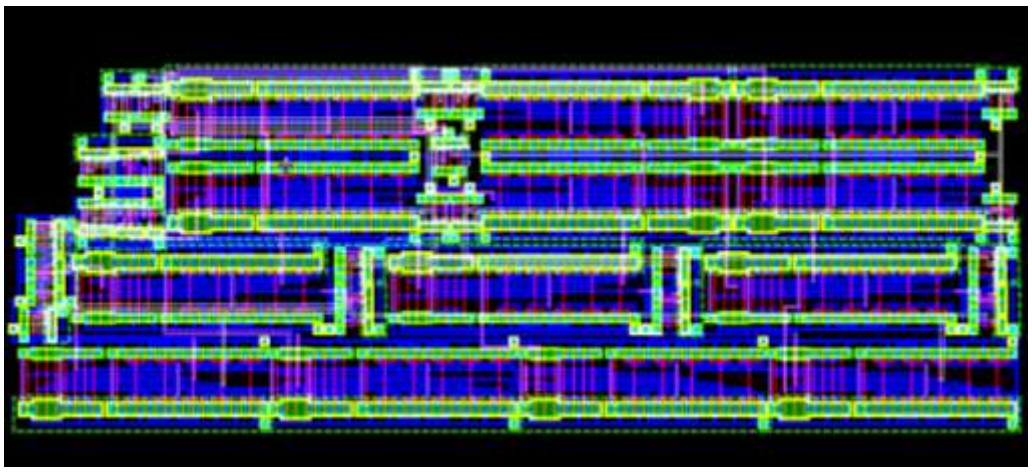
Mirror adder



MODE

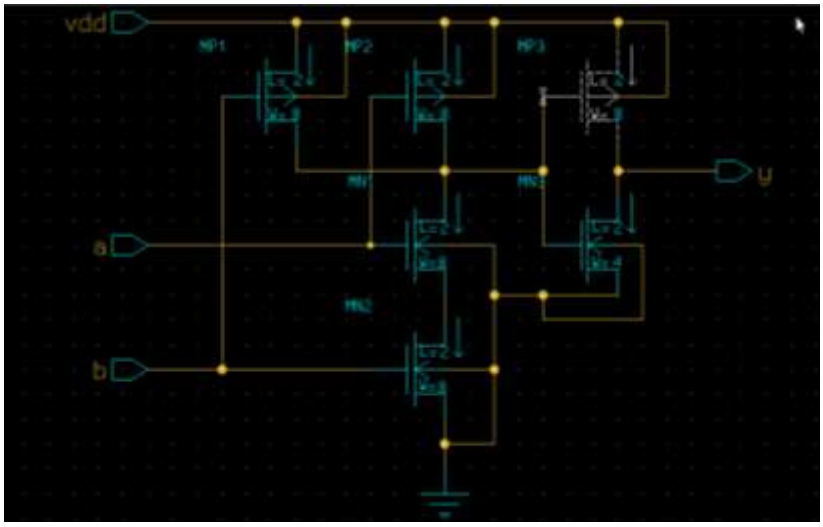


MUX

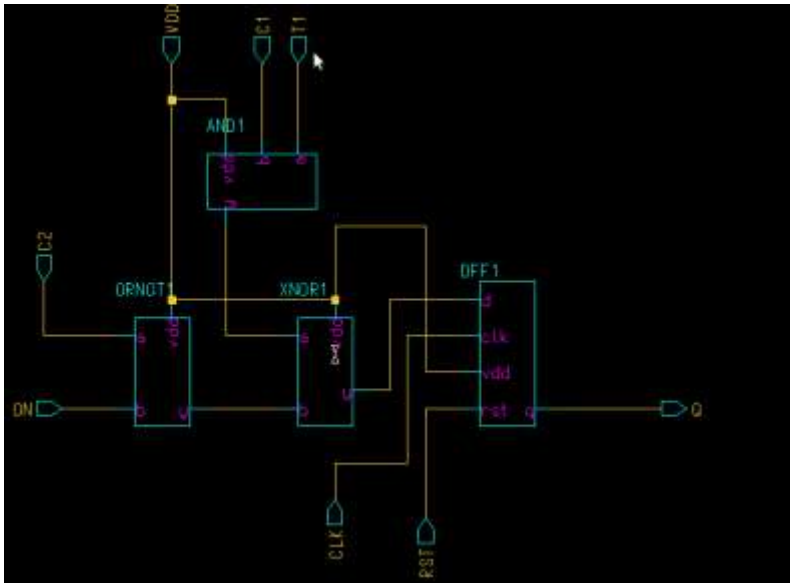


Multiplier Core layout

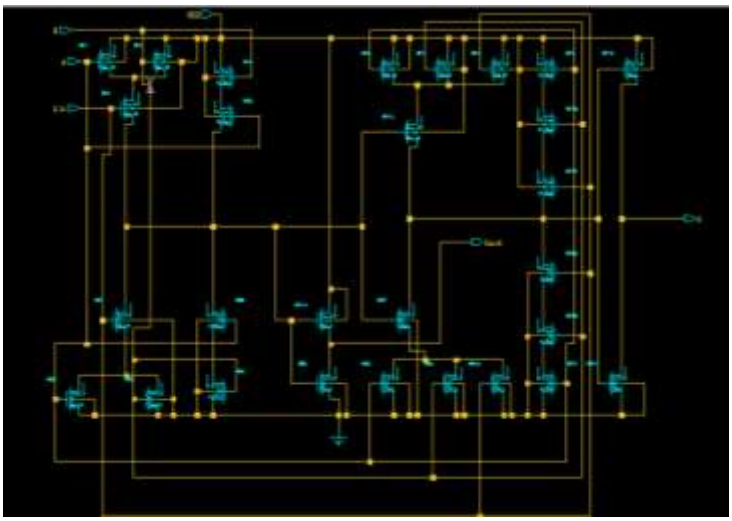
Schematic:



