

EE620 — DESIGN OF DIGITAL SYSTEMS

From: Dinesh Anand Bashkaran

Uid: 57000943

Instructor: Dr. Mark.A.indovina

Date: Performed: 10/28/2016 **Due:** 10/31/2016

Design Constraints and Discussions:

- Decimal Bit code: 001101
- The base size of pMOS and nMOS is 540/45nm and 240/45nm.

Project 1:

The idea proposed in project 1 is about the various standard cells, with respect to their schematic and layout. It helped in understanding how the generic standard cells should be designed and how the pins and abutment has to be placed. A good idea about DRC and LVS helped in understanding the constraints that has be followed with respect to design.

- **Inverter:**

The inverter is X2 drive which makes the size of pMOS and nMOS to 1080/45nm and 540/45nm. The scaling is done to maintain the equivalent resistance across the pMOS and nMOS. There were no difficulties faced during the layout.

- **NAND/TMUX/NOR/XOR/OAI:**

These cells go with the base size of pMOS and nMOS is 540/45nm and 240/45nm. The size of their individual cells is scaled accordingly to maintain the equivalent resistance across them. Considering the layout, issue is faced in sharing the diffusion part to max as possible and trying to minimize the width of the layout to max as possible. Also the part where the pins have to be placed in middle of track raised a challenge. Trying to maintain only pin across the track have forced

some changes in the layout. However, gave very good understanding about how the router tracks and having more than a pin can mess with routing of hierarchical cells.

Also in some cells, just to maintain the cell width in 8λ one track was used unnecessarily. Apart from these, not much issues was faced in design of the layout.

Project 2:

The project 2 is very interesting and has very good link with project 1. It is more of hierarchical design where the cells made are combined as a single block to work on a complex function. It gives a good knowledge how the standard cells functions can be combined together to get a complex function and makes the understanding in easier way.

The Routing technique is introduced in this project. Router is a tool which is used to route the connections between the pins automatically. Routing is done to reduce the complexity of the designing a block and also to reduce the connection space between the pins. Router is a great tool and does an excellent job except in some cases where the manual connection is better than router tool. Overall routing done by router is generally pretty good.

Design done in this project includes

- Full adder
- D-Flip Flop
- TIEHI/TIELO
- ADD16
- BSC

- BSR
- BSSUM
- BSTEST

Coming to the hierarchical part, this project not only shows how the hierarchical design is done but also shows few advantages of it and the reason behind this project.

- The hierarchical block once designed can also be used in other circuit as a single block which drives as a small part of that circuit.
- Simply saying, it can be reused.
- For instance, in this project the BSC is done using the DFF and TMUX. The instance of BSC is reused in BSR and BSSUM
- Hierarchical design of bigger circuit reduces the complexity of calculating the number of cells required, as it only consists of blocks where the blocks consist of unit cells interconnected between them.
- This reduces the connection between them and reduces the wire spacing which apparently reduces the delay in signal.
- It is also easy to debug and find which part of circuit is causing error.

Route Cells and Discussions:

- D Flip Flop:

The D Flip Flop is designed using TMUX which consists of Master and Slave circuit. 2 non overlapping clocks are used to drive this design. D acts as the input to the master circuit and the clock PHI2 is used to drive the master circuit. According to the logic of mux, whenever the clock signal is high the output in D is shown in Output side of the master circuit. Here the output is given as feedback to other input of the master circuit.

The slave circuit acts similar to the master but it is driven by other clock PHI1. The input to the slave circuit is the output of the master circuit. So whenever, the PHI1, goes high the output is shown on Q.

Generally, the output is shown on Q whenever the PHI2 goes high first followed by PHI1. The minimum time difference between falling edge of clock 2 and rising edge of clock 1 is called guard time. For the circuit to work ideal way it is good to maintain a high guard time. When the guard time is very low, the circuit gets transparent and output fails to show up with respect to input. This is called as the hazard condition of Flip Flop.

Coming to the layout of the design, the instances are generated respectively from the unit cells. The cells are then placed in a single PR boundary which then acts as a single block unit. After which pins are placed and they are routed using router. Not much difficulty was faced in the design of layout.

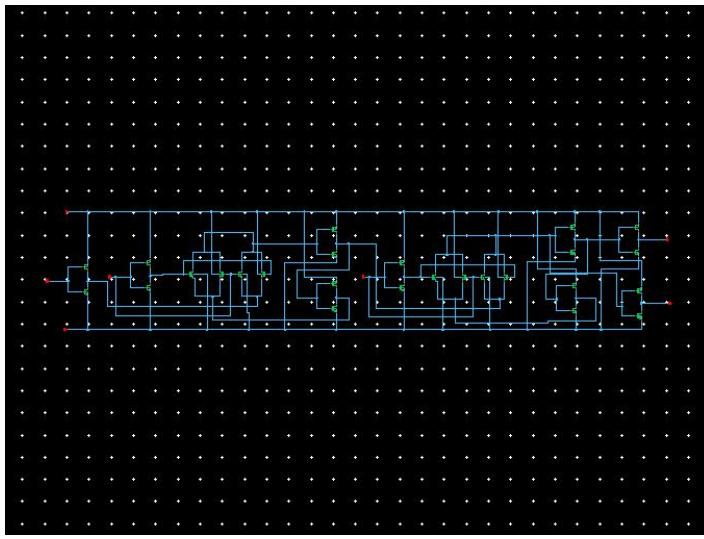


Fig:1 Schematic of D flip flop

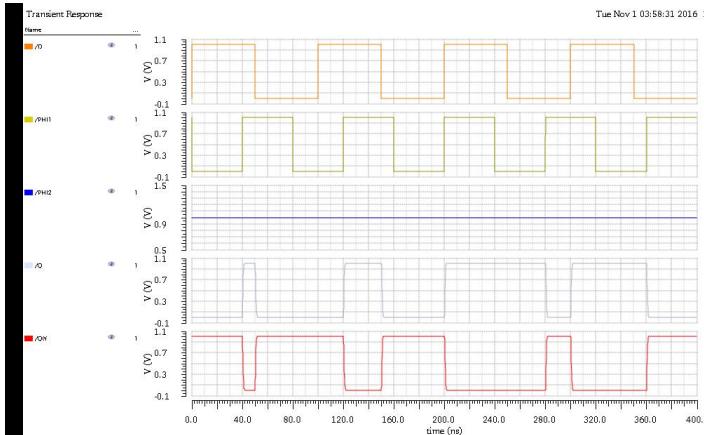


Fig 2: waveform of D flip flop

From the waveform, the Input D is shown on the output only when the PHI1 goes high and also the PHI2 is made to be high for complete cycle. So, the D input will always pass through master circuit and whenever the clock of the slave circuit goes high, the input is seen in the output end.

- TIEHI/TIELO:

TIEHI and TIELO are connected to the floating or un-used pins. The main reason of using them is to provide the ESD safety. They prevent the damage of the pins from static power, they also provide act as TIEHI-1 and TIELO-0.

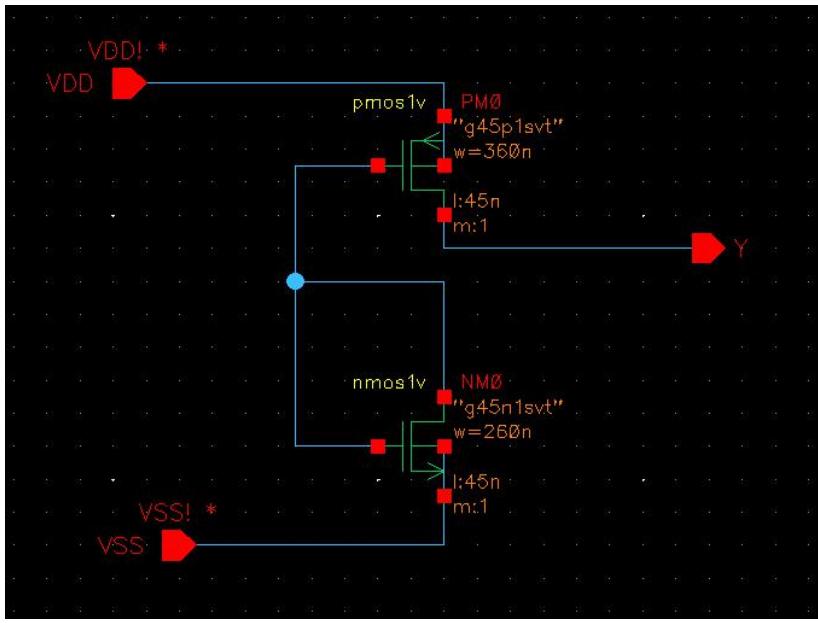


Fig 3: schematic of TIEHI

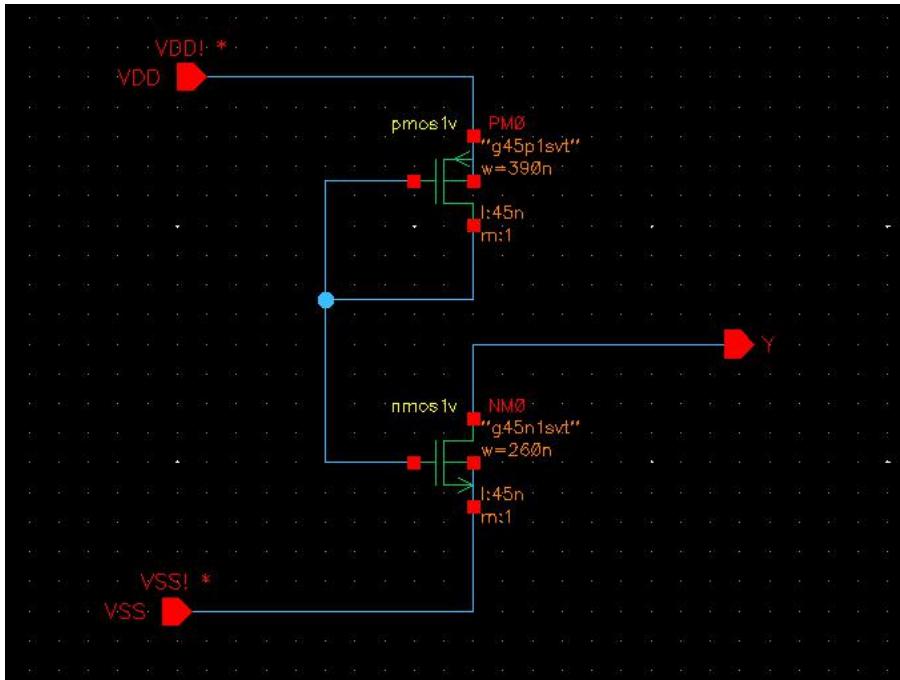


Fig4: schematic of TIELO

- Full adder:

Full adder is designed to calculate 3 inputs and show them in Output and carry respectively. Considering the layout, the design is hierarchical of 2 cells NAND and XOR. The layout of individual cells is generated and they are placed such that, they form a single block of cell which consists of 2 or more instance of other cells in them (In this case, 2XOR and 3 NAND). After which, the pins are placed respectively

from the schematic and they are placed in corners for routing. Once, the pins are placed, they are routed by Routing tool which find the shortest path to match the pins using different layers of metal. Routing Tool reduces the complexity pretty much however at some points it acts weird, which can be figured and routed manually. Once this is done, the PR boundary is drawn for this block which is done by calculating the PR boundary of the individual cells used.

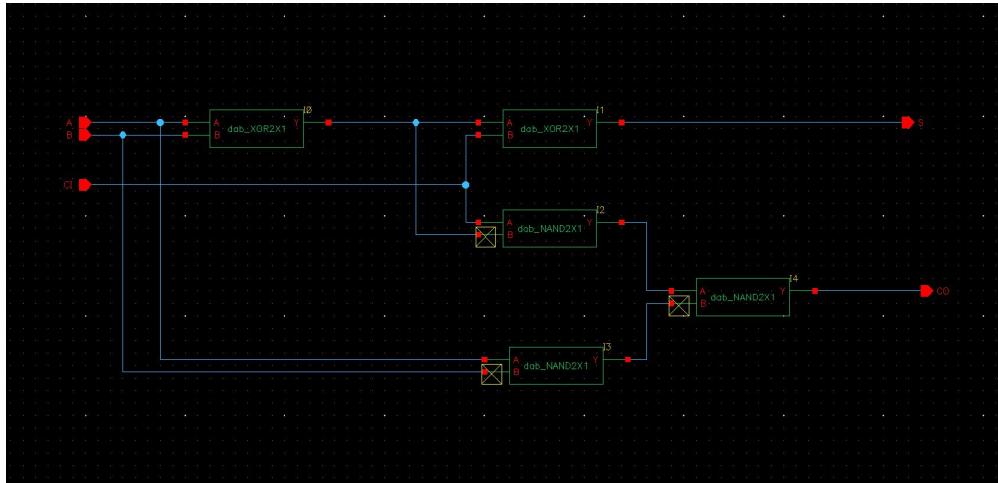


Fig5: schematic of Full adder

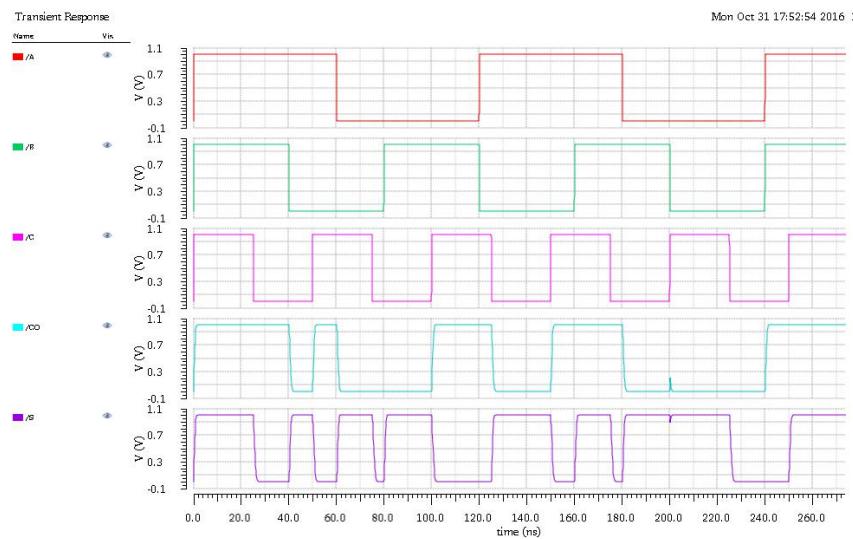


Fig 6: waveform of full adder

The waveform shows, whenever the inputs are high they provide a carry. For instance when all the inputs are 1 the carry is shown high.

- Adder 16:

16-bit carry select adder is designed using Full adder and TMUX. The schematic is made by 4 stages with 4-bit input on each stage. When it comes to layout, the design is generated in 4 rows where the Full adder and TMUX are placed respectively. When it comes to hierarchical design of more than one row, 16 track spacing is maintained between 2 rows and 8 track spacing is maintained between the two rows. This is done to provide a path for the router to route across different rows and also to avoid short between two metals. Once, all the conditions are satisfied the pins are placed and they are routed using router. Router again, does a great job in connect the pins respectively using the shortest path. However, at points it shorts the same metal, which has to be moved manually by figuring it in DRC. Once, DRC and LVS is cleared a single block hierarchical design of 16-bit adder is created which consists of instances from Full adder and TMUX.



Fig7: schematic of adder 16

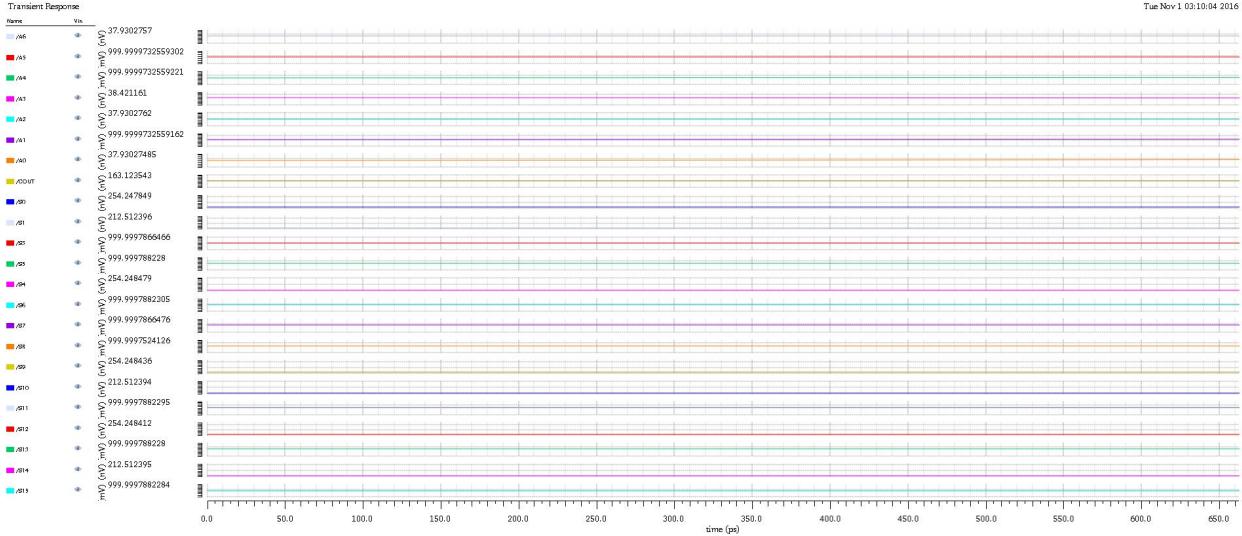


Fig8: waveform of adder 16

- BSC:

Boundary scan cell is generally used to test the circuit with respect to its logical function. The reason we are using BSC is to test the ADD16 we have designed. BSC was introduced by joint test action group in 1980 to test the interconnections of a circuit board. Generally, BSC tests the circuit without the use of any physical test probes. The way boundary scan cells work is by forcing the data to the pins and also by capturing the pins. The captured data is serially shifted out and these values are compared externally with expected values. The forced data is serially shifted in; all of these are controlled using scan path. Direct access to the pin nets reduces the need of large number of test vectors.

Boundary scan cell works in four modes:

- Normal Mode:

It passes the input directly to the output, for this condition the MODE has to be 0.

➤ Scan Mode:

It takes the input from the SIN and stores it on the register QA and then serially sends to SOUT. For this mode to occur the SDR clock should be 1 and CDR1 has to be 1, this will make the serial input to reach register QA and now the UPDR and MODE control should be high to show the SIN values serially on the SOUT.

➤ Capture Mode:

Capture mode captures the input value from the IN and stores it in the register QA.

For this mode, the SDR has 0 and CDR1 should be maintained high.

➤ Update Data:

Update mode updates the value from register QA to register QB and shows it serially on the OUT.

The Boundary scan cell is generated using TMUX and DFF. The layout is again done using Routing method as it is a hierarchical design. Single PR boundary is generated to show them as a single block unit. The Router does a good job in routing these pins with the shortest possible length. Not many difficulties were faced during the layout.

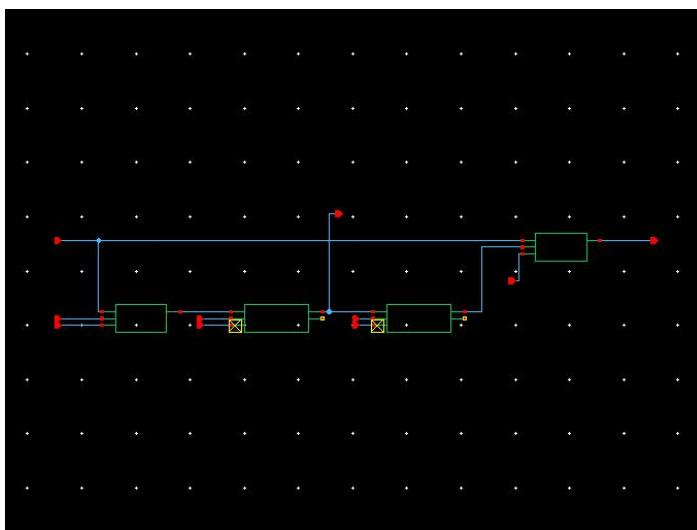


Fig9: schematic of BSC

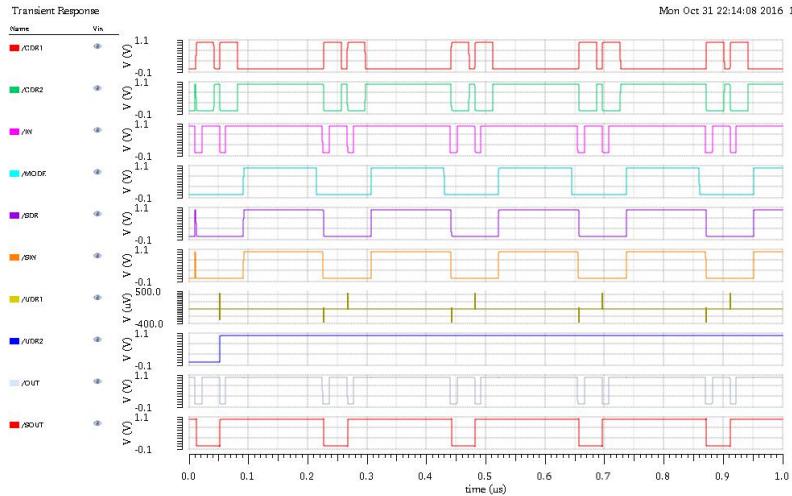


Fig 10: waveform of BSC

From the above waveform it is observed that when mode is 0 “out” follows “in” which explains Normal mode. for scan mode When sdr is low and cdr1 is high and cdr2 is low sout will follow “in”. For update mode when a clock is given to UDR pin the value stored in register Qb will be shifted to output.

For capture mode when sdr is high and a single scan clock is given to sdr and udr the “out” will follow “sin”.

BSR:

The Boundary scan register is a block which consists of 50 instances of Boundary scan cell.

The boundary scan register is a hierarchical design which is done in similar fashion but placed in 10 rows and the routing is done to make the interconnection between the pins. The router again does excellent job and reduces the complexity. The BSC is placed in 10 ROWS with 5 cells on each row, 16 track distance between the rows and 8 track between the PR boundary and top/bottom of the cell is maintained.

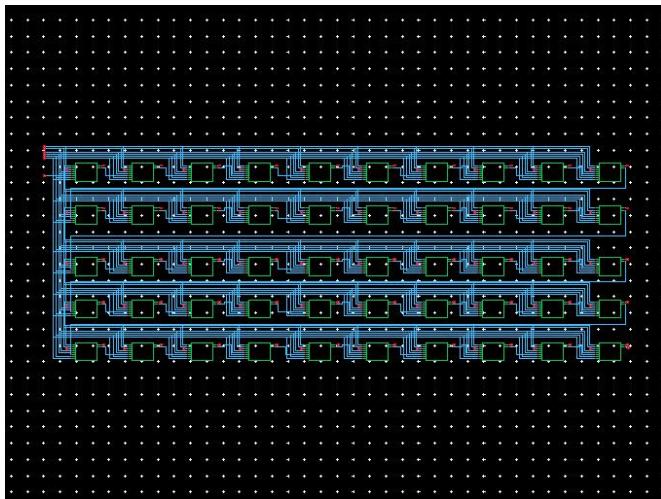


Fig 11: schematic of BSR

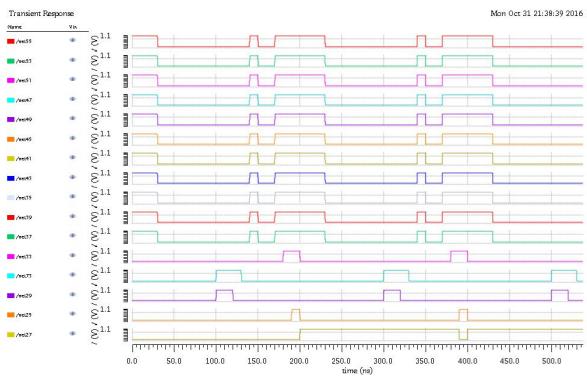


Fig 12: waveform of BSR

BSSUM:

The BSSUM is designed to test the 16 bit adder. The BSR and 16 Bit adder are connected together. The first 32 inputs of the BSR is connected to TIEHI and TIELO. The output of the BSR is connected to 32 inputs of the Full adder and the output of the full adder is connected to the input pins 33-50 of the BSR. The Test bench is written for all possible conditions and the output is tested respectively.

