# cjg_lib: Standard Cell Library

# &

# Corner Cell Design Report

EE-520

Connor Goldberg

October 30th, 2015

**cjg_lib: Standard Cell Library & Corner Cell Design Report**

## Design Constraints & Discussion

Bit Code: 000010

This project's main focus was creating a 16-bit ripple carry select adder, then testing it with a 50-bit boundary scale register cell. This design was then placed into a corner cell. To accomplish these immense tasks, only two additional flat cells were added to our library, then several cells were designed at the hierarchical level. This involved using the place and route tools. All of the designs had to be created with the specifications in mind.

Specifications: My base transistor size for NMOS was constrained to have an oxide width of 180nm, with a 45nm polysilicon gate width. In addition, the W/L ratio of the NMOS transistors in this library were constrained to be 1X the W/L of the PMOS. This causes the gates to be unbalanced in terms of rise and fall times. Two flat designs that were added to the library were a rising edge triggered D-Flip-Flop (CJG_DFFX1) and Full-Adder (CJG_FA). The full adder was made using the instantiation of other cell, however the pins were pulled inside of the PR boundary and the height was made to the same height as the other standard cells so it could be placed and routed alongside them easily.

The D-Flip-Flop was specifically a D type master-slave Flip-Flop with two non-overlapping clocks. This is basically a 2-stage design where the first stage locks the value at the falling edge of its clock, $\varphi_2$, and that value is sent into the output stage at the rising edge of the second clock, $\varphi_1$. The non-overlapping clocks are good for setup-hold times, however it is important to design the cells and understand the minimum timings that are required for the D-Flip-Flop to function properly. For this design, the important timings are the minimum pulse-widths for each of the clocks, to give enough time for the input value to be captured / output by the master or slave. The other important timing that is needed is the guard time, which is the time between the falling edge of $\varphi_2$ (when the input is captured) and the rising edge of$\varphi_1$ (when the value is output). If the time between these edges are too close, then the design will not function.

The Full Adder was just a standard full adder design with 2 XOR gates and 3 NAND gates, all of which were designed in Project 1. These cells were then placed into the layout and routed using the auto route functionality of Cadence Virtuoso. The timings for this gate were measured as well, and they are shown in the Timing Results section.

The next cell that was designed was the 16-bit ripple carry select adder (CJG_ADD16). This involved 32 of the full adders in addition to 17 multiplexers. The inputs go into 32 adders, where 16 are evaluated without a carry value, and the other 16 are evaluated with the carry value. Then the actual carry input just is a select signal to the multiplexers to choose the correct result. This design was easy to implement in the schematic, however it was the first larger design where the place and route tools were used to create the layout. At first it was extremely frustrating trying to work with the tools and all of the different settings/options. Once the schematic part was designed, the design was simulated to verify the functionality was correct. For the larger cells with many inputs and outputs, mixed signal was used to test the design. This involved creating an additional cell to serve as a test bench for the cell that was to be tested. This test bench was comprised of a functional view that described the output vectors of the test bench, which then became the input signals to the design under test (DUT). The test bench also included a symbol-view so that the two designs could be placed together in the test cell. A separate simulation had to be run for this test to support the special configuration of the test bench. There was definitely more overhead involved to get the cell ready for testing, however the test process is much more advanced and once it is set up, it is much easier to test different input combinations and then view the outputs.

The boundary scan cell (CJG_BSC) was the next cell to be designed. This cell involved two of the D-Flip-Flops that were designed in this project, then two additional multiplexers. This cell has two inputs, a standard input and a scan-in input, then two outputs which consist of a standard output and a scan-out output. There are also two select pins that control the mode of the cell. By itself, the cell is not very useful, however when combined with many other boundary scan cells it becomes extremely helpful.

The 50-bit boundary scan register (CJG_BSR50) was the design created using 50 separate boundary scan cells. This cell can scan in a vector of inputs from a single pin, then send those inputs to the output (which would then serve as the input to a cell that was under test). The results could then be scanned back into the boundary scan register, then scanned out of a single pin and read to check the results. This is exactly what was done to test the 16-bit adder. To create the layout for the boundary scan register, 50 of the boundary scan cells were instantiated and then routed using the auto route functionality. This cell was easier to route than the adder because it was the second time using the auto-route functionality for a cell that large.

The last main cell that was designed was the boundary scan sum cell (CJG_BSSUM). This design was one of the main purposes of this project. The 16-bit adder was placed into a cell with the boundary scan register to be tested. Some of the outputs of the BSR were tied to adder, then the outputs of the adder were wired back into some of the inputs of the BSR. The schematic and layout for this cell was relatively straightforward after completing the previous designs, however the test bench for this cell was extremely difficult to figure out at first. The test consisted of scanning in a large vector. This was shifted into place. Then this vector was output into the adder. The adder result was then captured into some of the inputs of the BSR. Once these inputs were captured, the values were shifted out of the register. Figuring out the timing for the clocks and select signals to the BSR was tough to figure out, however it was extremely satisfying to see the final result when it was correct.

The BSSUM was then placed into a Quarter Pad Frame cell (CJG_BSTEST). This design was manually routed due to complications with the quarter pad cell design. In addition, the design never passed DRC or LVS due to the same reason.

**cjg_lib: Standard Cell Library & Corner Cell Design Report**

With the exception of the BSTEST cell, every other cell was correctly designed, simulated, and they all passed DRC and LVS. This project was extremely time consuming even with the help of place and route tools. One of the most important thing learned from this project was that advanced tools can be extremely helpful, if they are used correctly. If the tools are not used with the proper settings or values, then the tools will cause a big mess of the designs and cause trouble. Once I gained familiarity with how these functions worked and how to use them, it became a much more pleasant experience. In addition, the mixed signal test bench work was very interesting. It made the larger cells much easier to verify. Overall, this project was a lot of work, but it was very rewarding to complete successfully.

## Timing Results

**Propagation Delay: CJG_FA**

| | |
|---|---|
| **A → S↑** | **1.488 nS** |
| **A → S↓** | **928.4 pS** |
| **B → S↑** | **1.448 nS** |
| **B → S↓** | **982.8 pS** |
| **CIN → S↑** | **1.340 nS** |
| **CIN → S↓** | **808.6 pS** |
| **A → COUT↑** | **1.229 nS** |
| **A → COUT ↓** | **980.5 pS** |
| **B → COUT ↑** | **1.232 nS** |
| **B → COUT ↓** | **898.3 pS** |
| **CIN → COUT ↑** | **1.230 nS** |
| **CIN → COUT ↓** | **986.2 pS** |

**S: Rise Time: 1.764 nS, Fall Time: 1.050 nS**

**COUT: Rise Time: 1.638 nS, Fall Time: 1.118 nS**

**Propagation Delay: CJG_DFFX1**

| | |
|---|---|
| **PHI1 → Q↑** | **1.237 nS** |
| **PHI1 → Q↓** | **866.4 pS** |
| **PHI1 → QN↑** | **1.316 nS** |
| **PHI1 → QN↓** | **858.0 pS** |

**Q: Rise Time: 1.639 nS, Fall Time: 1.050 nS**

**QN: Rise Time: 1.639 nS, Fall Time: 1.051 nS**

## Appendix

This following section contains the individual data sheets for each cell that was designed in addition to overview sheets for the hierarchical cell designs.

Name: Connor Goldberg

| | |
|---|---|
| **Library Name:** | **cjg_lib** |
| **Cell Name:** | **CJG_INVX2** |

**Function/Truth Table:**

| **A** | **Y** |
|---|---|
| **0** | **1** |
| **1** | **0** |

**Propagation Delay:**

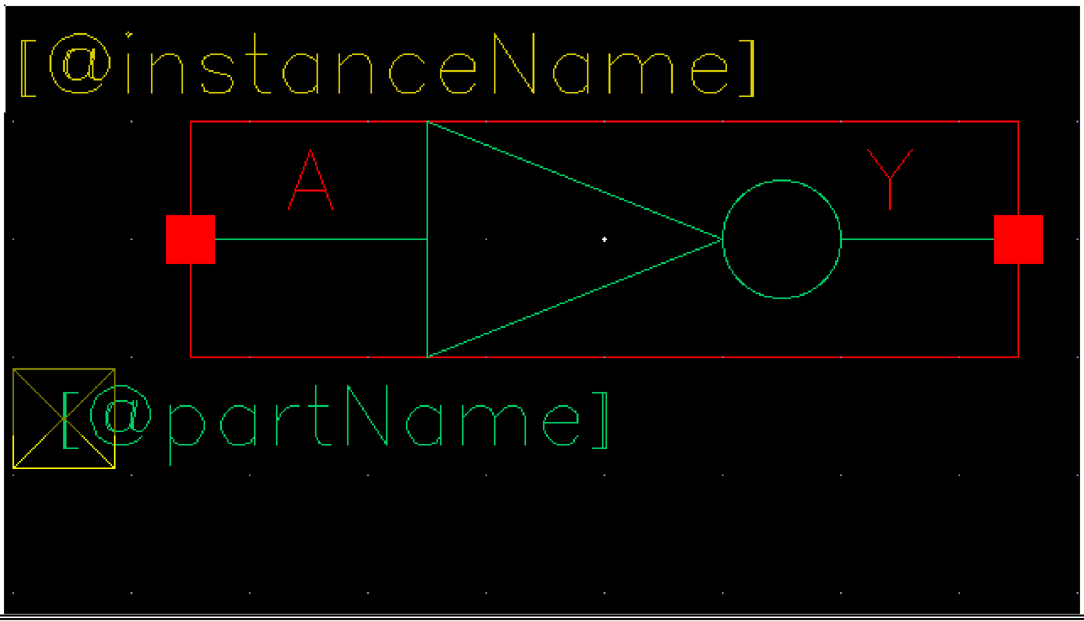
| | |
|---|---|
| **A → Y↑** | **636.4 pS** |
| **A → Y↓** | **459.4 pS** |

**Output Rise Time: 856.6 pS**

**Output Fall Time: 612.5 pS**

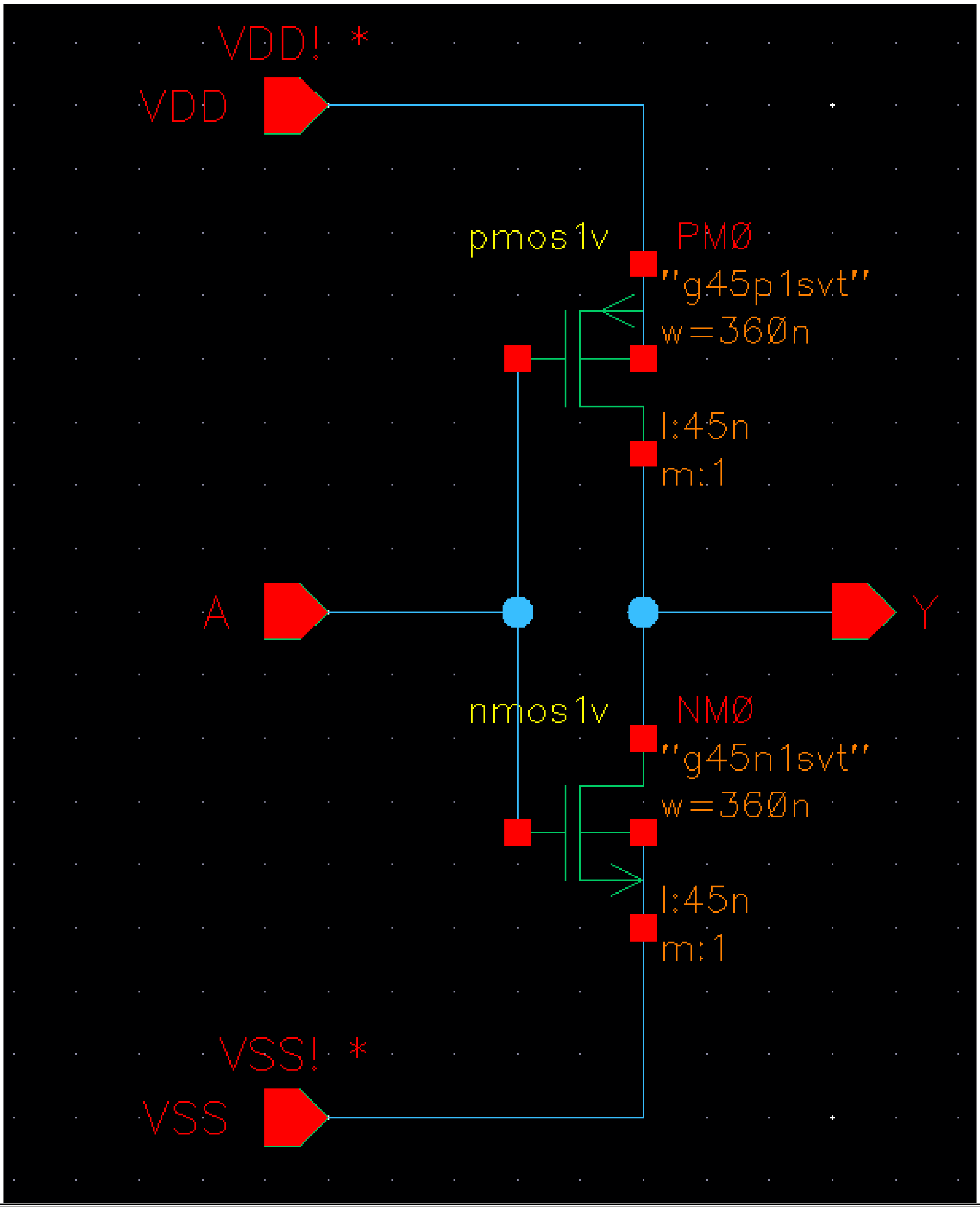
**Layout Area: 1.71 µm X 0.6 µm = 1.026 µm$^2$**

