VLSI DESIGN 1

Project Title: 4x4 Array Multiplier

Project Done Under: PROF.DR.D.MISRA

Project By

SANKET SHANKAR KULKARNI
NJIT ID: 31406022 EMAIL: sk2339@njit.edu

JIGNESH FRANCIS LAWRANCE
NJIT ID: 31390272 EMAIL: jl763@njit.edu

Table of Contents

List of Figures

ABSTRACT

Abstract: Multiplier is one of the most important arithmetic unit in Microprocessors and DSPs and also a major source of power dissipation. Reducing the power dissipation of multipliers is a key to satisfy the overall power budget of various digital circuits and systems. The project elaborates the steps required to design array multiplier. The fundamental units to design a multiplier are adders. The variants of adders used in this project are Carry Save Adder(CSA) and Carry Propagate Adder(CPA). The main objective of our work is to calculate the average power, delay and PDP of 4x4 multipliers. The design of Half adder and Full adder for low power is obtained and the low power units are implemented on the array multiplier and the results are analyzed for better performance. The designs are done using Mentor Graphics tool and are simulated using H-SPICE.

Keywords: Array Multipliers, Delay, Adders, Hspice and Mentor Graphics

1. INTRODUCTION

1.1 Multiplier

Multipliers assume a vital part in today's advanced flag preparing and different applications. With advances in innovation, numerous analysts have attempted and are attempting to outline multipliers which offer both of the accompanying plan targets – fast, low power utilization, normality of design and consequently less region or even blend of them in one multiplier in this way making them appropriate for different rapid, low power and minimized VLSI usage.

The regular increase technique is "include and move" calculation. In parallel multipliers number of halfway items to be included is the fundamental parameter that decides the execution of the multiplier. To decrease the quantity of halfway items to be included, Modified Booth calculation is a standout amongst the most well-known calculations. To accomplish speed changes Wallace Tree calculation can be utilized to diminish the quantity of successive including stages. Promote by joining both Modified Booth calculation and Wallace Tree procedure we can see preferred standpoint of both calculations in one multiplier. However, with expanding parallelism, the measure of movements between the incomplete items and moderate aggregates to be included will build which may bring about decreased speed, increment in silicon region because of abnormality of structure furthermore expanded power utilization because of increment in interconnect coming about because of complex steering. Then again "serial-parallel" multipliers trade off speed to accomplish better execution for territory and power utilization. The determination of a parallel or serial multiplier really relies on upon the way of utilization.

1.2 Types of Multipliers

A) Serial Multiplier
Where area and power is of utmost importance and delay can be tolerated the serial multiplier is used. This circuit uses one adder to add the m * n partial products.

B) Serial/Parallel Multiplier
One operand is fed to the circuit in parallel while the other is serial. N partial products are formed each cycle. On successive cycles, each cycle does the addition of one column of the multiplication table of M*N PPs. The final results are stored in the output register after N+M cycles. While the area required is N-1 for M=N.

C) Shift/Add Multiplier
Depending on the value of multiplier LSB bit, a value of the multiplicand is added and accumulated. At each clock cycle the multiplier is shifted one bit to the right and its value is tested. If it is a 0, then only a shift operation is performed. If the value is a 1, then the multiplicand is added to the accumulator and is shifted by one bit to the right. After all the multiplier bits have been tested the product is in the accumulator. The accumulator is 2N (M+N) in size and initially the N, LSBs contains the Multiplier.

D) Array Multiplier
Array multiplier is well known due to its regular structure. Multiplier circuit is based on add and shift algorithm. Each partial product is generated by the multiplication of the multiplicand with one multiplier bit. The partial product is shifted according to their bit orders and then added. The addition can be performed with normal carry propagate adder.

2.Array Multiplier

2.1 Array Multiplier Concept

An array multiplier is very regular in structure as shown in figure 4. It uses short wires that go from one full adder to adjacent full adders horizontally, vertically or diagonally an n × n array of AND gates can compute all the *i i a b* terms simultaneously. The shifting of partial products for their proper alignment is performed by simple routing and does not require any logic. The number of rows in array multiplier denotes length of the multiplier and width of each row denotes width of multiplicand. The output of each row of adders acts as input to the next row of adders. Fast multipliers use carry-save adders to sum the partial products. A CSA typically has a delay of 1.5–2 FO4 inverters independent of the width of the partial product, while a carry-propagate adder (CPA) tends to have a delay of 4–15+ FO4 inverters depending on the width, architecture, and circuit family.

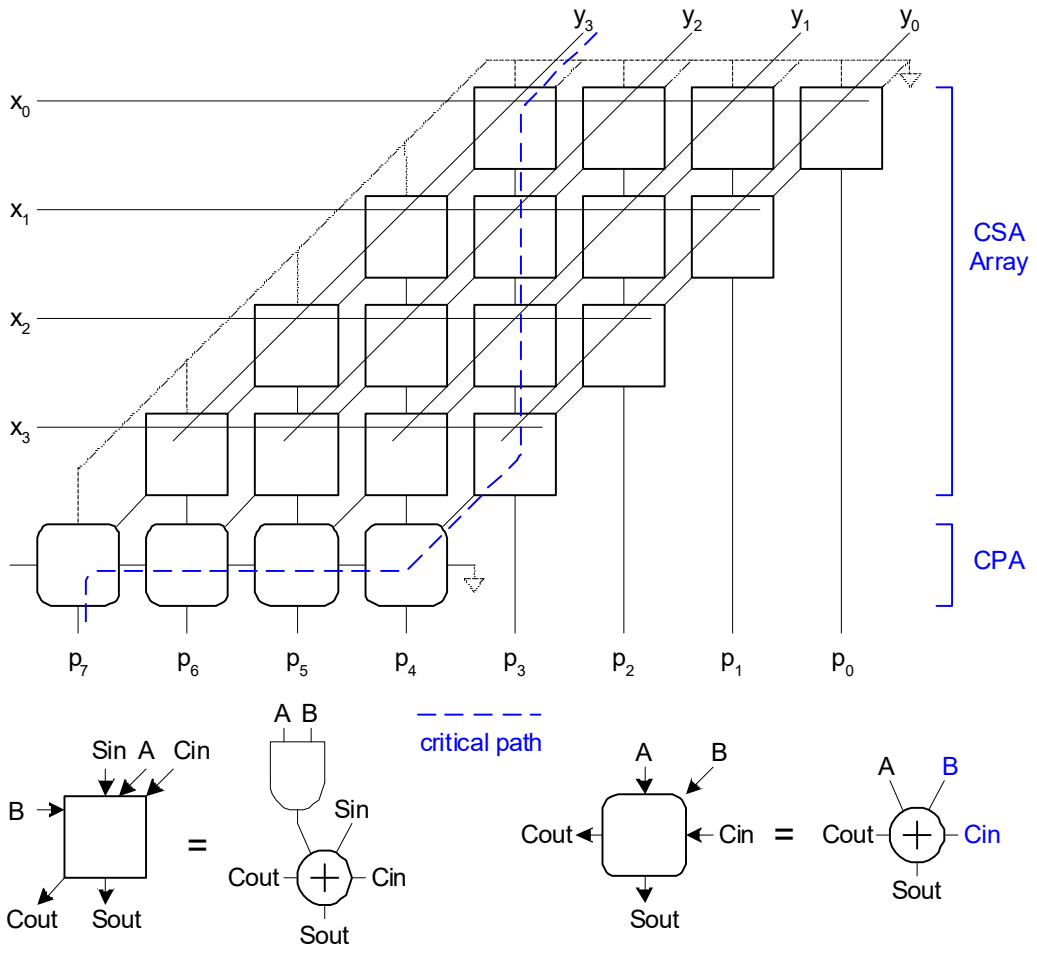


Fig 2.1: Array Multiplier

Each cell contains a 2-input AND gate that forms a partial product and a full adder (CSA) to add the partial product into the running sum. The first row converts the first partial product into carry-save redundant form. Each later row uses the CSA to add the corresponding partial product to the carry-save redundant result of the previous row and generate a carry-save redundant result. The least significant *N* output bits are available as sum outputs directly from CSAs. The most significant output bits arrive in carry-save redundant form and require an *M*-bit carry-propagate adder to convert into regular binary form. In Figure 2.1, the CPA is implemented as a carry-ripple adder. The array is regular in structure and uses a single type of cell, so it is easy to design and layout. Assuming the carry output
is faster than the sum output in a CSA, the critical path through the array is marked on the figure with a dashed line. The adder can easily be pipelined with the placement of registers between rows. In practice, circuits are assigned rectangular blocks in the floorplan so the parallelogram shape wastes space.