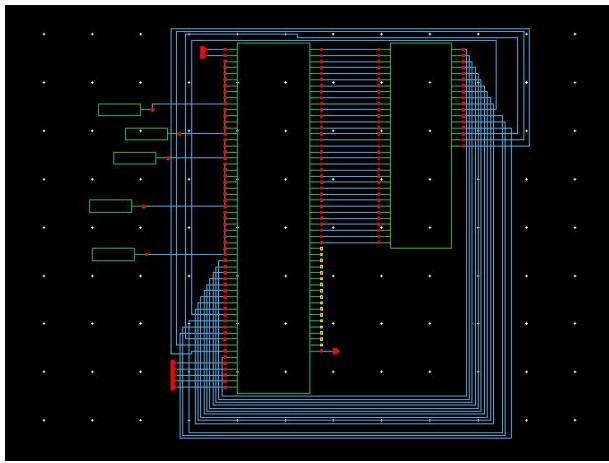


Fig 13:schematic of BSSUM

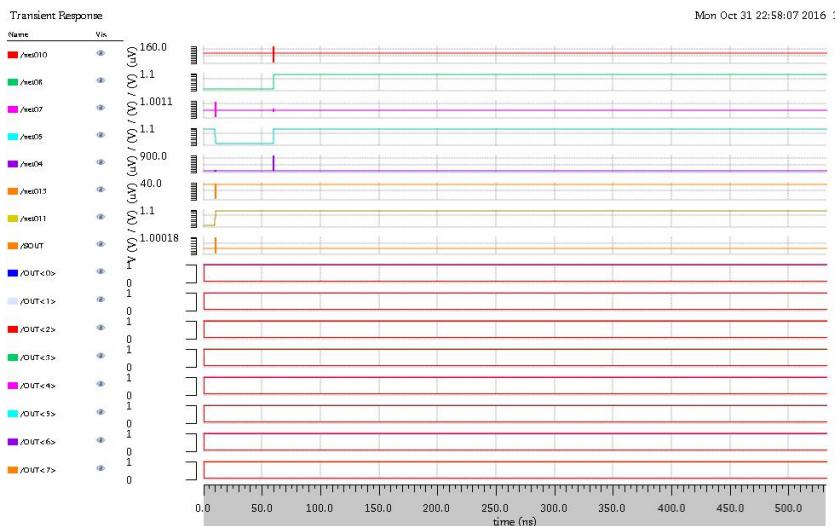


Fig14: waveform of BSSUM

When sdr is changed from 0 to 1 the and cdr is clocked 50 times the value stored in IN[49] will reach BSC cell 0.so The sum is obtained in series from sout pin.

BSTEST:

The BSSUM design is mounted on the quarter pad and once this is done the chip is finally designed successfully and is used in other logic functions to implement the addition.

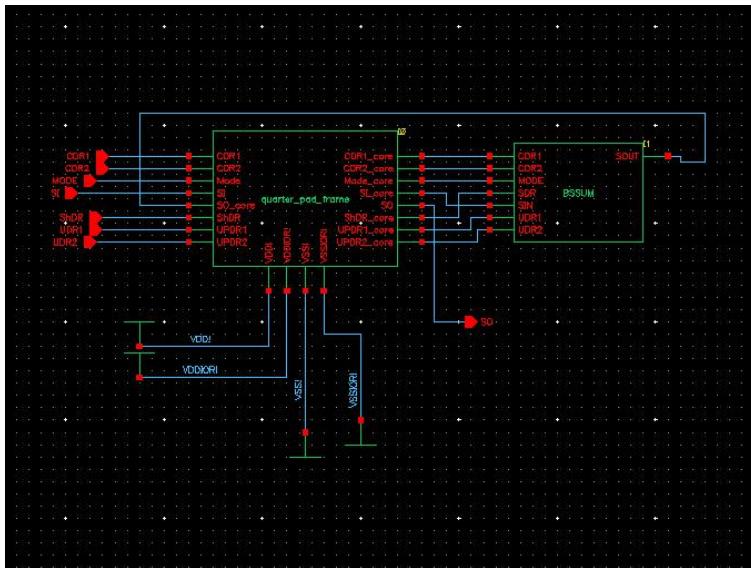


Fig 15: schematic of BSTEST

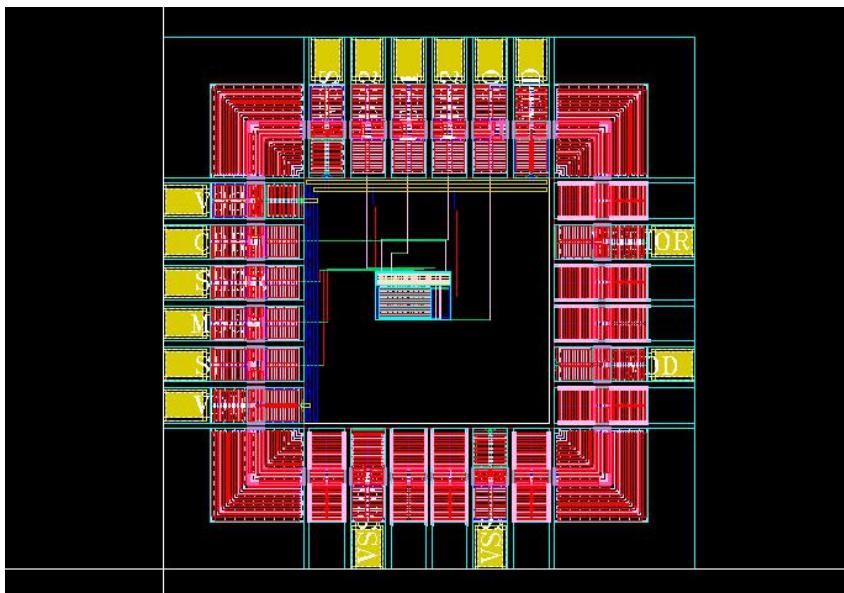


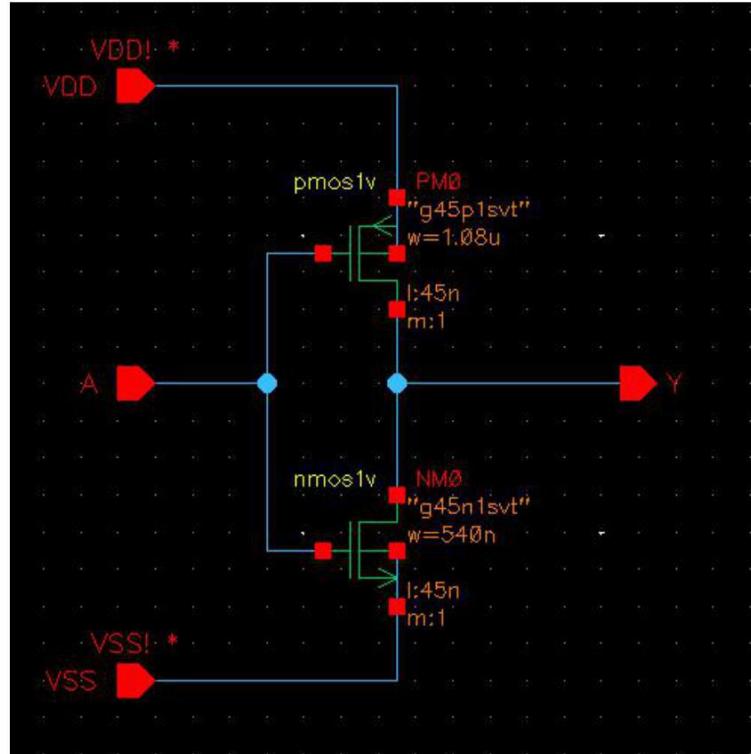
Fig 16: Layout of BSTEST

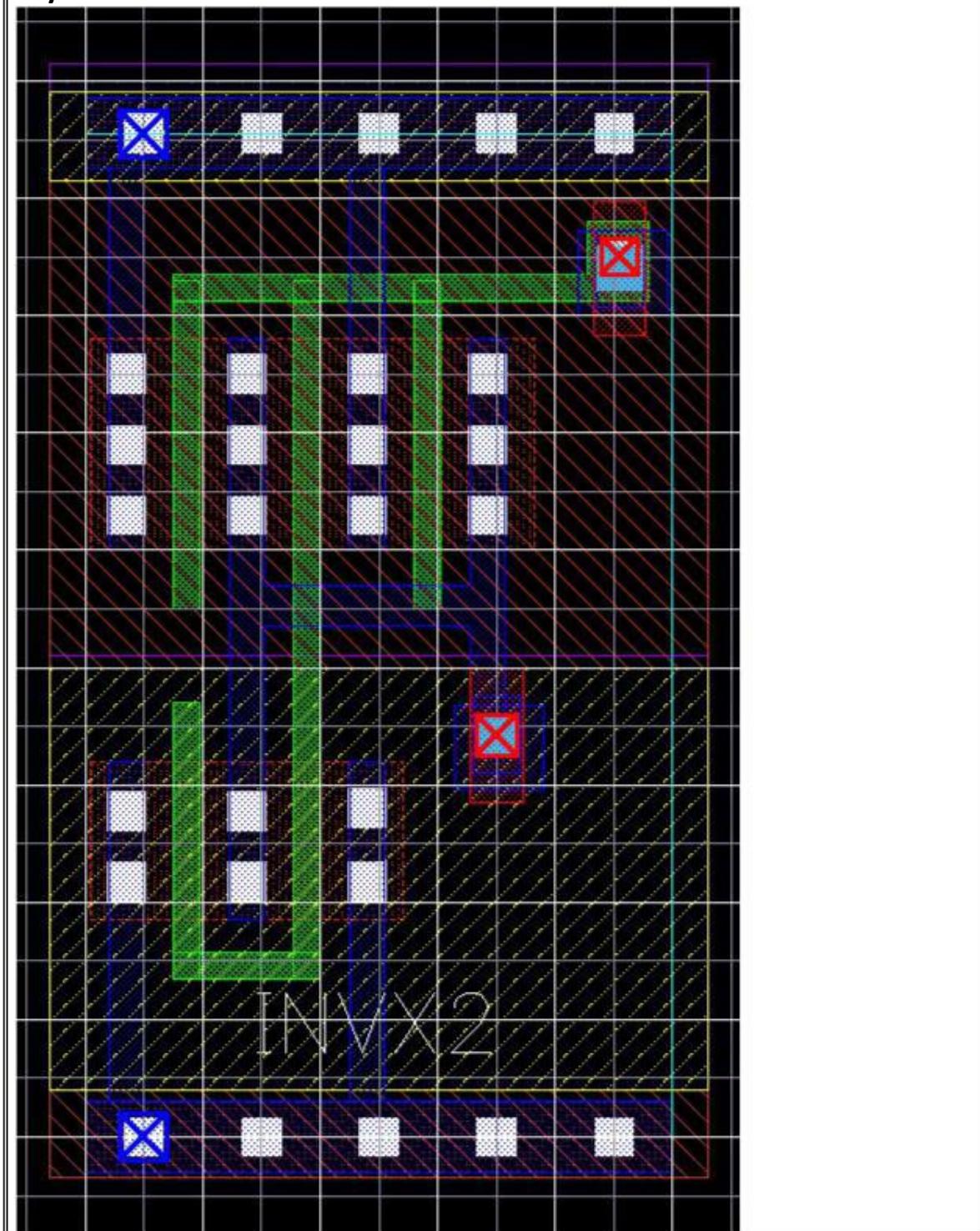
Conclusion:

The project 2 is very helpful in understanding the basics mechanism in manufacturing of cell. The path from the design of unit cell till the packing of circuit into the quarter pad provided a path of knowledge. The few techniques like Routing, placement and Boundary scan cell provided good knowledge in understanding few

things like how the bigger cells are routed by reducing the complexity and then how the cell is tested logically using boundary scan cell. I personally, feel both the project provided the very good understanding of the subject.

Library Name:	dab8730_dab_lib
Cell Name:	dab_INVX2
Function/Truth Table:	
A	Y
0	1
1	0
Propagation Delay:	
1. Signal A → Falling edge	247.9E-12
Signal Y → Rising edge	
2. Signal A → Rising edge	279.5E-12
Signal Y → Falling edge	
Output Rise Time: 310.8E-12	
Output Fall Time: 354.6E-12	
Layout Area: 1.71μm	

Symbol with Port Names:**Schematic:**

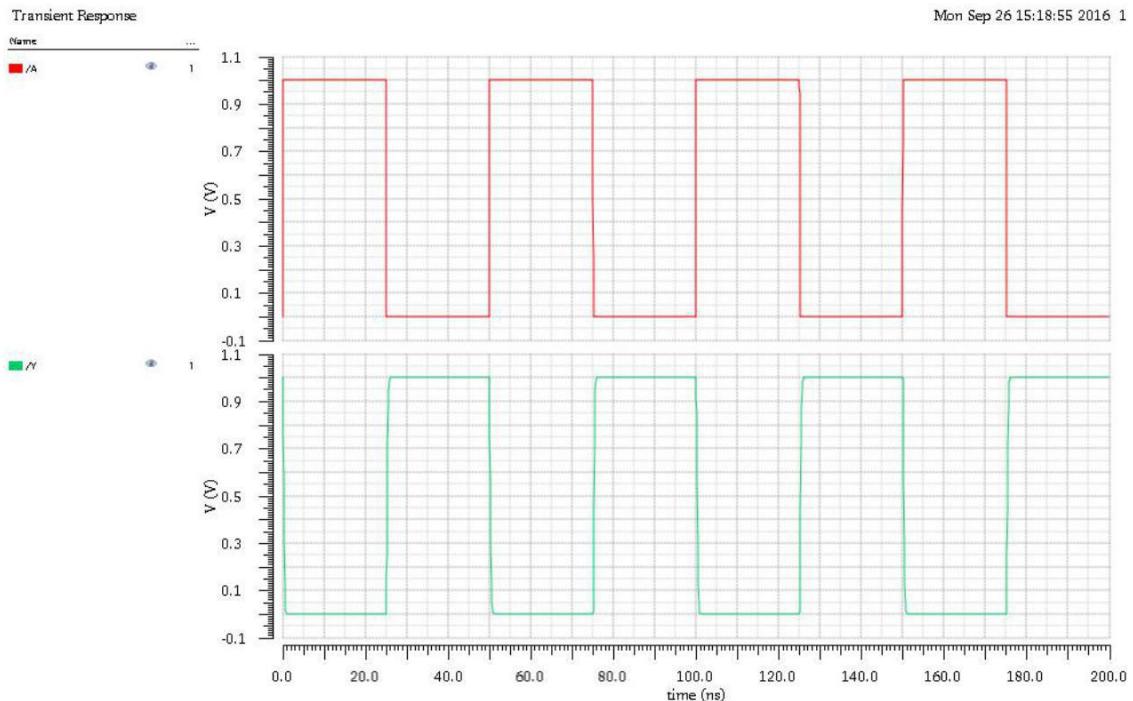
Layout:

Verilog Model:

```
//Verilog HDL for "dab8730_dab.lib", "dab_INVX2" "functional"

module dab_INVX2 ( Y, A, .VDD(\VDD!), .VSS(\VSS!) );
    input A;
    output Y;
    input
    `ifdef INCA
        (* integer inh_conn_prop_name = "VDD";
           integer inh_conn_def_value = "cds_globals.\VDD!" ; *)
    `endif
    \VDD!;
    input
    `ifdef INCA
        (* integer inh_conn_prop_name = "VSS";
           integer inh_conn_def_value = "cds_globals.\VSS!" ; *)
    `endif
    \VSS!;
    not U1 (Y, A);
endmodule
```

Functional Simulation Waveforms:



Comments/Notes:

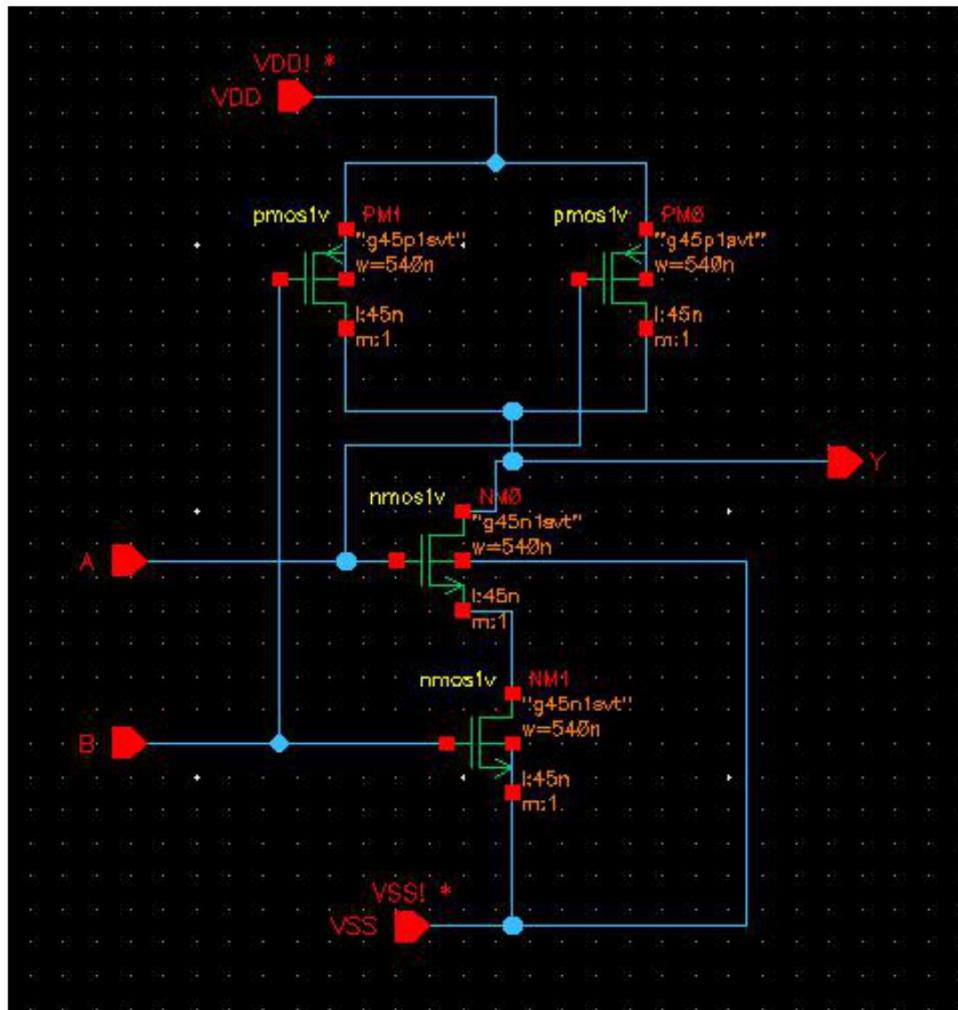
**Reference inverter : nMOS Wn/L= 270nm/45nm
pMOS wp/l= 540nm/45nm**

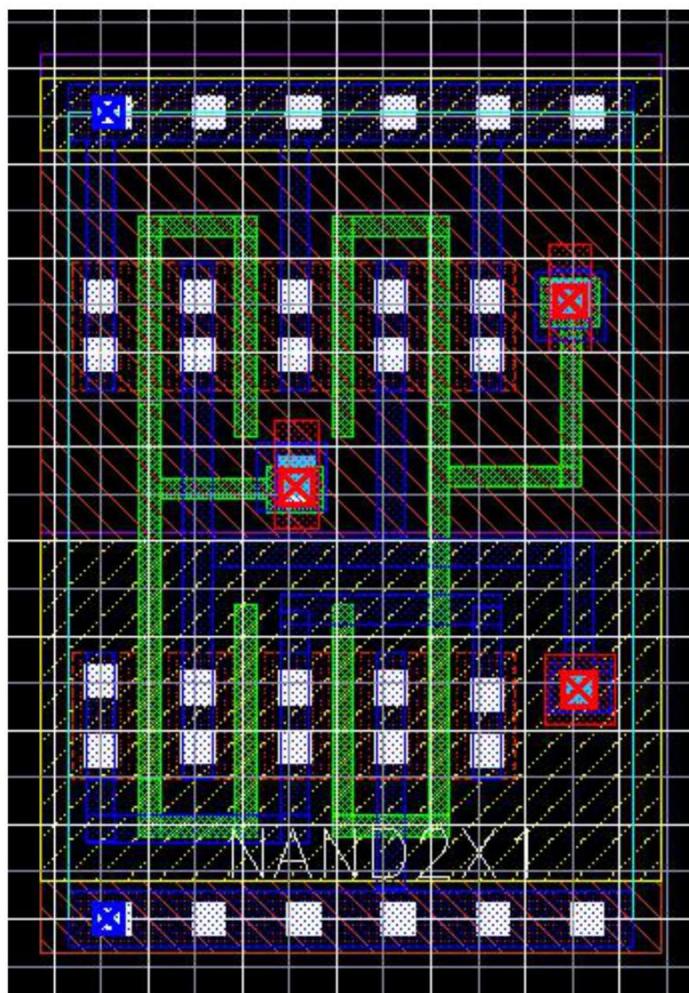
**To maintain the equivalent resistance between pMOS and nMOS,
scaling is done**

**After scaling: nMOS Wn/L= 540nm/45nm
pMOS wp/l= 1080nm/45nm.**

Tpd= 332.4ps

Library Name:	dab8730_dab_lib	
Cell Name:	dab_NAND2X1	
Function/Truth Table:		
A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0
Propagation Delay:		
1. Signal A → Falling edge Signal Y → Rising edge	430.8E-12	
2. Signal A → Rising edge Signal Y → Falling edge	551.7E-12	
3. Signal B → Falling edge Signal Y → Rising edge	431.7E-12	
4. Signal B → Rising edge Signal Y → Falling edge	551.7E-12	
Output Rise Time: 562.2E-12		
Output Fall Time: 709.3E-12		
Layout Area: 2.052 μm		

Symbol with Port Names:**Schematic:**

Layout:

Verilog Model:

```
//Verilog HDL for "dab8730_dab.lib", "dab_NAND2X1" "functional"

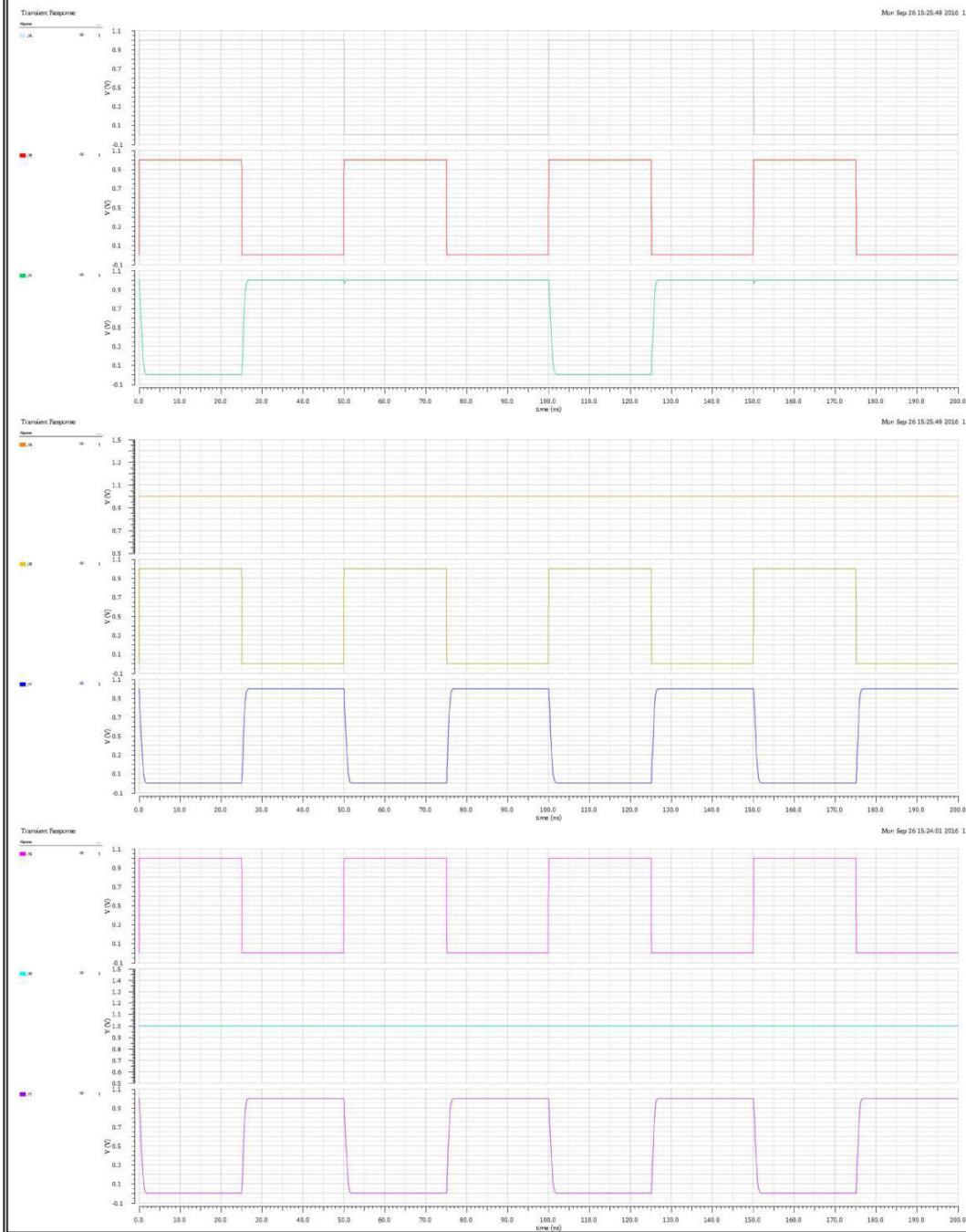
module dab_NAND2X1 ( Y, A, B, .VDD(\VDD!), .VSS(\VSS!) );

    input A;
    output Y;
    input
`ifdef INCA
    (* integer inh_com_prop_name = "VDD";
       integer inh_com_def_value = "cds_globals.\VDD! " ; *)
`endif
    \VDD!;
    input
`ifdef INCA
    (* integer inh_com_prop_name = "VSS";
       integer inh_com_def_value = "cds_globals.\VSS! " ; *)
`endif
    \VSS!;
    input B;

nand U1 (Y, A, B);

endmodule
```

Functional Simulation Waveforms:



Comments/Notes:

With respect to the reference sizes, the nMOS width is doubled as they are in series with each other to maintain the equivalent resistance R across the layout.

pMOS W_p/L=540nm/45nm

nMOS W_n/L= 540nm/45nm

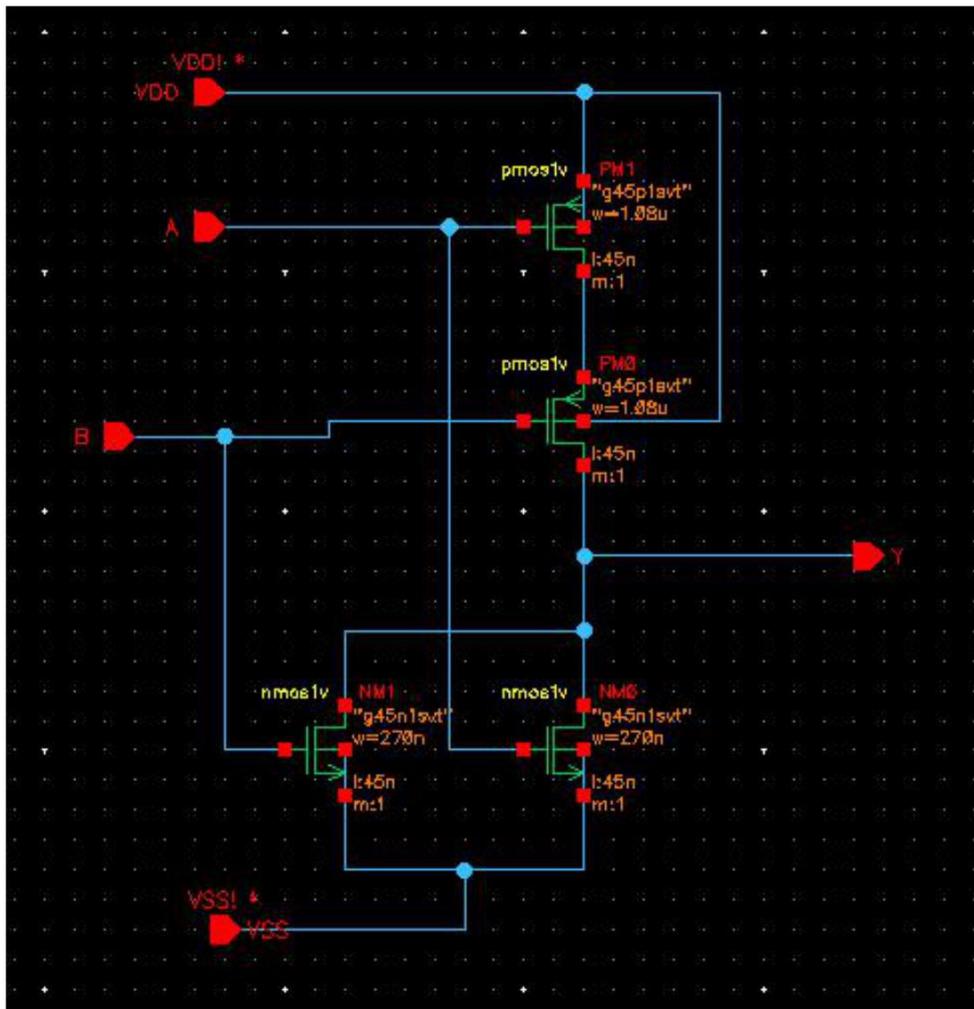
$$T_{pd} = (431.7 + 551.7) / 2 = 491.7 \text{ ps}$$

Library Name:	dab8730_dab_lib	
Cell Name:	dab_NOR2X1	
Function/Truth Table:		
A	B	Y
0	0	1
0	1	0
1	0	0
1	1	0
Propagation Delay:		
1. Signal A → Falling edge Signal Y → Rising edge	560.2E-12	
2. Signal A → Rising edge Signal Y → Falling edge	486.2E-12	
3. Signal B → Falling edge Signal Y → Rising edge	484.5E-12	
4. Signal B → Rising edge Signal Y → Falling edge	548.2E-12	
Output Rise Time: 624.1E-12		
Output Fall Time: 741.2E-12		
Layout Area: $1.6 \times 1.71 = 2.736 \mu\text{m}$		

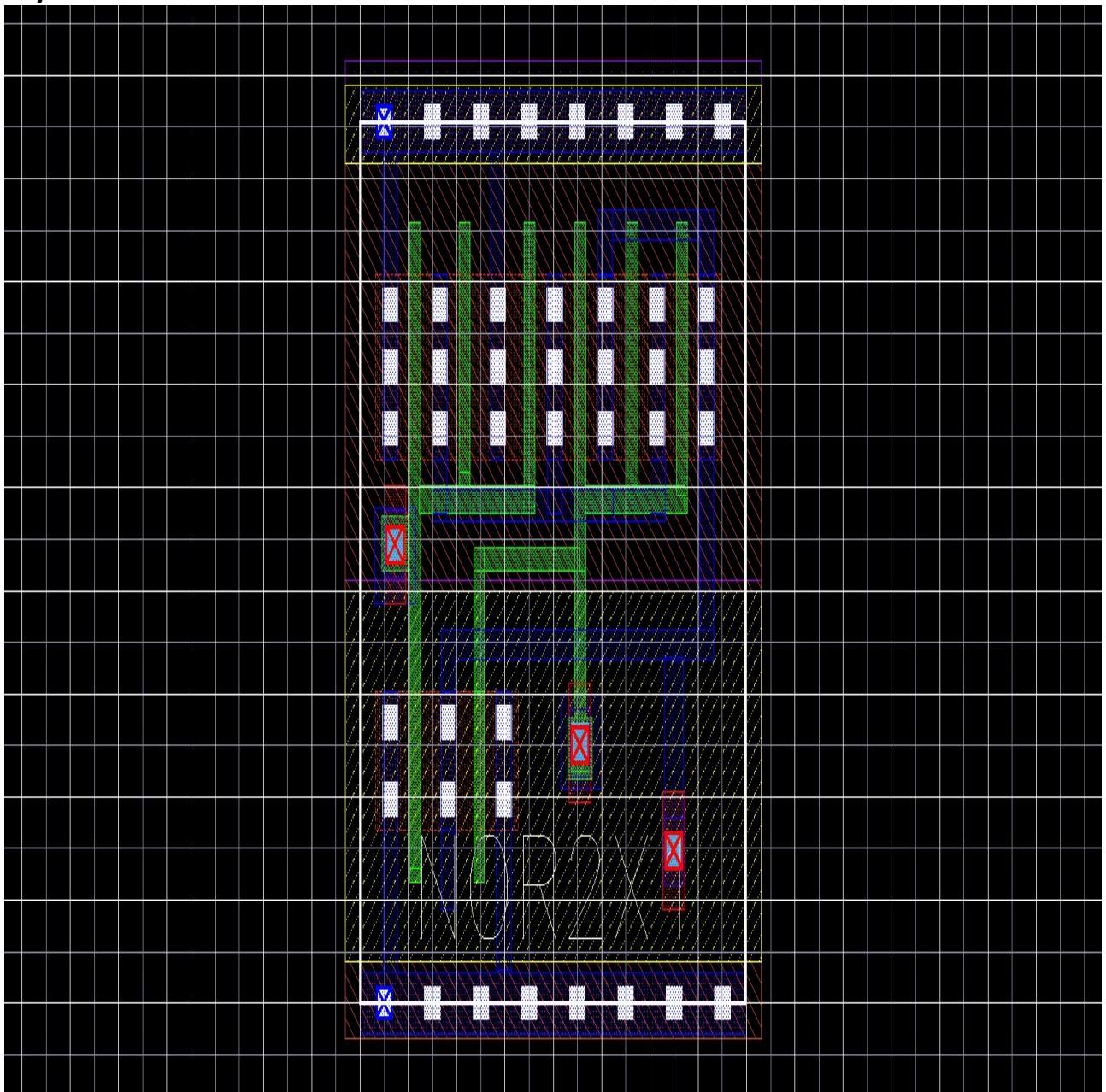
Symbol with Port Names:



Schematic:



Layout:



Verilog Model:

```
//Verilog HDL for "dab8730_dab.lib", "dab_NOR2x1" "functional"

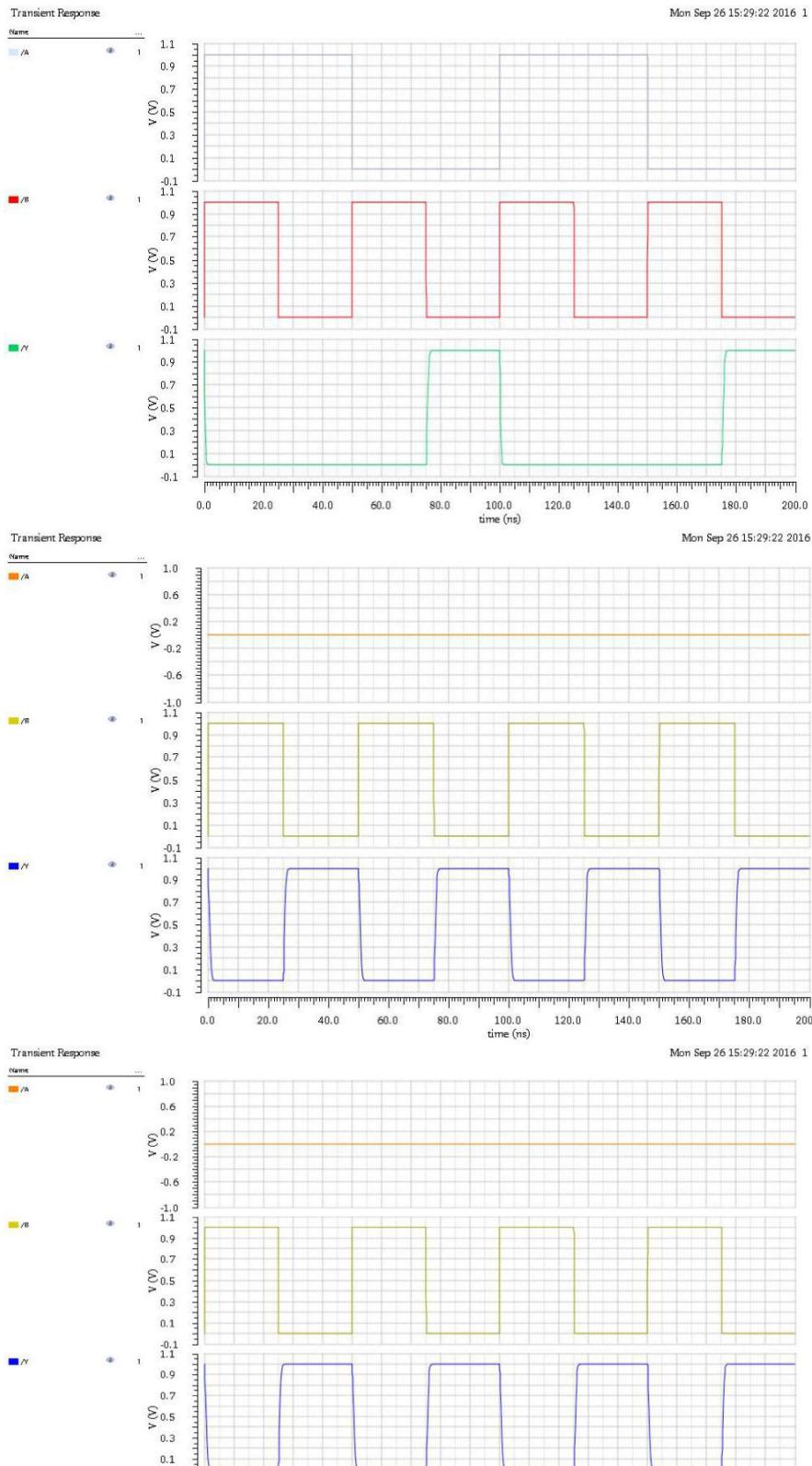
module dab_NOR2x1 ( .VSS(\VSS! ), Y, A, B, .VDD(\VDD! ) );

    input A;
    output Y;
    input
`ifdef INCA
    (* integer inh_comm_prop_name = "VDD";
       integer inh_comm_def_value = "cds_globals.\VDD! "; *)
`endif
    \VDD! ;
    output
`ifdef INCA
    (* integer inh_comm_prop_name = "VSS";
       integer inh_comm_def_value = "cds_globals.\VSS! "; *)
`endif
    \VSS! ;
    input B;

nor U1 (Y, A, B);

endmodule
```

Functional Simulation Waveforms:



Comments/Notes:

**With respect to the reference sizes, pMOS size is doubled to
1080nm/45nm to keep the equivalent R across the nMOS and pMOS.**

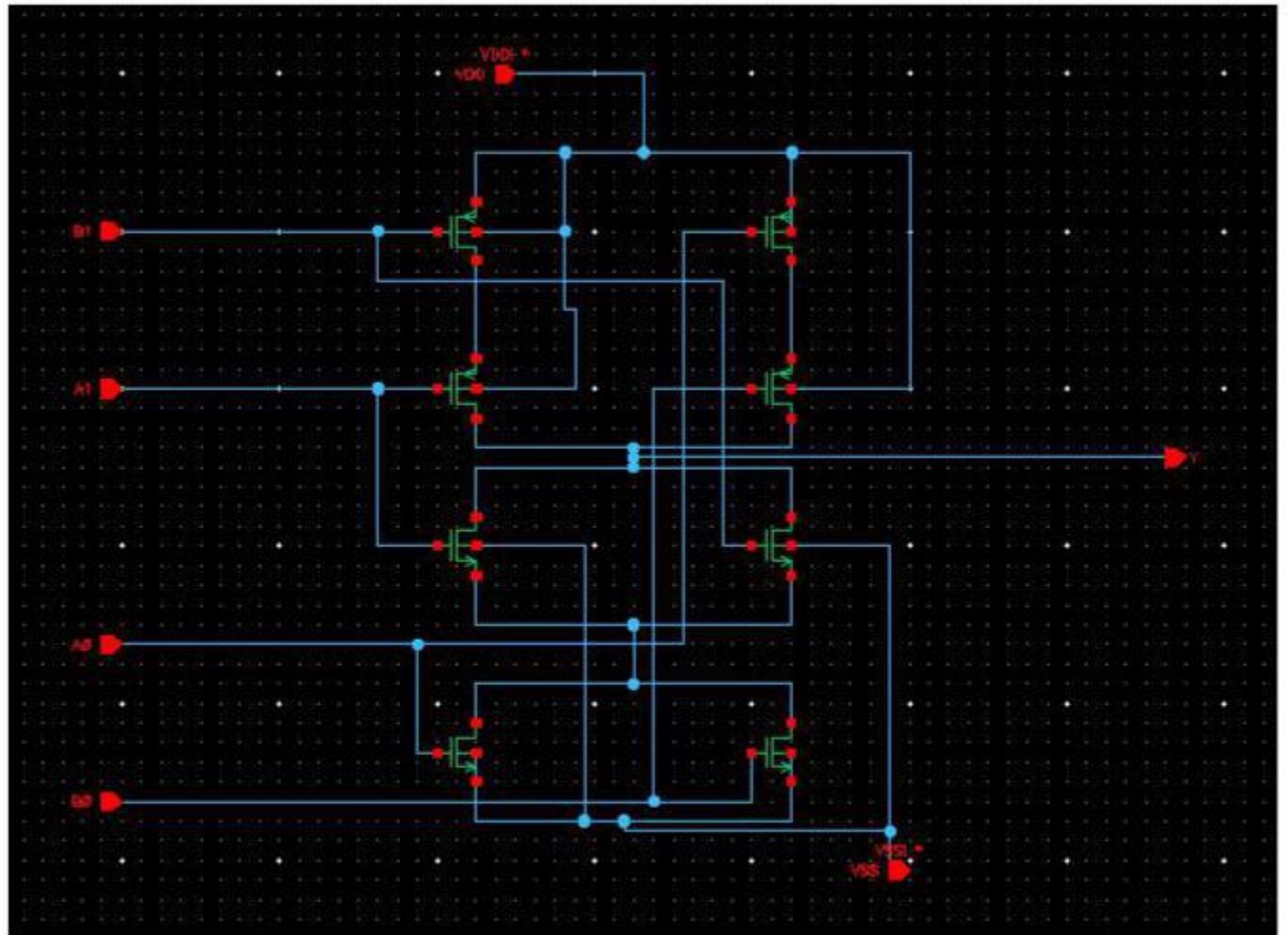
pMOS W_p/I= 1080nm/45nm

nMOS W_n/I= 270nm/45nm

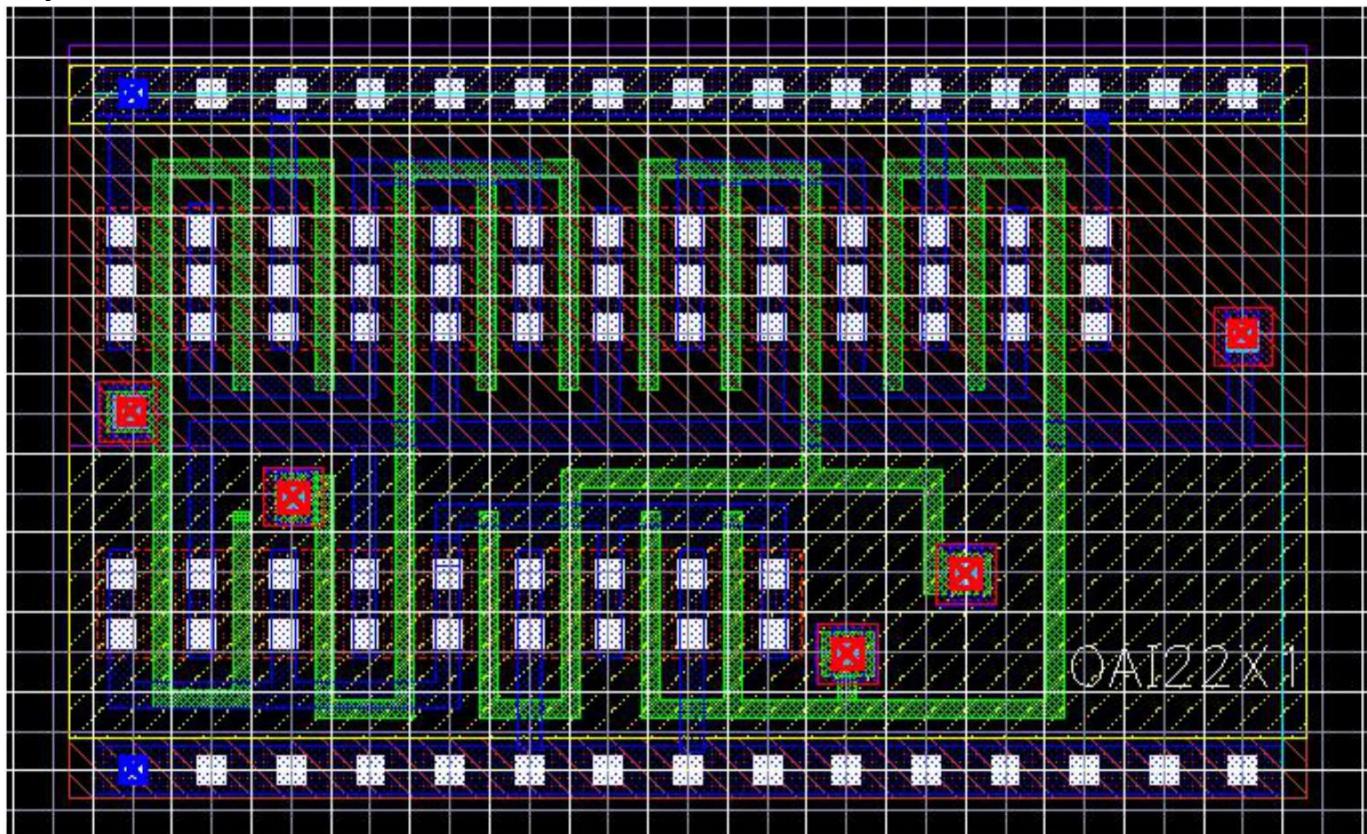
Library Name:	dab8730_dab_lib			
Cell Name:	dab_OAI22X1			
Function/Truth Table:				
A0	A1	B0	B1	Y
0	0	0	0	1
0	0	0	1	1
0	0	1	0	1
0	0	1	1	1
0	0	0	0	1
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	1	0	0	1
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	1
1	1	0	1	0
1	1	1	0	0
1	1	1	1	0
Propagation Delay:				
1. Signal A0 -> Falling Signal Y-> Rising				451.9E-12