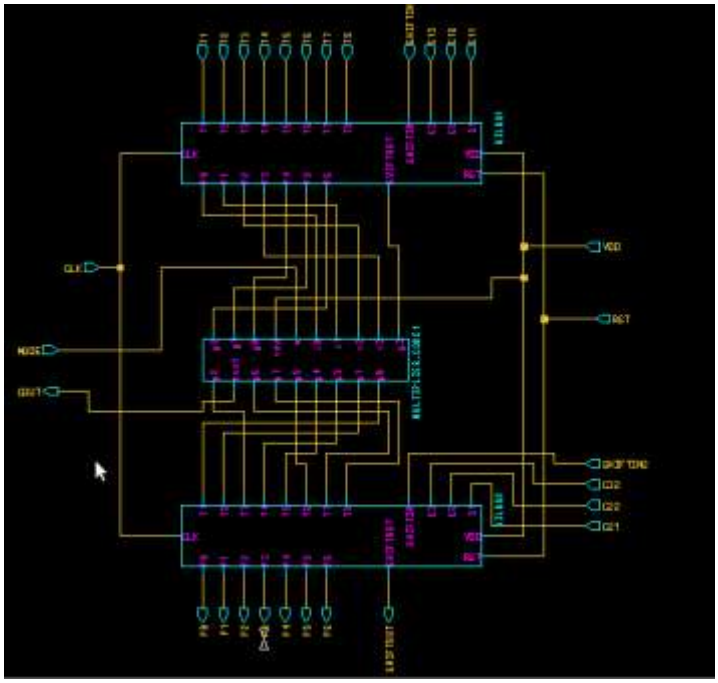
And



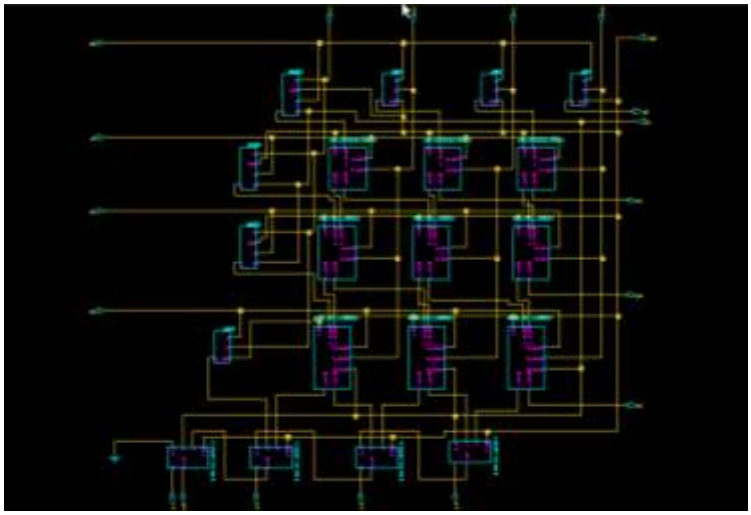
Bilbo



Mirror adder



Core