2.2 Principles of Array Multiplier

There are two sets of 4-bit inputs i.e. x0, x1, x2, x3 and y0, y1, y2, y3. All input bits are ANDed with each other which require 16 AND gates 16 Full Adder as CSA and 4 Full Adder as CPA are used in the design to collect all partial products. The various min terms obtained while multiplication as are as follows

			X_3	X_2	X_1	X_0		Multiplicand
			Y_3	Y_2	Y_1	Y_0		Multiplier
			X_3Y_0	X_2Y_0	X_1Y_0	X_0Y_0		Partial product 0
		X_3Y_1	X_2Y_1	X_1Y_1	X_0Y_1			Partial Product 1
		C_{12}	C_{11}	C_{10}				1st row carries
	C_{13}	S_{13}	S_{12}	S_{11}	S_{10}			1st row sums
	X_3Y_2	X_2Y_2	X_1Y_2	X_0Y_2				partial product 2
	C_{22}	C_{21}	C_{20}					2nd row carries
C_{23}	S_{23}	S_{22}	S_{21}	S_{20}				2nd row sums
X_3Y_3	X_2Y_3	X_1Y_3	X_0Y_3					partial product 3
C_{32}	C_{31}	C_{30}						3rd row carries
C_{33}	S_{33}	S_{32}	S_{31}	S_{30}				3rd row sums
P_7	P_6	P_5	P_4	P_3	P_2	P_1	P_0	Final Product

Fig 2.2 Array Multiplier Products.

3.DESIGN APPROACH

3.1 Creating Schematic

All the gates and Cmos transistor are arranged in such a way to get an array multiplier. The Transistor level circuit is made to form gate and gate level Schematic is used in creating half adder and full adder. Using Half adder and full adder carry save adder is created and finally array multiplier schematic is sketched

3.2 Generating Symbol

For a simplified approach of the schematic, symbols are generated of simple basic transistor level schematic. These symbols are further used as a instance in the bigger complex circuit to make the approach simple and convenient

3.3 Making Layouts

The Mentor graphics IC layout editor is used for creating layouts according to the schematic. Different types of approaches are available for creating schematic, In this project the hierarchical approach is followed for the successful execution of the layouts.

3.4 Extracting Parameters

Successfully executed layouts are followed by the extraction of parameters from the respective layouts. These parameters contain the width, length of the Pmos and Nmos transistor. These parameters also provide the resistor value and capacitor value. These values can be used in simulating the layout and analyzing the graph

3.5 Simulation

The process of simulation includes execution of parameters extracted from layout in Hspice. This steps gives the results in the form of waveforms. The waveforms can be used to obtain the results such as delay, power.

4.Basic Blocks

4.1 AND Gate

The ‘and’ gate is the basic building block required in the Array Multiplier. The and gate is also used for executing the carry part of the half adder.

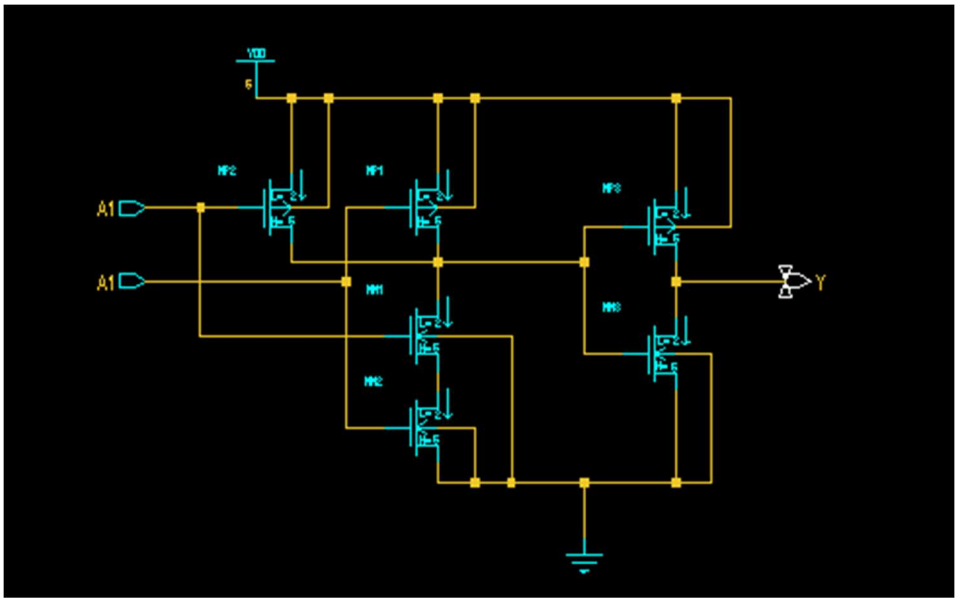


Fig 4.1 Schematic of AND Gate

The above circuit is converted into symbol of and gate using the Generate Symbol option in Mentor Graphics Schematic Editor.

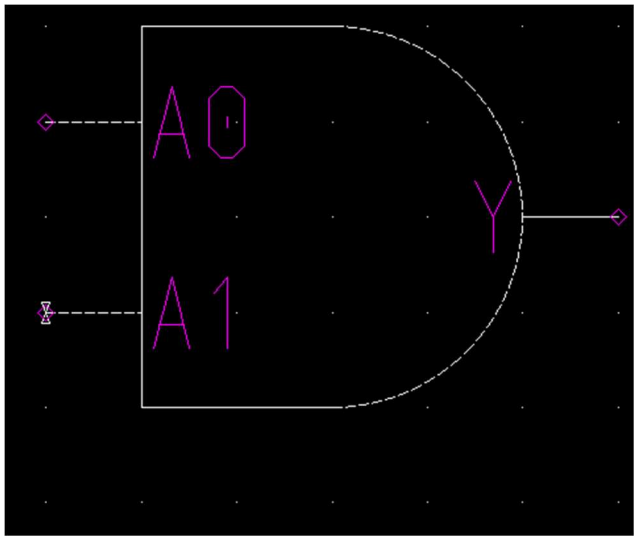


Fig 4.2 Symbol of AND Gate

4.2 OR Gate

The OR gate is further used in the circuit diagram of the full adder. Hence the schematic of the OR gate is created using the NOR2 gate.

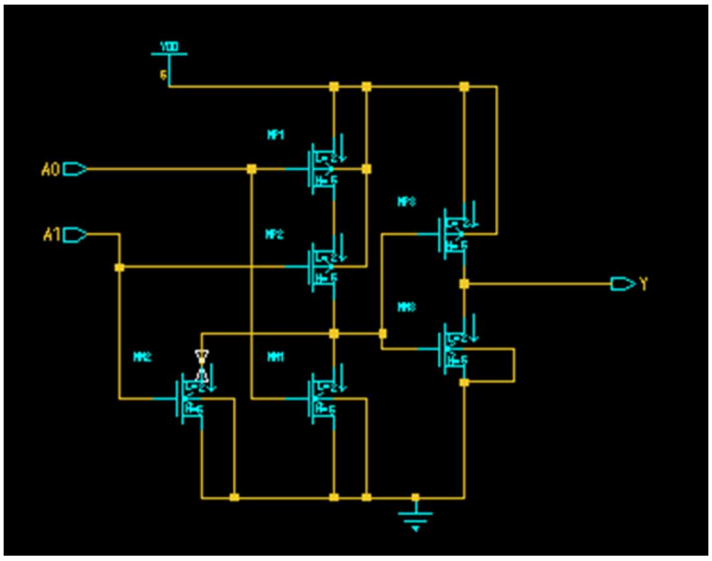


Fig 4.3: Schematic of OR Gate

The above circuit is converted into symbol of and gate using the Generate Symbol option in Mentor Graphics Schematic Editor.

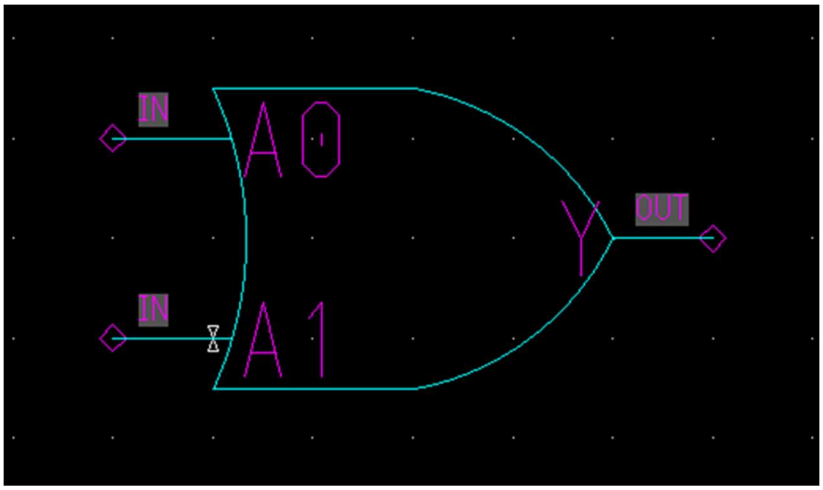


Fig 4.4 Symbol of OR Gate

4.3 XOR Gate

The XOR gate is the basic building block of the half adder. The XOR block will be used to generate the Sum part of the half adder.