2. Signal A0 -> Rising Signal Y-> Falling	643.3E-12
3. Signal A1 -> Falling Signal Y-> Rising	449.4E-12
4. Signal A1 -> Rising Signal Y->Falling	462.8E-12
5. Signal B0 -> Falling Signal Y-> Rising	447.9E-12
6. Signal B0 -> Rising Signal Y-> Falling	632.7E-12
7. Signal B1 -> Falling Signal Y-> Rising	450.6E-12
8. Signal B1 -> Rising Signal Y->Falling	627.1E-12

Output Rise Time: 563.6E-12**Output Fall Time: 406.1E-12****Layout Area: Cell Width*Cell Height = $3.1\mu\text{m} * 1.71 \mu\text{m} = 5.13\mu\text{m}$** **Symbol with Port Names:**

Schematic:

Layout:



Verilog Model:

```
//Verilog HDL for "dab8730_dab_lib", "dab_OAI22X1" "functional"

module dab_OAI22X1 ( Y, A0, A1, B0, B1, .VDD(\VDD!), .VSS(\VSS!) );
    input A0;
    output Y;
    input
`ifdef INCA
    (* integer inh_conn_prop_name = "VDD";
       integer inh_conn_def_value = "cds_global.s.\VDD! " ; *)
`endif
    \VDD!;
    input B0;
    input B1;
    input A1;
    input
`ifdef INCA
    (* integer inh_conn_prop_name = "VSS";
       integer inh_conn_def_value = "cds_global.s.\VSS! " ; *)
`endif
    \VSS!;
    assign y= ~ ((A1|B1)&(A0|B0));
endmodule
```

Functional Simulation Waveforms:



Comments/Notes:

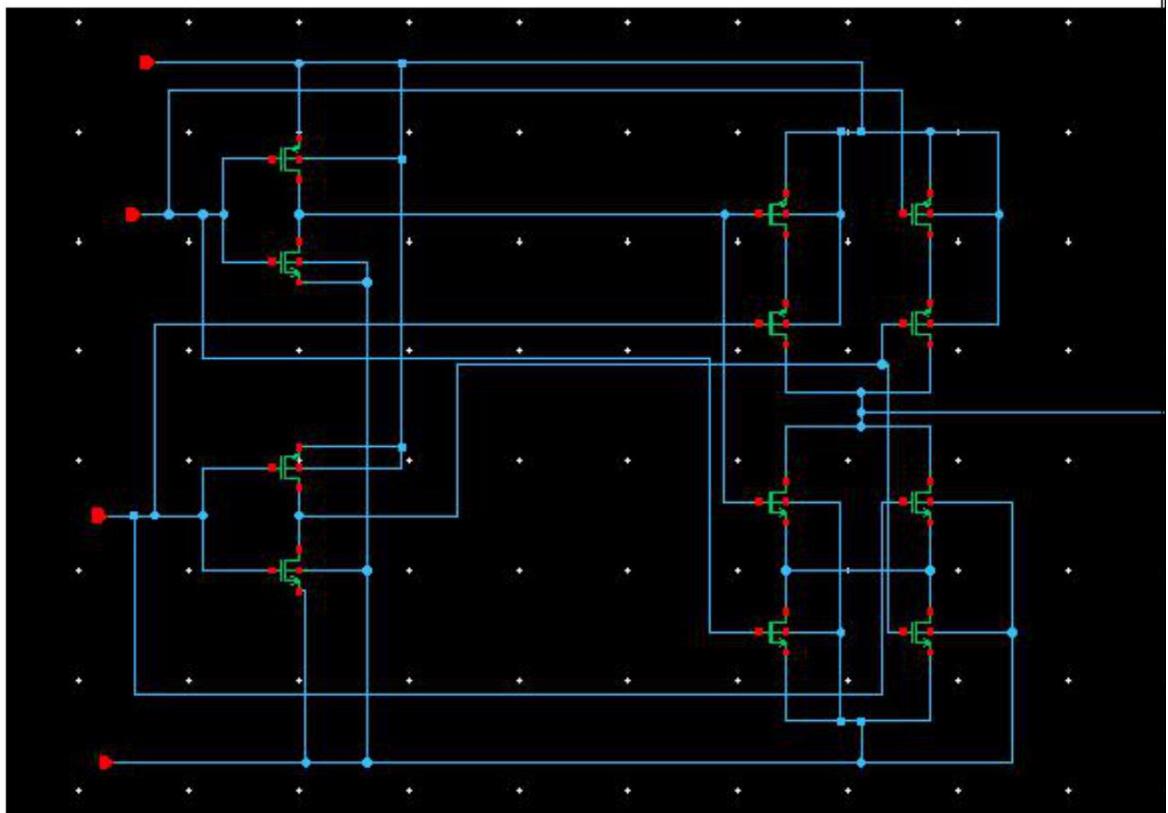
The width of nMOS and pMOS are doubled with respect to the reference size as, they are in series combination the equivalent resistance across them has to be same.

pMOS W_p/L= 540nm/45nm

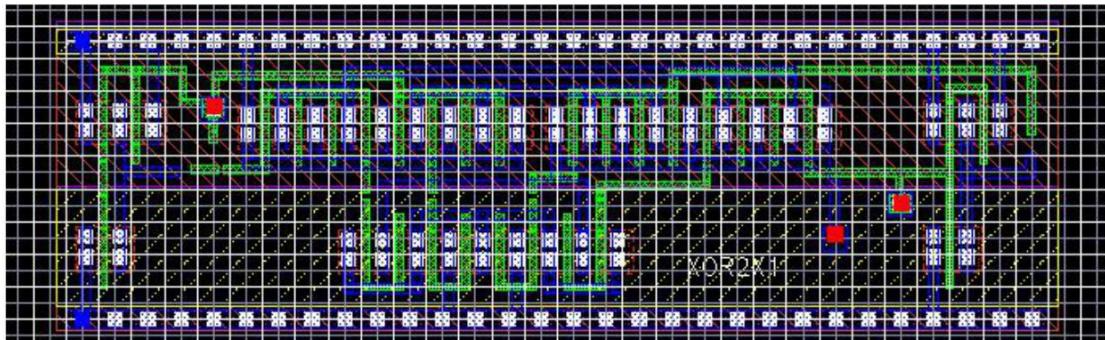
nMOS W_n/L=1080nm/45nm

$$T_{pd} = (451.9\text{ps} + 643.3\text{ns})/2 = 547.6\text{ns}$$

Library Name:	dab8730_dab_lib	
Cell Name:	dab_XOR2X1	
Function/Truth Table:		
A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0
Propagation Delay:		
1. Signal A → Falling edge Signal Y → Rising edge	517E-12	
2. Signal A → Rising edge Signal Y → Falling edge	547.7E-12	
3. Signal B → Falling edge Signal Y → Rising edge	537.7E-12	
4. Signal B → Rising edge Signal Y → Falling edge	513.4E-12	
Output Rise Time: 621.5E-12		
Output Fall Time: 664.6E-12		
Layout Area: 10.26 μm		

Symbol with Port Names:**Schematic:**

Layout:

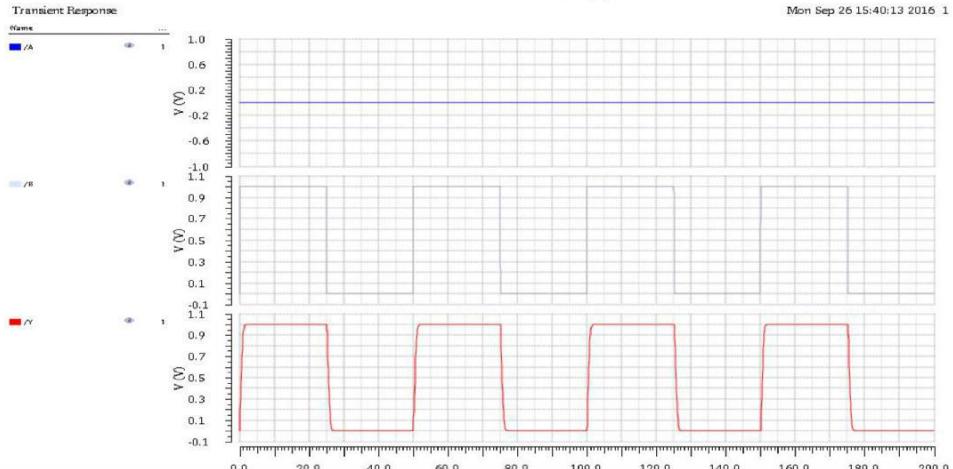
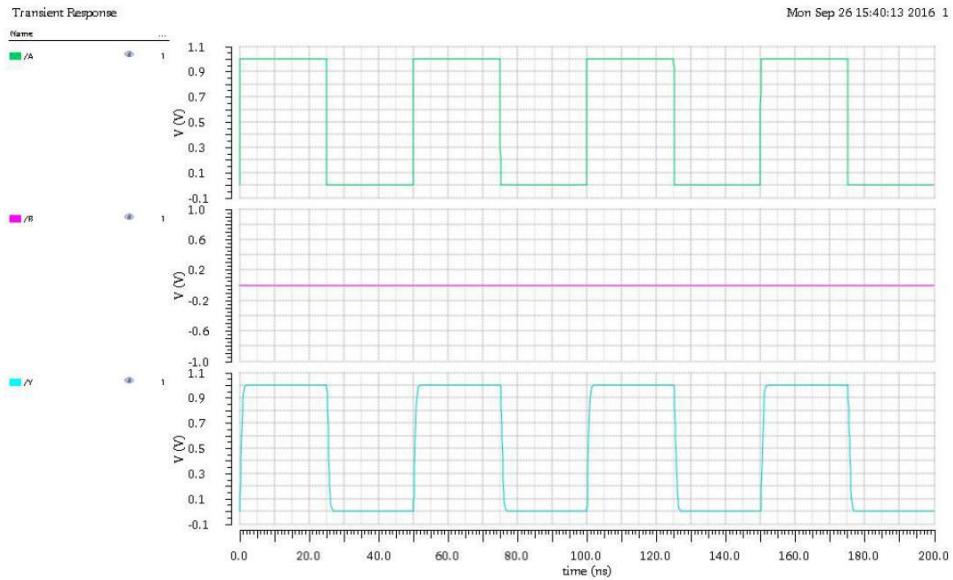
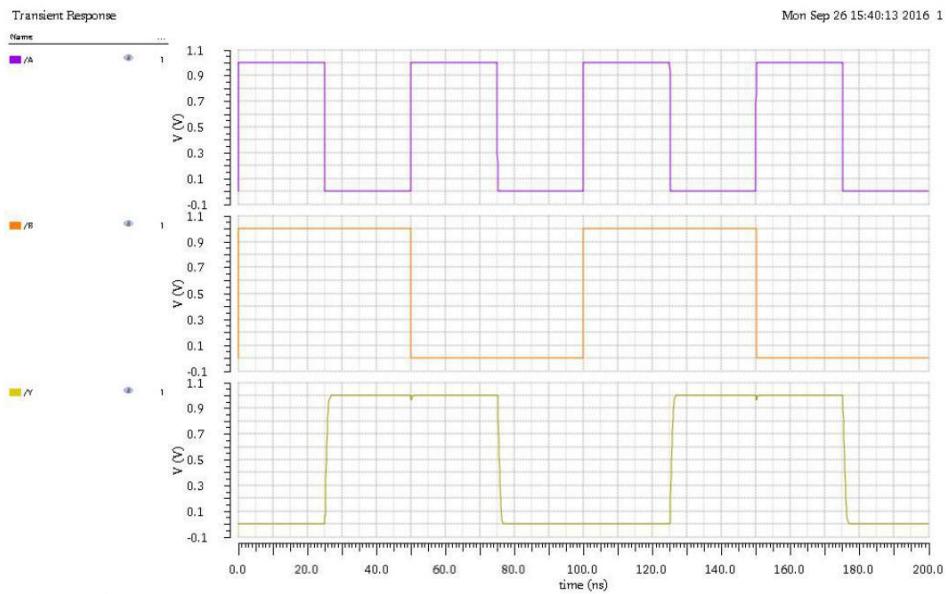


Verilog Model:

```
//Verilog HDL for "dab8730_dab_lib", "dab_XOR2X1" "functional"

module dab_XOR2X1 ( Y, A, B, .VDD(\VDD!), .VSS(\VSS!) );

    input A;
    output Y;
    input
`ifdef INCA
    (* integer inh_conn_prop_name = "VDD";
       integer inh_conn_def_value = "cds_globals.\VDD! "; *)
`endif
    \VDD!;
    input
`ifdef INCA
    (* integer inh_conn_prop_name = "VSS";
       integer inh_conn_def_value = "cds_globals.\VSS! "; *)
`endif
    \VSS!;
    input B;
    xor U1 (Y, A, B);
endmodule
```

Functional Simulation Waveforms:

Comments/Notes:

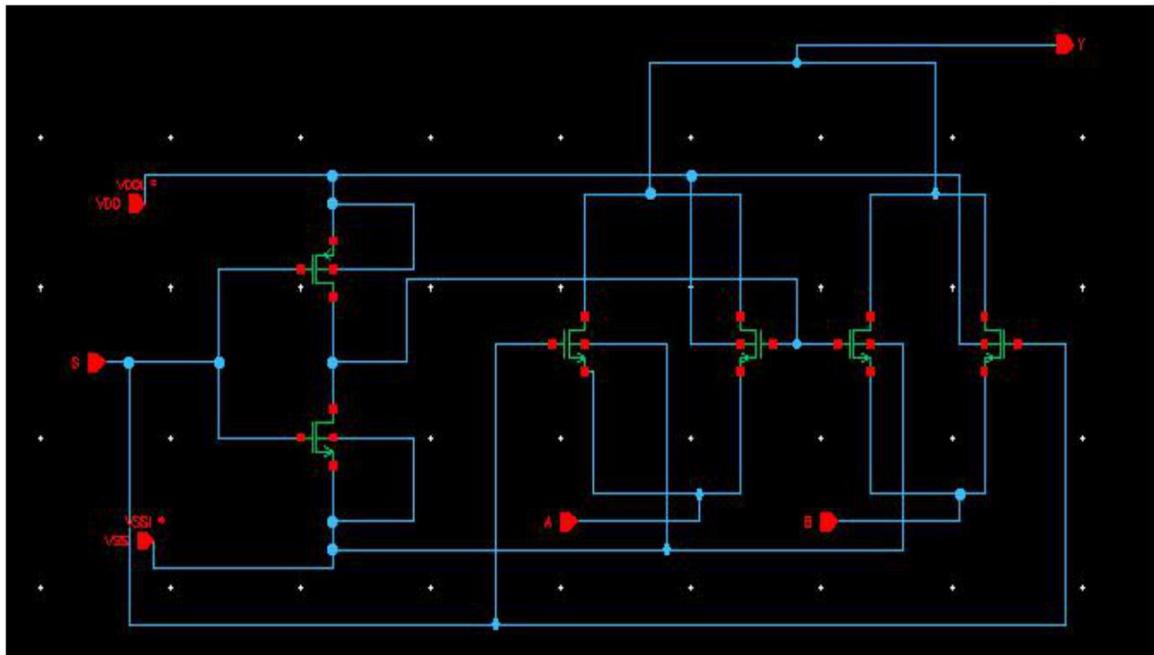
With respect to the reference size, the pMOS and nMOS are doubled to maintain the equivalent R across them. Since they are in series the resistance is r/2 which increases the width over x2.

nMOS Wn/L= 540nm/45nm

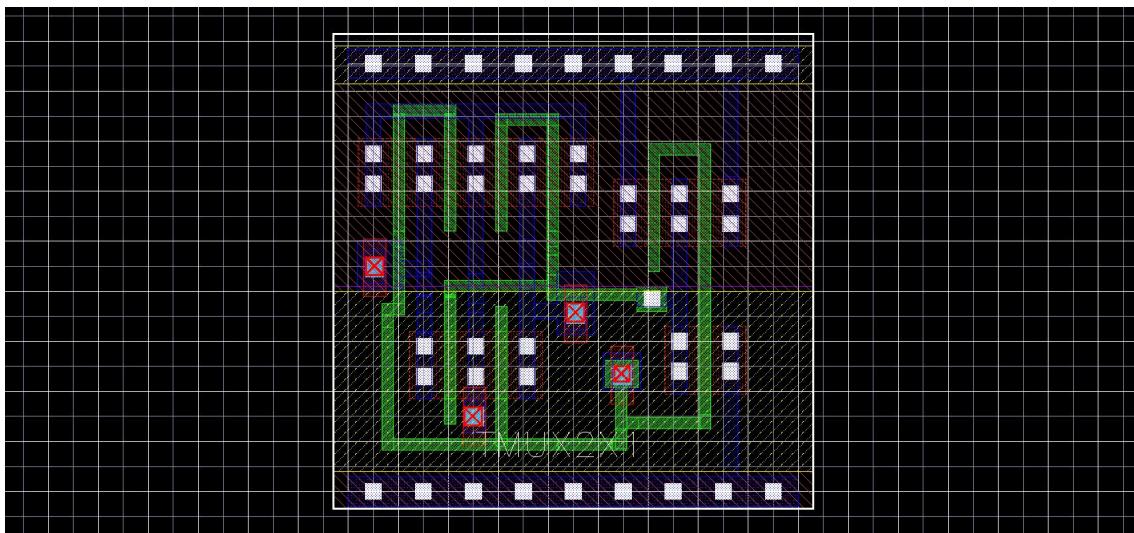
pMOS WI/L = 1080nm/45nm

$$Tpd = (537.7\text{ps} + 547.7\text{ps}) = 542.35\text{ps}$$

Library Name:	dab8730_dab_lib
Cell Name:	dab_TMUX2X1
Function/Truth Table:	
S A B	Y
0 0 X	0
0 1 X	1
1 X 0	0
1 X 1	1
Propagation Delay:	
1. Signal S → Falling edge Signal Y → Rising edge	25.3E-9
2. Signal S → Rising edge Signal Y → Falling edge	24.3E-9
3. Signal A → Falling edge Signal Y → Rising edge	49.75E-9
4. Signal A → Rising edge Signal Y → Falling edge	50.5E-9
5. Signal B → Falling edge Signal Y → Rising edge	25.5E-9
6. Signal B → Rising edge Signal Y → Falling edge	49.7E-9
Output Rise Time: 573.2E-12	
Output Fall Time: 699.5E-12	
Layout Area: 1.8x1.71=3.078 μm	

Symbol with Port Names:**Schematic:**

Layout:

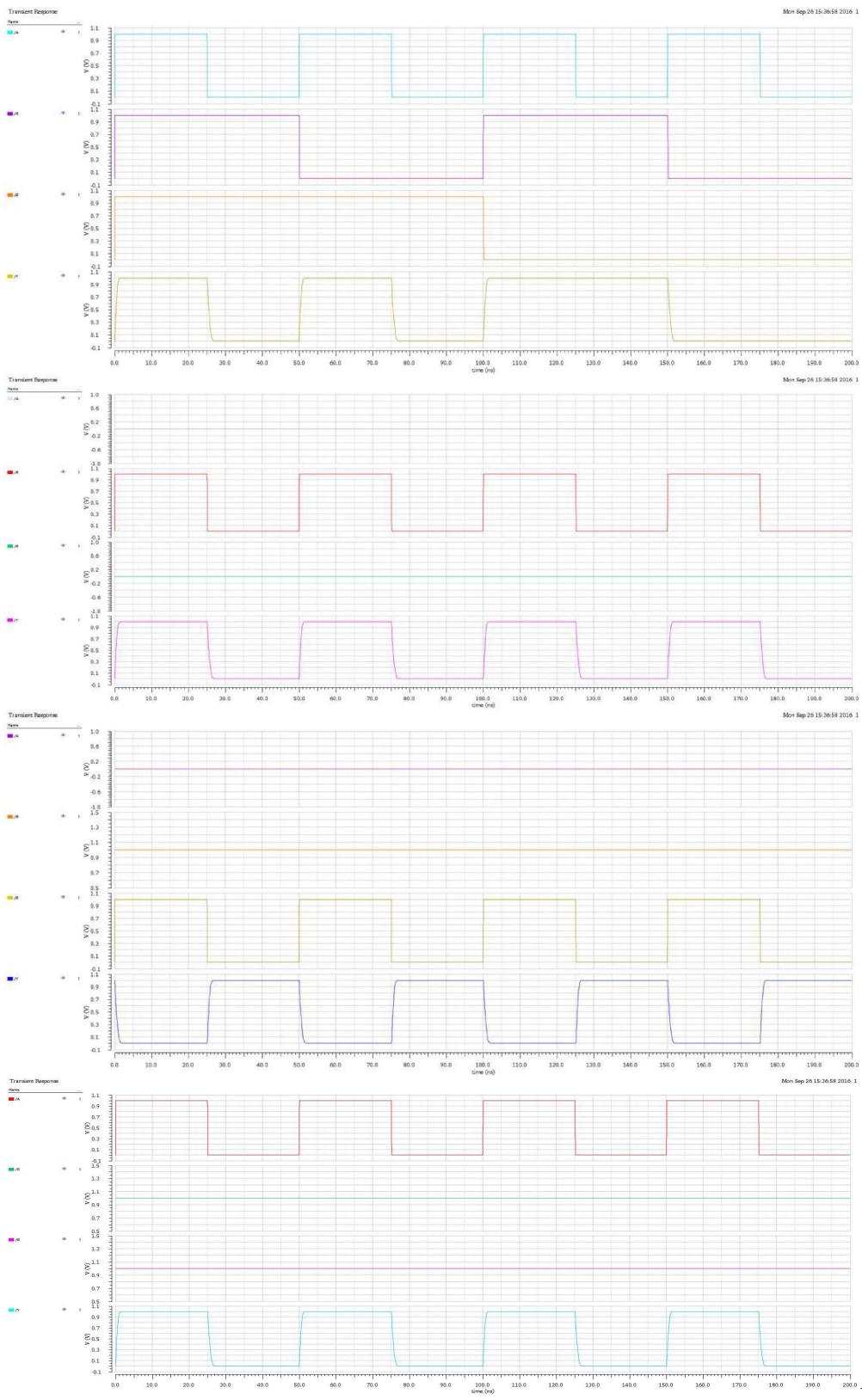


Verilog:

The screenshot shows a Cadence Verilog Editor window with the following code:

```
//Verilog HDL for "dab730_dab_tmux2x1"
module dab_TMUX2X1 ( Y, A, B, S, .VDD(VDD), .VSS(VSS) );
    input A;
    input B;
    output Y;
    input S;
    input VDD;
    input VSS;
    assign Y = ((~S) & A)|(S & B);
endmodule
```

Functional Simulation Waveforms:



Comments/Notes:

Multiplexer acts as a unit where the output is totally controlled with respect to the signals which is fed into the switch. Multiplexer are cheap ,reduces complexity , reduces the usage of wires and also various implementation can be done.