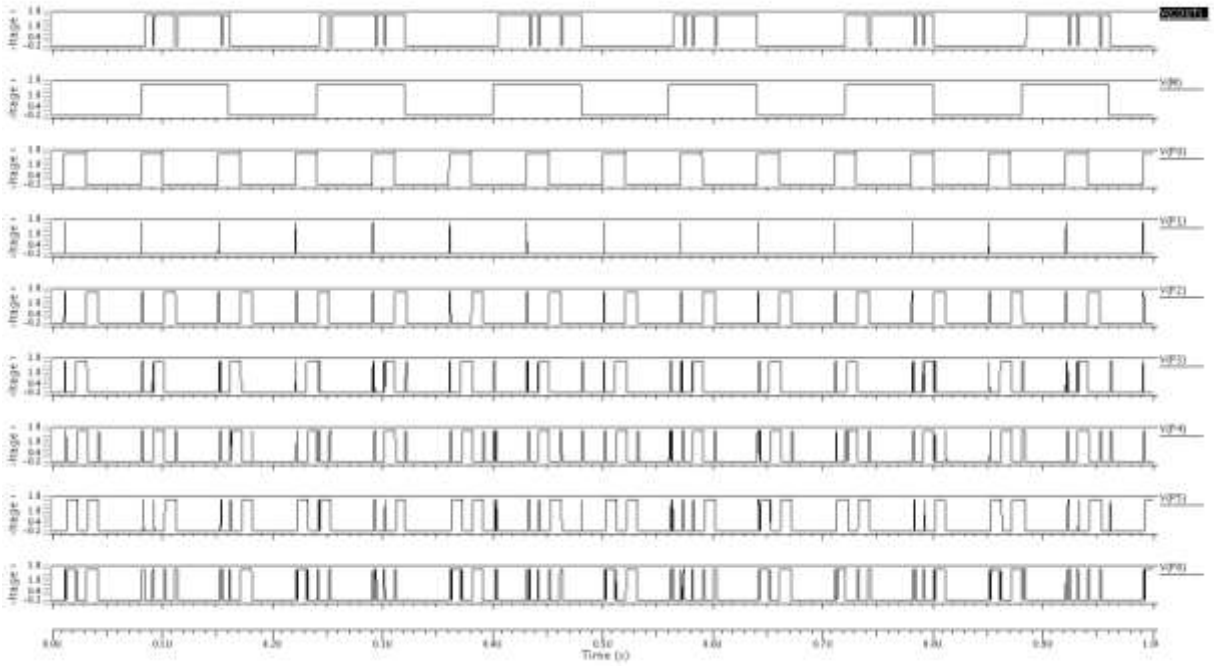


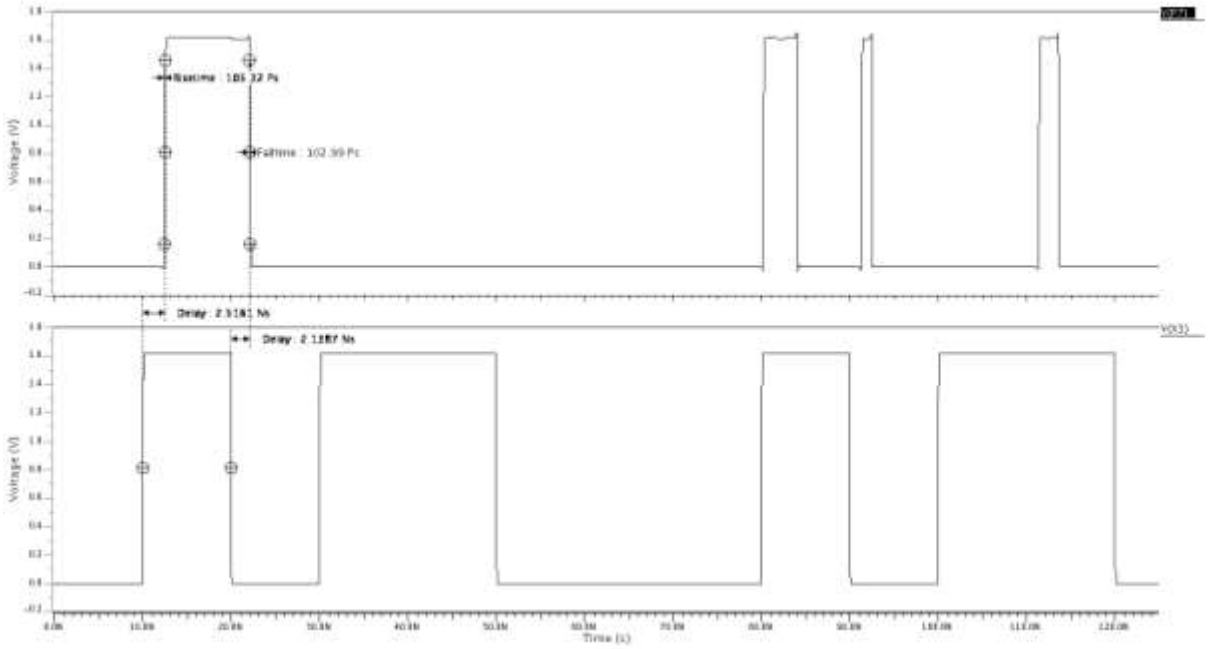
Multiplier Core

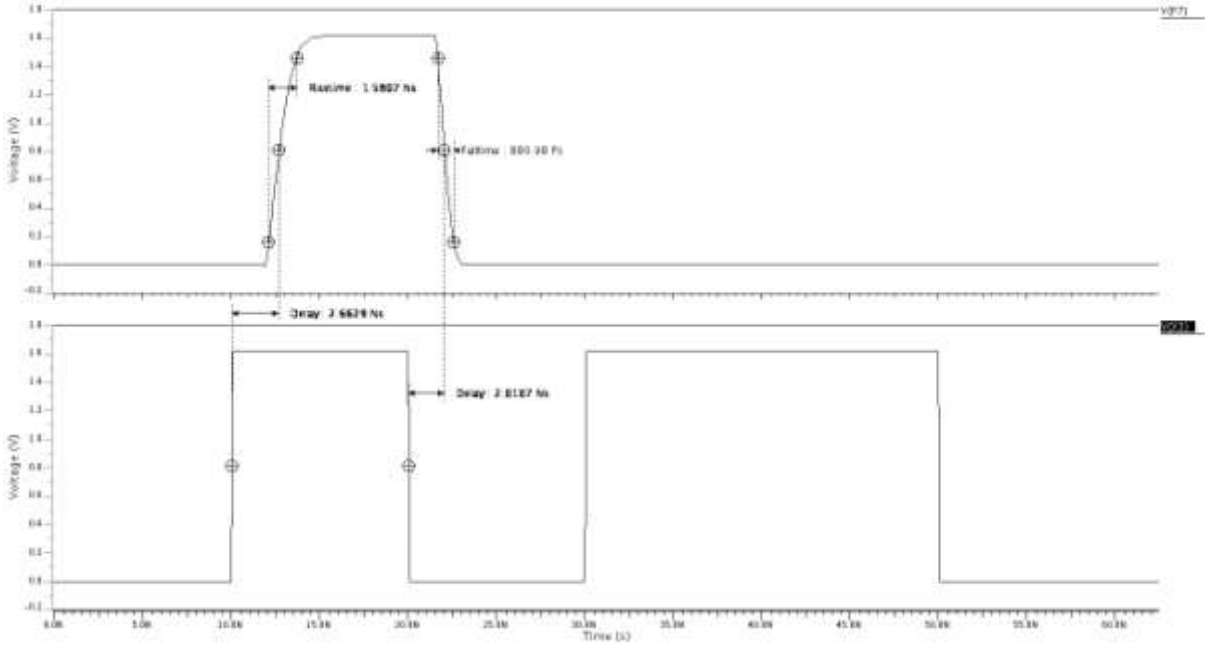
Waveforms:

Multiplier Core wave forms

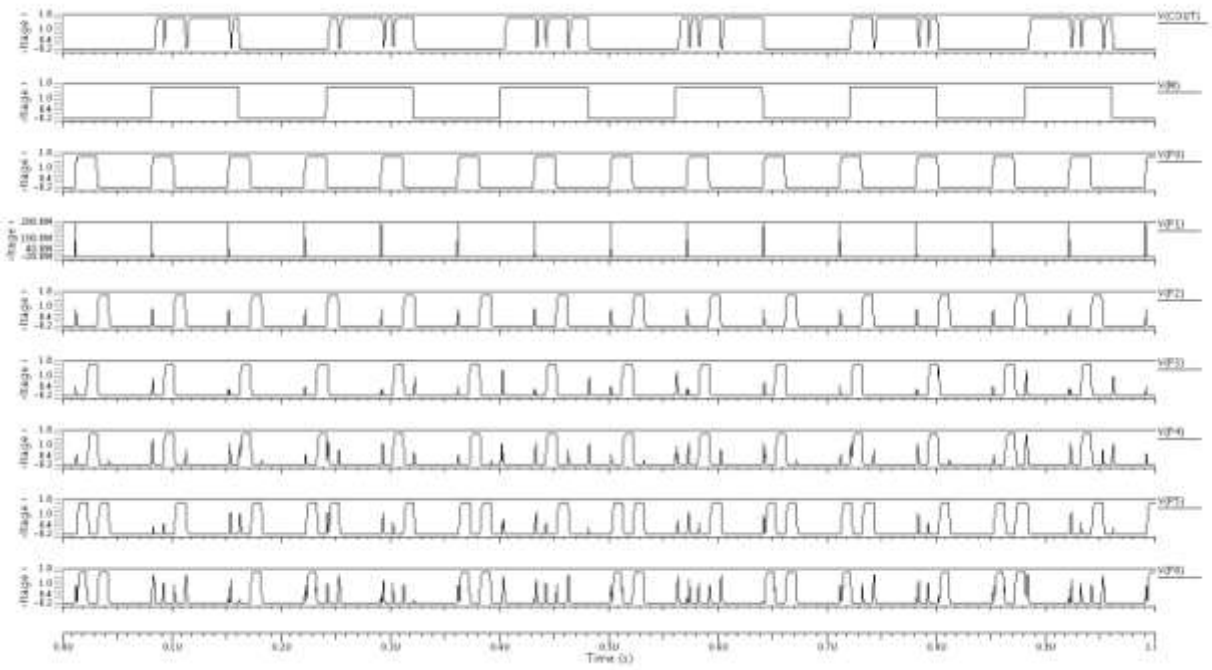
Post-layout



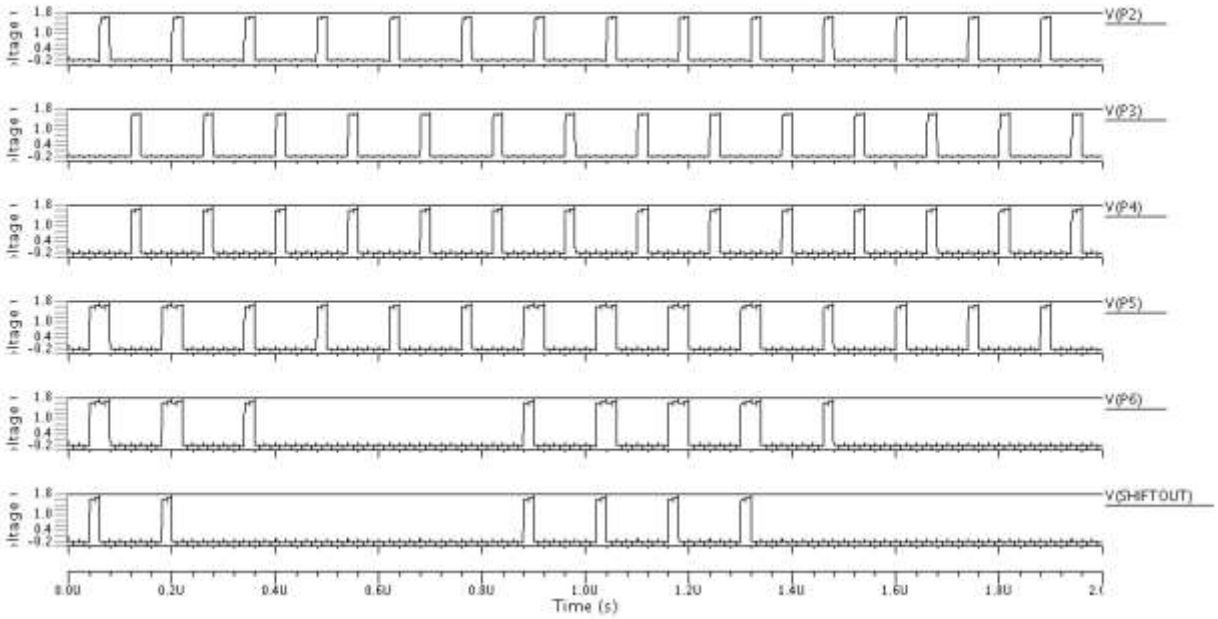

Voltage (V)
Time (s)