**Symbol with Port Names:**



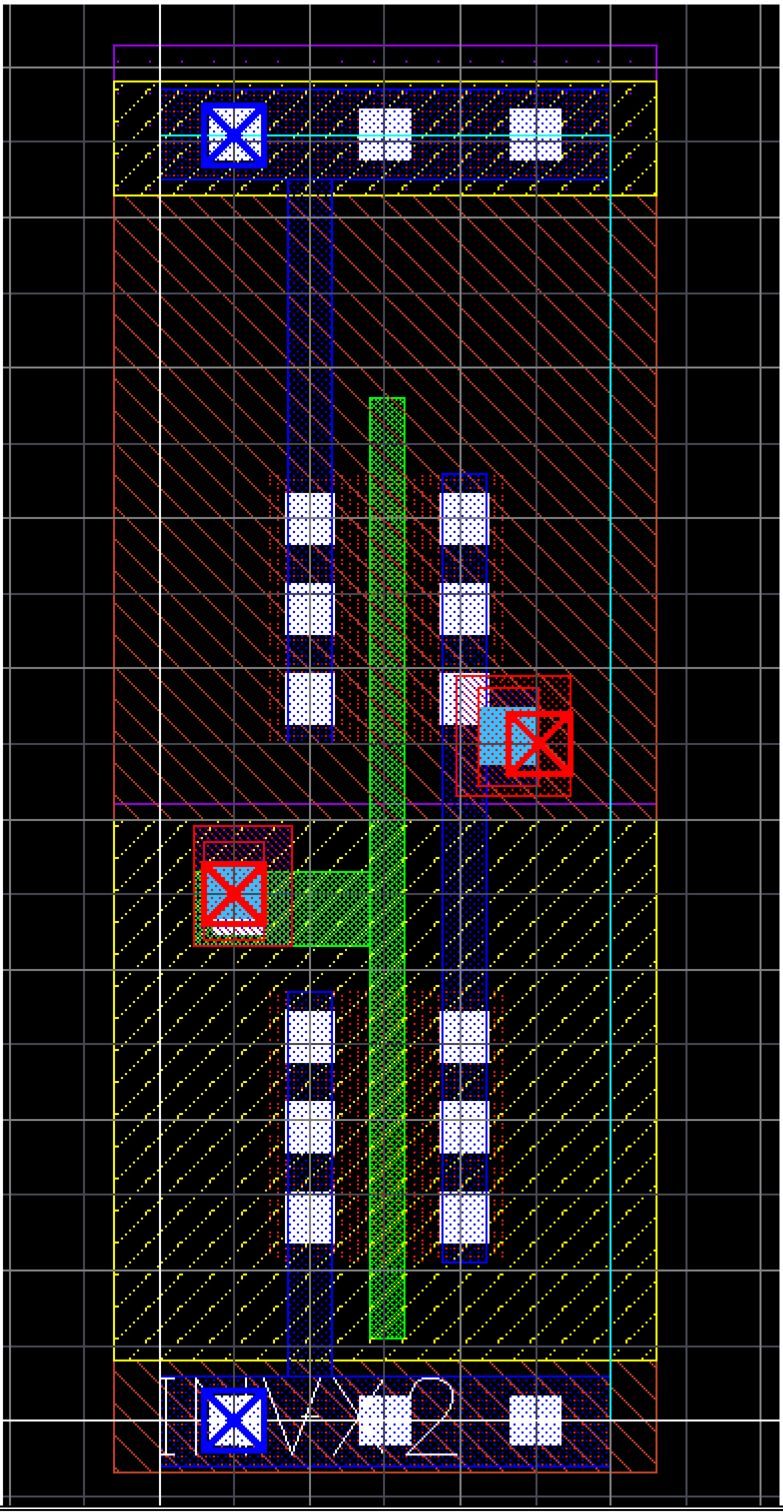
Name: Connor Goldberg

**Schematic:**



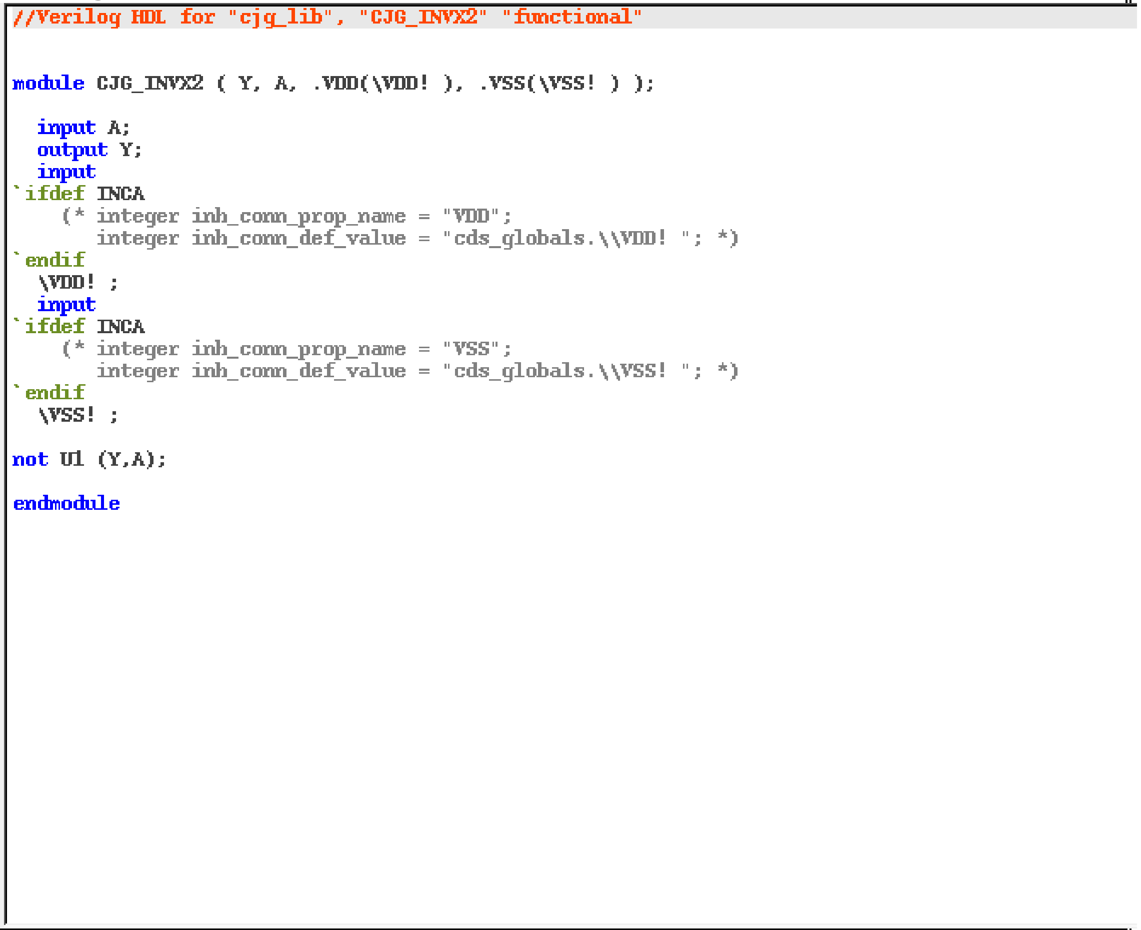
Name: Connor Goldberg

**Layout:**



Name:_Connor Goldberg__

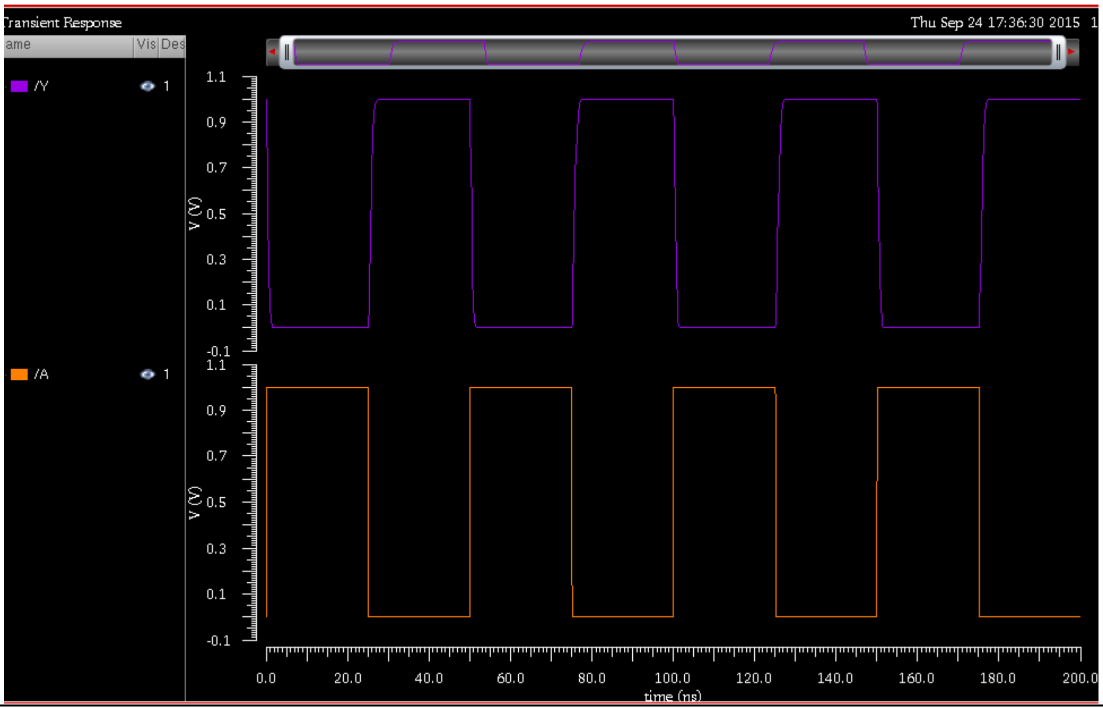
## Verilog Model:



```
//Verilog HDL for "cjg_lib", "CJG_INVX2" "functional"


module CJG_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
```

Name:_Connor Goldberg__

## Functional Simulation Waveforms:



## Comments/Notes:

24 September 2015

Name: Connor Goldberg

| | |
|---|---|
| **Library Name:** | **cjg_lib** |
| **Cell Name:** | **CJG_NAND2X1** |

**Function/Truth Table:**

| **A** | **B** | **Y** |
|---|---|---|
| **0** | **0** | **1** |
| **0** | **1** | **1** |
| **1** | **0** | **1** |
| **1** | **1** | **0** |

**Propagation Delay:**

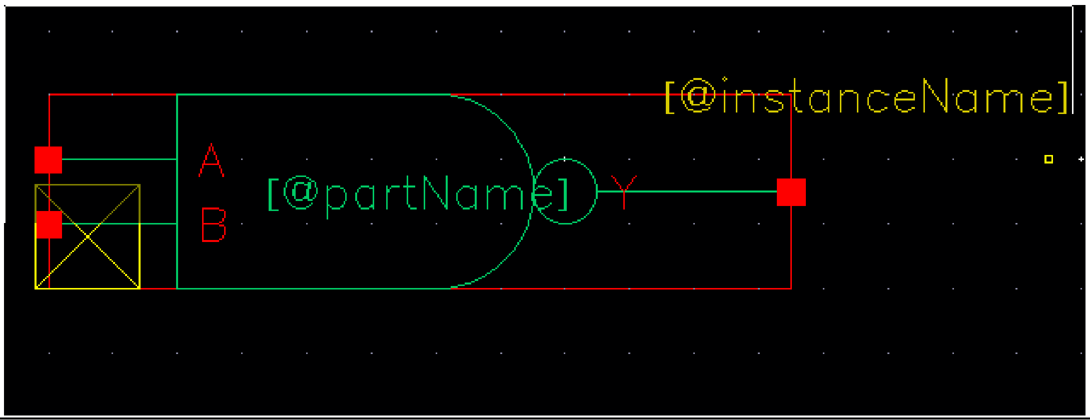
| | |
|---|---|
| **A → Y↑** | **1.206 nS** |
| **A → Y↓** | **854.0 pS** |
| **B → Y↑** | **1.200 nS** |
| **B → Y↓** | **862.7 pS** |

**Output Rise Time: 1.653 nS**

**Output Fall Time: 1.115 nS**

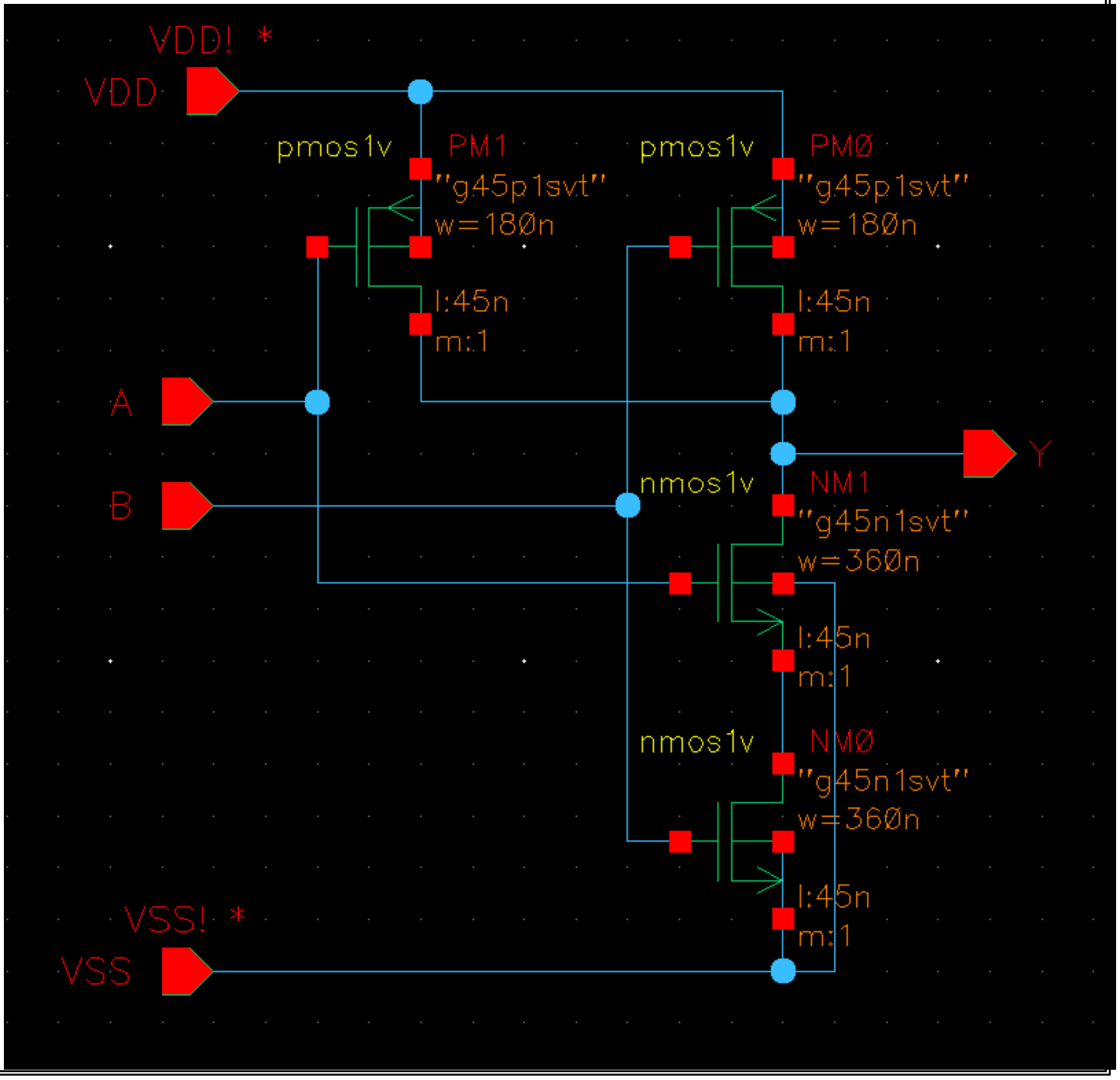
**Layout Area: 1.71 µm X 0.8 µm = 1.368 µm²**