pMOS W_p/L= 540nm/45nm

nMOS W_n/L=270nm/45nm

Library Name:	dab8730_dab_lib			
Cell Name:	dab_FA			
Function/Truth Table:				
A	B	CIN	S	COUT
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1
Propagation Delay:				
A -> S(Rising)	24.5nsec			
A-> S(Falling)	25.8 n sec			
B->S(Rising)	24.58n sec			
B->S (Falling)	25.8 n sec			
CIN->S(Rising)	49.8n sec			
CIN->S(Falling)	50.7n sec			
A -> CO(Rising)	50.7 n sec			
A-> CO(Falling)	49.6 n sec			
B->CO(Rising)	49.5n sec			
B->CO(Falling)	50.8 n sec			
CIN->CO(Rising)	50.6 n sec			
CIN->CO(Falling)	49.6 nsec			

Output Rise Time of S: 570 p sec

Output Fall Time of S: 763.9 p sec

Output Rise Time of CO: 565 p sec

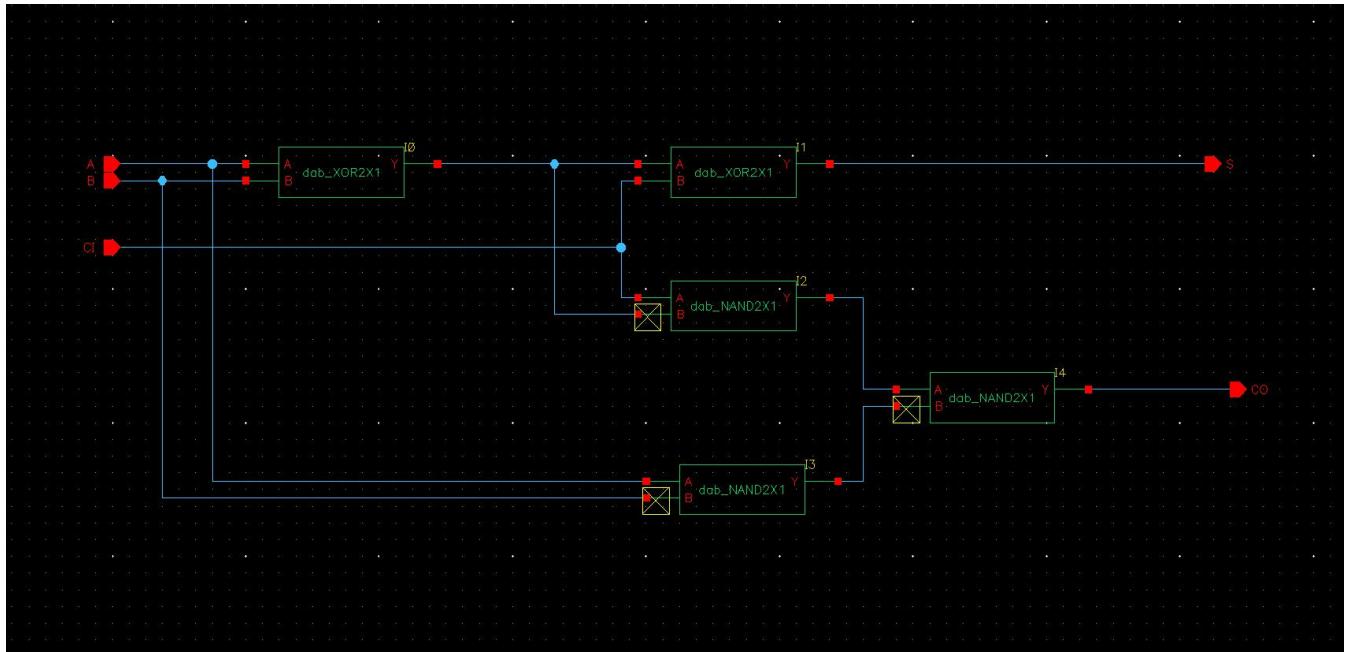
Output Fall Time of CO: 693 p sec

Layout Area: 14nm X 1.71 nm

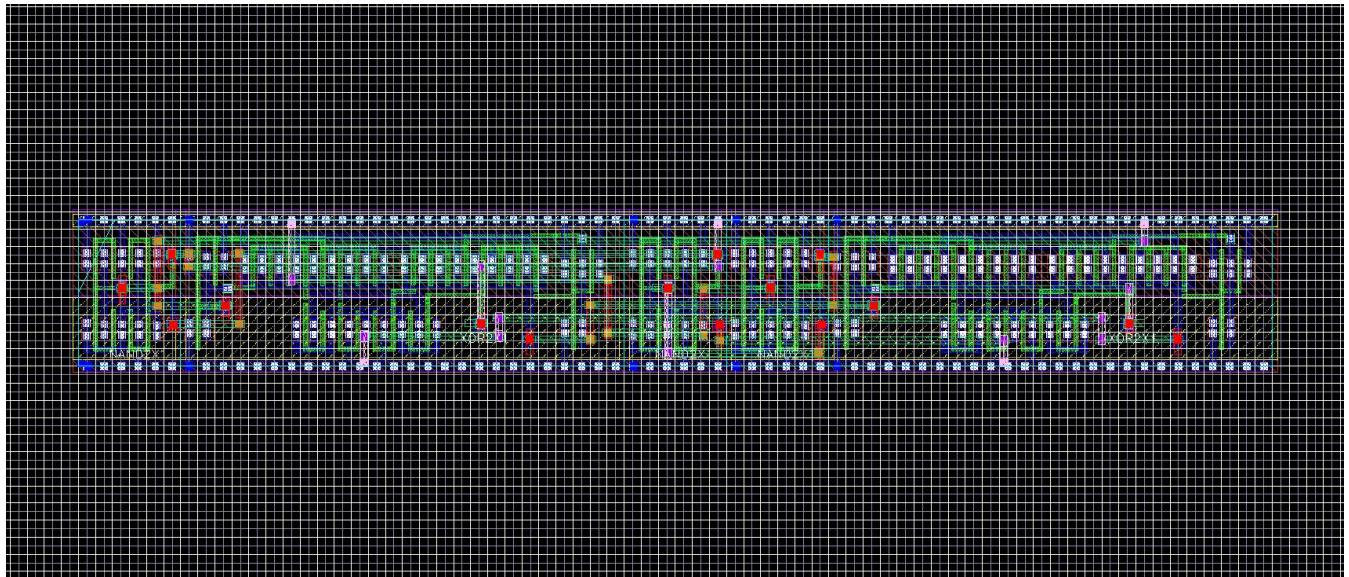
Symbol with Port Names:



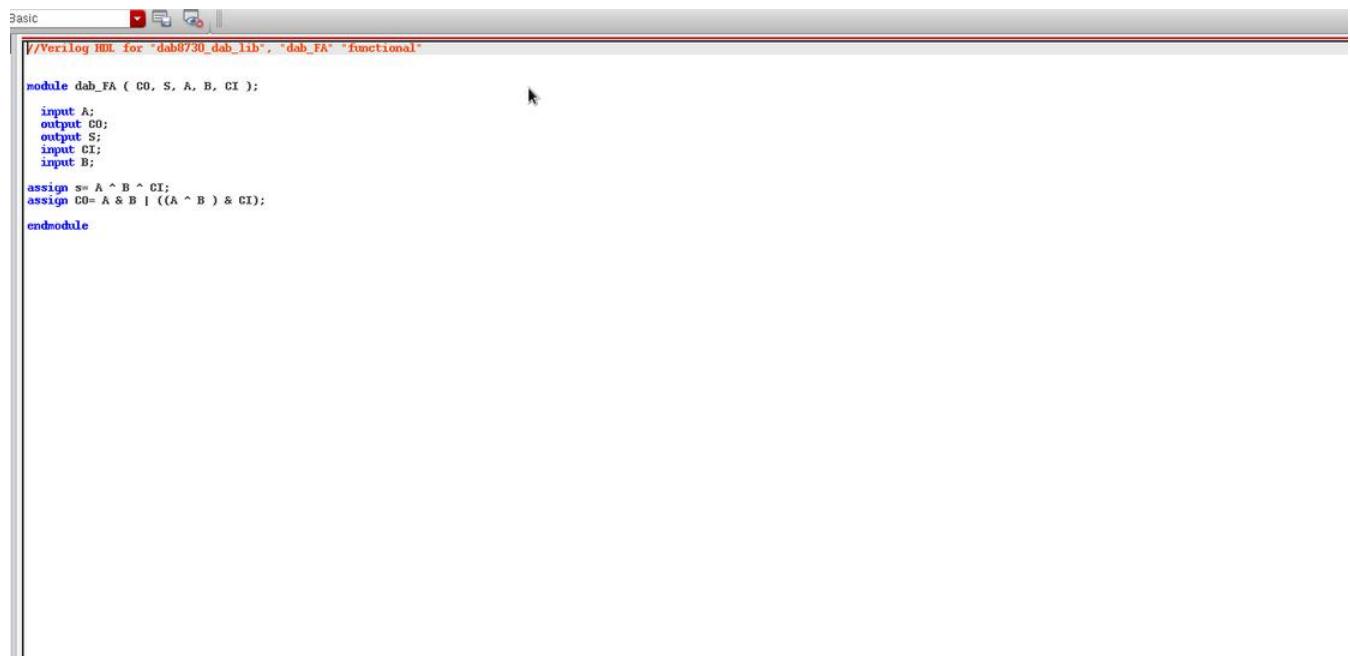
Schematic:



Layout:



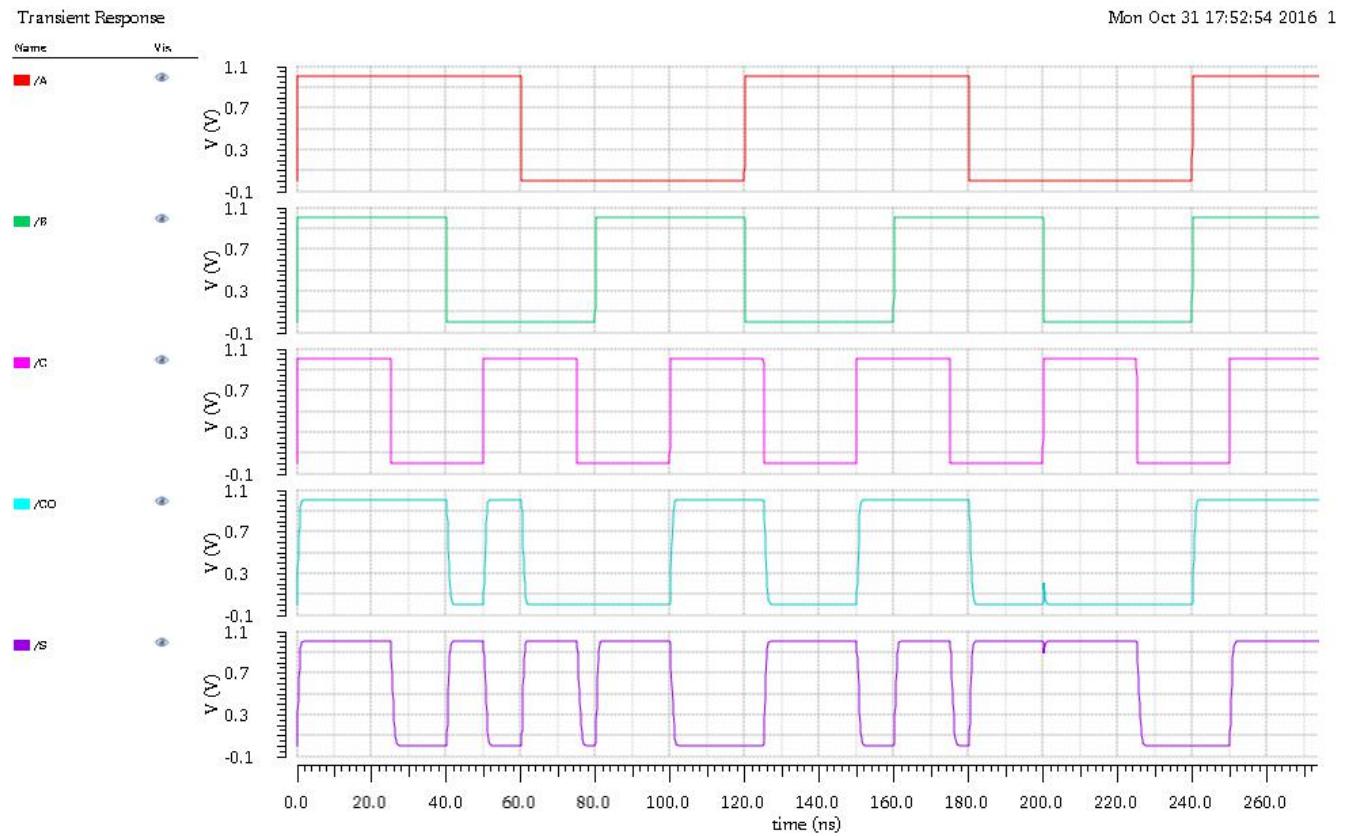
Verilog Model:



The screenshot shows a Verilog HDL editor window titled "Basic". The code area contains the following Verilog module definition:

```
/*Verilog HDL for "dab8730_dab_l1b", "dab_FA" "functional"
module dab_FA ( C0, S, A, B, CI );
  input A;
  output C0;
  output S;
  input CI;
  input B;
  assign S= A ^ B ^ CI;
  assign C0= A & B | ((A ^ B ) & CI);
endmodule
```

Functional Simulation Waveforms



Comments/Notes:

Size of layout : 14nm X 1.71 nm

Library Name: dab8730_dab_lib

Cell Name: dab_TIEHI

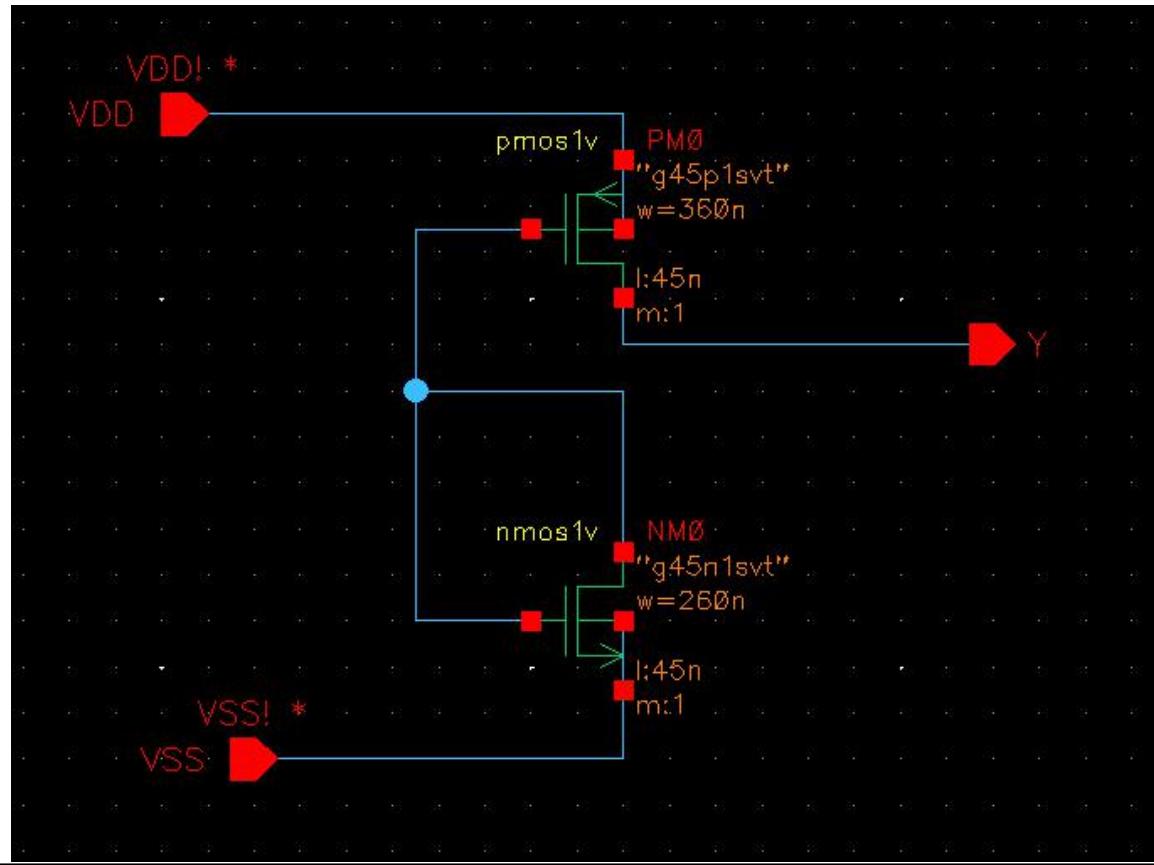
Layout Area: 0.6(WIDTH)*1.71(height)

Symbol:

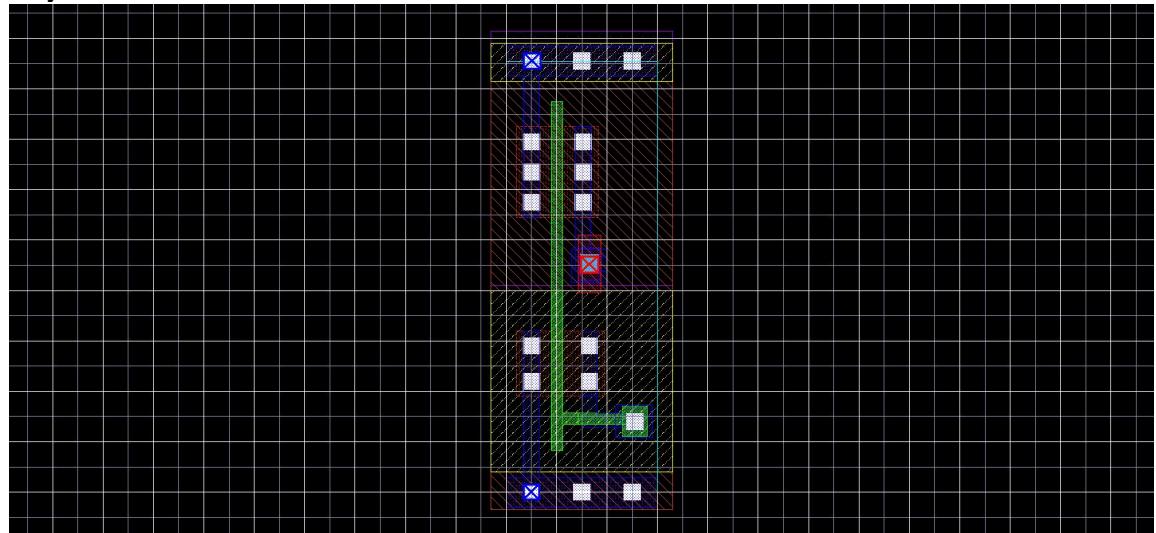


[@instanceName]

Schematic:

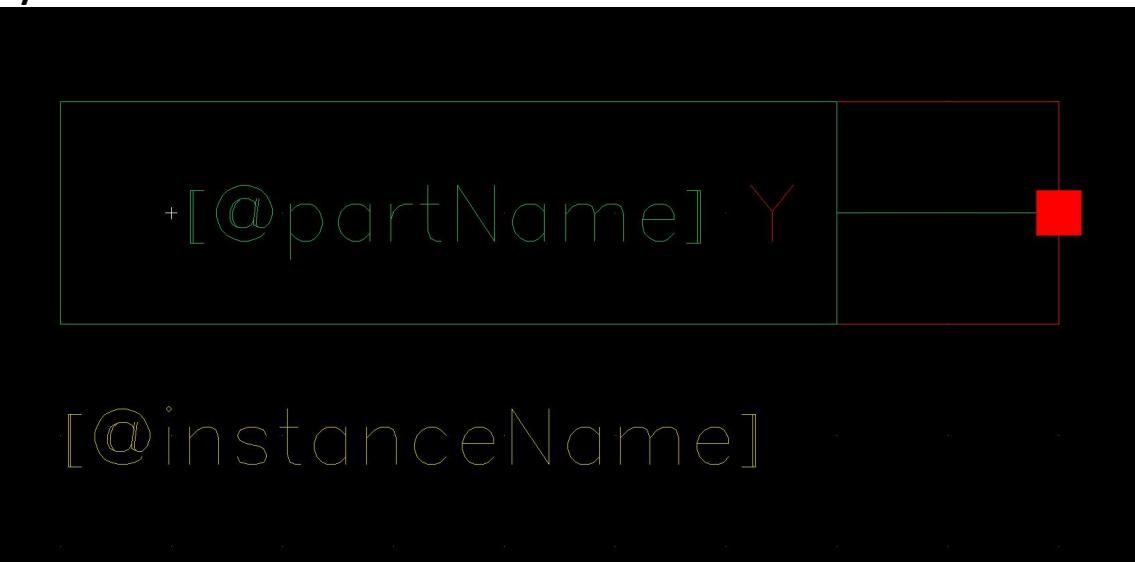
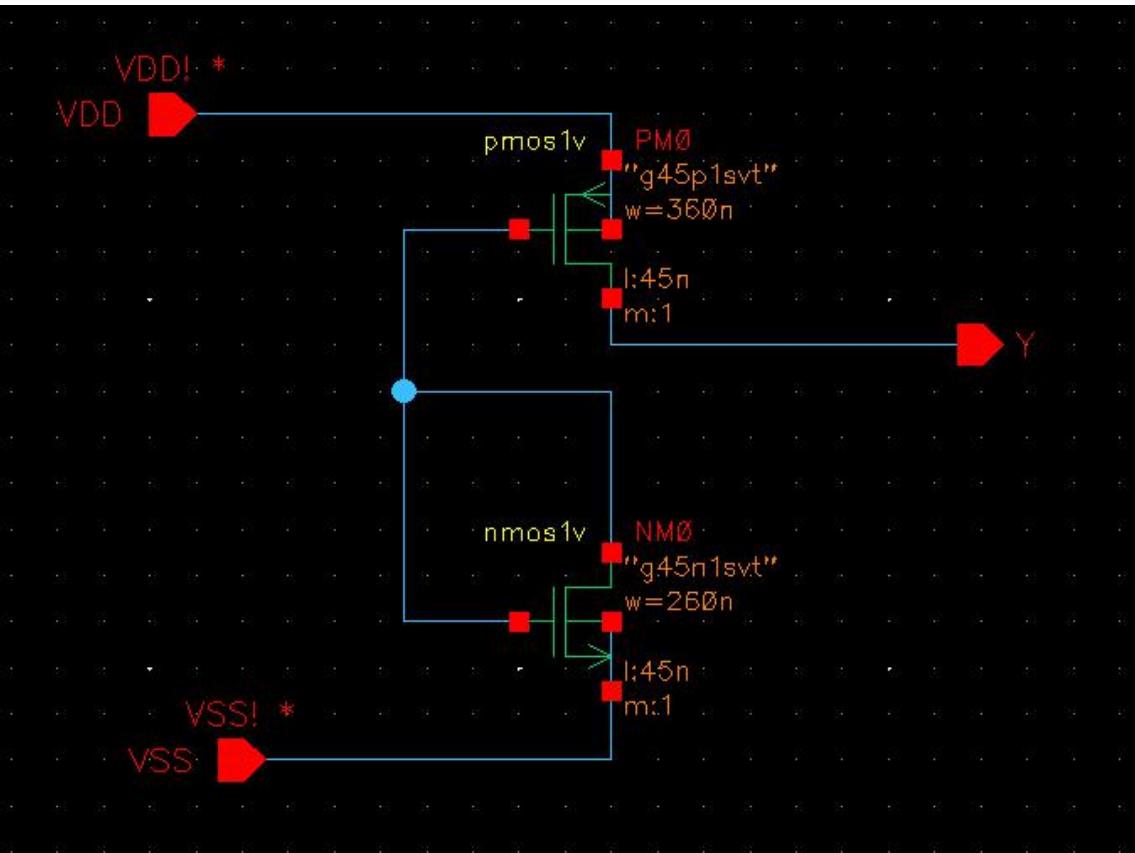


Layout:

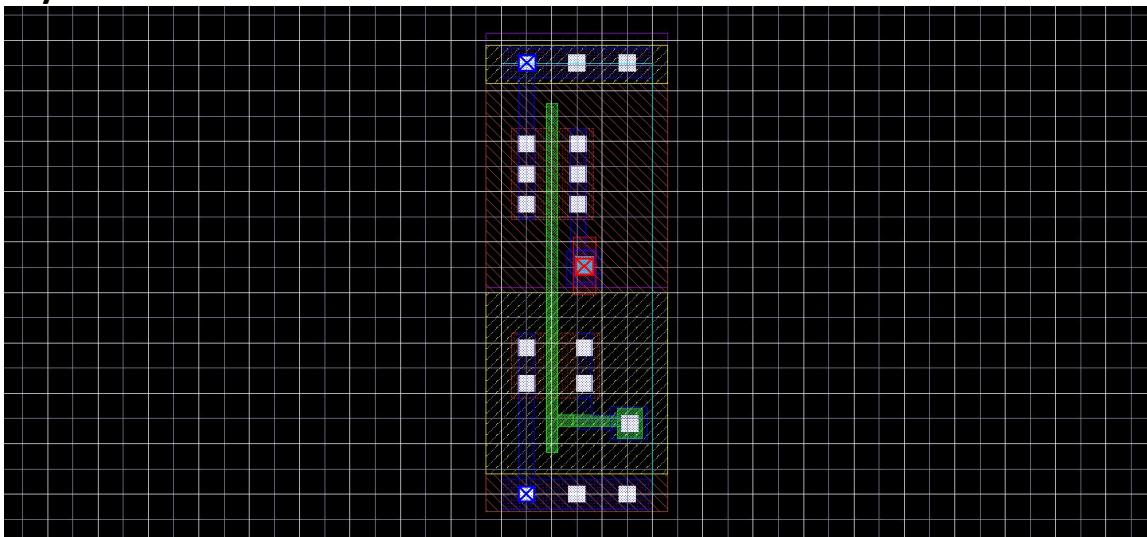


Comments/Notes:

TIE-HI cell is used to connect unused input to High.