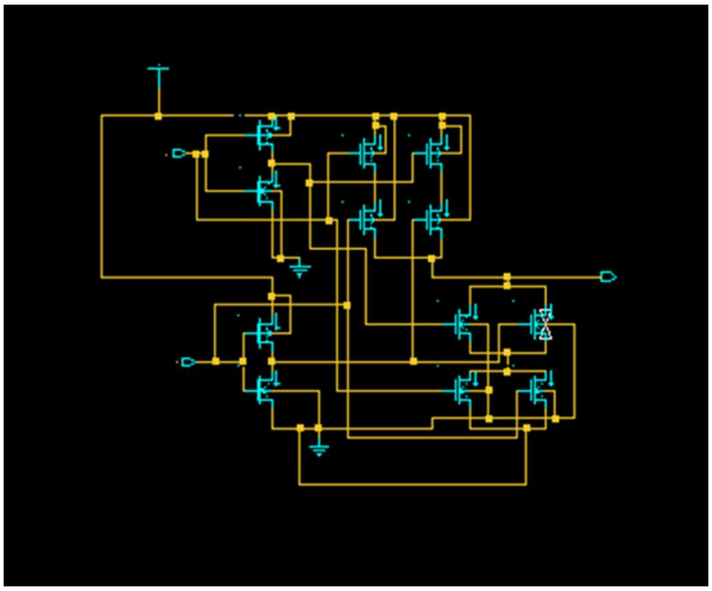


Fig4.5 Schematic of XOR gate

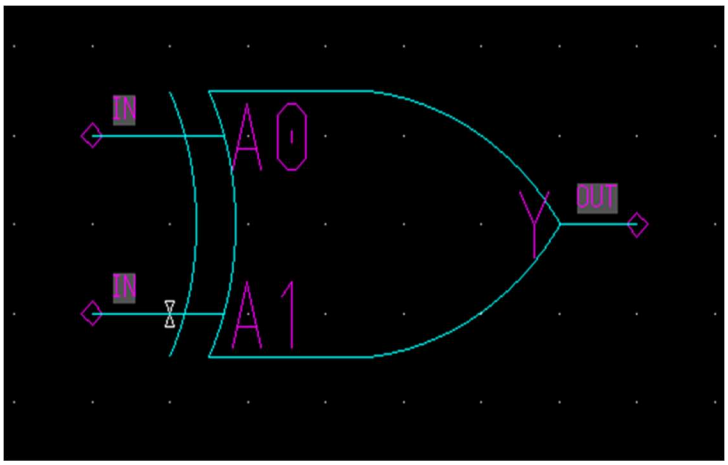


Fig 4.6: Symbol of XOR Gate

5.Detailed Design Approach

In this chapter, the detailed design of each of the component of the system will be discussed. Design criteria, delay considerations, circuits and layout will be discussed.

5.1 General Design Process

The general process that was used for the design of the 4x4 Array multiplier was to divide the multiplier into small functional blocks which could be then designed and executed. The 4x4 Multiplier was divided into two major components

A) Carry Save Adder (CSA)

The CSA consists of the AND Gate followed by the Full Adder. The And gate is used to obtain the products of multiplication and the pass it on the Full Adder.

B) Carry Propagate Adder(CPA)

The Carry Propagate Adder is array of simple full adders in horizontal way.

The CSA and CPA consist of two sub-blocks in the design. They are

A) Full Adder
B) Half Adder

5.2 Subsystem Design

5.2.1 Half Adder

The complex adders can be built using the basic adder known as Half Adder. The half adder is the most basic block of adder. The main function of the half adder is to perform addition of two binary bits. The schematic of the half adder can be represented as follows. The schematic is created using the basic blocks shown previously in section 4.

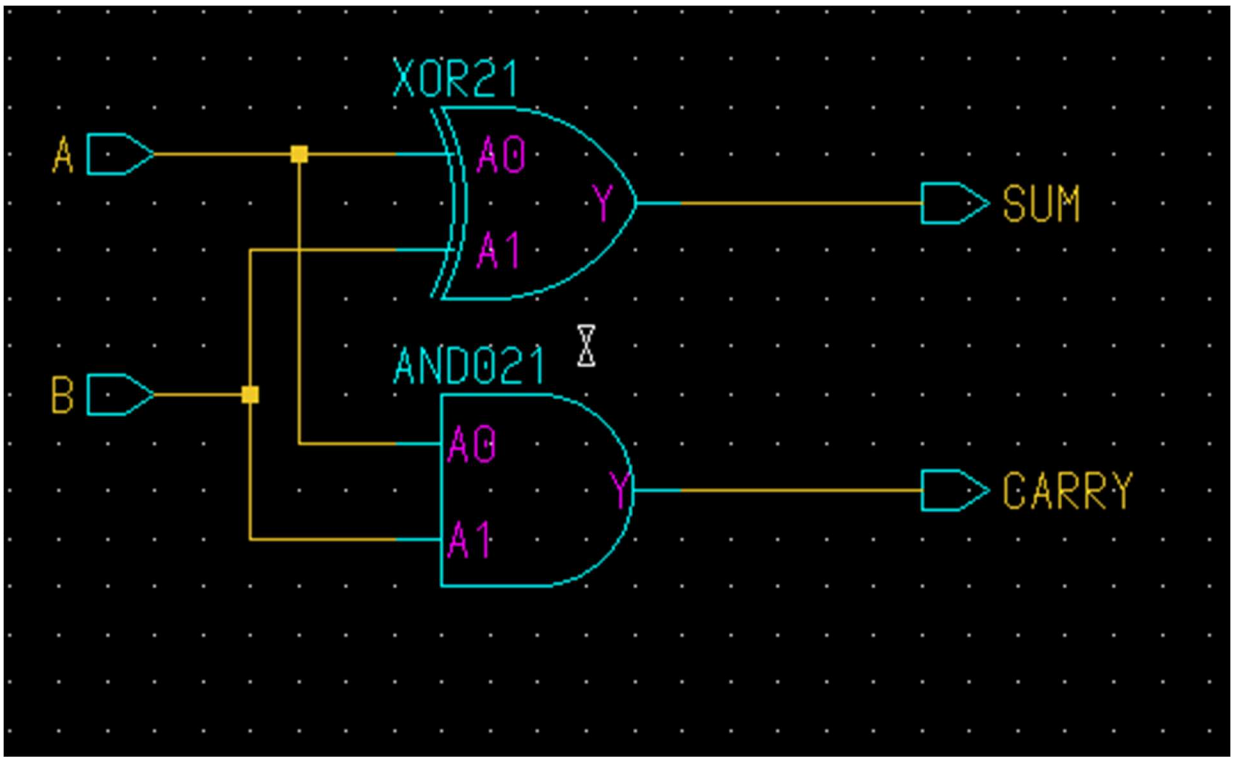


Fig 5.1: Schematic of Half Adder

The output port SUM provides the addition of A & B and the output port CARRY gives the carry if any in the process of the addition.

$S = \overline{A} \cdot B + A \cdot \overline{B}$

$C = A \cdot B$

Boolean equation

Fig 5.2: Half adder equations

A	B	Sum	Carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Table5.1: Half Adder Truth table

The simulation for the layout was carried out and by using appropriate geometrical approaches the layout of the half adder was created. The layout of the half adder is as follows.

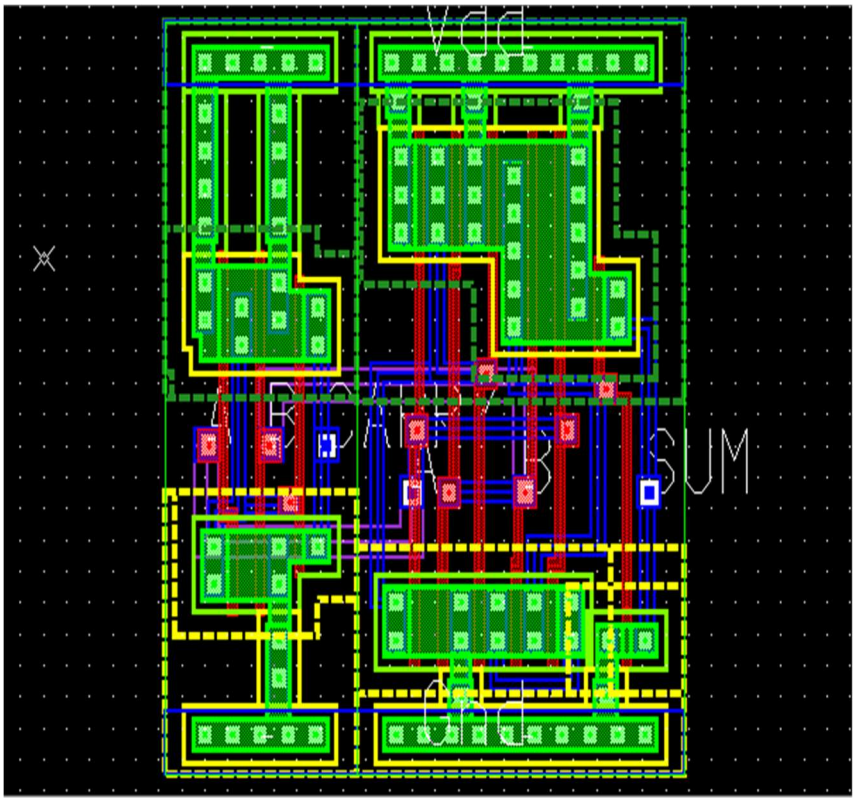


Fig5.3: Layout of Half Adder

The Symbol or the pin diagram of the half adder can be represented as follows:

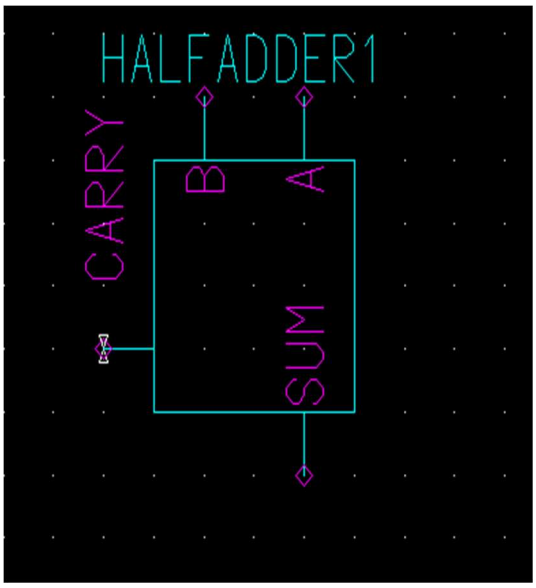


Fig5.4: Symbol of Half Adder

The simulation of the parameters was carried out using Hspice and desired delay results which can be shown through the timing diagram of half adder

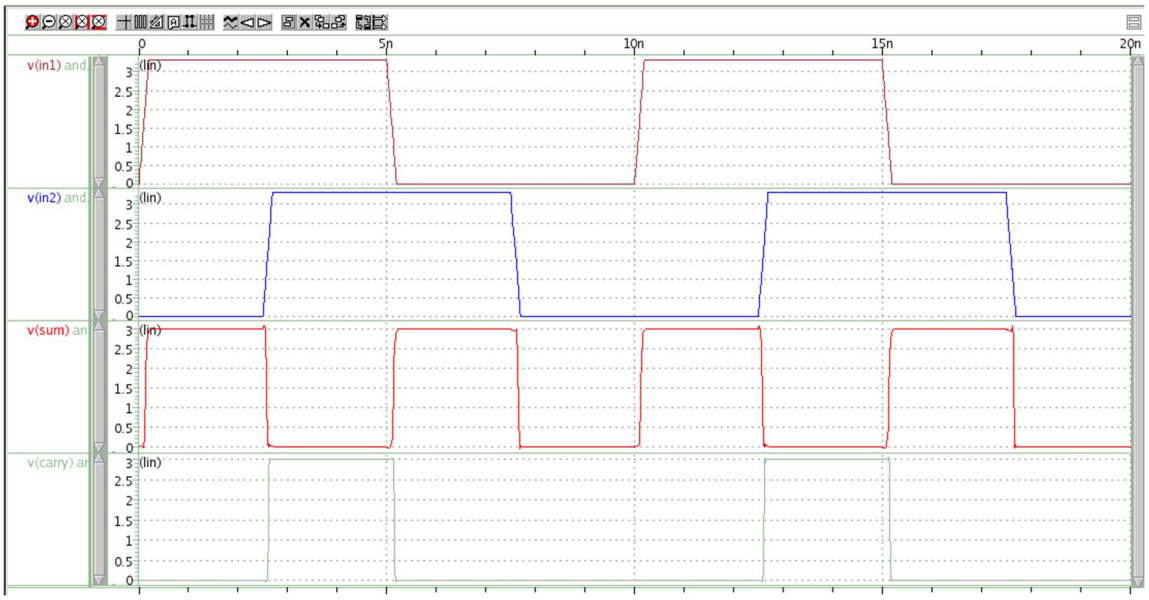


Fig5.5: Simulation waveform half adder

5.2.2 Full Adder

The Full Adder circuit is used for adding three one bit binary numbers such as A, B and Cin as inputs. The outputs are two one bit binary numbers such as SUM and Cout. The full adder is actually a component is cascade of adders which add 8,16 and 32 bit numbers. The carry output from the full adder is fed to another full adder as input if the full adders are in cascade. The full adder consists of two half adder connected with the OR gate. The circuit diagram of full adder can be represented as follows

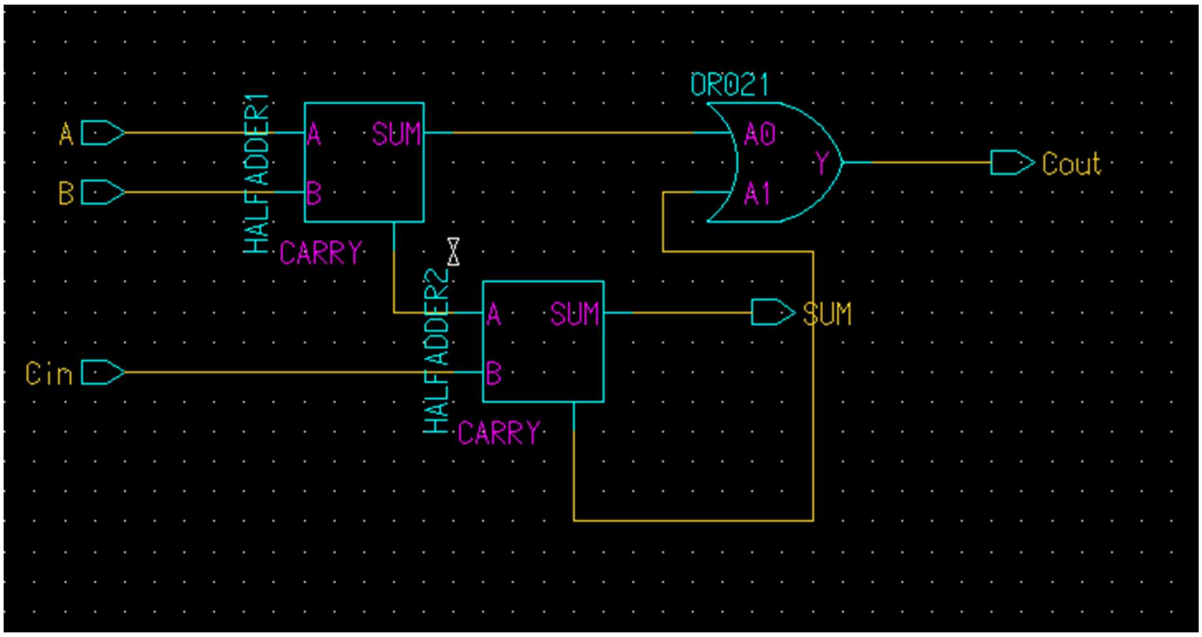


Fig5.6: Schematic of Full Adder

The OR gate provides the output as Cout which is representation for the Carry generated in the addition process. The Sum of one of the Half Adder gives SUM output. The SUM and Cout can be expressed as follows,

$$S = A \oplus B \oplus C_{in}$$

$$C_{out} = (A \cdot B) + (C_{in} \cdot (A \oplus B))$$

Truth Table (Full Adder) (Table 5.2)

INPUT			OUTPUT	
A	B	Cin	Cout	S
0	0	0	0	0
1	0	0	0	1
0	1	0	0	1
1	1	0	1	0
0	0	1	0	1
1	0	1	1	0
0	1	1	1	0
1	1	1	1	1

The simulation for the layout was carried out and by using appropriate geometrical approaches the layout of the full adder was created. The layout of the full adder is as follows

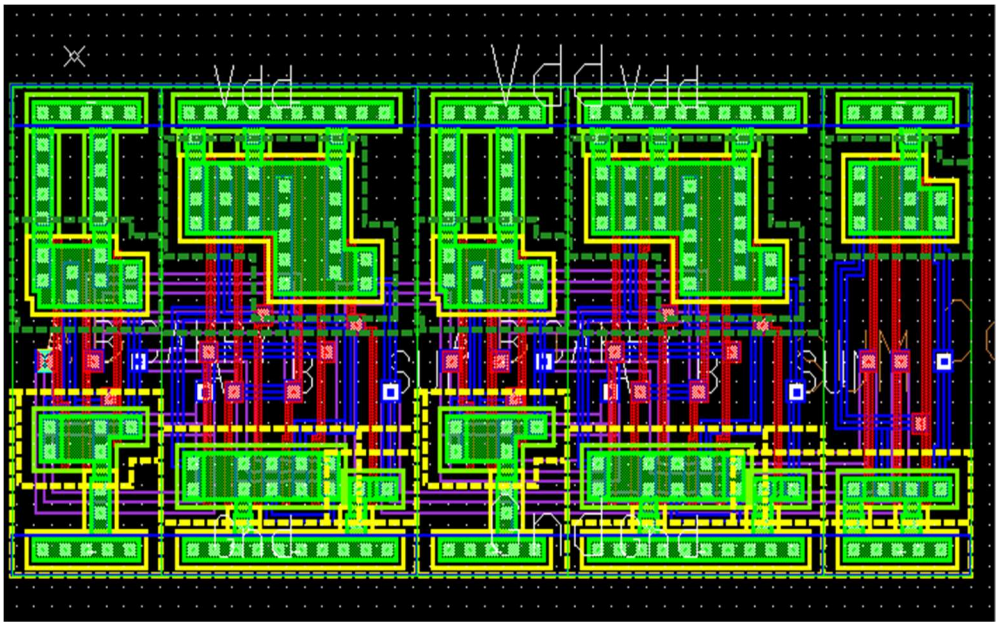


Fig5.7 Layout of Full Adder

The Symbol of the Full adder can be represented as follows:

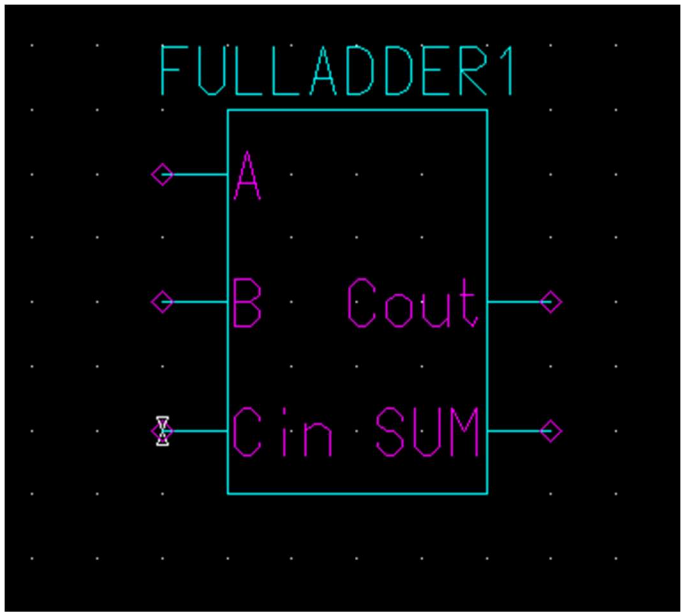


Fig5.8: Symbol of Full Adder

The simulation of the parameters was carried out using Hspice and desired delay results which can be shown through the timing diagram of Full adder

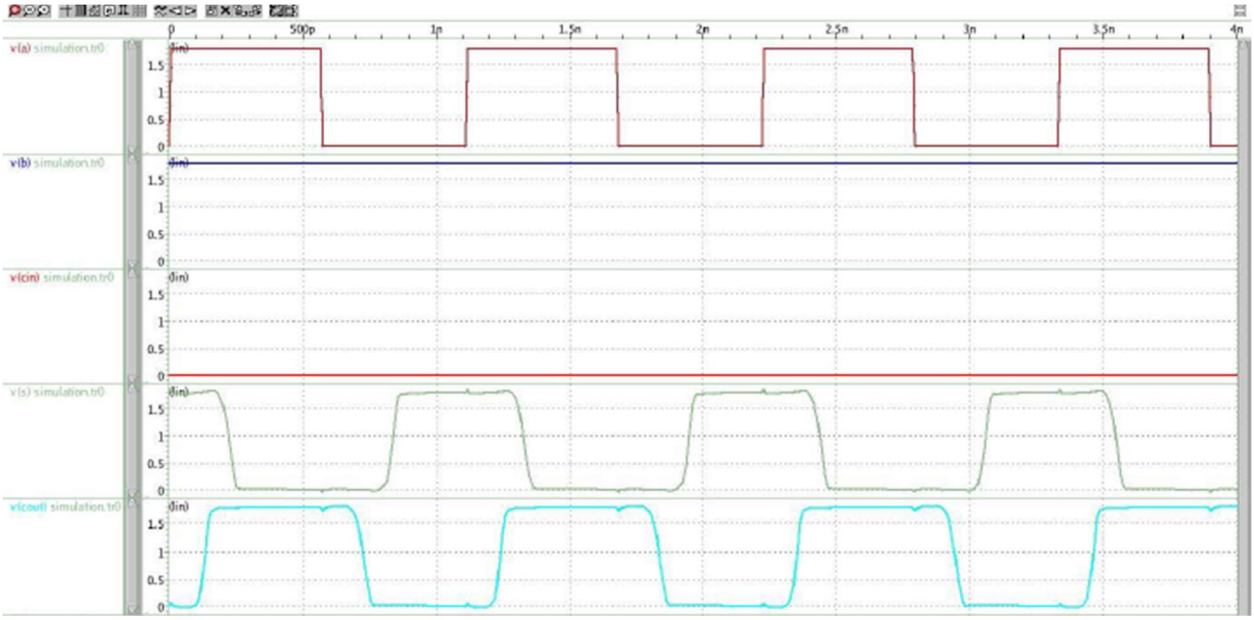


Fig 5.9 Wave form Simulation Full Adder

5.2.3 Carry Save Adder(CSA)

The CSA part of the 4x4 Array Multiplier consist of the two input and gate followed by the full adder. The CSA is used to extract output products from the least significant bit and moreover to send the carry output to full adder in the next row. The difference from normal circuits to this is that CSA gives the carry output as input to the full adder in the next row instead of adjacent full adder. Each cell contains a 2-input AND gate that forms a partial product and a full adder (CSA) to add the partial product into the running sum. The first row converts the first partial product into carry-save redundant form. Each later row uses the CSA to add the corresponding partial product to the carry-save redundant result of the previous row and generate a carry-save redundant result. The least significant *N* output bits are available as sum outputs directly from CSAs. The most significant output bits arrive in carry-save redundant form and require an *M*-bit carry-propagate adder to convert into regular binary form.

The Schematic of a part of CSA array can be shown with an AND gate followed by the full adder previously designed and executed.

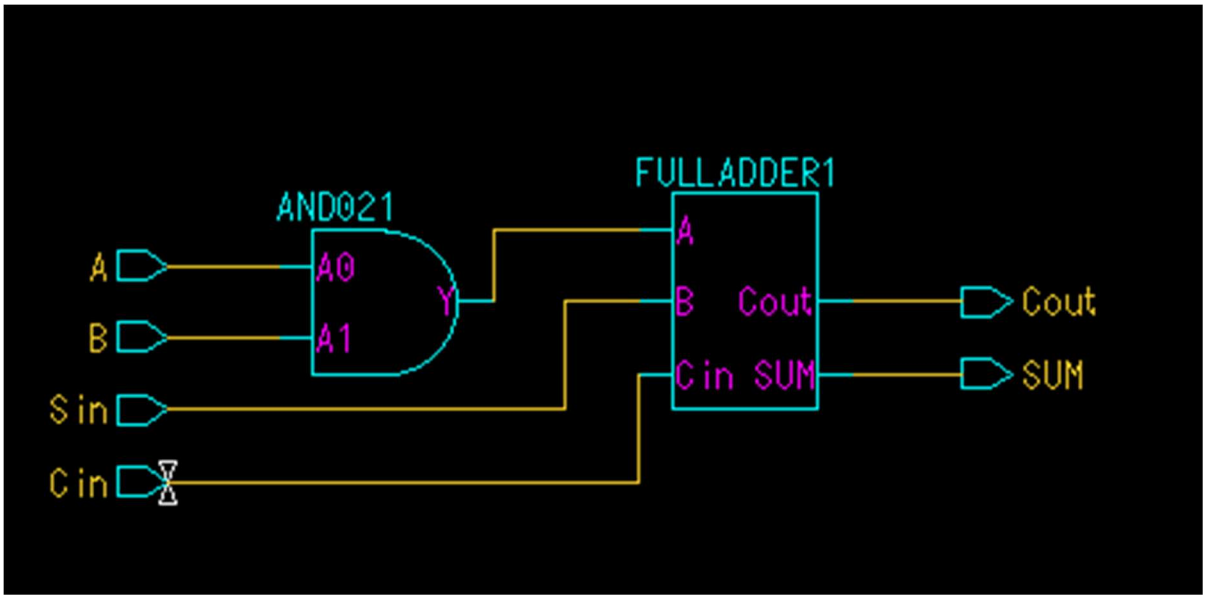


Fig5.10: Schematic of CSA Full adder

A key element of the design is a compact CSA. This not only benefits area but also helps performance because it leads to short wires with low wire capacitance. An ideal CSA design has approximately equal sum and carry delays because the greater of these two delays limits performance.