**Symbol with Port Names:**



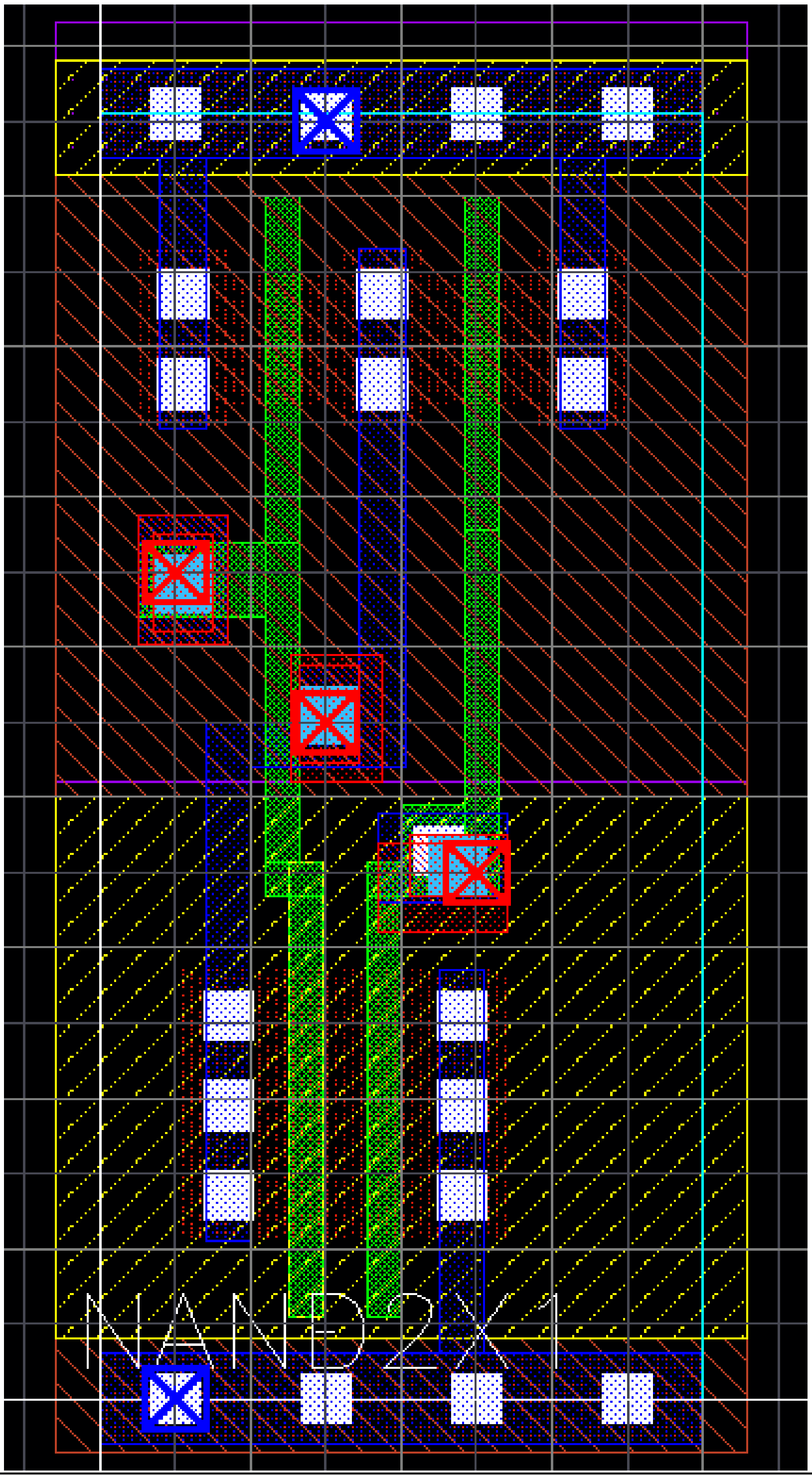
Name:_Connor Goldberg__

**Schematic:**



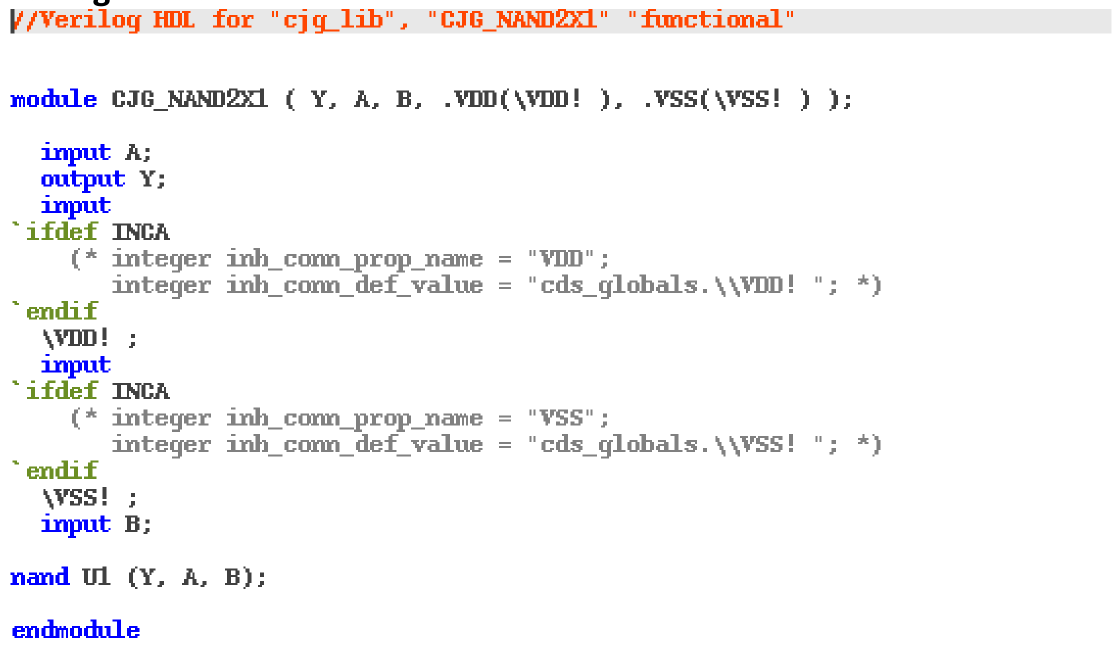
Name: Connor Goldberg

**Layout:**



Name: Connor Goldberg

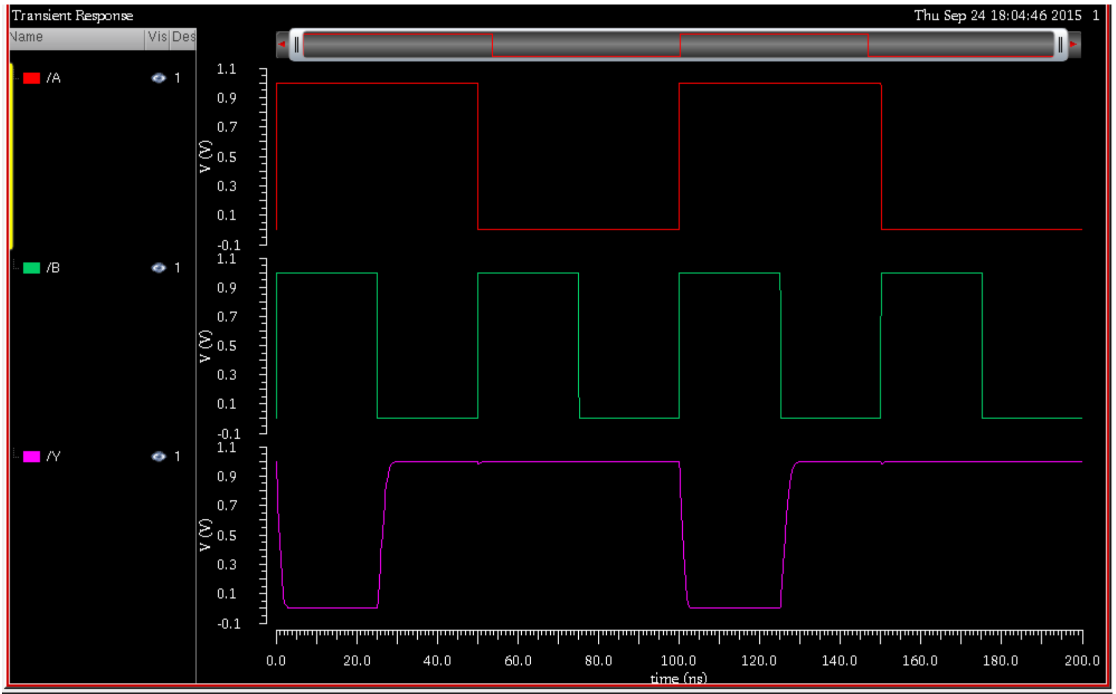
## Verilog Model:



```
//Verilog HDL for "cjg_lib", "CJG_NAND2X1" "functional"


module CJG_NAND2X1 ( 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;

nand U1 (Y, A, B);

endmodule
```

Name: Connor Goldberg

## Functional Simulation Waveforms:



## Comments/Notes:

Name: Connor Goldberg

| | |
|---|---|
| **Library Name:** | **cjg_lib** |
| **Cell Name:** | **CJG_NOR2X1** |

**Function/Truth Table:**

| A | B | Y |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 0 |

**Propagation Delay:**

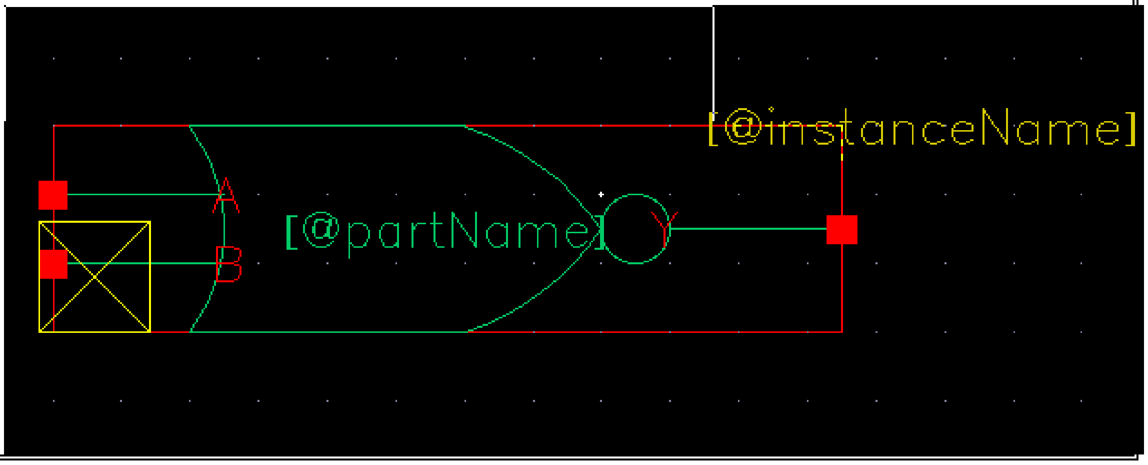
| | |
|---|---|
| A → Y↑ | 1.291 nS |
| A → Y↓ | 774.0 pS |
| B → Y↑ | 1.306 nS |
| B → Y↓ | 780.4 pS |

**Output Rise Time: 1.739 nS**

**Output Fall Time: 528.8 pS**

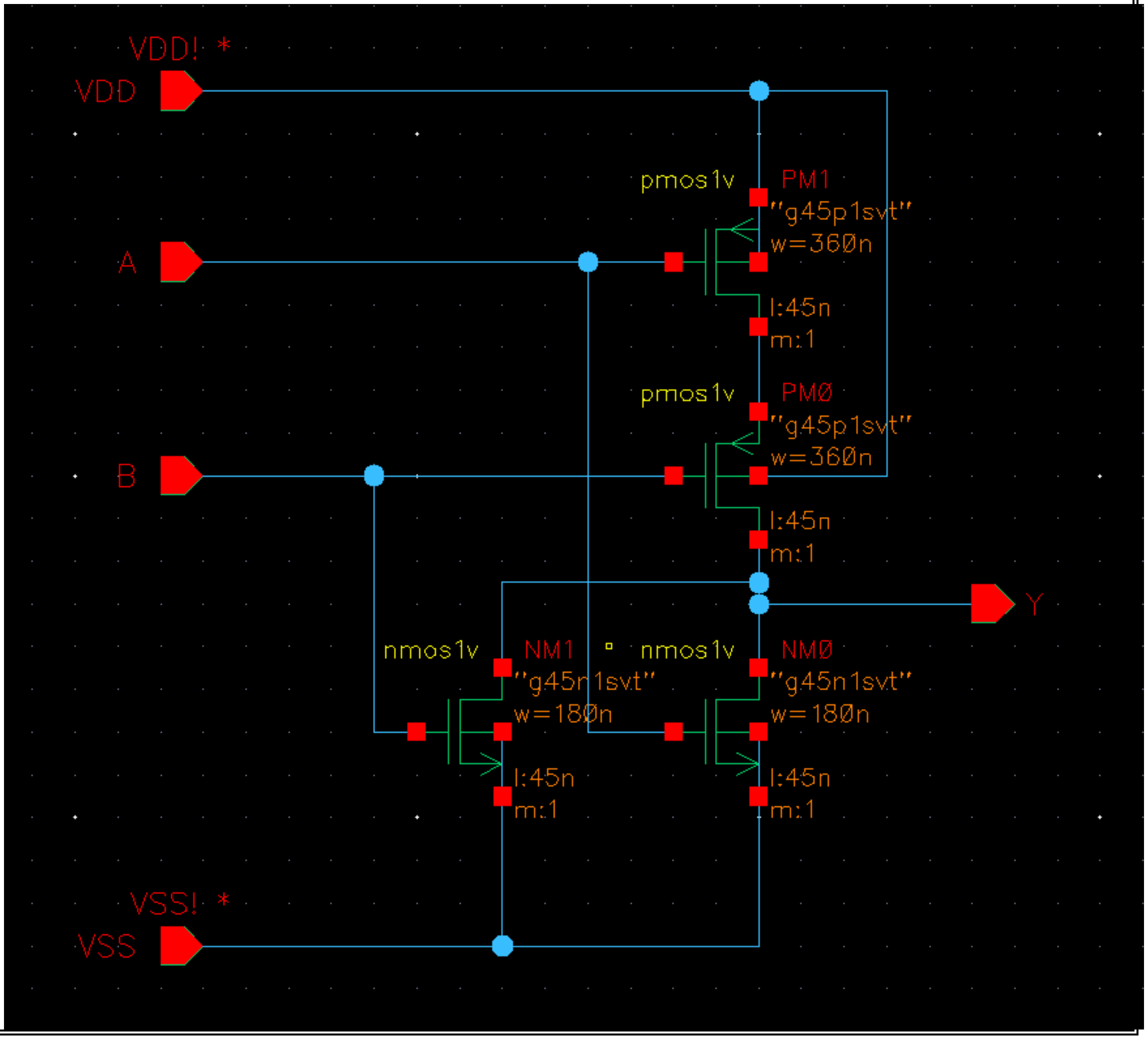
**Layout Area: 1.71 µm X 0.8 µm = 1.368 µm$^2$**