Library Name:	dab8730_dab_lib
Cell Name:	dab_TIEHI
Layout Area: 0.6(WIDTH)*1.71(height)	
Symbol:	
 <p>[@instanceName]</p>	
Schematic:	
	

Layout:

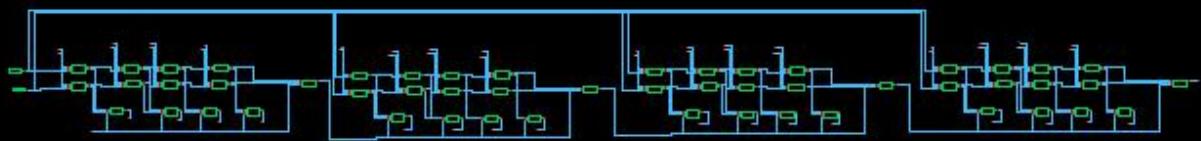


Comments/Notes:

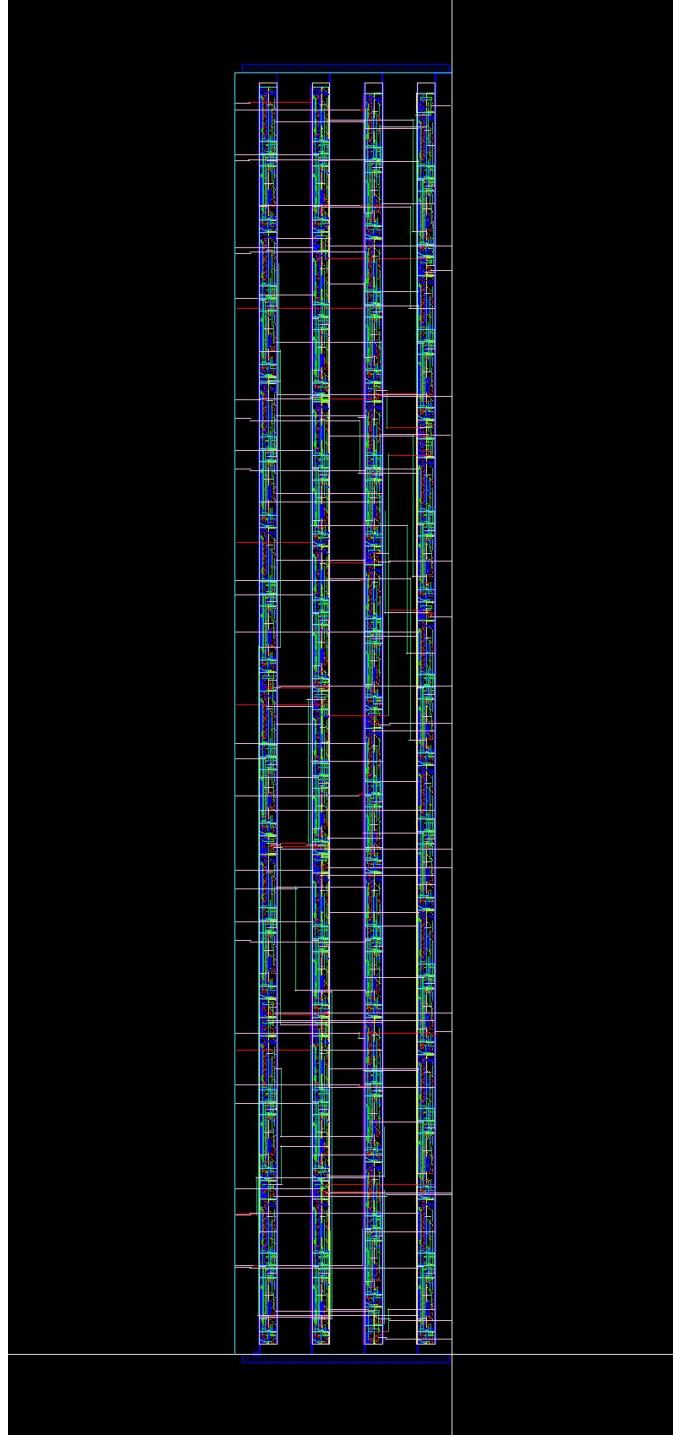
TIE-HI cell is used to connect unused input to High.

Library Name:	dab8730_dab_lib
Cell Name:	dab_ADD16
Layout Area: 124nmx20.9nm	
Symbol with Port Names:	
<p>The diagram shows a symbol with port names. It consists of two columns of pins. The left column is labeled A0, A1, A2, A3, A4, A5, A6, A7, A8, A9, A10, A11, A12, A13, A14, A15, B0, B1, B2, B3, B4, B5, B6, B7, B8, B9, B10, B11, B12, B13, B14, B15, and CRN. The right column is labeled COUT, S0, S1, S2, S3, S4, S5, S6, S7, S8, S9, S10, S11, S12, S13, S14, and S15. A vertical red line connects the A and B columns. A green box highlights the S0-S15 row.</p>	

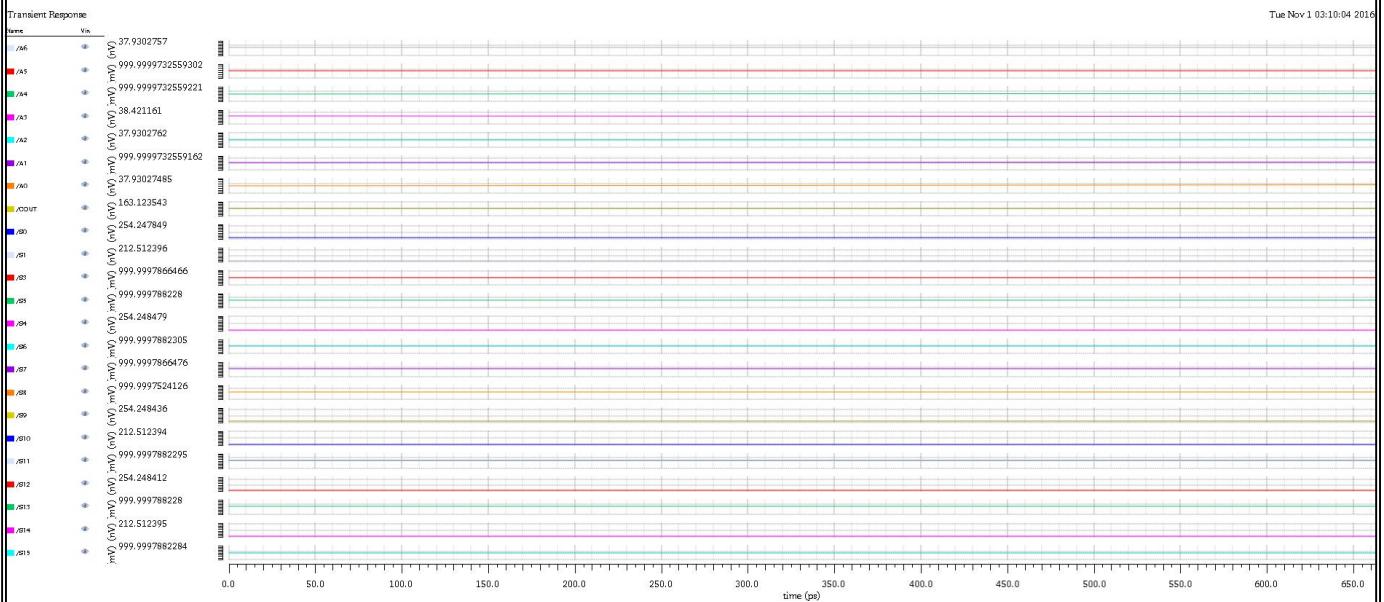
Schematic:



ayout:



Functional Simulation Waveforms:



Comments/Notes:

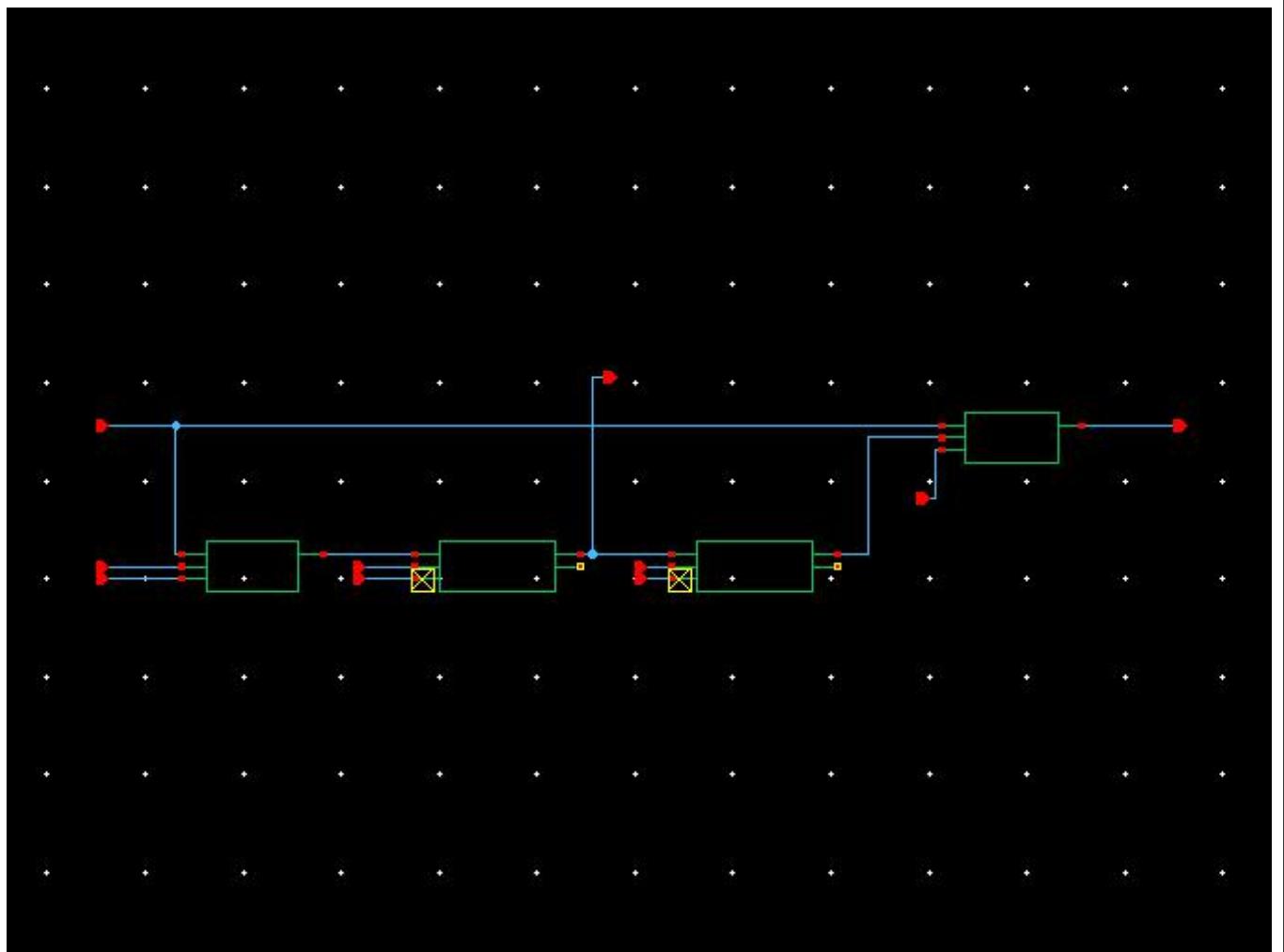
The layout was generated using the auto routing tool in virtuoso.

Dimensions of the layout are 124nmx20.9nm

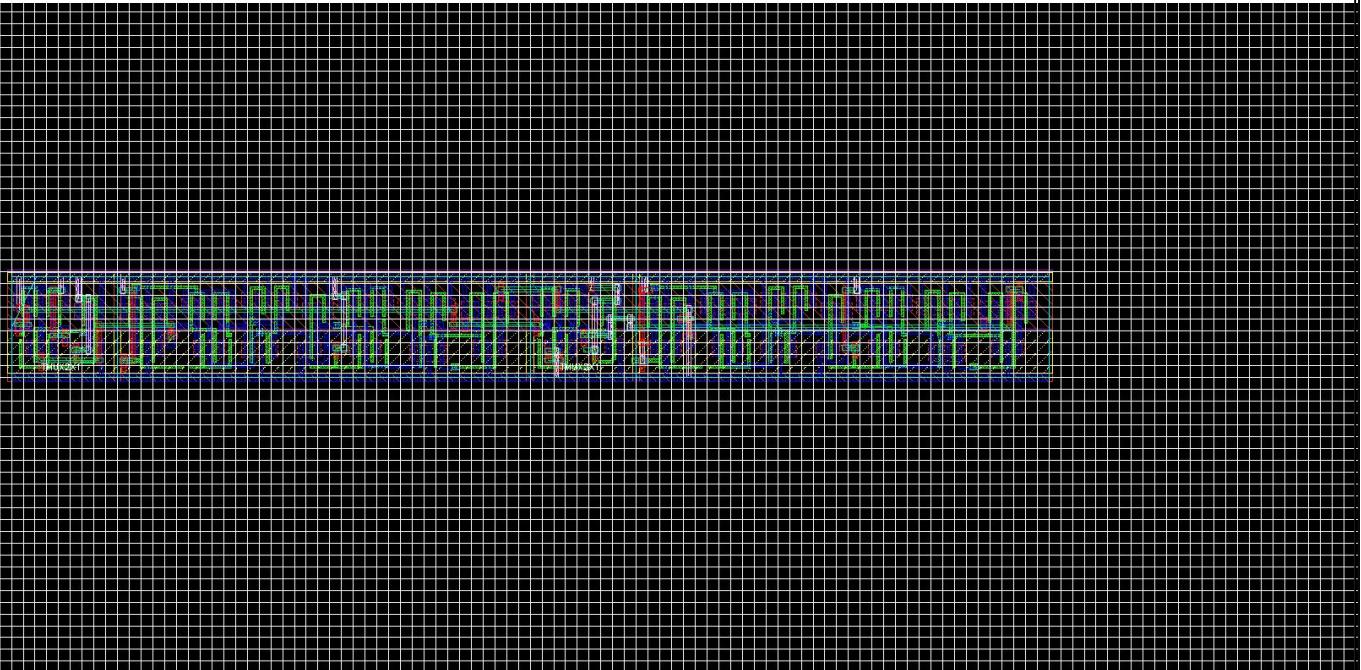
16 bit Carry select adder is made of 32 Full adders and 20 2:1 MUX.

Library Name:	dab8730_dab_lib
Cell Name:	dab_BSC
Layout Area: 17.6x1.71nm	
Symbol with Port Names:	
<p>The symbol consists of two vertical columns of pins. The left column has eight pins labeled CDR1, CDR2, IN, MODE, SDR, SIN, UDR1, and UDR2 from top to bottom. The right column has two pins labeled OUT and SOUT. A green box highlights the left column, and a red box highlights the right column. Labels @instanceName and @partName are placed near the respective columns.</p>	<p>[@instanceName]</p> <p>CDR1 CDR2 IN MODE SDR SIN UDR1 UDR2</p> <p>[@partName]</p> <p>OUT SOUT</p>

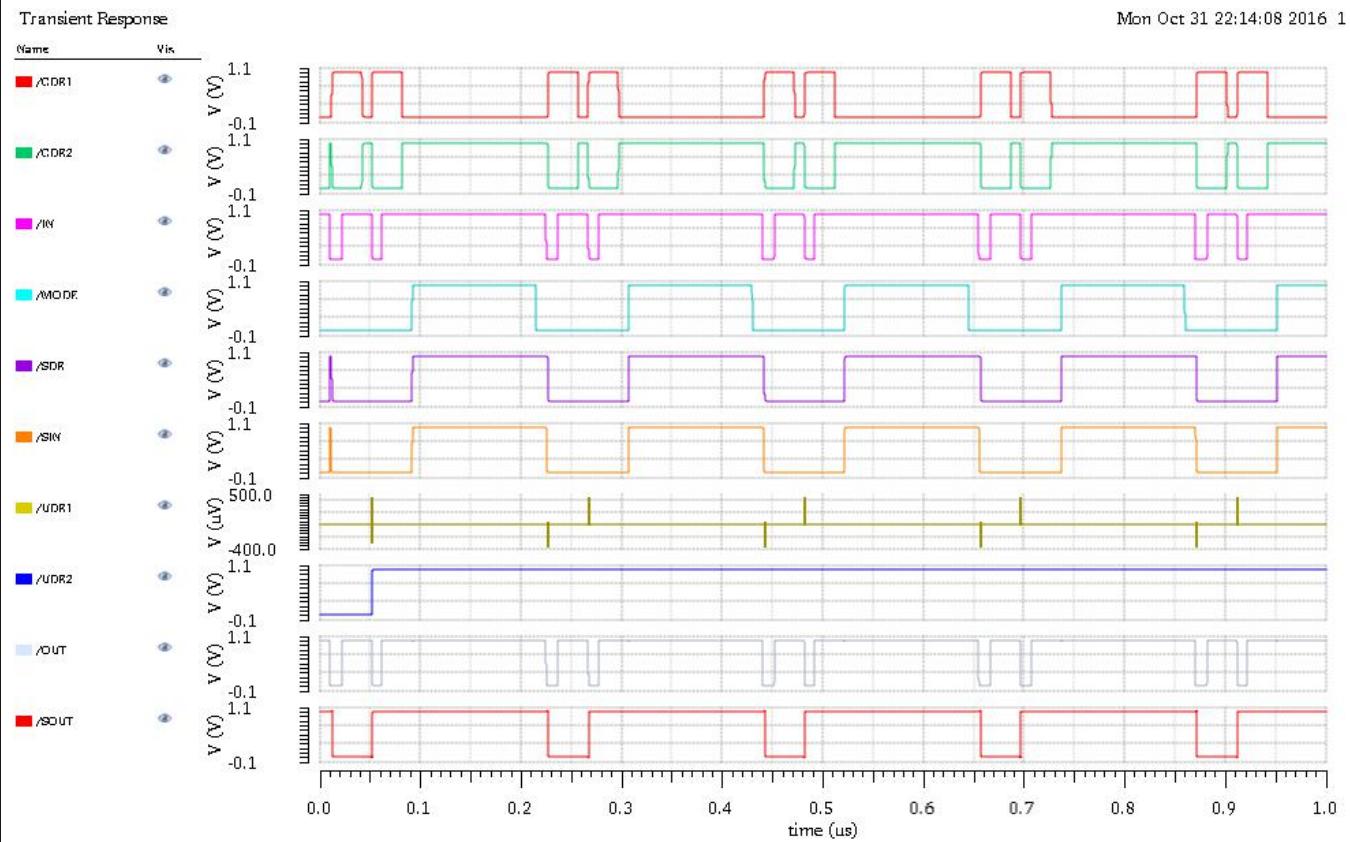
Schematic:



Layout:



Functional Simulation Waveforms:



Comments/Notes:

Dimensions of the layout **17.6x1.71nm**.

Since it is a combination of 2 TMUX and 2 D Flipflops.