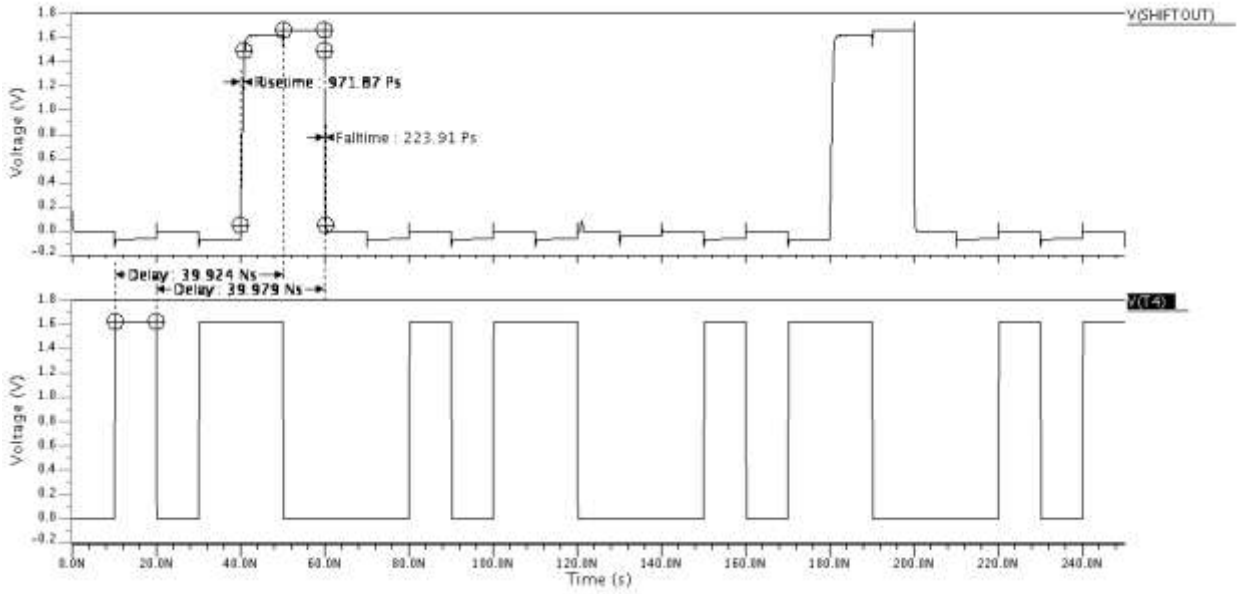

Voltage (V)
Time (s)