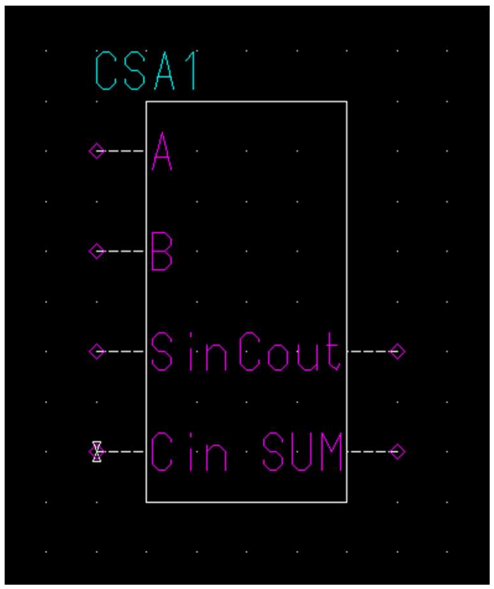


Fig5.11 Symbol of CSA

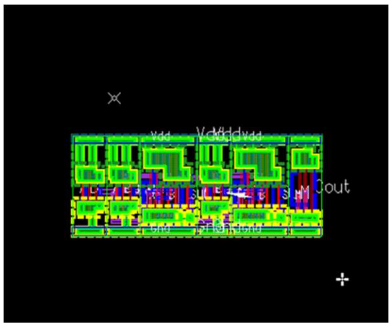


Fig5.12: Layout of CSA

5.2.4 Carry Propagate Adder (CPA)

The CPA consist of the full adders arranged in adjacent manner to each other. The CPA in 4x4 array multiplier is used as carry ripple adder. The array is regular in structure and uses a single type of cell, so it is easy to design and lay out The adder can easily be pipelined with the placement of registers between rows. In practice, circuits are assigned rectangular blocks in the floorplan so the parallelogram shape wastes space. In CPA the Cout of one full adder is passed to Cin of the adjacent full adder. The most significant output bits arrive in carry-save redundant form and require an *M*-bit carry-propagate adder to convert into regular binary form. Finally, the output of the array Multiplier is extracted from the sum outputs of the full adders in the CPA array.

5.2.5 4x4 Array Multiplier

The Schematic of 4x4 Array multiplier is created using the array of CSA and CPA. The Complete Schematic can be represented as follows

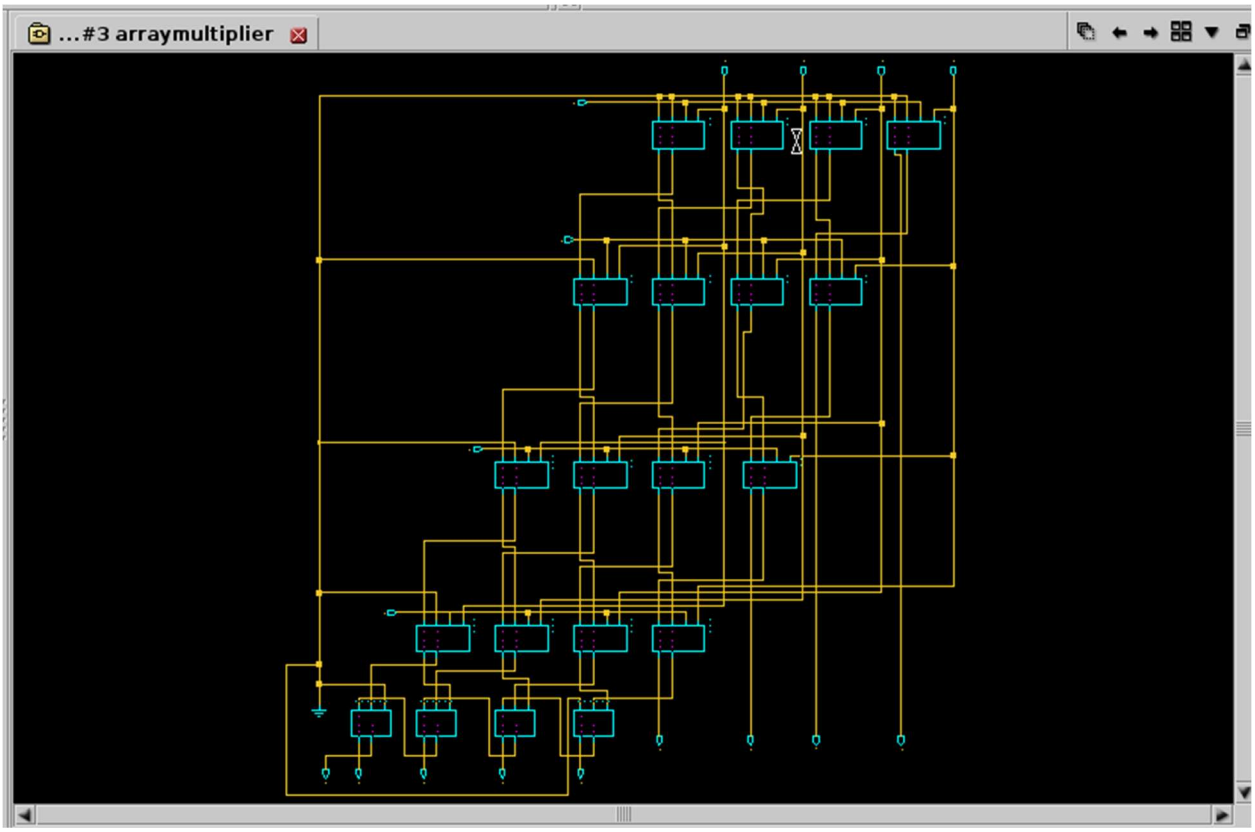


Fig5.13: Schematic of Array Multiplier

The layout of the 4x4 array multiplier was completed using the layouts of the previously completed layouts of the sub blocks used in the circuit. The layouts were specifically arranged in such a way that the area occupied is minimum. The arrangement of layout avoids interference of the metal wires. Extreme care was taken to avoid the joining of nets. The LVS test was carried for each of the layouts describes for which the test results are attached in later section of the report.