Library Name:

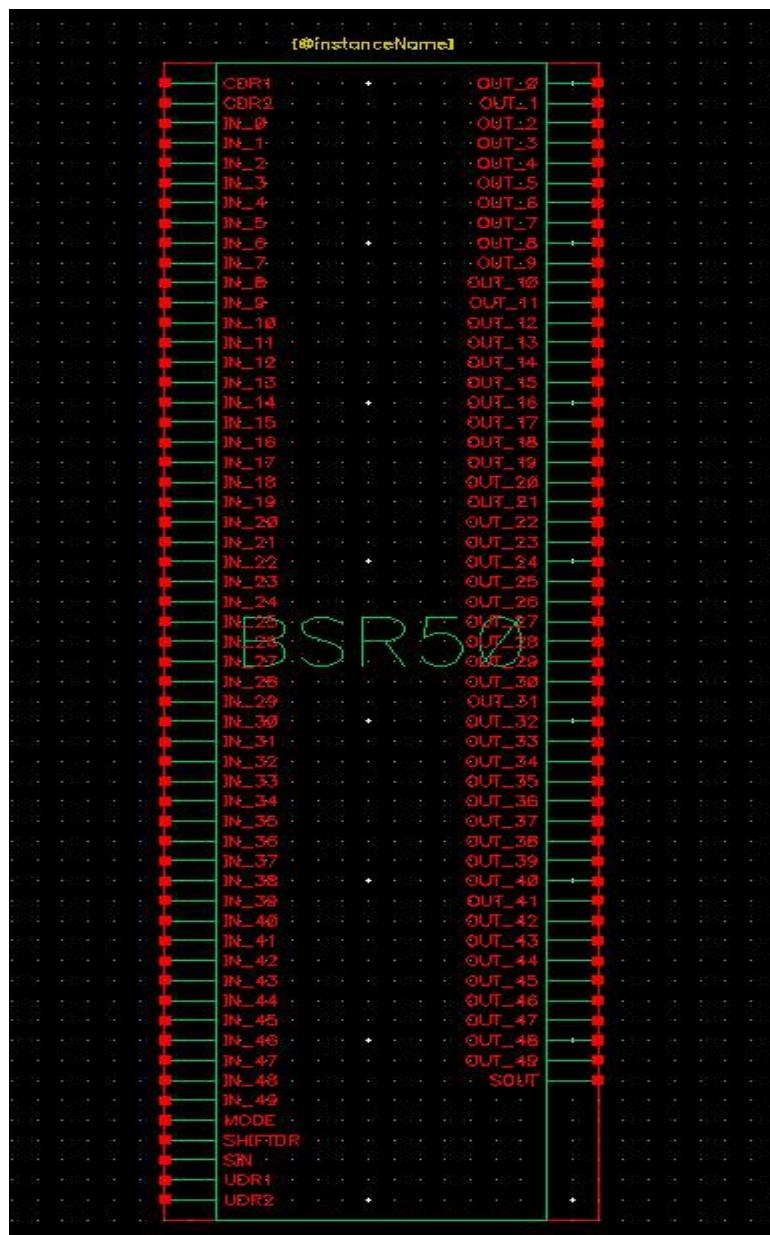
dab8730_dab_lib

Cell Name:

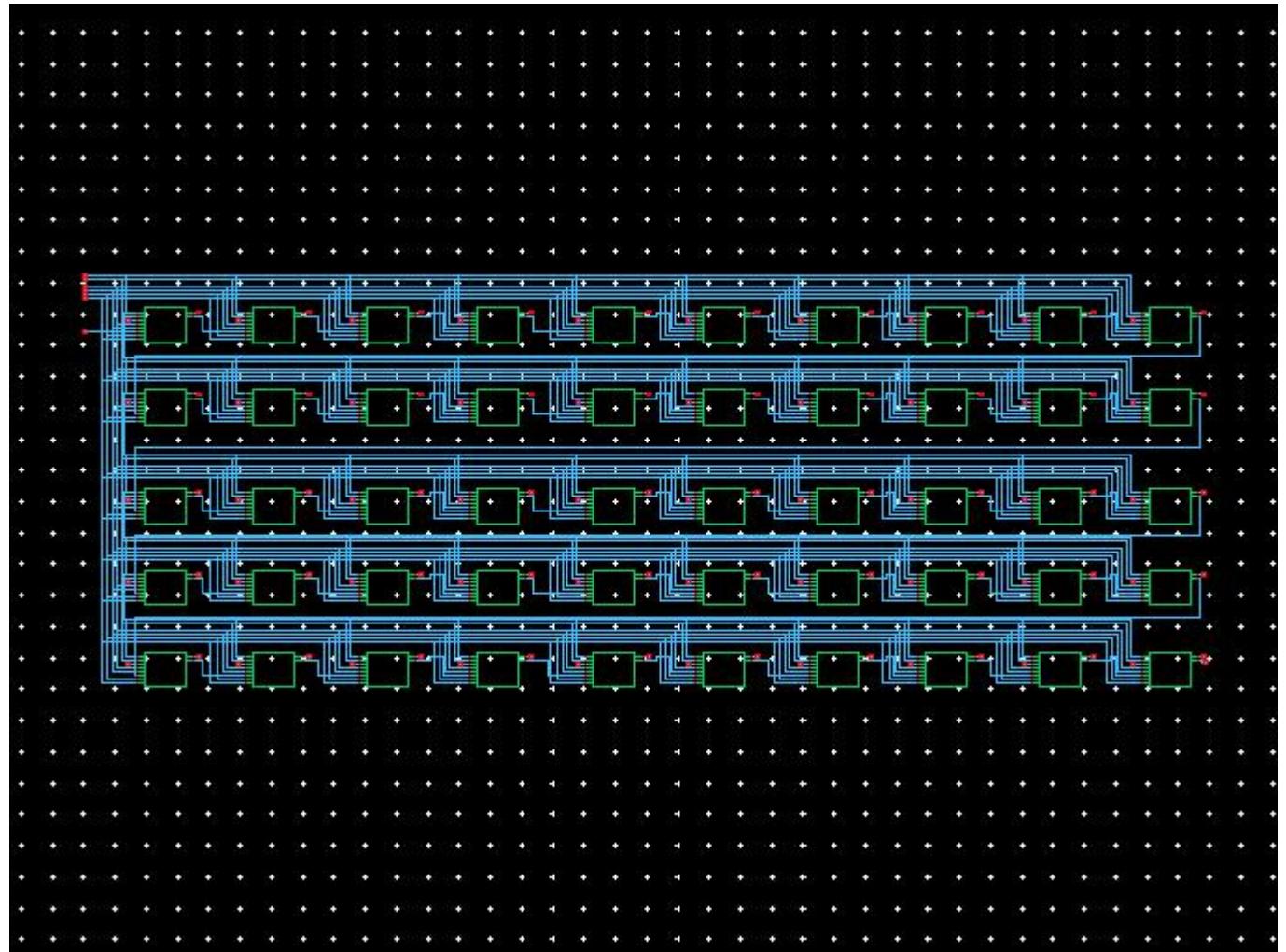
dab_BSR

Layout Area: 90nmx52nm

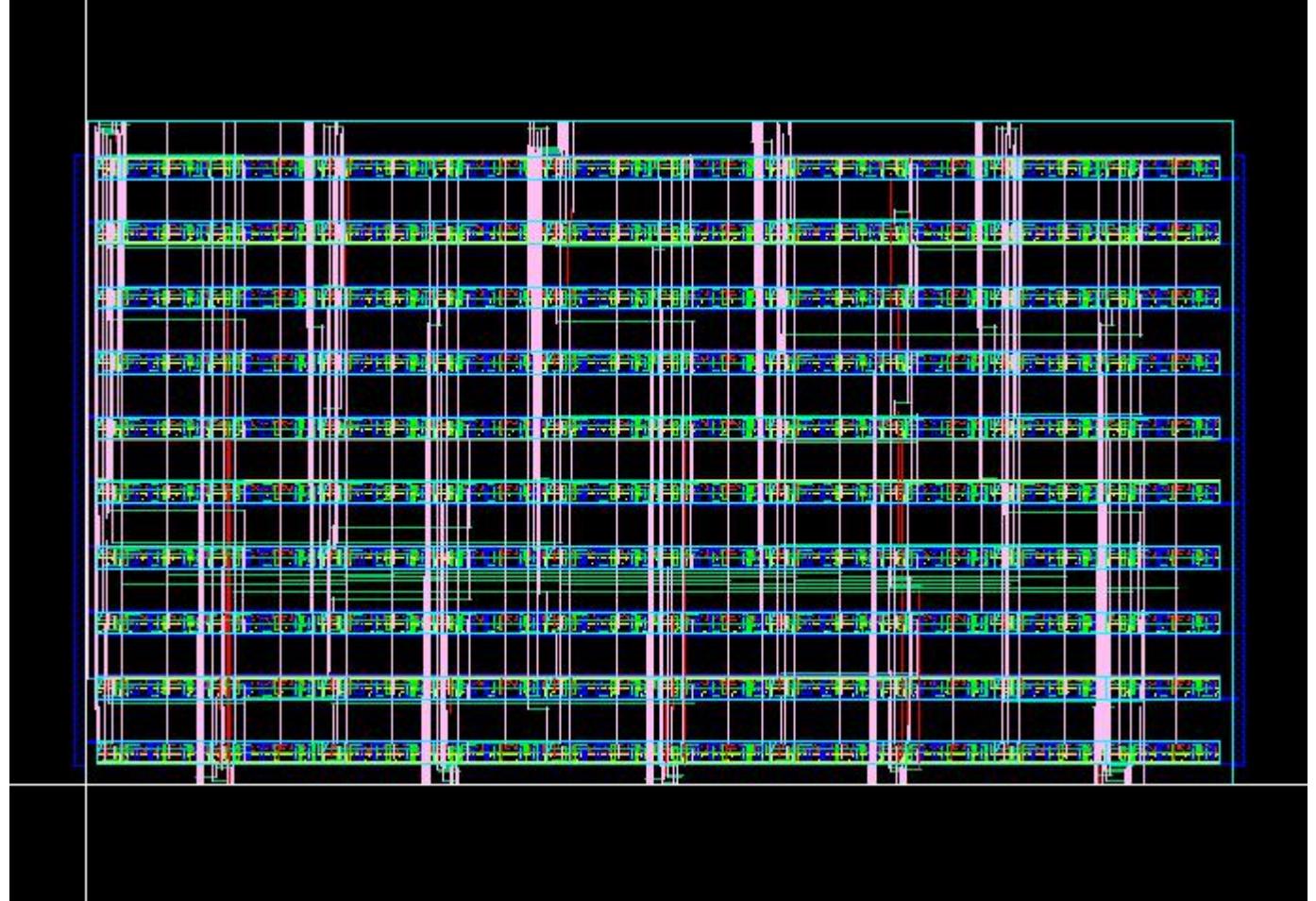
Symbol with Port Names:



Schematic:



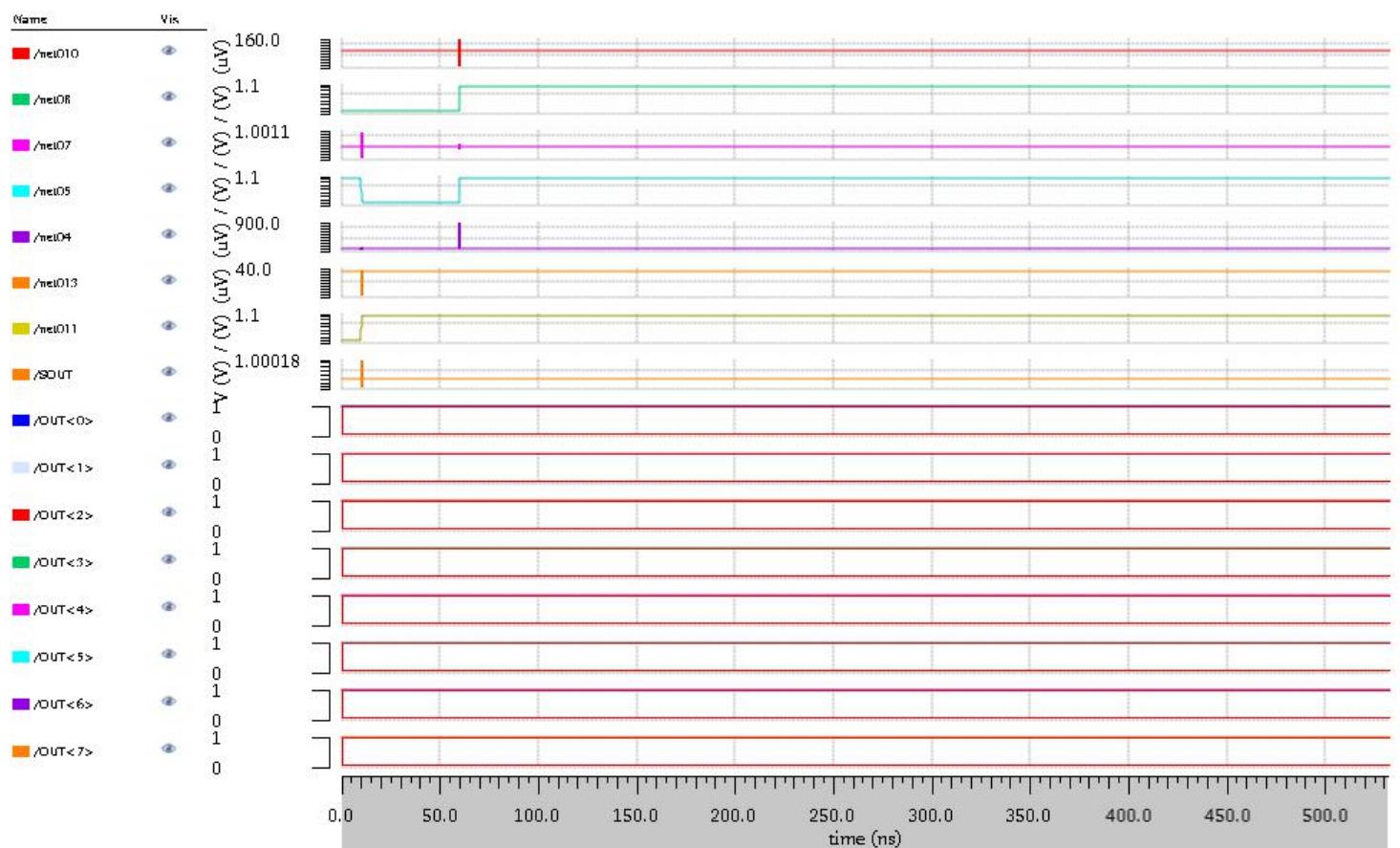
Layout:



Functional Simulation Waveforms:

Transient Response

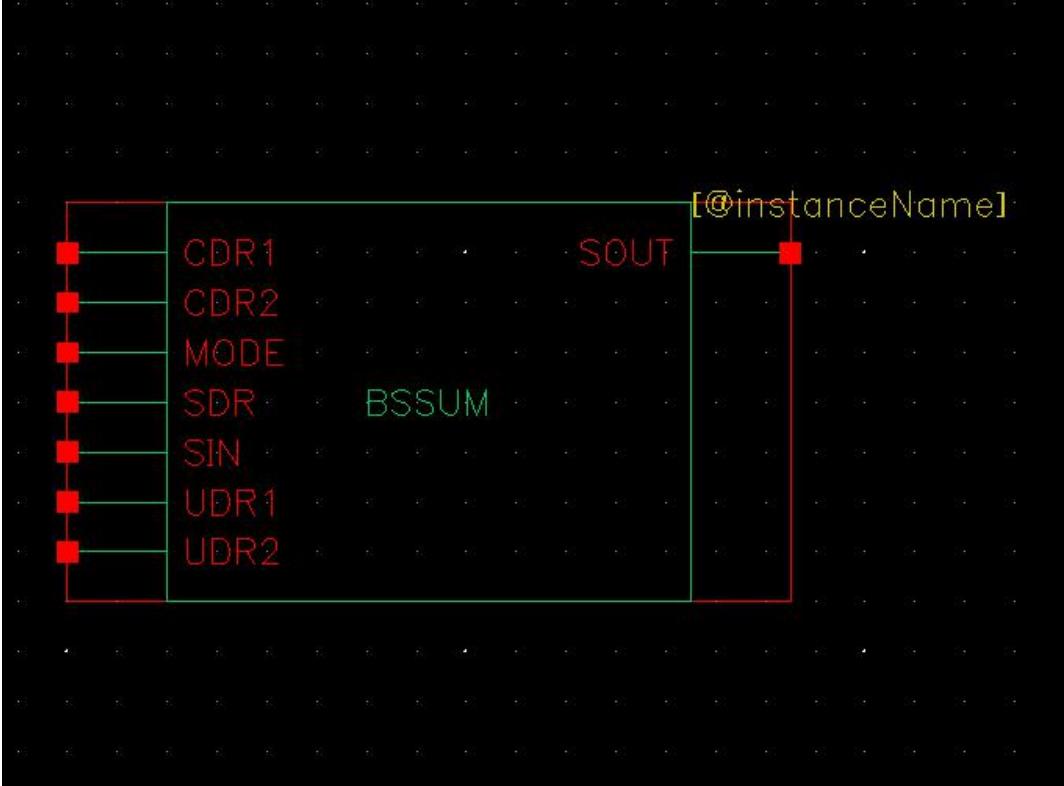
Mon Oct 31 22:58:07 2016 1



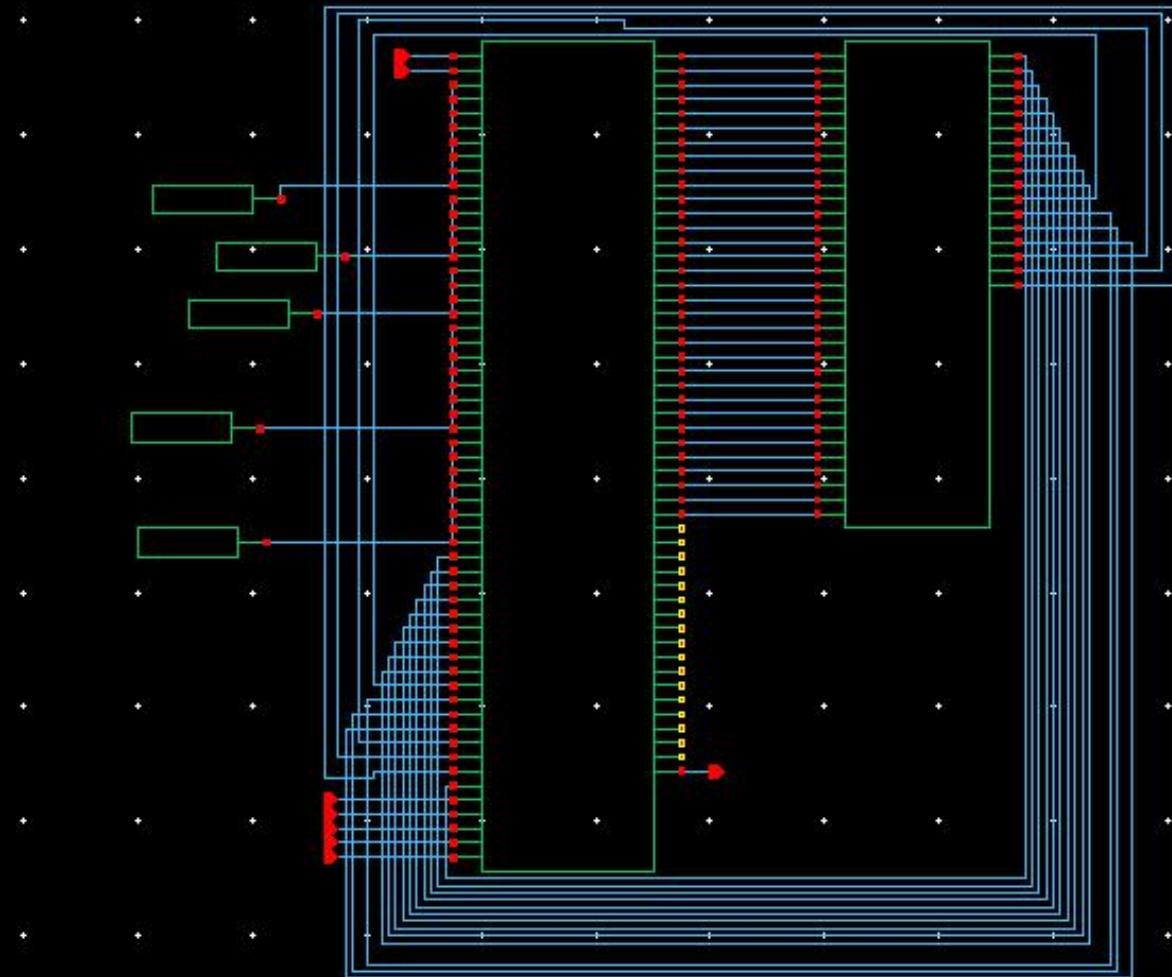
Comments/Notes:

Dimensions of the layout **90nmx52nm**

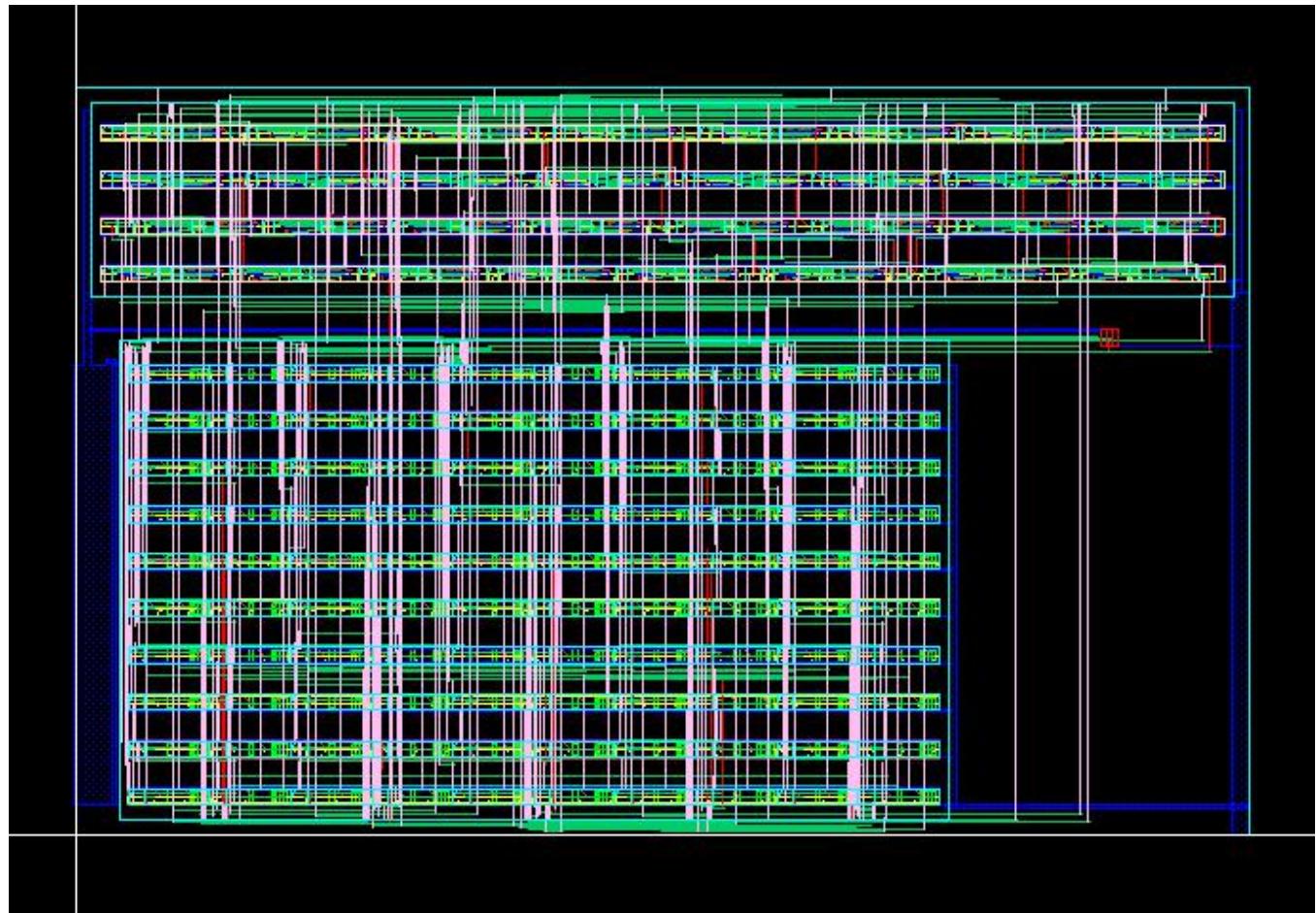
BSR50 is a merger of 50 BSC cells. 50th BSC cell SOUT is connected to SIN of 49th cell. And the SOUT of 2nd BSC cell is connected to SIN 1st cell

Library Name:	dab8730_dab_lib
Cell Name:	dab_BSSUM
Layout Area: 127.6 nm X 21 nm	
Symbol with Port Names:	
 <p>The symbol for the dab_BSSUM cell is a rectangular box containing several port names and their corresponding connection points. On the left side, there are seven red squares connected to vertical green lines, labeled from top to bottom: CDR1, CDR2, MODE, SDR, SIN, UDR1, and UDR2. On the right side, there is one red square connected to a vertical red line, labeled SOUT. Above the SOUT label, the text '@instanceName' is written in brackets. The entire symbol is set against a black background.</p>	