Pre-layout

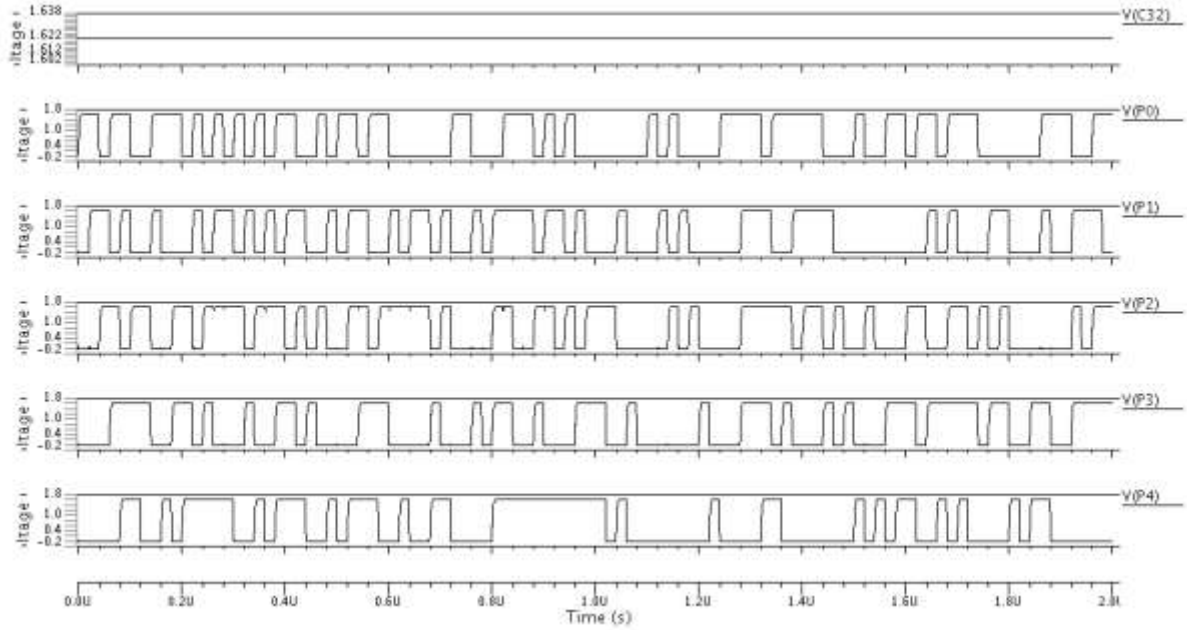


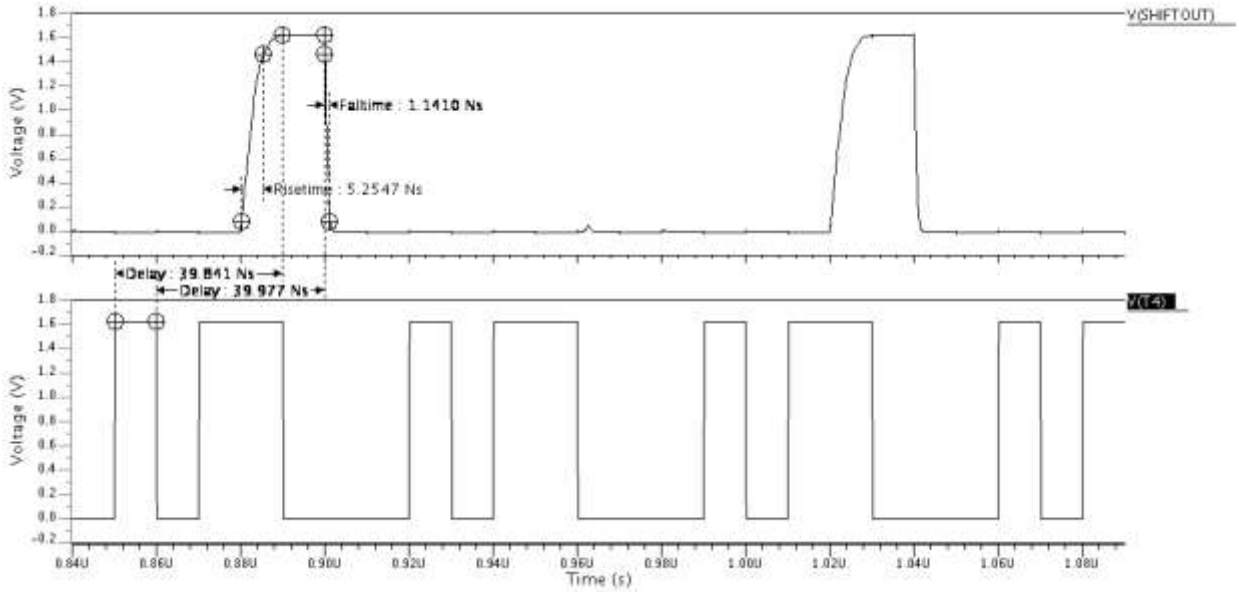
CORE:
Pre-layout


V(P2)
V(P3)
V(P4)
V(P5)
V(P6)
V(SHIFTOUT)
Time (s)


V(SHIFTOUT)
Riseime : 971.87 Ps
Falltime : 223.91 Ps
Delay : 39.924 Ns
Delay : 39.979 Ns
Voltage (V)
Time (s)

Post-layout


V(C32)
V(P0)
V(P1)
V(P2)
V(P3)
V(P4)
Time (s)


V(SHIFTOUT)
Falltime : 1.1410 Ns
Risetime : 5.2547 Ns
Delay : 39.841 Ns
Delay : 39.977 Ns
V(C4)
Voltage (V)
Time (s)

LVS:

```
##################################################
##                                              ##
##     C A L I B R E   S Y S T E M      ##
##                                              ##
##       L V S   R E P O R T        ##
##                                              ##
##################################################
```

REPORT FILE NAME: /shared/eecc630/eecc630-axmsxg/project_workspace/final_core_beta/lvs/lvs.rep

LAYOUT NAME: /shared/eecc630/eecc630-axmsxg/project_workspace/final_core_beta/layout/final_core_beta