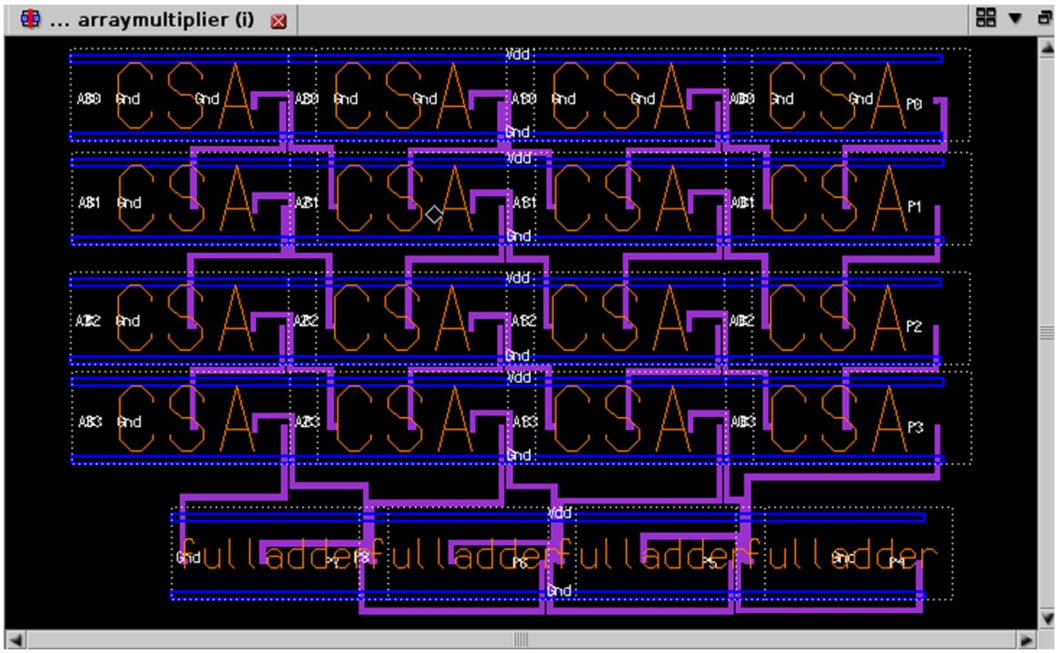


Fig5.14: Layout of Array Multiplier

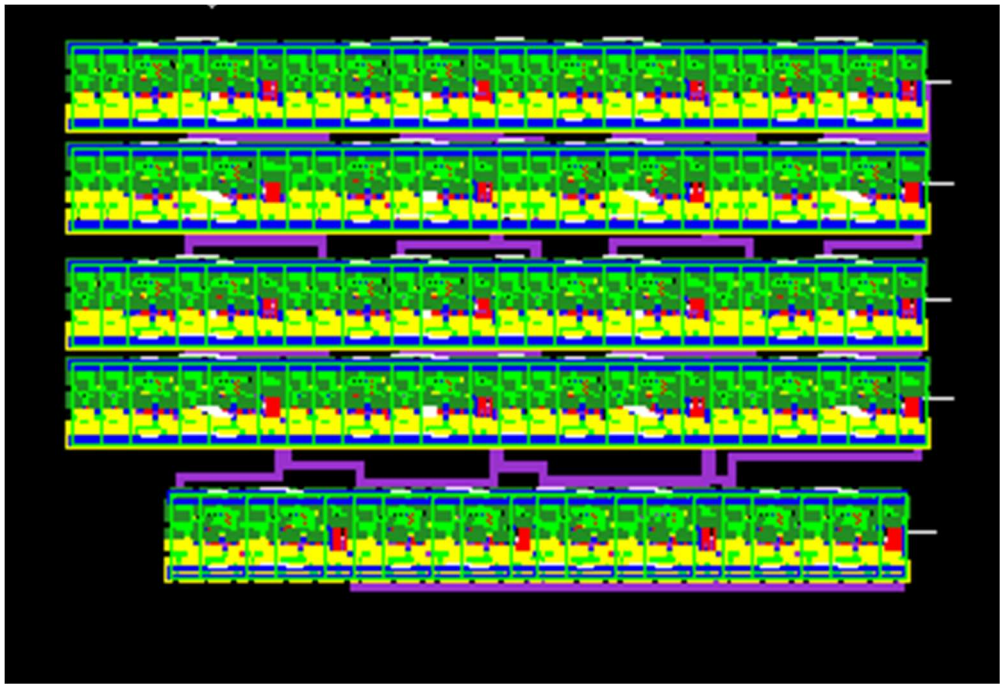


Fig 5.15 Layout of Array Multiplier

The symbol for schematic of 4x4 Array Multiplier is show below. The symbol represents 8 inputs i.e. A0 to A3 and B0 to B3. There are 9 output pins which gives the product of the above inputs. The Pins P0 to P7 gives the 8-bit output of the multiplication. If there is any carry generated during the complete process, the carry can be obtained at the port P8.

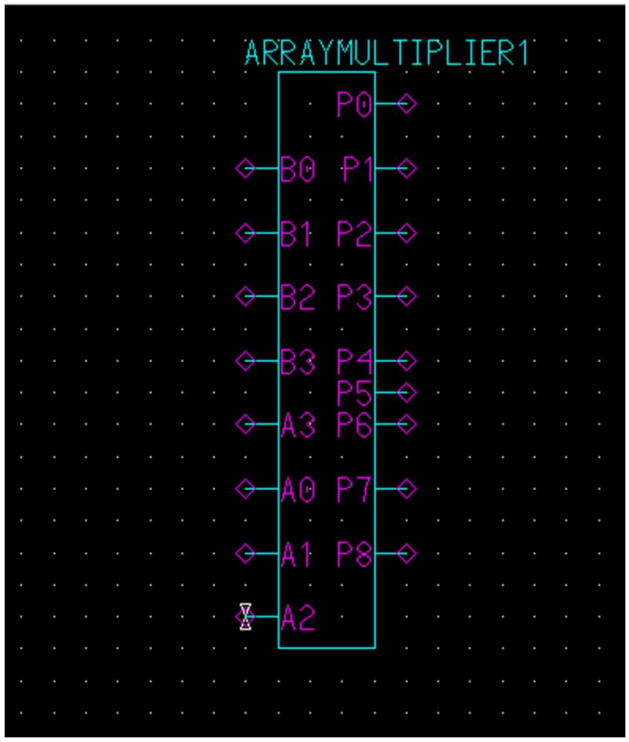


Fig5.16: Symbol of Array Multiplier

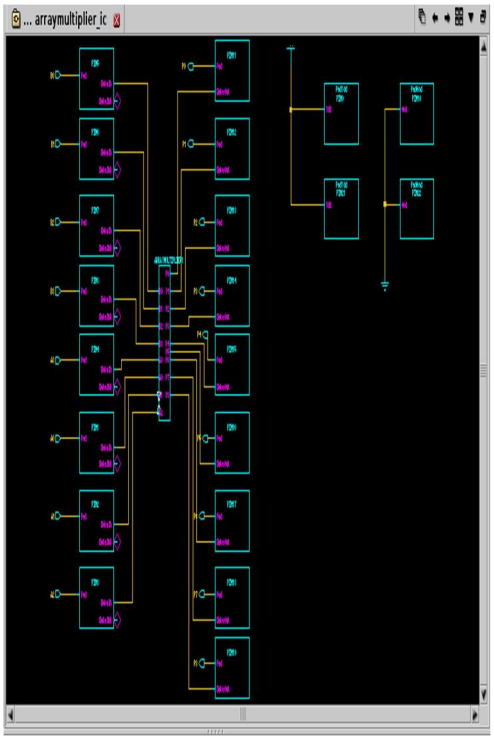


Fig5.17: Schematic with input output pads

The Schematic of the 4x4 array multiplier with input output pads attached along with the Vdd and Gnd source is shown. The IC of the 4x4 array multiplier will consist of 21 pins including the 8 inputs pins, 9 output pins and 2 pins for Vdd and Gnd each.

5.3 Results

The process of parameter extraction was executed on the final layout for simulation purpose . The results obtained are as follows:

Logic Style : CMOS

Delay (ns) : 8.30

Number of Pins : 21

8-input pins

9-output pins

2-Vdd pins

2-Gnd Pins

Appendix A Design Verification Reports

LVS Report of Half adder

```
  ################################################
##                                ##
##      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: /afs/cad.njit.edu/u/s/k/sk2339/model/project_array/halfadder/lvs.rep
LAYOUT NAME: /afs/cad/u/s/k/sk2339/model/project_array/halfadder/halfadder
SOURCE NAME:
/afs/cad.njit.edu/u/s/k/sk2339/model/project_array/halfadder/halfadder/halfadder_IC_Station_SDL_Flat
RULE FILE: /afs/cad/sw.common/mentor.2012/adk3_1/technology/ic/process/tsmc018.rules
LVS MODE: Mask
RULE FILE NAME: /afs/cad/sw.common/mentor.2012/adk3_1/technology/ic/process/tsmc018.rules
CREATION TIME: Wed Dec 14 22:46:38 2016

CURRENT DIRECTORY: /afs/cad.njit.edu/u/s/k/sk2339/model/project_array/halfadder
USER NAME: sk2339

```
*****************************************************************************************************
  ***************
                  OVERALL COMPARISON RESULTS
*****************************************************************************************************
  ***************
```

```
          #    ###################    _ _
         #     #          #     * *
       # #     #   CORRECT   #      |
       ##      #          #    \___/
        #      ###################
```

INITIAL NUMBERS OF OBJECTS

	Layout	Source	Component Type
Ports:	6	6	

```
 Nets:            13       13

 Instances:        9        9       mn (4 pins)
                   9        9       mp (4 pins)
                 ------   ------
 Total Inst:      18       18
NUMBERS OF OBJECTS AFTER TRANSFORMATION
---------------------------------------

                 Layout   Source       Component Type
                 ------   ------       --------------
 Ports:           6        6

 Nets:            9        9


 Instances:       2        2       mn (4 pins)
                  7        7       mp (4 pins)
                  1        1       SUP2 (3 pins)
                  2        2       SMN2 (4 pins)
                  1        1       SPMN_2_1 (5 pins)
                 ------   ------
 Total Inst:     13       13



*********************************************************************************************
  **************
                          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                    "VSS"
  LVS CELL SUPPLY                   NO
  LVS RECOGNIZE GATES                ALL
  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                  YES

  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
  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 SCALE X PARAMETERS          NO
  LVS SPICE STRICT WL                   NO
  // LVS SPICE OPTION
  LVS EDDM PROCESS M                    NO
  LVS STRICT SUBTYPES                   NO
  LVS EXACT 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
```

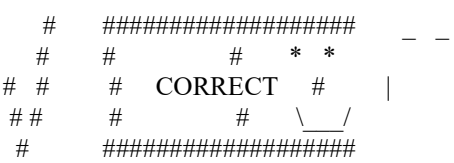
```
  // Trace Property


// 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   Matched   Unmatched   Unmatched   Component
             Layout    Source     Layout      Source    Type
            -------   -------   ---------   ---------   ---------
  Ports:          6         6           0           0

  Nets:           9         9           0           0

  Instances:      2         2           0           0    mn(nmos4)
                  7         7           0           0    mp(pmos4)

      1         1           0           0    SUP2
                2         2           0           0    SMN2
                1         1           0           0    SPMN_2_1
            -------   -------   ---------   ---------
  Total Inst:    13        13           0           0


o Initial Correspondence Points:

  Ports:      VDD A B CARRY SUM Gnd
```

LVS Report of Full Adder

```
          ################################################
          ##                                            ##
          ##      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: /afs/cad/u/s/k/sk2339/model/project_array/fulladder1/lvs.rep
LAYOUT NAME: /afs/cad/u/s/k/sk2339/model/project_array/fulladder1/fulladder
SOURCE NAME: /afs/cad/u/s/k/sk2339/model/project_array/fulladder1/fulladder_IC_Station_SDL_Flat
RULE FILE: /afs/cad/sw.common/mentor.2012/adk3_1/technology/ic/process/tsmc018.rules
LVS MODE: Mask
RULE FILE NAME: /afs/cad/sw.common/mentor.2012/adk3_1/technology/ic/process/tsmc018.rules
CREATION TIME: Mon Dec 12 20:07:03 2016
CURRENT DIRECTORY: /afs/cad.njit.edu/u/s/k/sk2339/model/project_array/fulladder1
USER NAME: sk2339