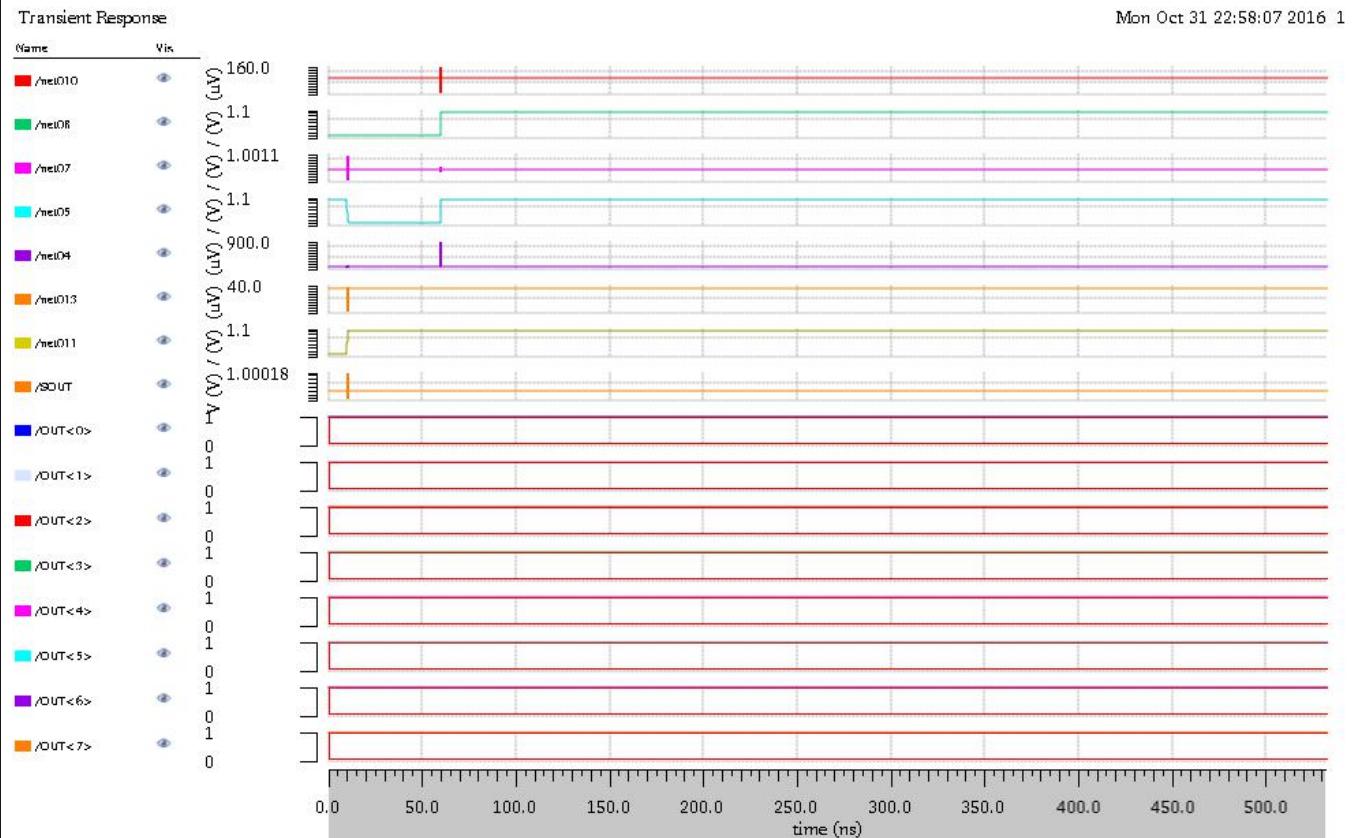
Schematic:



Layout:



Functional Simulation Waveforms:



Comments/Notes:

Dimensions of the layout **127.6 nm X 21 nm**

BSSUM is used for testing the Adder 16. BSSUM uses a BSR5cell which is combination of 50 BSC cells which is used for testing a small unit. Since we have 50 inputs and outputs we use 50 cells to check the data.

Library Name:

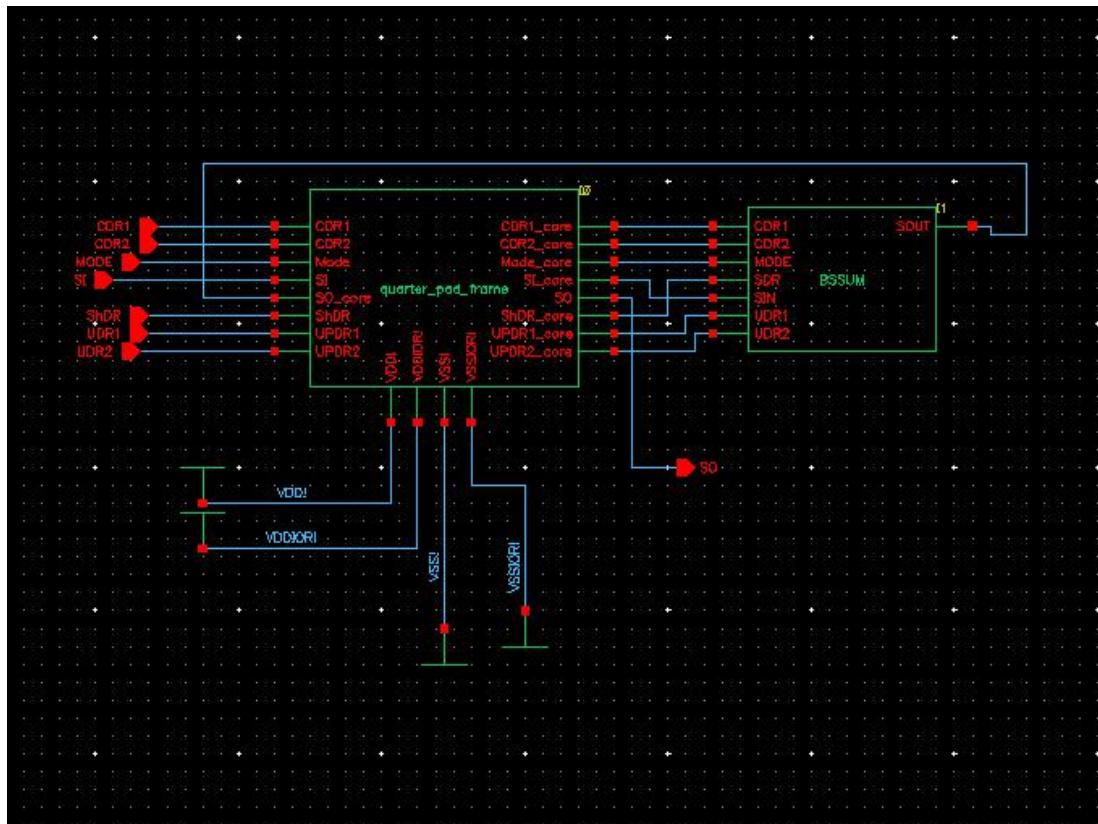
dab8730_dab_lib

Cell Name:

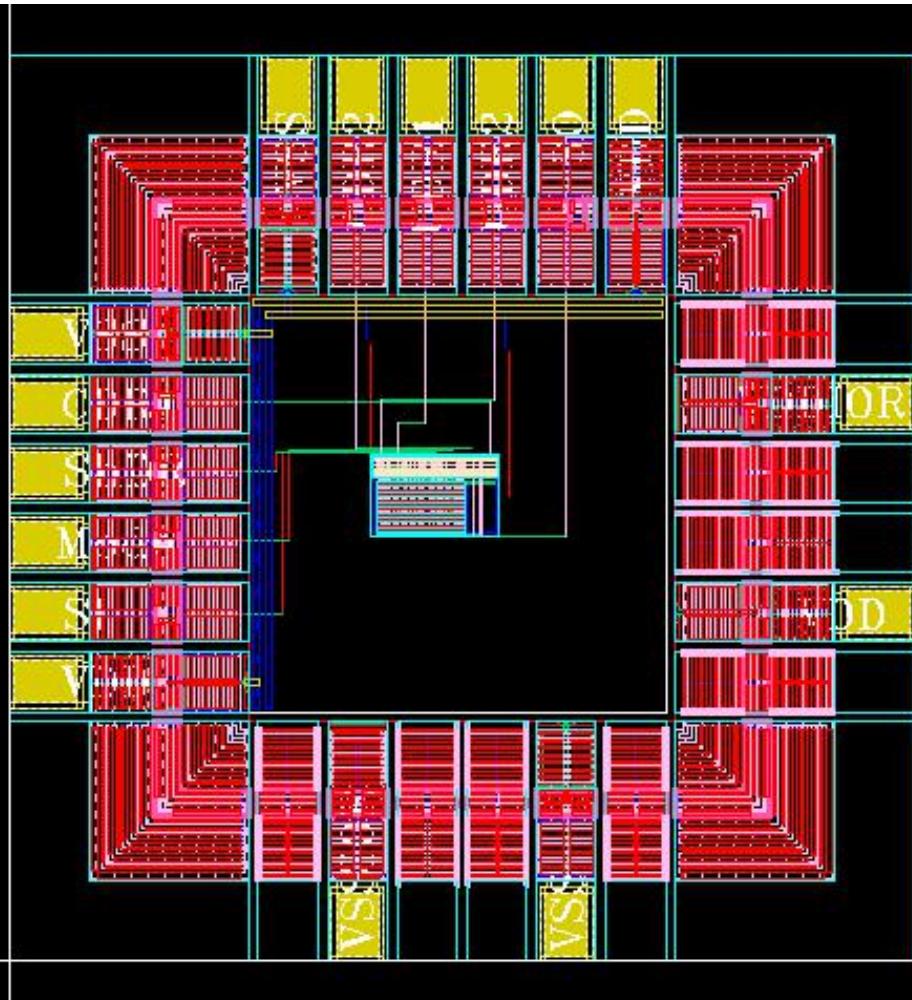
dab_BTEST

Layout Area: 910 nm X 910 nm

Schematic:



Layout:



Comments/Notes:

Dimensions of the layout 910 nm X 910 nm

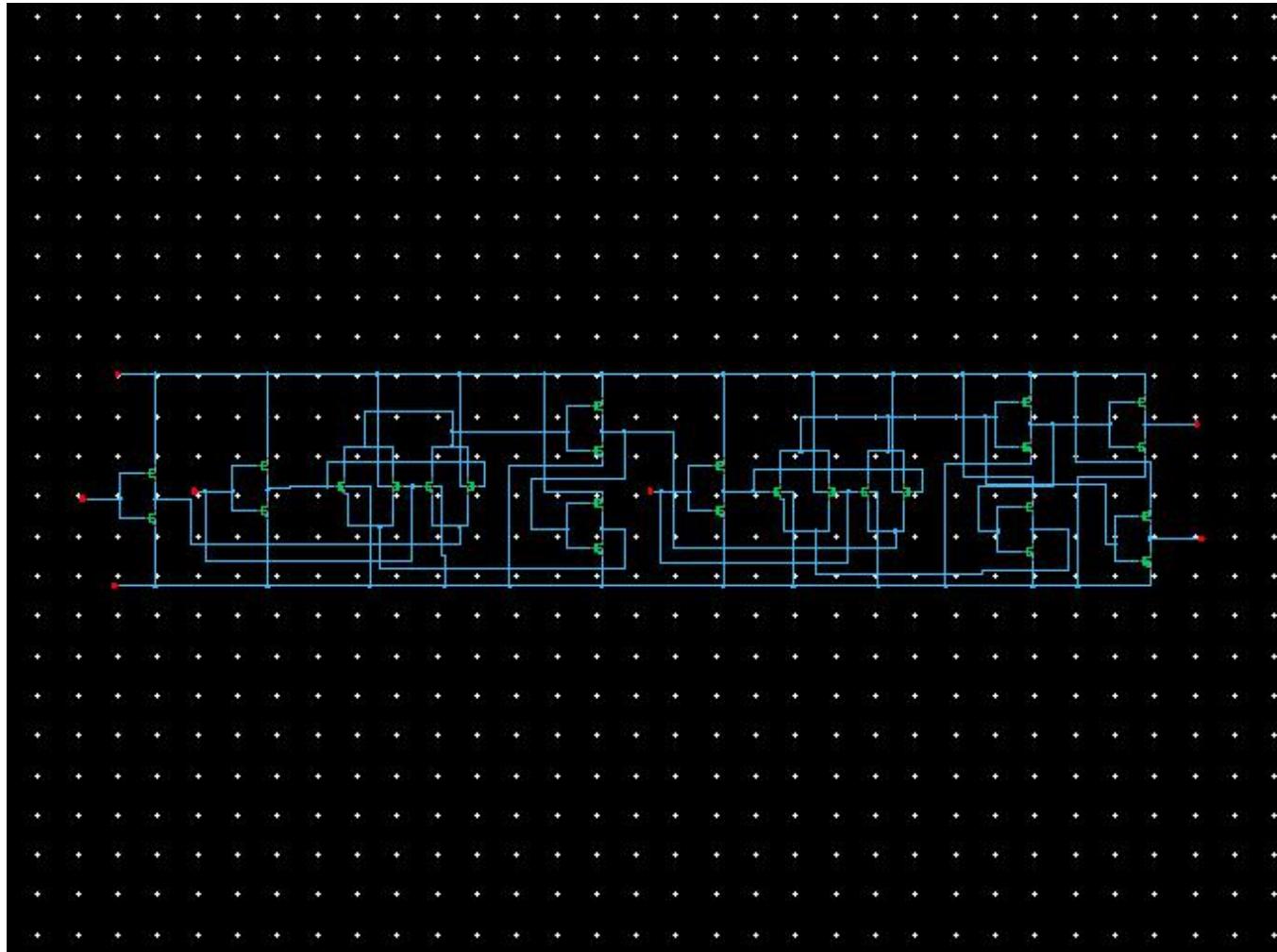
BSTEST is a cell with the pads connected to the functional cell. The pins are connected respectively to the pads and they are manufactured and sent for usage .

Library Name:	dab8730_dab_lib	
Cell Name:	dab_DFF	
Function/Truth Table:		
D	Q(rising edge of PHI1)	QN(Rising edge of PHI1)
0	0	1
1	1	0
Propagation Delay:		
PHI1(Falling) Q(Rising)	40.6 n sec	
PHI1(Rising) Q(falling)	610.2 P sec	
PHI1(Falling) QN(Rising)	50.6 n sec	
PHI1(Rising) QN(falling)	520.6 p sec	
Minimum pulse width of PHI1	52 p sec	
Minimum pulse width of PHI2	54 p sec	
Output Rise Time of Q: 1.947 n sec		
Output Fall Time of Q: 1.8 n sec		
Output Rise Time of QN: 1.5 n sec		
Output Fall Time of QN: 1.06 n sec		
Layout Area: 7 nm X 1.71 nm		

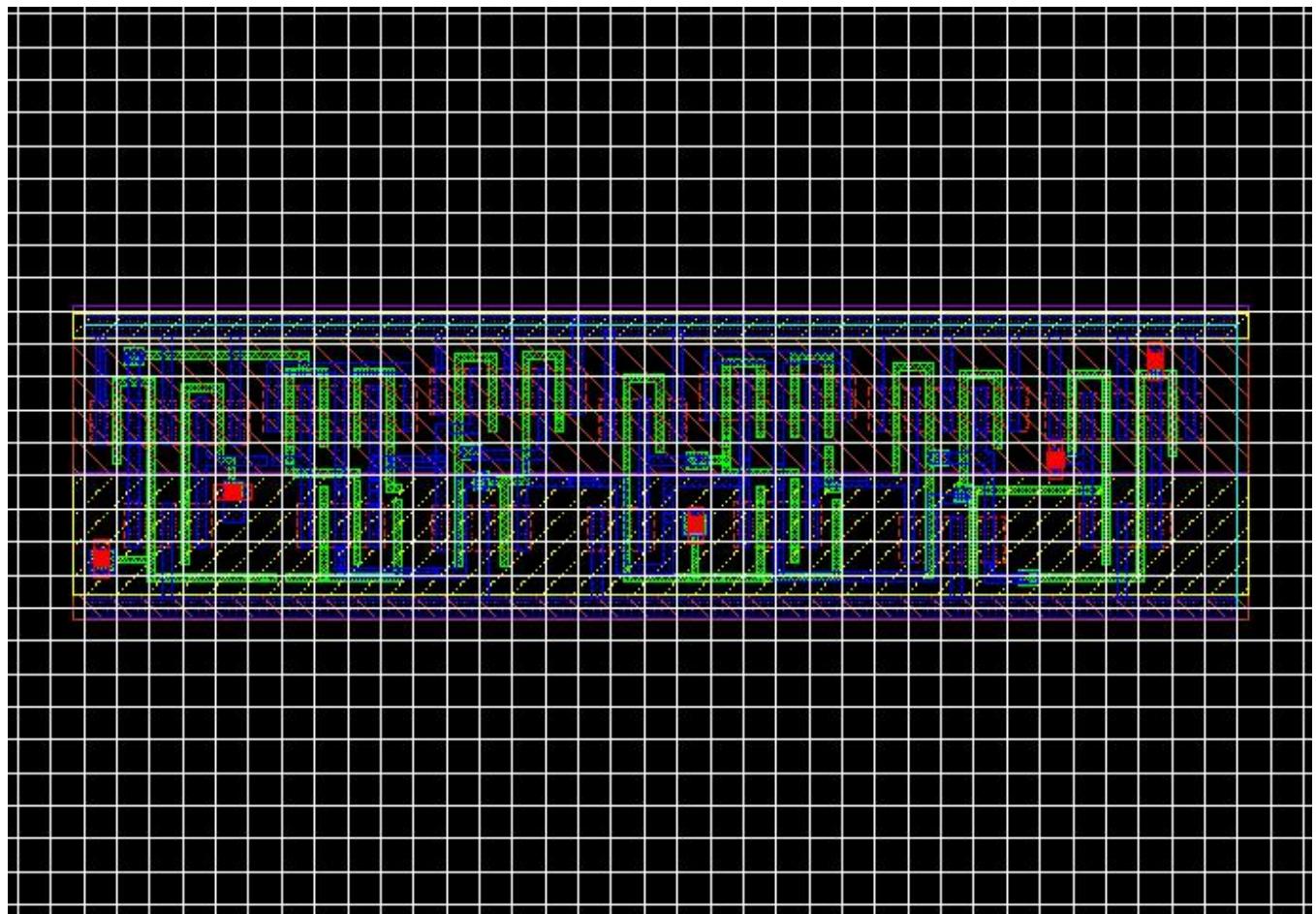
Symbol with Port Names:



Schematic:



Layout:



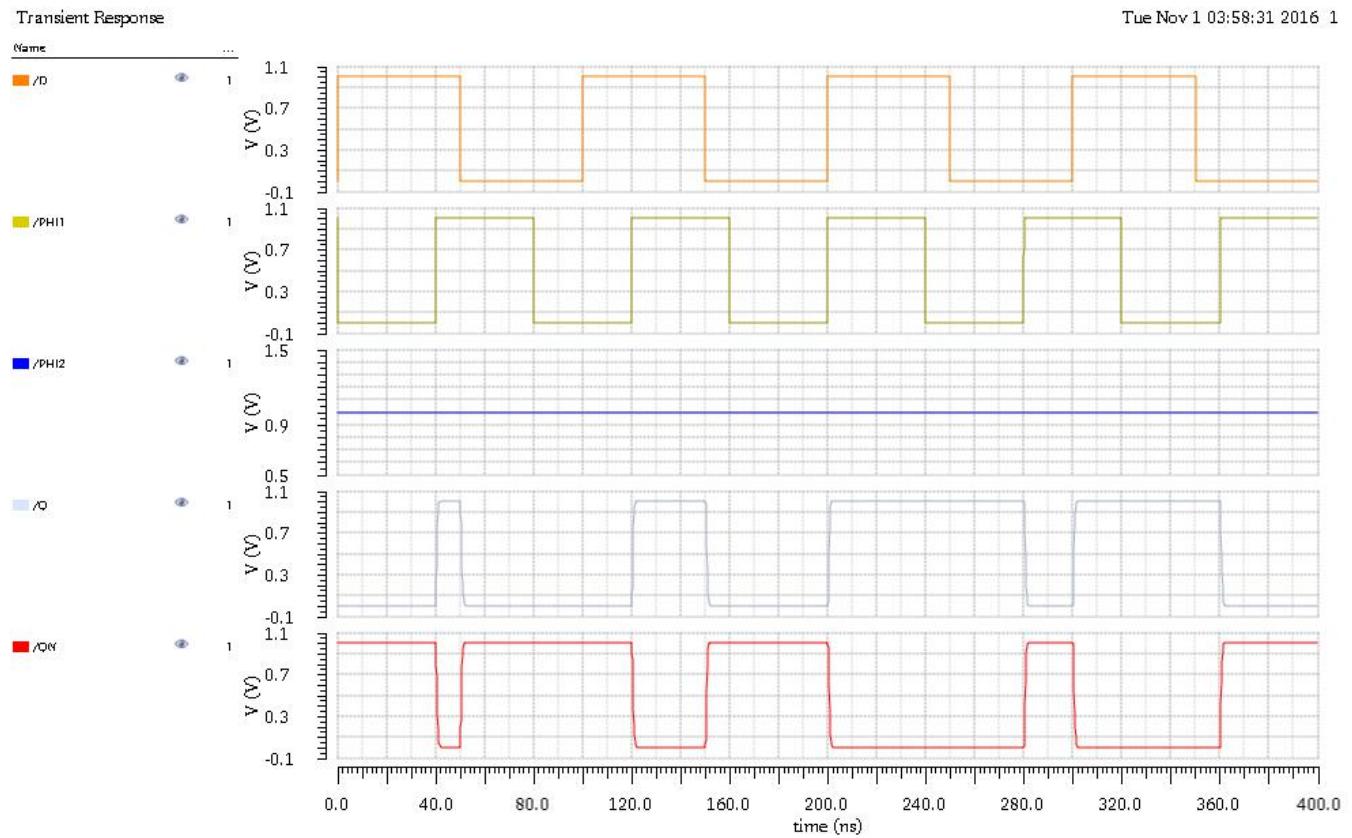
Verilog Model:

```
//Verilog HDL for "dab0730_dab.lib", "dab_BFFC2" "functional"
module dab_BFFC2 ( Q, QN, D, PHI1, PHI2, .VDD(\VDD! ), .VSS(\VSS! ) );
    input PHI1;
    input D;
    `ifndef INCA
        (* integer inh_conn_prop_name = "VDD";
           integer inh_conn_def_value = "cds_globals.\VDD! " : *)
    `endif
    VDD! ;
    output Q, QN;
    wire w1,w2,w3,w4,w5,w6,w7,w8,c1,c2,o,x,y,z;
    input
    `ifndef INCA
        (* integer inh_conn_prop_name = "VSS";
           integer inh_conn_def_value = "cds_globals.\VSS! " : *)
    `endif
    VSS! ;
    input PHI2;
    input D;

    not a(w1,w4);
    not U2(w3,Phi2);
    and U3(o,Phi2,w1);
    and U4(x,w5,w3);
    not U5(w3,w2);
    or a5(w2,x,o);
    not U6(w6,w3);
    not U7(w3,Phi1);
    and U8(x,Phi2,w3);
    and U9(y,w3,w6);
    or a6(w4,y,y);
    not a10(w7,w4);
    not a11(yt,w7);
    not a12(w8,w7);
    not a13(w5,w8);
    not a14(Q,w6);

`endmodule
```

Functional Simulation Waveforms:



Comments/Notes:

The D flip flop has master and slave circuit. Both of them are driven by a non overlapping pulse. The Input D is shown on the output whenever the the PHI1 goes high following PHI2. But when both the signal is high at the same time they get transparent and the output fails.