SOURCE NAME: /shared/eecc630/eecc630-axmsxg/project_workspace/final_core_beta/lvs

RULE FILE: /tools/mentor/adk/technology/ic/process/tsmc018.rules

LVS MODE: Mask

RULE FILE NAME: /tools/mentor/adk/technology/ic/process/tsmc018.rules

CREATION TIME: Tue Feb 28 13:46:31 2012

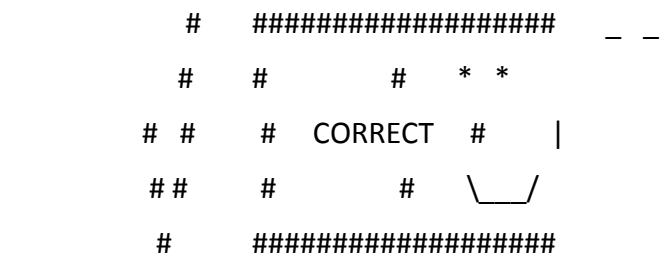
CURRENT DIRECTORY: /shared/eecc630/eecc630-axmsxg/project_workspace

USER NAME: axm9958

****

OVERALL COMPARISON RESULTS

****
****



INITIAL NUMBERS OF OBJECTS

	Layout	Source	Component Type
Ports:	30	30	
Nets:	594	589 *	
Instances:	696	566 *	mn (4 pins)
	696	566 *	mp (4 pins)
Total Inst:	1392	1132	

NUMBERS OF OBJECTS AFTER TRANSFORMATION

	Layout	Source	Component Type
Ports:	30	30	
Nets:	589	589	
Instances:	566	566	mn (4 pins)
	566	566	mp (4 pins)
Total Inst:	1132	1132	

* = Number of objects in layout different from number in source.

****

LVS PARAMETERS

****

o LVS Setup:

LVS COMPONENT TYPE PROPERTY phy_comp element comp
LVS COMPONENT SUBTYPE PROPERTY model
LVS PIN NAME PROPERTY phy_pin
LVS POWER NAME "VDD"
LVS GROUND NAME "GND"