****
****
OVERALL COMPARISON RESULTS
‘
****
****



INITIAL NUMBERS OF OBJECTS

	Layout	Source	Component Type
Ports:	7	7	
Nets:	26	26	
Instances:	21	21	mn (4 pins)
	21	21	mp (4 pins)
Total Inst:	42	42	

NUMBERS OF OBJECTS AFTER TRANSFORMATION

	Layout	Source	Component Type
Ports:	7	7	
Nets:	17	17	
Instances:	7	7	mn (4 pins)
	15	15	mp (4 pins)

```
   3        3        SUP2 (3 pins)


4        4        SMN2 (4 pins)
                  2        2        SPMN_2_1 (5 pins)
               ------   ------
Total Inst:       31       31



*****************************************************************************************************
  ***************
                              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                     "VSS"
  LVS CELL SUPPLY                   NO
  LVS RECOGNIZE GATES                 ALL
  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                  YES
  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
  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 SCALE X PARAMETERS        NO
  LVS SPICE STRICT WL               NO
  // LVS SPICE OPTION
  LVS EDDM PROCESS M                  NO

 LVS STRICT SUBTYPES                NO
```

```
LVS EXACT 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 Propert
// 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
```

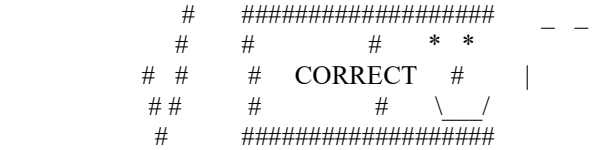
```
// TRACE PROPERTY  d  instpar(p) perimeter p 0

*******************************************************************************************************
  **************

   INFORMATION AND WARNINGS
*******************************************************************************************************
  **************



 Matched   Matched   Unmatched   Unmatched   Component
           Layout    Source     Layout      Source    Type
          -------   -------   ---------   ---------   ---------
  Ports:          7         7          0          0

  Nets:          17        17          0          0

  Instances:      7         7          0          0    mn(nmos4)

   15        15          0          0    mp(pmos4)
               3         3          0          0    SUP2
               4         4          0          0    SMN2
               2         2          0          0    SPMN_2_1
          -------   -------   ---------   ---------
  Total Inst:     31        31          0          0
   o Initial Correspondence Points:
  Ports:      VDD A B Cin SUM Cout Gnd
*******************************************************************************************************
  **************
                          SUMMARY
*******************************************************************************************************
  **************
Total CPU Time:     0 sec
Total Elapsed Time:  0 sec
```

LVS Report of CSA

```
        ##################################################
        ##                                              ##
        ##      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:        /afs/cad.njit.edu/u/s/k/sk2339/model/project_array/CSAfulladder/lvs.rep
LAYOUT NAME:            /afs/cad.njit.edu/u/s/k/sk2339/model/project_array/CSAfulladder/CSAfulladder
SOURCE NAME:
   /afs/cad.njit.edu/u/s/k/sk2339/model/project_array/CSAfulladder/CSAfulladder_IC_Station_SDL_Flat
RULE FILE:              /afs/cad/sw.common/mentor.2012/adk3_1/technology/ic/process/tsmc018.rules
LVS MODE:               Mask

RULE FILE NAME:         /afs/cad/sw.common/mentor.2012/adk3_1/technology/ic/process/tsmc018.rules
```

CREATION TIME: Mon Dec 12 20:48:09 2016
CURRENT DIRECTORY: /afs/cad.njit.edu/u/s/k/sk2339/model/project_array/CSAfulladder
USER NAME: sk2339

****
OVERALL COMPARISON RESULTS
****



INITIAL NUMBERS OF OBJECTS

	Layout	Source	Component Type
Ports:	9	9	
Nets:	34	34	
Instances:	27	27	mn (4 pins)
	27	27	mp (4 pins)
Total Inst:	54	54	

NUMBERS OF OBJECTS AFTER TRANSFORMATION

	Layout	Source	Component Type
Ports:	9	9	
Nets:	23	23	
Instances:	9	9	mn (4 pins)
	21	21	mp (4 pins)
	3	3	SUP2 (3 pins)
	6	6	SMN2 (4 pins)
	2	2	SPMN_2_1 (5 pins)
Total Inst:	41	41	

```
*************************************************************************************************
  **************
                        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                   "VSS"
  LVS CELL SUPPLY                  NO
  LVS RECOGNIZE GATES               ALL
  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                 YES
  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
  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 SCALE X PARAMETERS         NO
  LVS SPICE STRICT WL                NO
  // LVS SPICE OPTION

 LVS EDDM PROCESS M                  NO
  LVS STRICT SUBTYPES                NO
  LVS EXACT 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   Matched   Unmatched   Unmatched   Component
               Layout    Source     Layout      Source    Type
              -------   -------   ---------   ---------   ---------
  Ports:            9         9           0           0

  Nets:            23        23           0           0


 Instances:         9         9           0           0   mn(nmos4)
                   21        21           0           0   mp(pmos4)
                    3         3           0           0   SUP2
                    6         6           0           0   SMN2
                    2         2           0           0   SPMN_2_1
                -------   -------   ---------   ---------
  Total Inst:      41        41           0           0
```

o Layout Names That Are Missing In The Source:

Ports: Cout

o Initial Correspondence Points:

Ports: VDD P Q R S Cin SUM Gnd

```
********************************************************************************************************
  ***************
                          SUMMARY
********************************************************************************************************
  ***************
Total CPU Time:     0 sec
Total Elapsed Time:  0 sec
```

LVS Report of 4x4 Array Multiplier

```
        ##################################################
        ##                                              ##
        ##      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: /afs/cad.njit.edu/u/s/k/sk2339/model/project_array/arraymultiplier/lvs1.rep
LAYOUT NAME: /afs/cad.njit.edu/u/s/k/sk2339/model/project_array/CSA/CSA
SOURCE NAME: /afs/cad.njit.edu/u/s/k/sk2339/model/project_array/CSA/CSA/CSA_IC_Station_SDL_Flat
RULE FILE: /afs/cad/sw.common/mentor.2012/adk3_1/technology/ic/process/tsmc018.rules
LVS MODE: Mask

RULE FILE NAME: /afs/cad/sw.common/mentor.2012/adk3_1/technology/ic/process/tsmc018.rules
CREATION TIME: Tue Dec 13 21:35:38 2016
CURRENT DIRECTORY: /afs/cad.njit.edu/u/s/k/sk2339/model/project_array/arraymultiplier

USER NAME: sk2339

****
****
OVERALL COMPARISON RESULTS
****
****

INITIAL NUMBERS OF OBJECTS

	Layout	Source	Component Type
Ports:	8	8	
Nets:	30	30	
Instances:	24	24	mn (4 pins)
	24	24	mp (4 pins)
Total Inst:	48	48	

NUMBERS OF OBJECTS AFTER TRANSFORMATION

	Layout	Source	Component Type
Ports:	8	8	
Nets:	20	20	
Instances:	8	8	mn (4 pins)
	18	18	mp (4 pins)
	3	3	SUP2 (3 pins)
	5	5	SMN2 (4 pins)
	2	2	SPMN_2_1 (5 pins)
Total Inst:	36	36	

****
****
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                    "VSS"
  LVS CELL SUPPLY                   NO
  LVS RECOGNIZE GATES                ALL
  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                  YES
  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
  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 SCALE X PARAMETERS       NO
  LVS SPICE STRICT WL                NO
  // LVS SPICE OPTION

 LVS EDDM PROCESS M                 NO
  LVS STRICT SUBTYPES                NO
  LVS EXACT 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    Matched    Unmatched    Unmatched    Component
            Layout     Source       Layout       Source    Type
          -------    -------    ---------    ---------    ---------
  Ports:         8          8            0            0

  Nets:         20         20            0            0

  Instances:     8          8            0            0    mn(nmos4)
                18         18            0            0    mp(pmos4)
                 3          3            0            0    SUP2
                 5          5            0            0    SMN2
                 2          2            0            0    SPMN_2_1
          -------    -------    ---------    ---------
  Total Inst:   36         36            0            0
```

o Initial Correspondence Points:

  Ports: VDD A B Sin Cin SUM Cout Gnd

```
********************************************************************************************************
  ***************
                          SUMMARY
********************************************************************************************************
  ***************

Total CPU Time:      0 sec
Total Elapsed Time:  0 sec
```

6.References

1)Cmos VLSI Design; A Circuit and Systems Perspective Neil H Weste, David Money Harris

2) https://en.wikipedia.org/wiki/XOR_gate'

3)http://relayfiles.njit.edu/yl333/NJIT_Default/MP4%20with%20Smart%20Player%20%28Large%29/VLSI_Video4_-_20140810_234650_15.html

4)http://relayfiles.njit.edu/brateris/Faculty__For_Modules/MP4_with_Smart_Player_(Large)/Mentor_Graphics_Hierarchical_Design_-_20151124_114039_15.html