**Symbol with Port Names:**

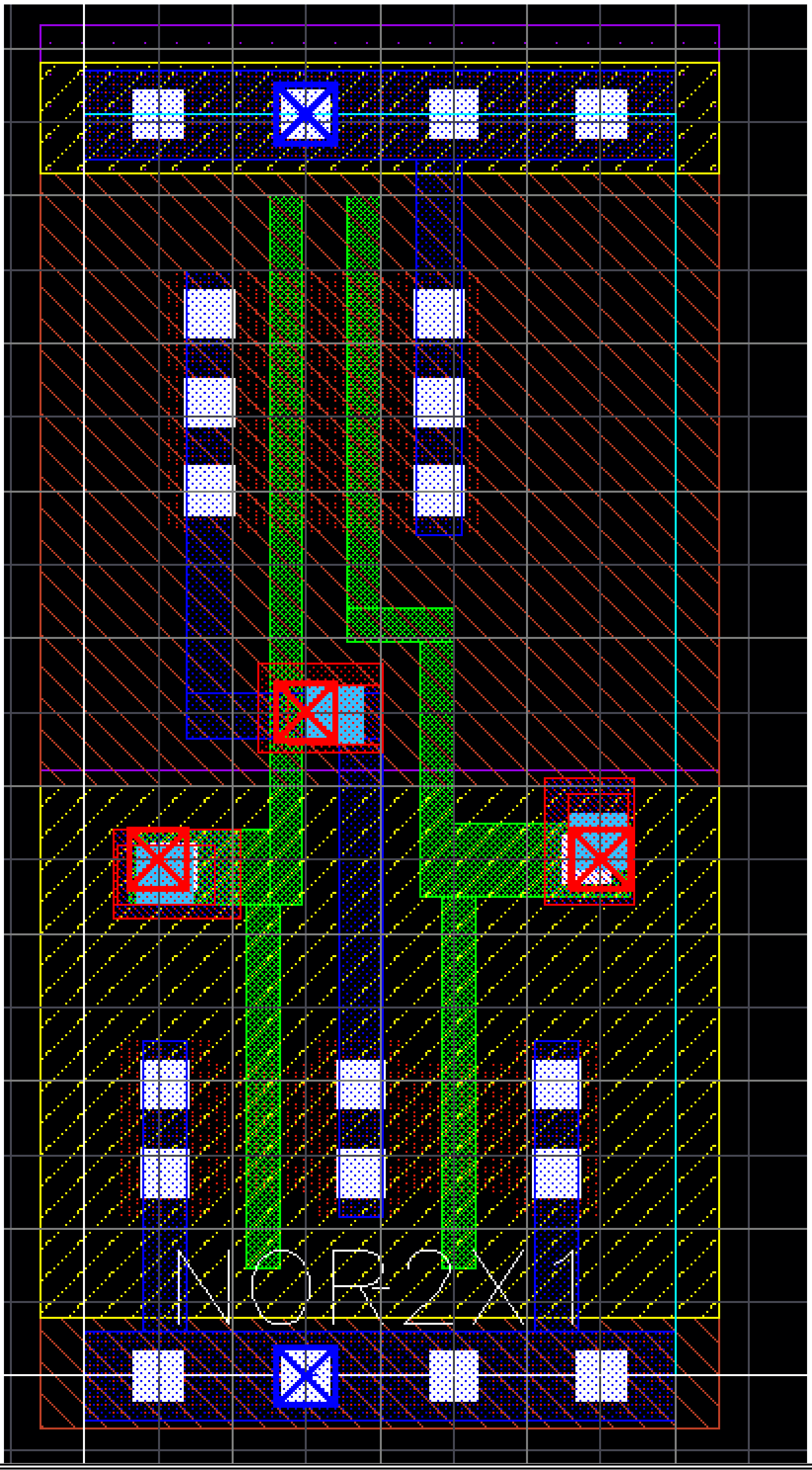


Name:_Connor Goldberg__

**Schematic:**

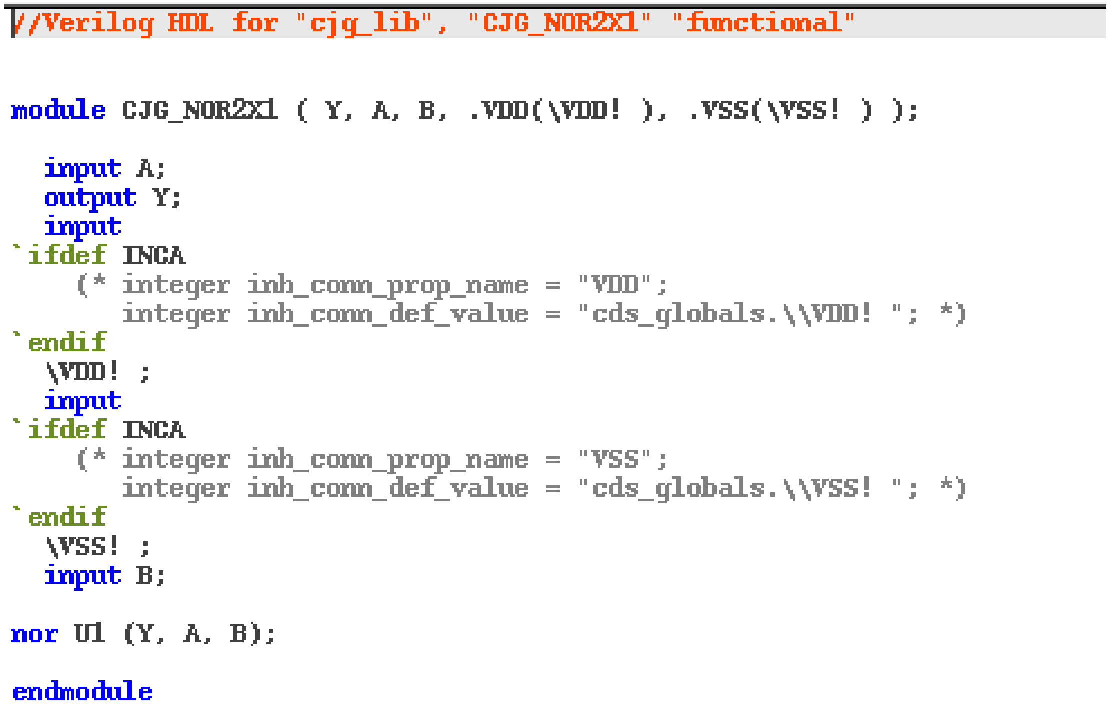


**Layout:**



Name: Connor Goldberg

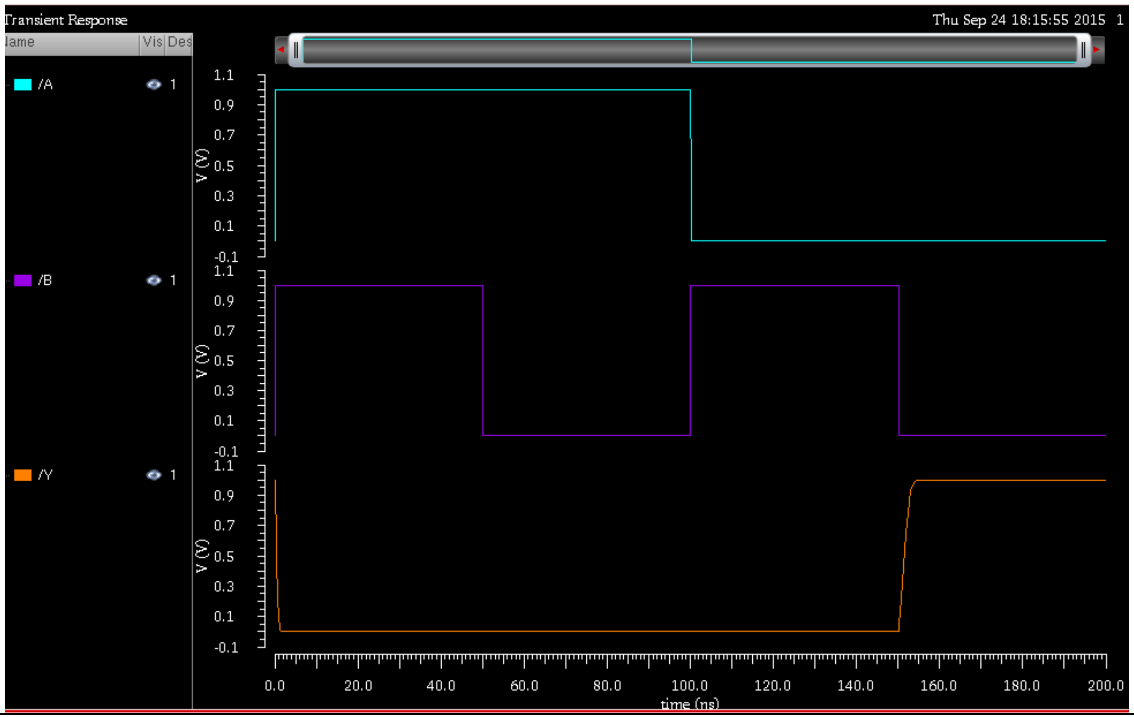
## Verilog Model:



```
//Verilog HDL for "cjg_lib", "CJG_NOR2X1" "functional"


module CJG_NOR2X1 ( 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;

nor U1 (Y, A, B);

endmodule
```

Name:_Connor Goldberg__

## Functional Simulation Waveforms:



## Comments/Notes:

Name: Connor Goldberg

| | |
|---|---|
| **Library Name:** | **cjg_lib** |
| **Cell Name:** | **CJG_MUX2X1** |

**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:**

| | |
|---|---|
| **A → Y↑** | **1.197 nS** |
| **A → Y↓** | **861.1 pS** |
| **B → Y↑** | **1.189 nS** |
| **B → Y↓** | **854.4 pS** |
| **S → Y↑** | **1.190 nS** |
| **S → Y↓** | **849.0 pS** |

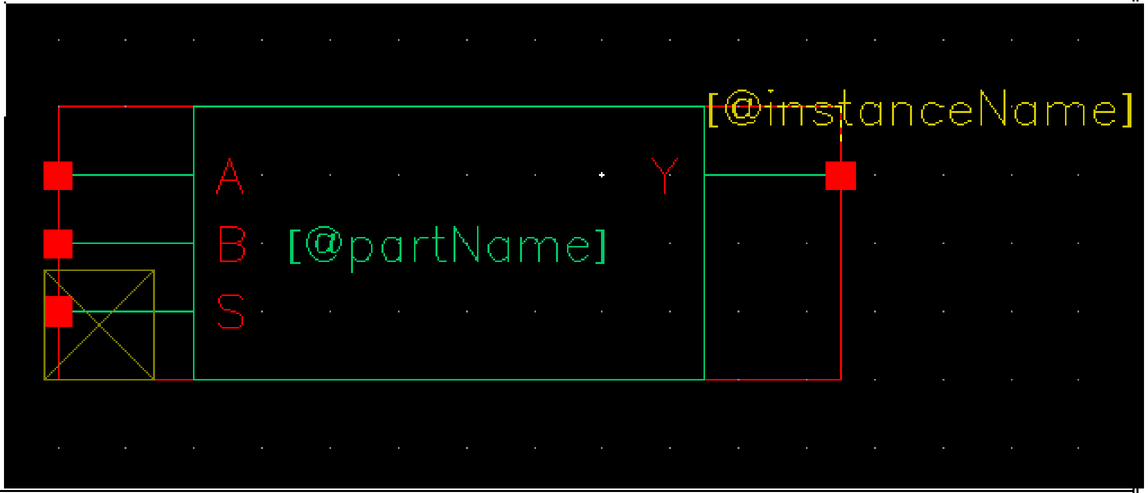
**Output Rise Time: 1.603 nS**

**Output Fall Time: 1.108 nS**

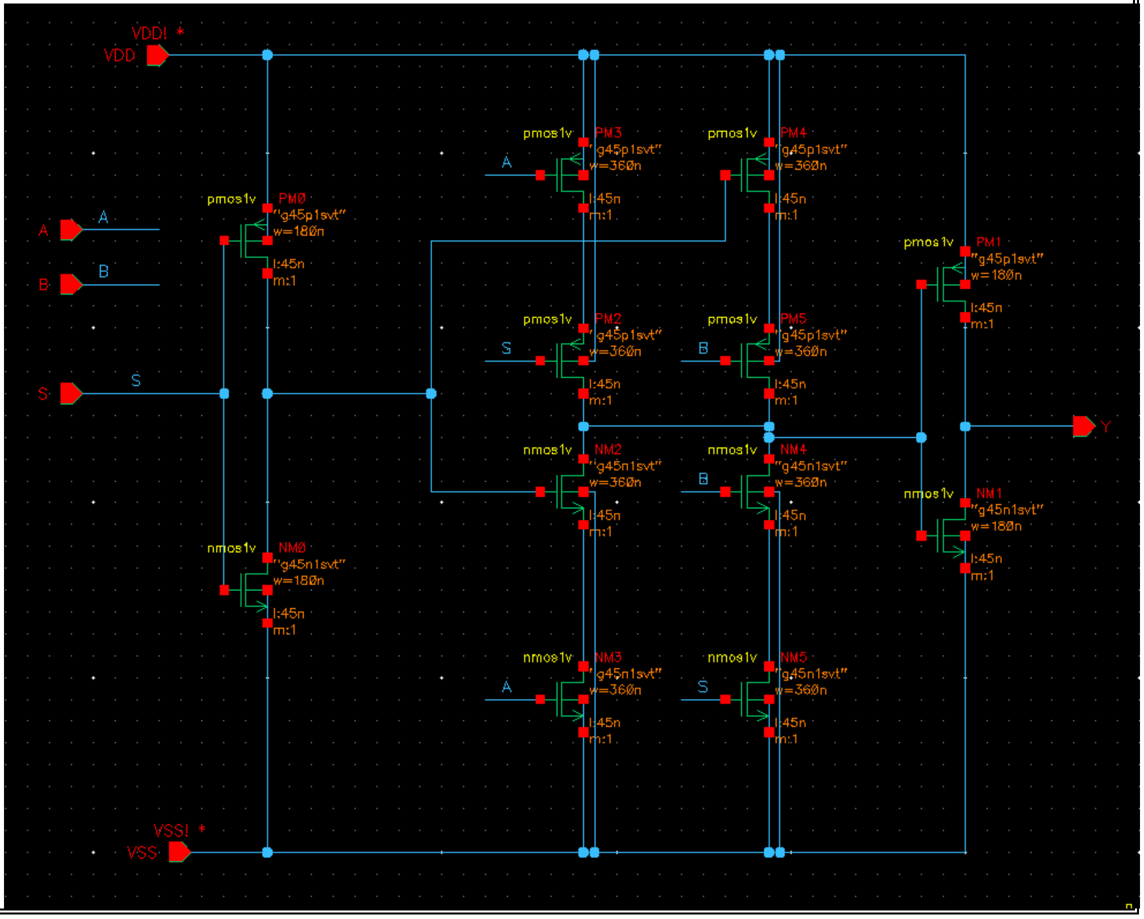
**Layout Area: 1.71 µm X 2 µm = 3.42 µm$^2$**