LVS CELL SUPPLY NO
LVS RECOGNIZE GATES NONE
LVS IGNORE PORTS NO
LVS CHECK PORT NAMES NO
LVS IGNORE TRIVIAL NAMED PORTS NO
LVS BUILTIN DEVICE PIN SWAP YES
LVS ALL CAPACITOR PINS SWAPPABLE NO
LVS DISCARD PINS BY DEVICE NO
LVS SOFT SUBSTRATE PINS NO
LVS INJECT LOGIC NO
LVS EXPAND UNBALANCED CELLS YES
LVS EXPAND SEED PROMOTIONS NO
LVS PRESERVE PARAMETERIZED CELLS NO
LVS GLOBALS ARE PORTS YES
LVS REVERSE WL NO
LVS SPICE PREFER PINS NO
LVS SPICE SLASH IS SPACE YES
LVS SPICE ALLOW FLOATING PINS YES
LVS SPICE ALLOW INLINE PARAMETERS UNSPECIFIED
LVS SPICE ALLOW UNQUOTED STRINGS NO
LVS SPICE CONDITIONAL LDD NO
LVS SPICE CULL PRIMITIVE SUBCIRCUITS NO
LVS SPICE IMPLIED MOS AREA NO
// LVS SPICE MULTIPLIER NAME
LVS SPICE OVERRIDE GLOBALS NO
LVS SPICE REDEFINE PARAM NO
LVS SPICE REPLICATE DEVICES NO
LVS SPICE STRICT WL NO
// LVS SPICE OPTION
LVS STRICT SUBTYPES NO
LAYOUT CASE NO

```
SOURCE CASE                     NO
LVS COMPARE CASE                NO
LVS DOWNCASE DEVICE             NO
LVS REPORT MAXIMUM              50
LVS PROPERTY RESOLUTION MAXIMUM 32
// LVS SIGNATURE MAXIMUM
// LVS FILTER UNUSED OPTION
// LVS REPORT OPTION
LVS REPORT UNITS                YES
// LVS NON USER NAME PORT
// LVS NON USER NAME NET
// LVS NON USER NAME INSTANCE

// Reduction

LVS REDUCE SERIES MOS           NO
LVS REDUCE PARALLEL MOS         YES
LVS REDUCE SEMI SERIES MOS      NO
LVS REDUCE SPLIT GATES          YES
LVS REDUCE PARALLEL BIPOLAR     YES
LVS REDUCE SERIES CAPACITORS    YES
LVS REDUCE PARALLEL CAPACITORS  YES
LVS REDUCE SERIES RESISTORS     YES
LVS REDUCE PARALLEL RESISTORS   YES
LVS REDUCE PARALLEL DIODES      YES
LVS REDUCTION PRIORITY          PARALLEL

// Filter

LVS FILTER  sch_filter_direct_open  OPEN SOURCE DIRECT
LVS FILTER  sch_filter_direct_short  SHORT SOURCE DIRECT
```

```
LVS FILTER  sch_filter_mask_open  OPEN SOURCE MASK
LVS FILTER  sch_filter_mask_short  SHORT SOURCE MASK
LVS FILTER  lay_filter_direct_open  OPEN LAYOUT DIRECT
LVS FILTER  lay_filter_direct_short  SHORT LAYOUT DIRECT
LVS FILTER  v  OPEN
LVS FILTER  i  OPEN
LVS FILTER  e  OPEN
LVS FILTER  f  OPEN
LVS FILTER  g  OPEN

// Trace Property

// TRACE PROPERTY  mn  instpar(w) width w 0
// TRACE PROPERTY  mp  instpar(w) width w 0
// TRACE PROPERTY  me  instpar(w) width w 0
// TRACE PROPERTY  md  instpar(w) width w 0
// TRACE PROPERTY  mn  instpar(l) length l 0
// TRACE PROPERTY  mp  instpar(l) length l 0
// TRACE PROPERTY  me  instpar(l) length l 0
// TRACE PROPERTY  md  instpar(l) length l 0
// TRACE PROPERTY  r  instpar(r) resistance r 0
// TRACE PROPERTY  c  instpar(c) capacitance c 0
// TRACE PROPERTY  d  instpar(a) area a 0
// TRACE PROPERTY  d  instpar(p) perimeter p 0
```

****
****

INFORMATION AND WARNINGS

****
****

	Matched Layout	Matched Source	Unmatched Layout	Unmatched Source	Component Type
Ports:	30	30	0	0	
Nets:	589	589	0	0	
Instances:	566	566	0	0	mn(nmos4)
	566	566	0	0	mp(pmos4)
Total Inst:	1132	1132	0	0	

o Statistics:

5 isolated layout nets were deleted.

390 layout mos transistors were reduced to 130.
260 mos transistors were deleted by parallel reduction.

o Isolated Layout Nets:

(Layout nets which are not connected to any instances or ports).

36(-253.000,-443.000) 37(917.000,261.000) 41(-284.000,268.000) 46(481.000,-282.000) 47(946.000,-436.000)

o Initial Correspondence Points:

Ports: VDD GND MODE SHIFTIN T2 T3 T4 T5 T6 T7 T8 CLK SHIFTOUT P2 P1 P0 P4 P6 P5 P3 SHIFTIN2 T1 C12 C11 C13 C22 C21 COUT RST C32

****
****

SUMMARY

****
****

Total CPU Time: 0 sec

Total Elapsed Time: 0 sec