Name: Connor Goldberg

## Symbol with Port Names:

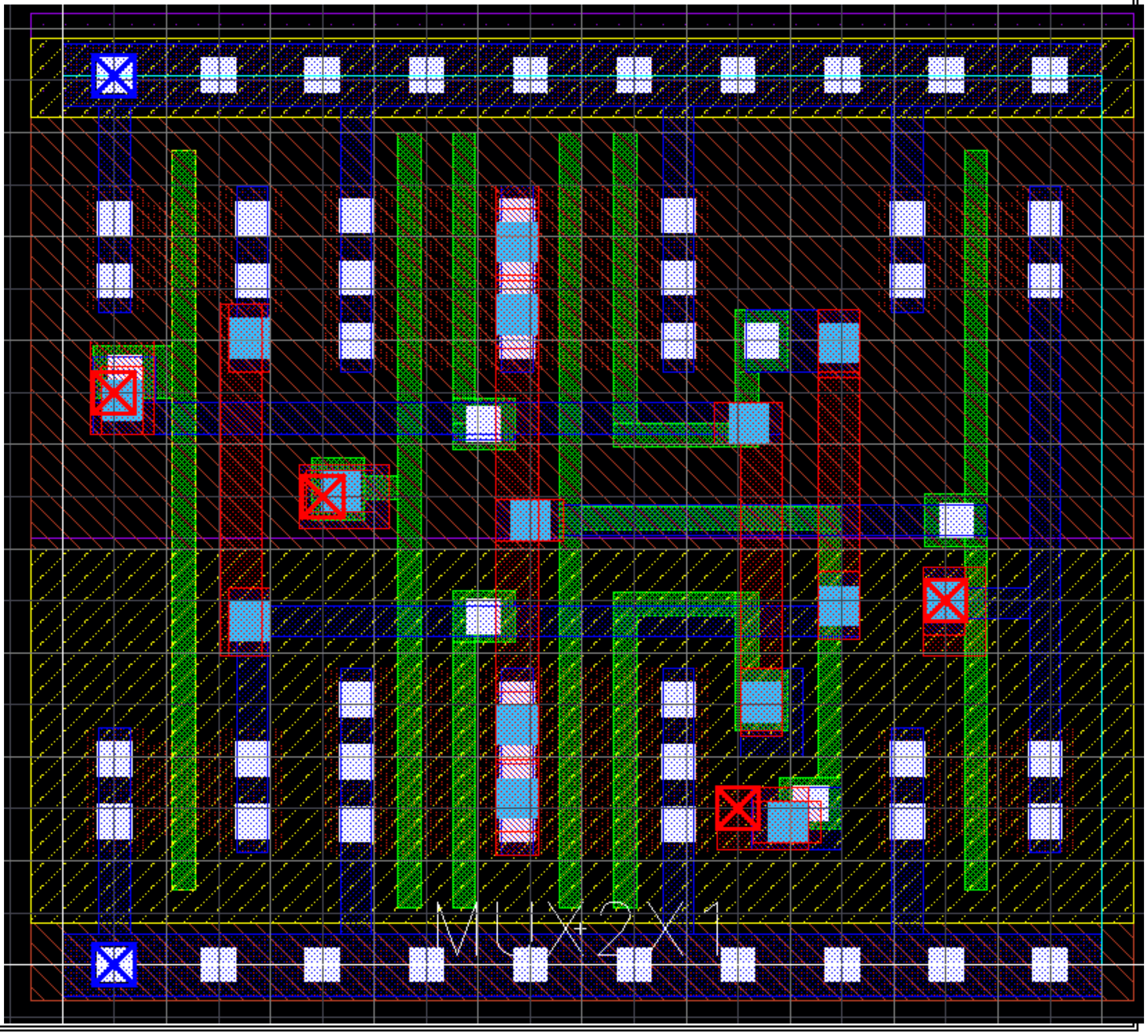


## Schematic:


VDD! *
VDD
VSS! *
VSS
A
B
S
Y
pmos1v PM0 "g45p1svt" w=180n l:45n m:1
nmos1v NM0 "g45n1svt" w=180n l:45n m:1
pmos1v PM3 "g45p1svt" w=360n l:45n m:1
pmos1v PM4 "g45p1svt" w=360n l:45n m:1
pmos1v PM2 "g45p1svt" w=360n l:45n m:1
pmos1v PM5 "g45p1svt" w=360n l:45n m:1
nmos1v NM2 "g45n1svt" w=360n l:45n m:1
nmos1v NM4 "g45n1svt" w=360n l:45n m:1
nmos1v NM3 "g45n1svt" w=360n l:45n m:1
nmos1v NM5 "g45n1svt" w=360n l:45n m:1
pmos1v PM1 "g45p1svt" w=180n l:45n m:1
nmos1v NM1 "g45n1svt" w=180n l:45n m:1

Name: _Connor Goldberg__

**Layout:**



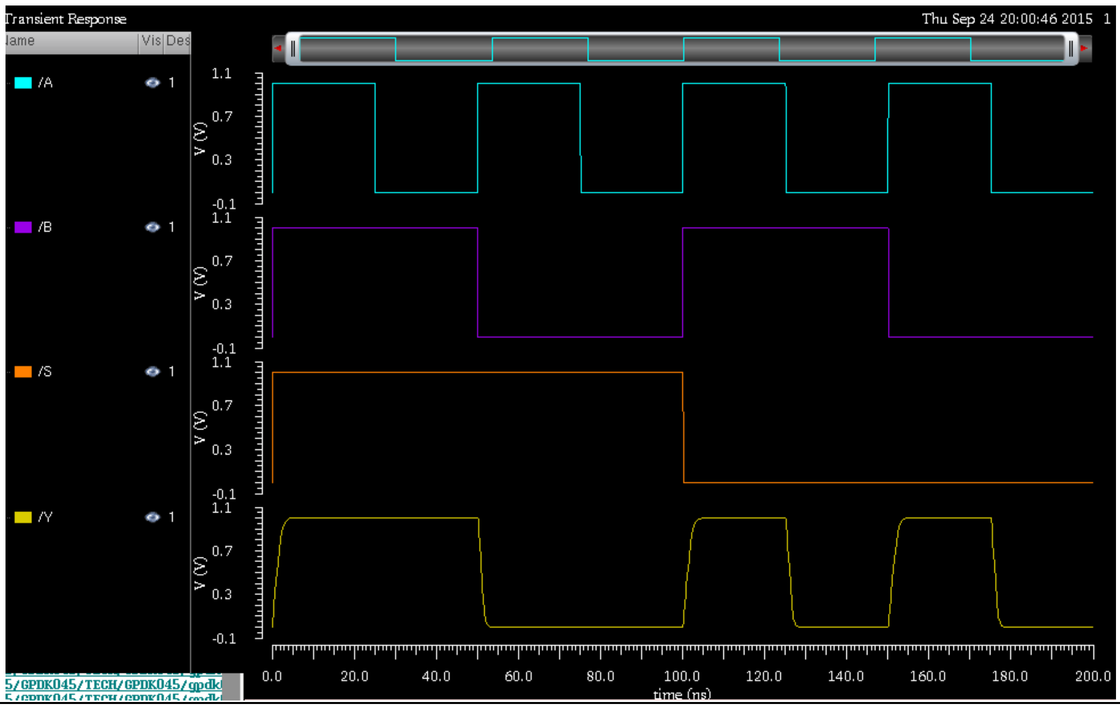
Name: Connor Goldberg

## Verilog Model:

```
//Verilog HDL for "cjg_lib", "CJG_MUX2X1" "functional"


module CJG_MUX2X1 ( Y, A, B, S, .VDD(\VDD! ), .VSS(\VSS! ) );

   input A;
   input S;
   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;

wire and1, and2;

and U1 (and1, A, ~S);
and U2 (and2, B, S);
or U3 (Y, and1, and2);


endmodule
```

Name:_Connor Goldberg__

## Functional Simulation Waveforms:



## Comments/Notes:

24 September 2015

Name: Connor Goldberg

| | |
|---|---|
| **Library Name:** | **cjg_lib** |
| **Cell Name:** | **CJG_XOR2X1** |

**Function/Truth Table:**

| A | B | Y |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

**Propagation Delay:**

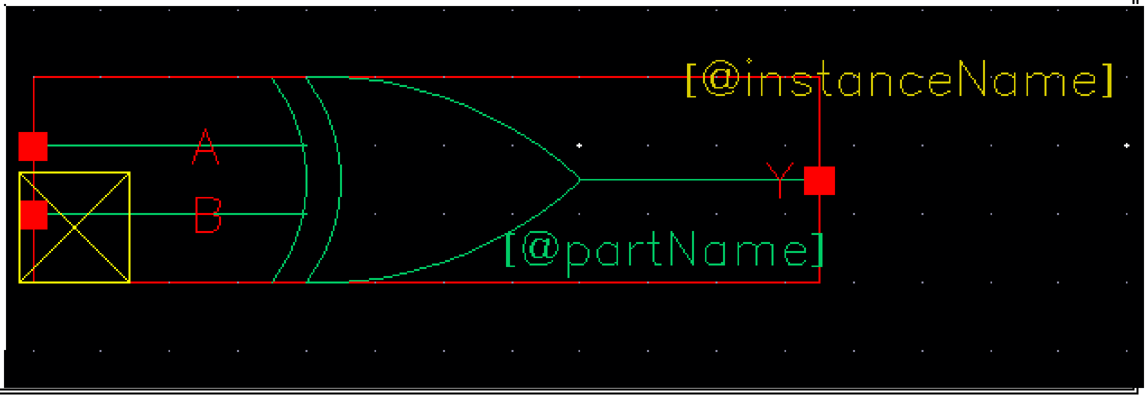
| | |
|---|---|
| A → Y↑ | 1.374 nS |
| A → Y↓ | 838.7 pS |
| B → Y↑ | 1.365 nS |
| B → Y↓ | 841.0 pS |

**Output Rise Time: 1.816 nS**

**Output Fall Time: 1.050 nS**

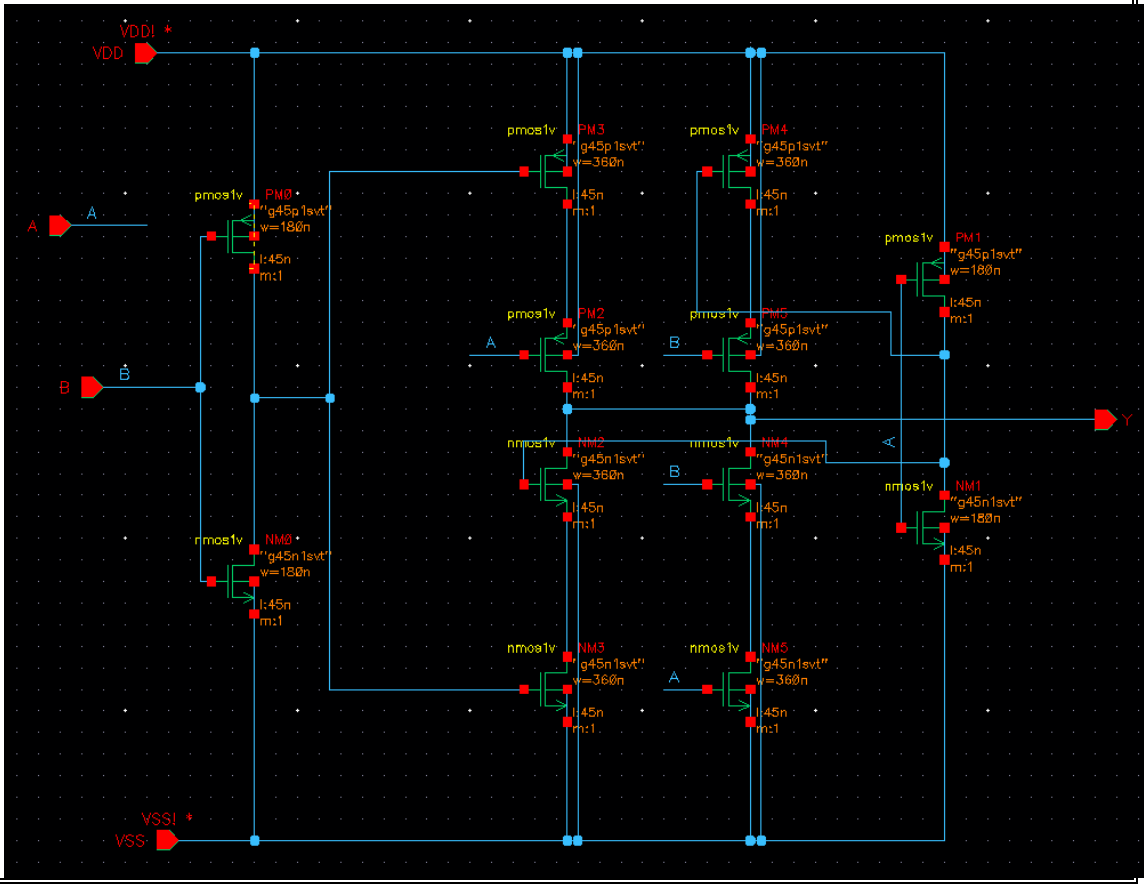
**Layout Area: 1.71 µm X 2 µm = 3.42 µm$^2$**

**Symbol with Port Names:**



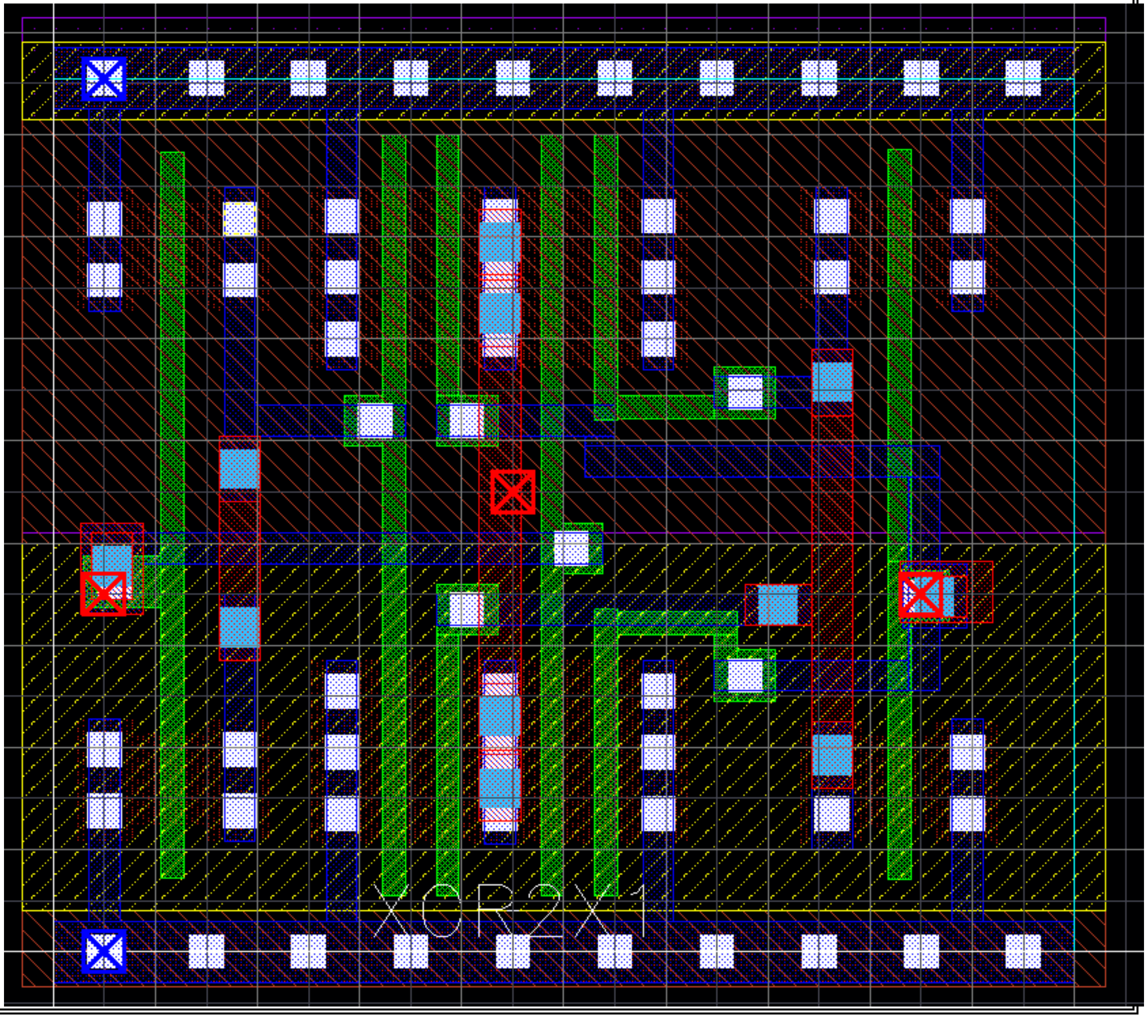
Name: Connor Goldberg

## Schematic:


VDD! *
VDD
A
B
Y
VSS! *
VSS
pmos1v PM0 "g45p1svt" w=180n l:45n m:1
nmos1v NM0 "g45n1svt" w=180n l:45n m:1
pmos1v PM3 "g45p1svt" w=360n l:45n m:1
pmos1v PM2 "g45p1svt" w=360n l:45n m:1
pmos1v PM4 "g45p1svt" w=360n l:45n m:1
pmos1v PM5 "g45p1svt" w=360n l:45n m:1
pmos1v PM1 "g45p1svt" w=180n l:45n m:1
nmos1v NM2 "g45n1svt" w=360n l:45n m:1
nmos1v NM4 "g45n1svt" w=360n l:45n m:1
nmos1v NM1 "g45n1svt" w=180n l:45n m:1
nmos1v NM3 "g45n1svt" w=360n l:45n m:1
nmos1v NM5 "g45n1svt" w=360n l:45n m:1

Name: Connor Goldberg

**Layout:**

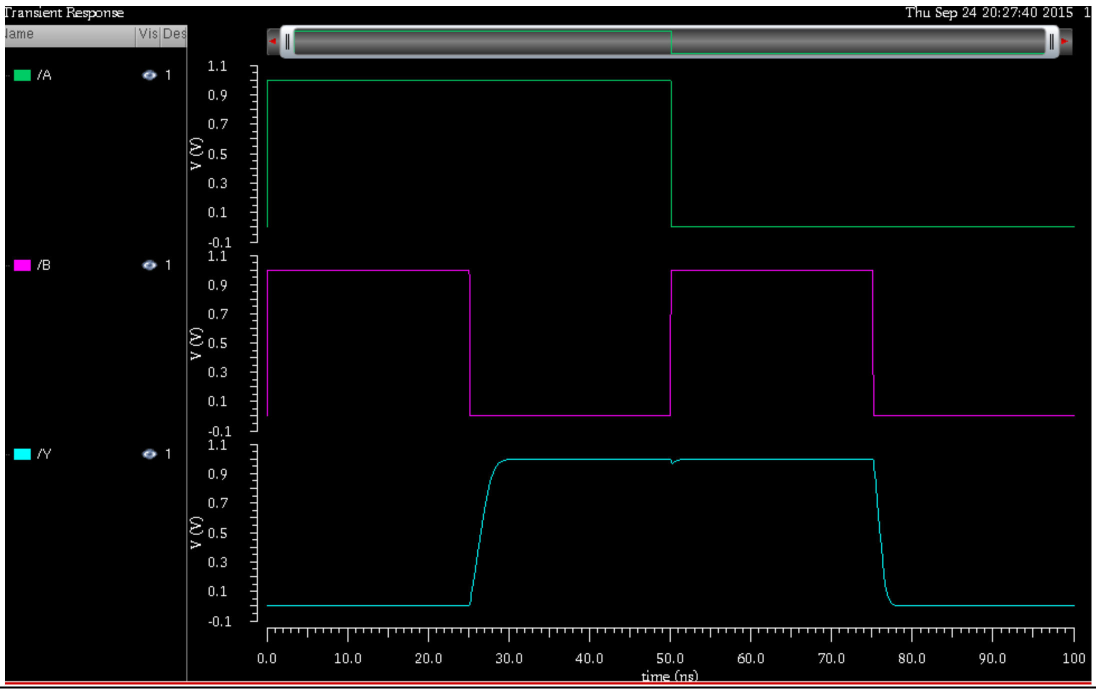


Name:_Connor Goldberg__

## Verilog Model:

```
//Verilog HDL for "cjg_lib", "CJG_XOR2X1" "functional"


module CJG_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:



Name: Connor Goldberg

**Comments/Notes:**

Name: Connor Goldberg

| | |
|---|---|
| **Library Name:** | **cjg_lib** |
| **Cell Name:** | **CJG_AOI22X1** |

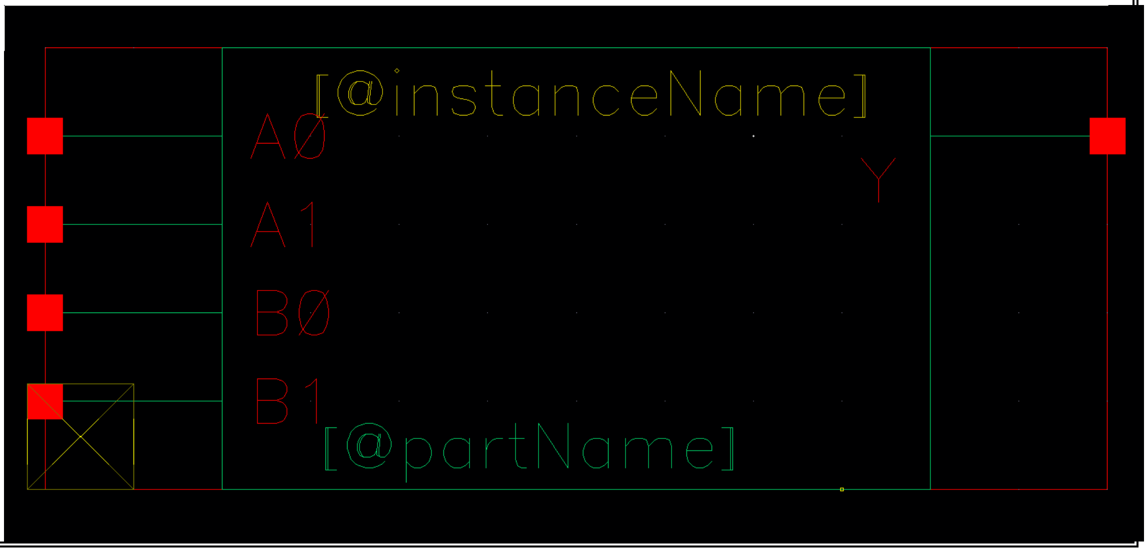
**Function/Truth Table:**

| A0 | A1 | B0 | B1 | Y |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 0 | 1 | 1 |
| 0 | 0 | 1 | 0 | 1 |
| 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 1 | 1 |
| 1 | 0 | 1 | 0 | 1 |
| 1 | 0 | 1 | 1 | 0 |
| 1 | 1 | 0 | 0 | 0 |
| 1 | 1 | 0 | 1 | 0 |
| 1 | 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 | 0 |

**Propagation Delay:**

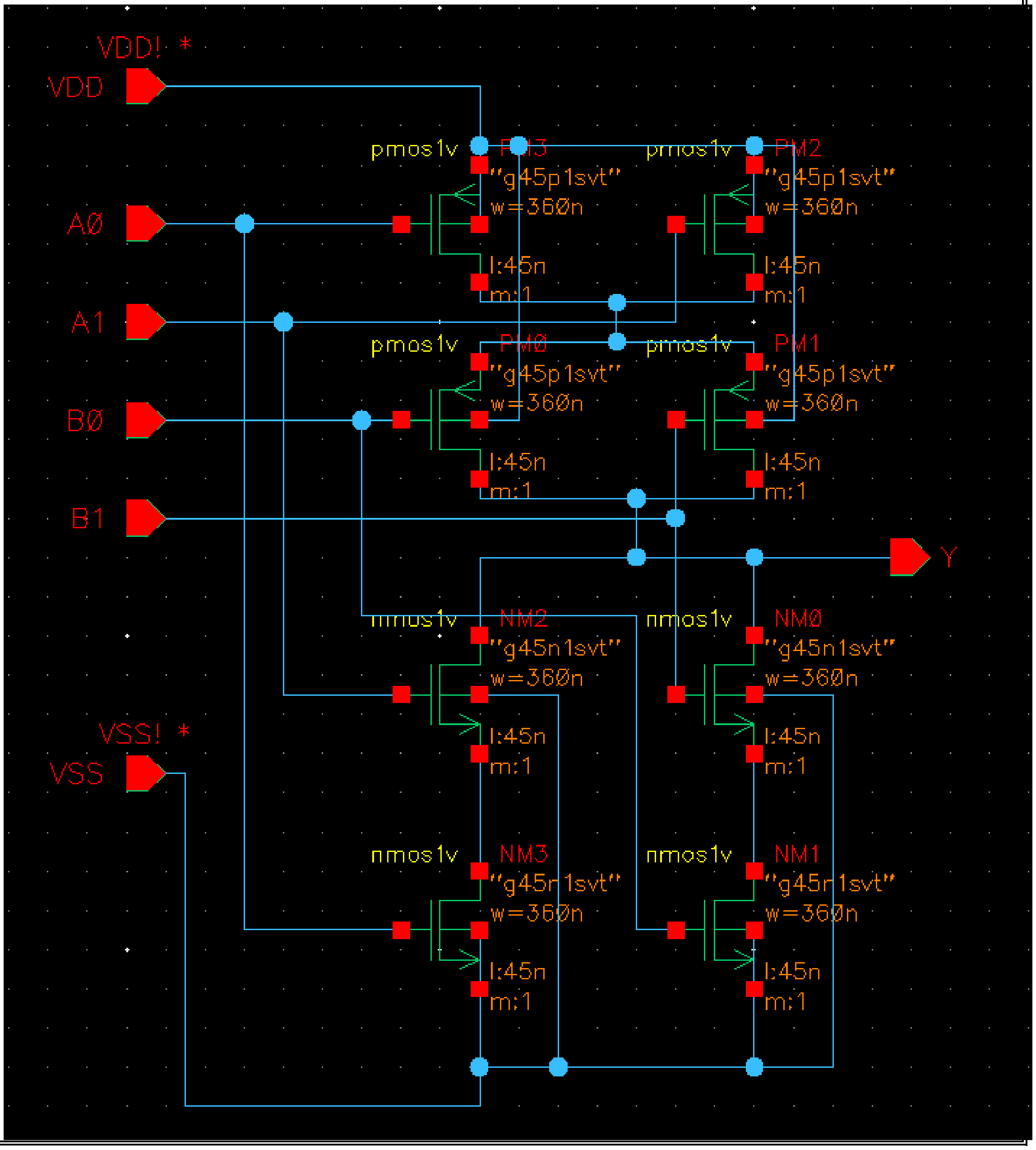
| | |
|---|---|
| A0 → Y↑ | 1.056 nS |
| A0 → Y↓ | 819.8 pS |
| A1 → Y↑ | 1.055 nS |
| A1 → Y↓ | 815.0 pS |
| B0 → Y↑ | 1.317 nS |
| B0 → Y↓ | 813.9 pS |
| B1 → Y↑ | 1.321 nS |
| B1 → Y↓ | 812.8 pS |

Name:_Connor Goldberg___

| |
|---|
| **Output Rise Time: 1.79 nS** |
| **Output Fall Time: 539.6 pS** |
| **Layout Area: 1.71 µm X 1.4 µm = 2.394 µm$^2$** |
| **Symbol with Port Names:**  |

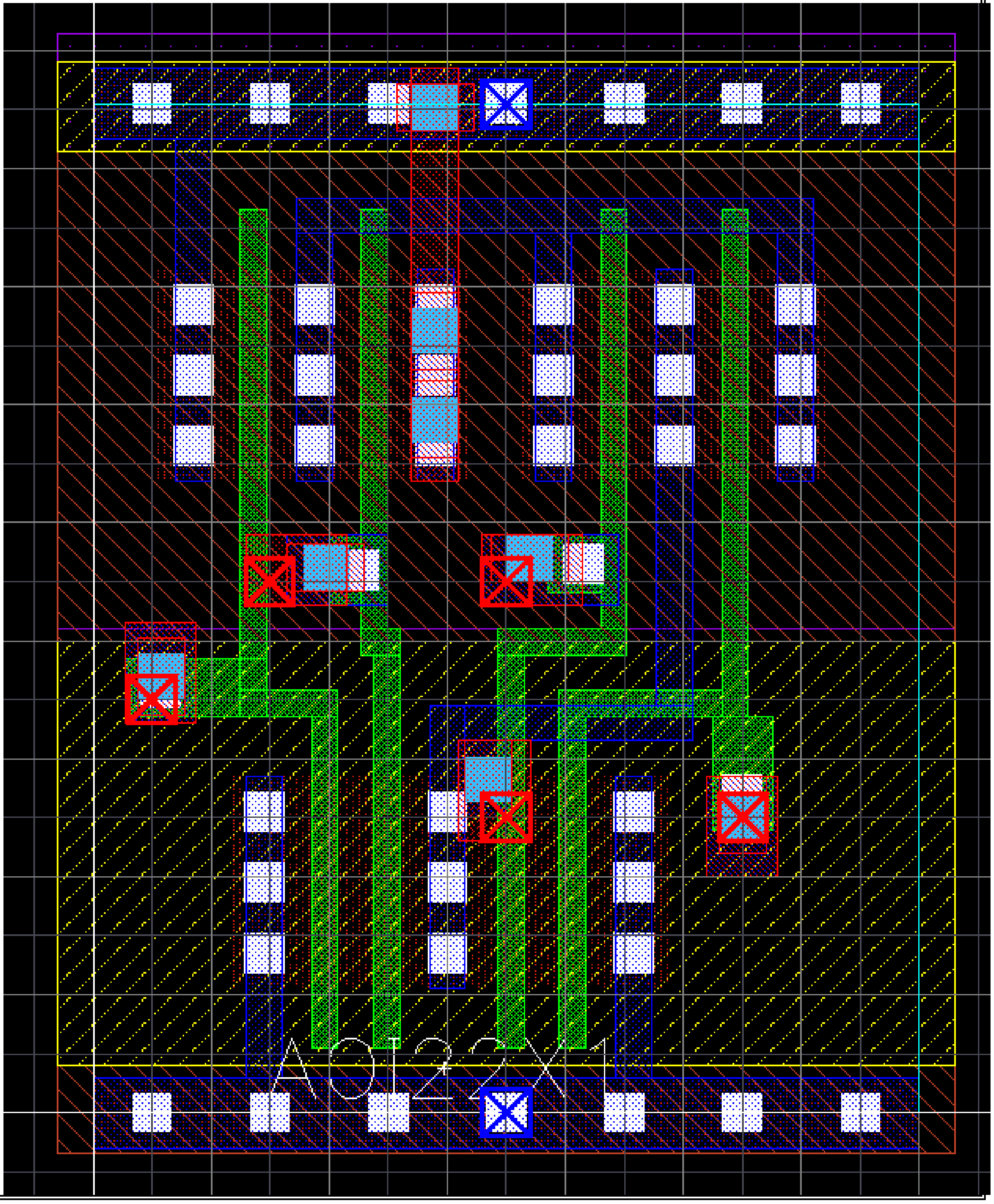
Name: Connor Goldberg

**Schematic:**



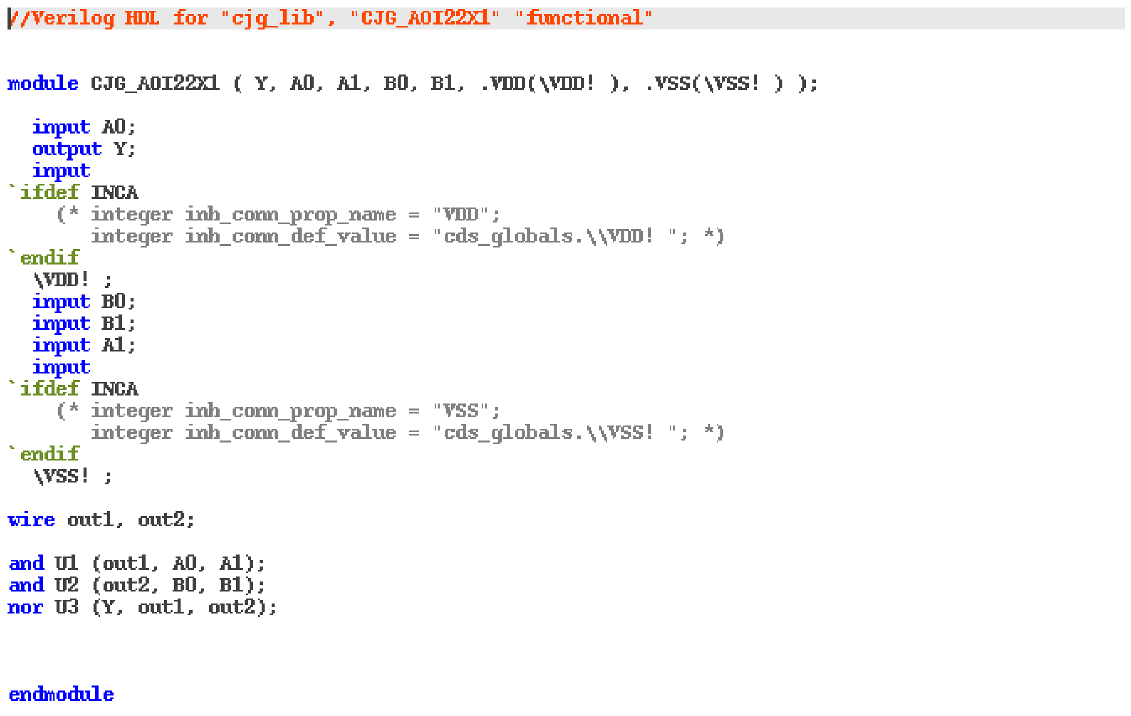
Name:_Connor Goldberg__

**Layout:**



Name: Connor Goldberg

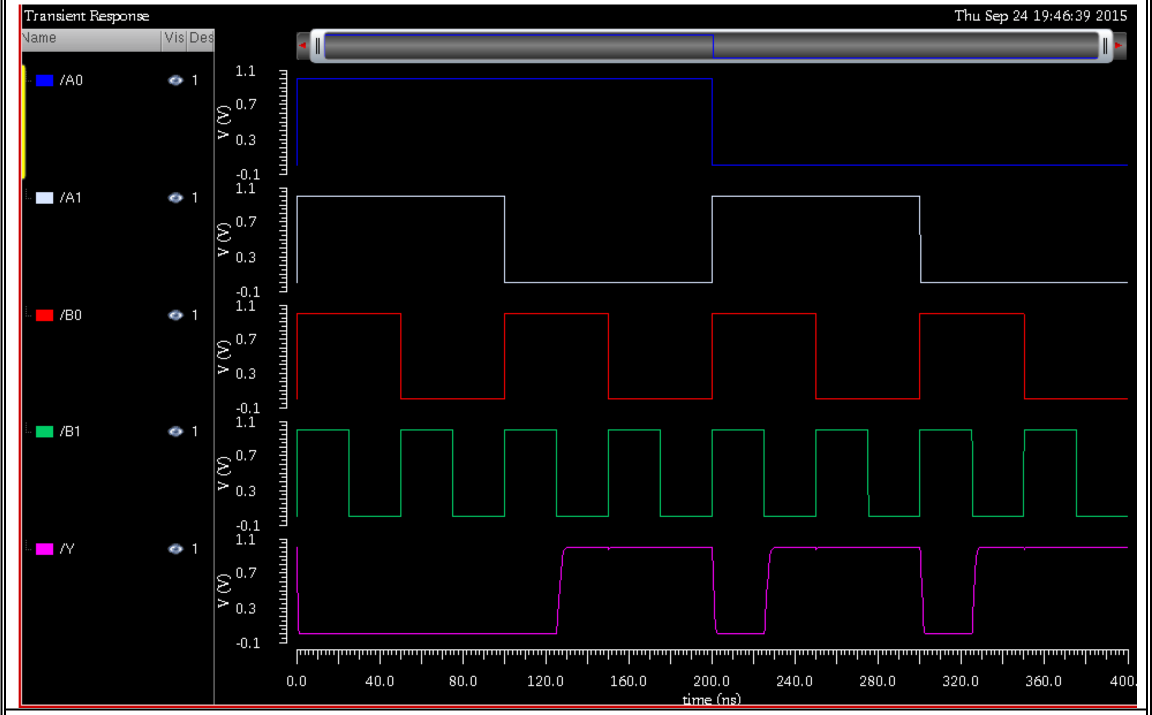
## Verilog Model:



```
//Verilog HDL for "cjg_lib", "CJG_AOI22X1" "functional"


module CJG_AOI22X1 ( 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_globals.\\VDD! "; *)
`endif
  \VDD! ;
  input B0;
  input B1;
  input A1;
  input
`ifdef INCA
    (* integer inh_conn_prop_name = "VSS";
       integer inh_conn_def_value = "cds_globals.\\VSS! "; *)
`endif
  \VSS! ;

wire out1, out2;

and U1 (out1, A0, A1);
and U2 (out2, B0, B1);
nor U3 (Y, out1, out2);



endmodule
```

Name: Connor Goldberg

## Functional Simulation Waveforms:



## Comments/Notes:

Name: Connor Goldberg

| | |
|---|---|
| **Library Name:** | **cjg_lib** |
| **Cell Name:** | **CJG_DFFX1** |

**Function/Truth Table:**

**Assuming a non-overlapping clock with at least the minimum pulse widths and guard time, Q and QN (Q inverted) will reflect the input at the falling edge of PHI2 and be seen at the output on the rising edge of PHI1.**

**Propagation Delay:**

| | |
|---|---|
| **PHI1 → Q↑** | **1.237 nS** |
| **PHI1 → Q↓** | **866.4 pS** |
| **PHI1 → QN↑** | **1.316 nS** |
| **PHI1 → QN↓** | **858.0 pS** |

**Q: Rise Time: 1.639 nS, Fall Time: 1.050 nS**

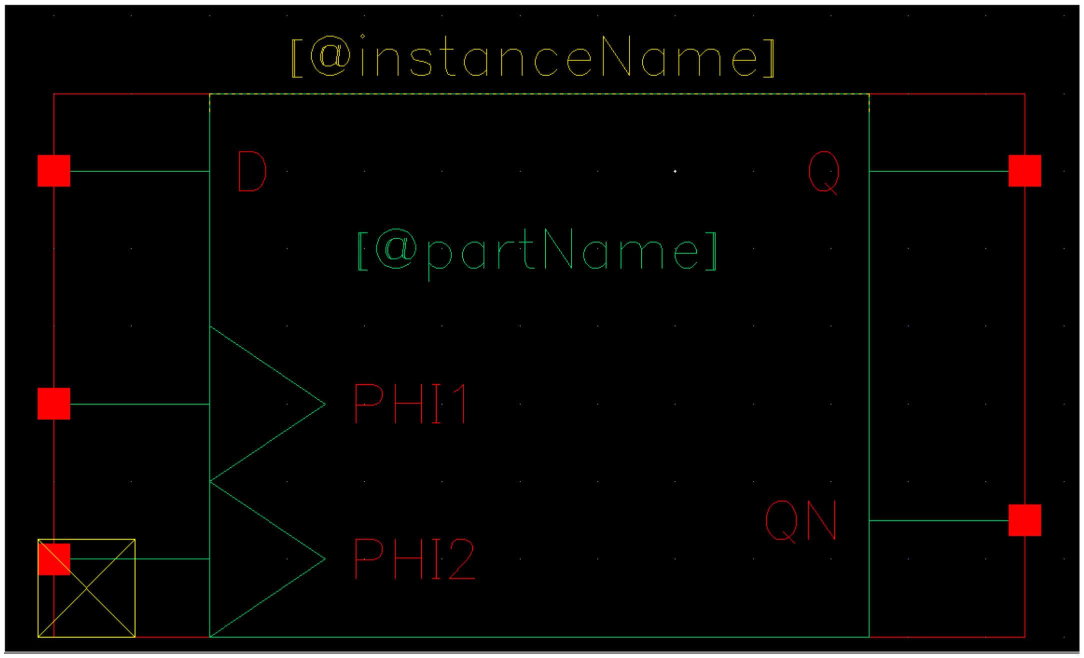
**QN: Rise Time: 1.639 nS, Fall Time: 1.051 nS**

**Minimum Timings:**

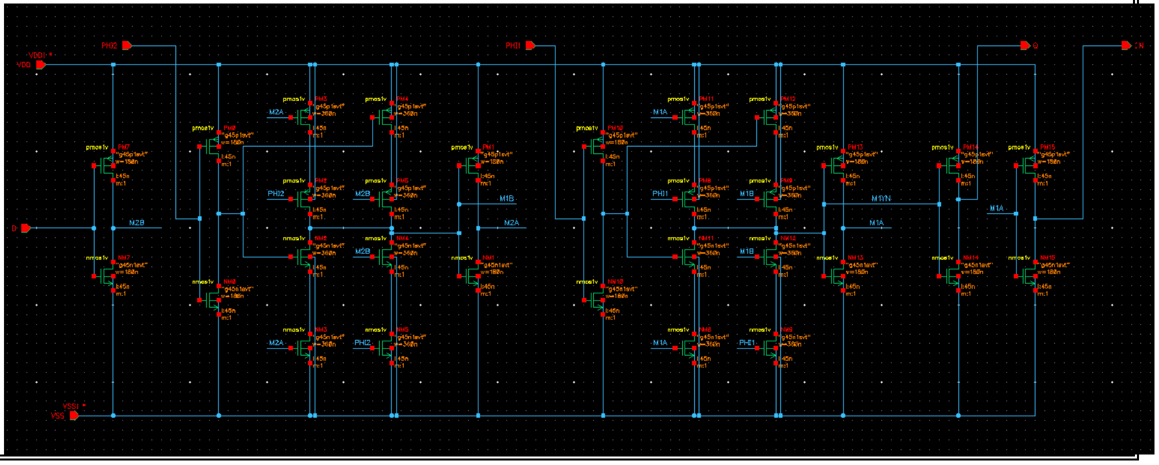
- **PHI1 = 70 pS**
- **PHI2 = 75 pS**
- **Guard Time = 89 pS**

Name:_Connor Goldberg__

## Symbol with Port Names:

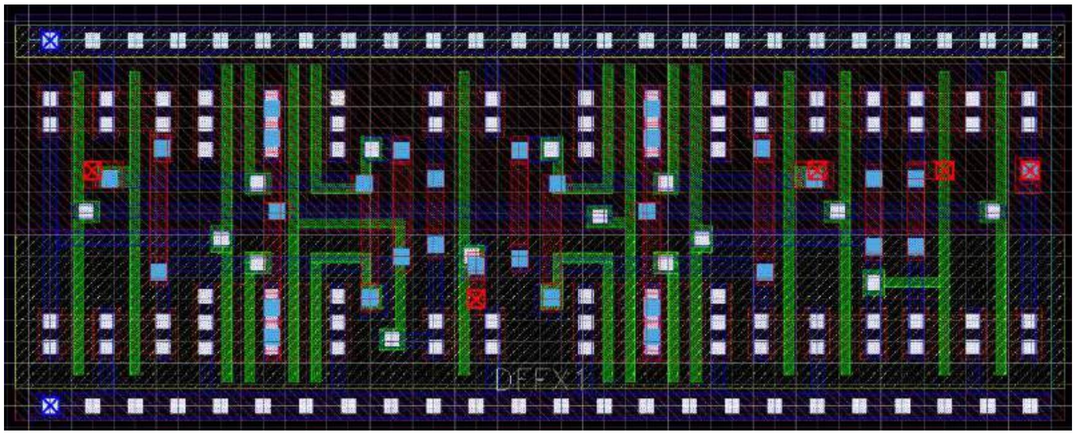


## Schematic:



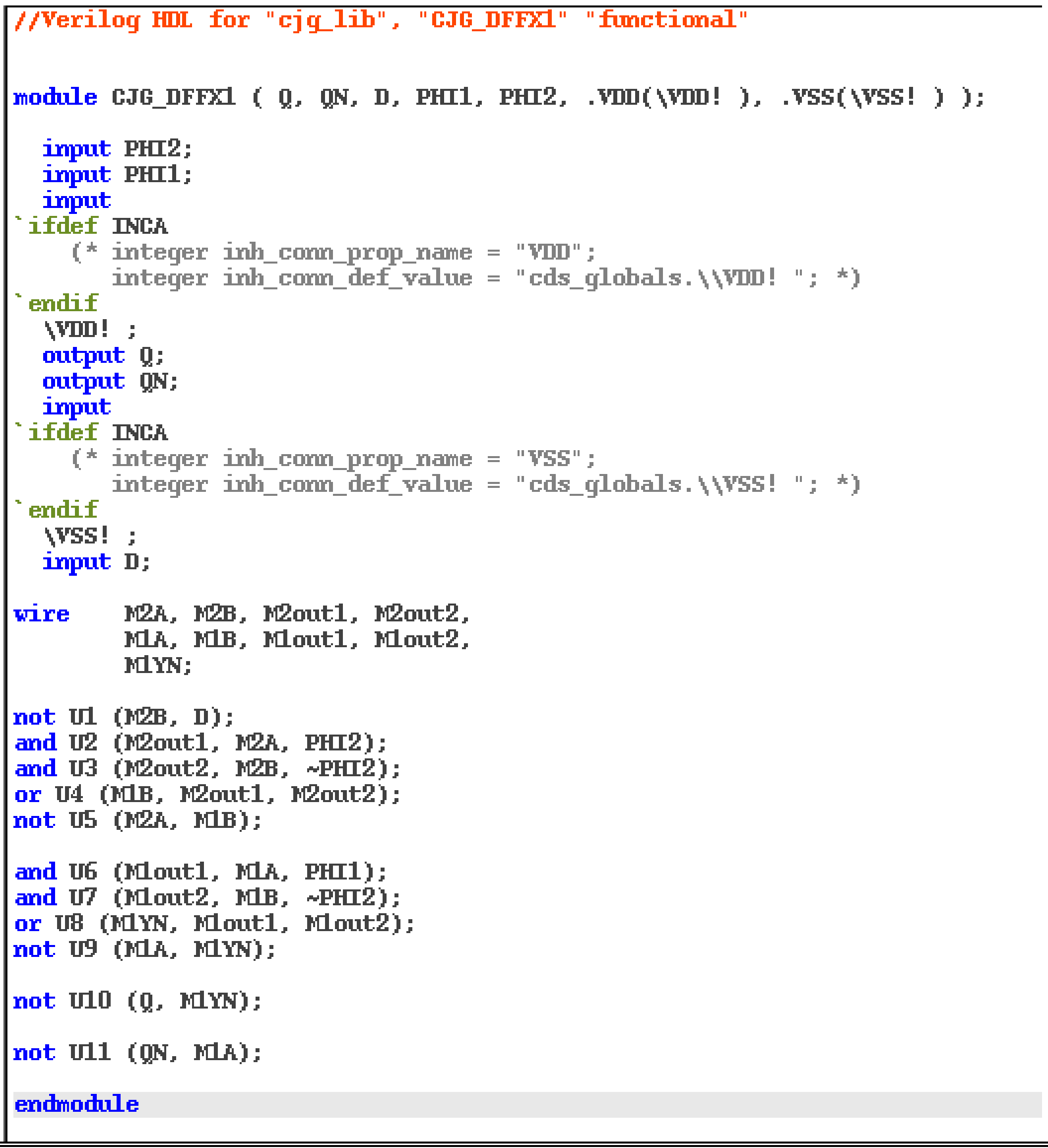
Name:_Connor Goldberg__

**Layout:**



Name: Connor Goldberg

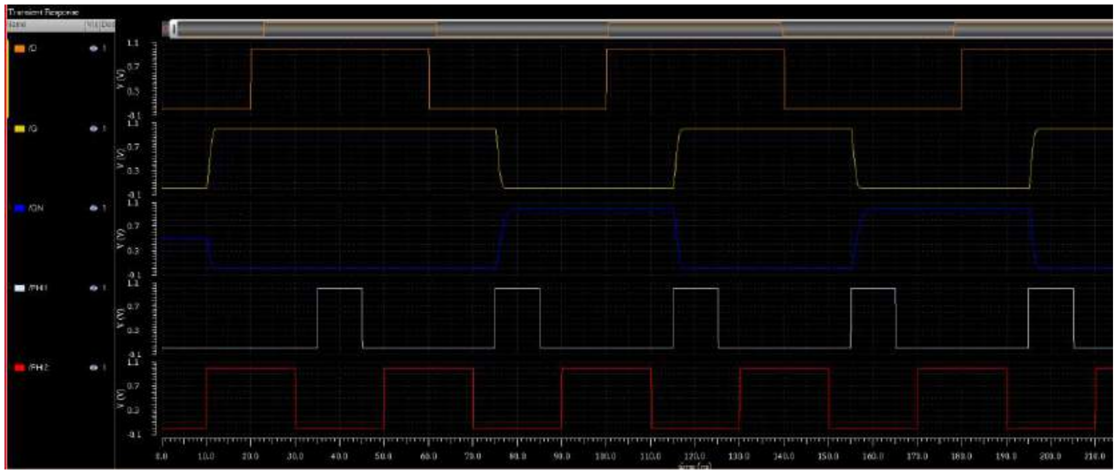
## Verilog Model:



```
//Verilog HDL for "cjg_lib", "CJG_DFFX1" "functional"


module CJG_DFFX1 ( Q, QN, D, PHI1, PHI2, .VDD(\VDD! ), .VSS(\VSS! ) );

  input PHI2;
  input PHI1;
  input
`ifdef INCA
    (* integer inh_conn_prop_name = "VDD";
       integer inh_conn_def_value = "cds_globals.\\VDD! "; *)
`endif
  \VDD! ;
  output Q;
  output QN;
  input
`ifdef INCA
    (* integer inh_conn_prop_name = "VSS";
       integer inh_conn_def_value = "cds_globals.\\VSS! "; *)
`endif
  \VSS! ;
  input D;

wire    M2A, M2B, M2out1, M2out2,
        M1A, M1B, M1out1, M1out2,
        M1YN;

not U1 (M2B, D);
and U2 (M2out1, M2A, PHI2);
and U3 (M2out2, M2B, ~PHI2);
or U4 (M1B, M2out1, M2out2);
not U5 (M2A, M1B);

and U6 (M1out1, M1A, PHI1);
and U7 (M1out2, M1B, ~PHI2);
or U8 (M1YN, M1out1, M1out2);
not U9 (M1A, M1YN);

not U10 (Q, M1YN);

not U11 (QN, M1A);

endmodule
```

Name: Connor Goldberg

## Functional Simulation Waveforms:



## Comments/Notes:

Name: Connor Goldberg

| | |
|---|---|
| **Library Name:** | **cjg_lib** |
| **Cell Name:** | **CJG_FA** |

**Function/Truth Table:**

| CIN | B | A | 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:**

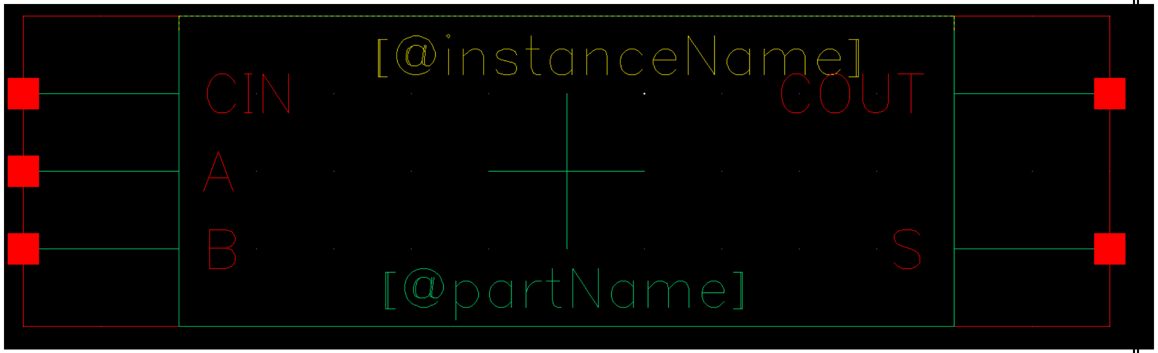
| | |
|---|---|
| A → S↑ | 1.488 nS |
| A → S↓ | 928.4 pS |
| B → S↑ | 1.448 nS |
| B → S↓ | 982.8 pS |
| CIN → S↑ | 1.340 nS |
| CIN → S↓ | 808.6 pS |
| A → COUT↑ | 1.229 nS |
| A → COUT ↓ | 980.5 pS |
| B → COUT ↑ | 1.232 nS |
| B → COUT ↓ | 898.3 pS |
| CIN → COUT ↑ | 1.230 nS |
| CIN → COUT ↓ | 986.2 pS |

**S: Rise Time: 1.764 nS, Fall Time: 1.050 nS**

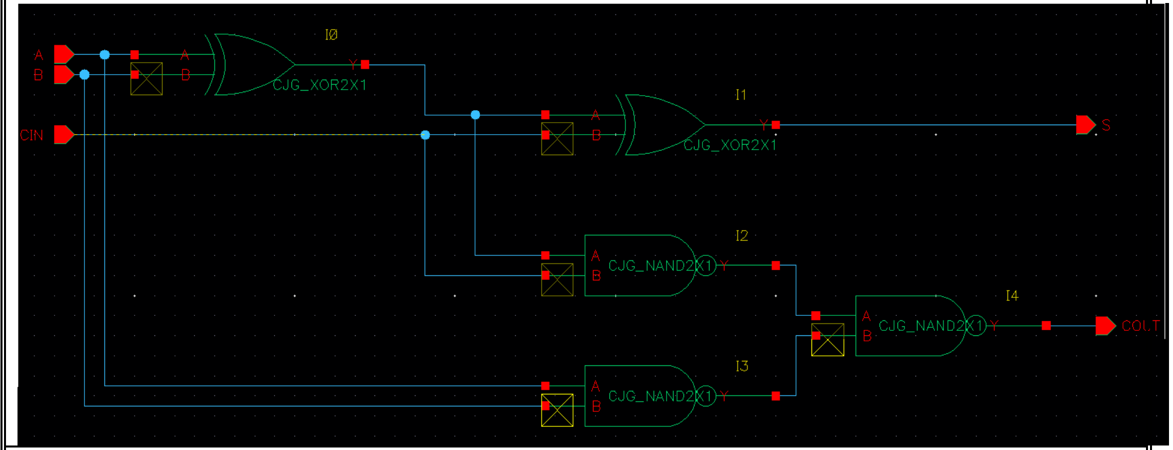
**COUT: Rise Time: 1.638 nS, Fall Time: 1.118 nS**

Name:_Connor Goldberg__

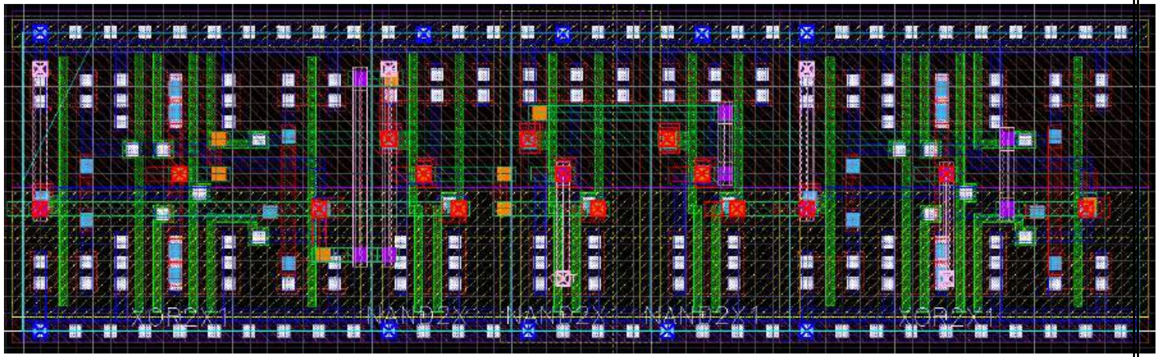
## Symbol with Port Names:



## Schematic:

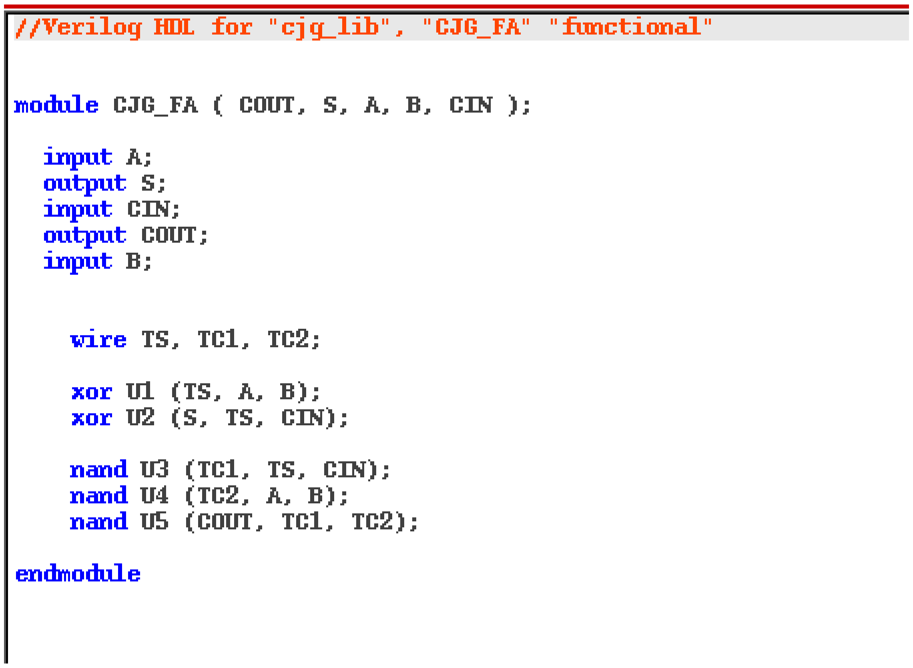


## Layout:



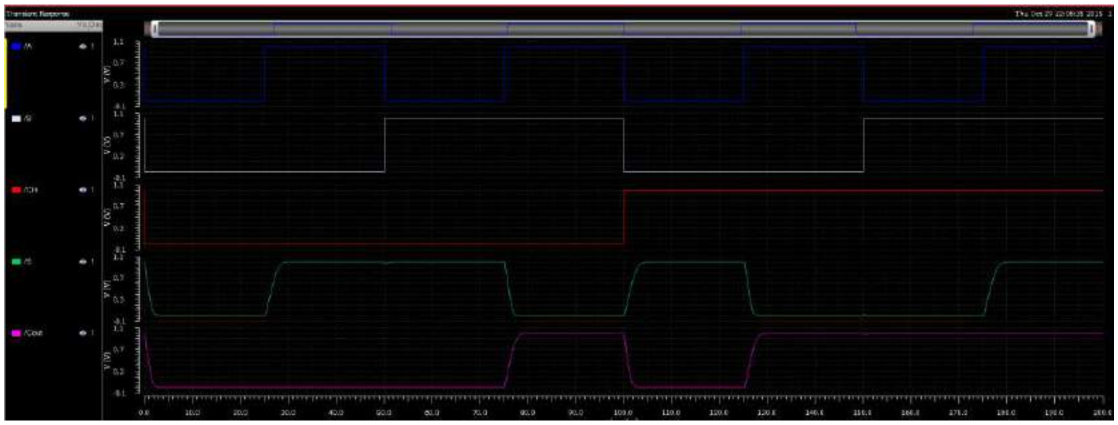
Name:_Connor Goldberg__

## Verilog Model:



```verilog
//Verilog HDL for "cjg_lib", "CJG_FA" "functional"


module CJG_FA ( COUT, S, A, B, CIN );

  input A;
  output S;
  input CIN;
  output COUT;
  input B;


    wire TS, TC1, TC2;

    xor U1 (TS, A, B);
    xor U2 (S, TS, CIN);

    nand U3 (TC1, TS, CIN);
    nand U4 (TC2, A, B);
    nand U5 (COUT, TC1, TC2);

endmodule
```

## Functional Simulation Waveforms:

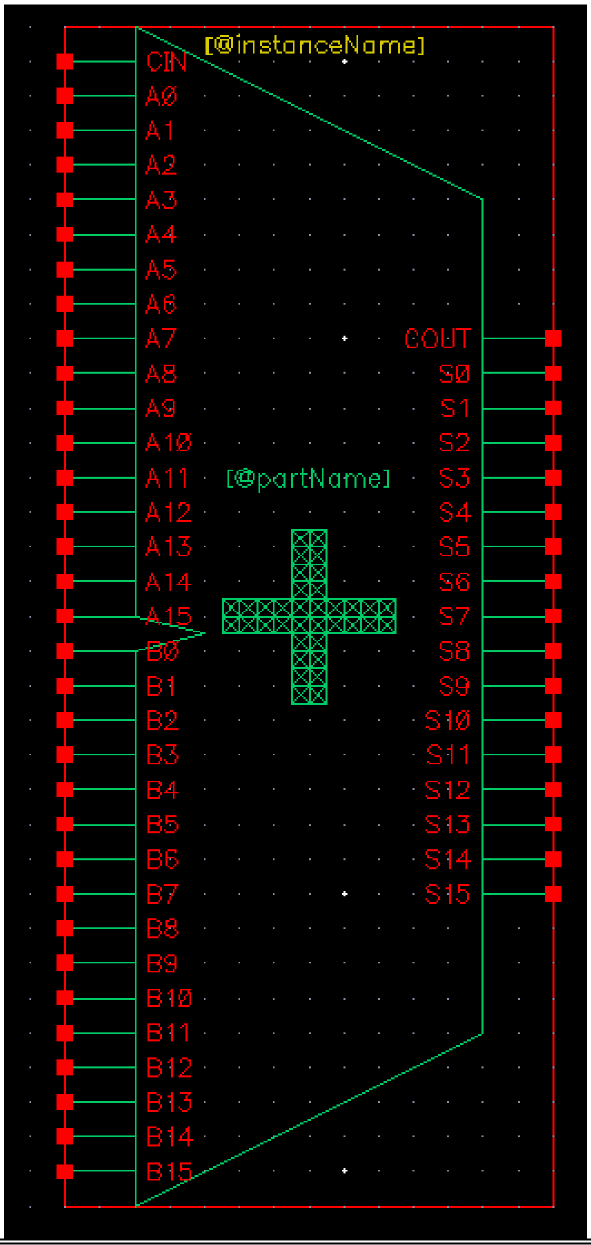


## Comments/Notes:

Name: Connor Goldberg

| | |
|---|---|
| **Library Name:** | **cjg_lib** |
| **Cell Name:** | **CJG_ADD16** |

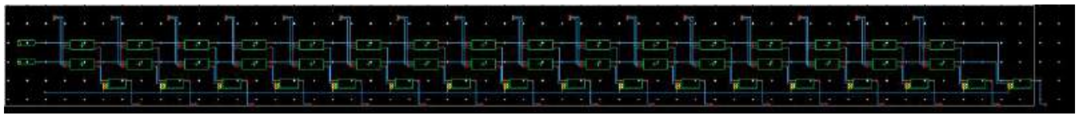
**Symbol:**



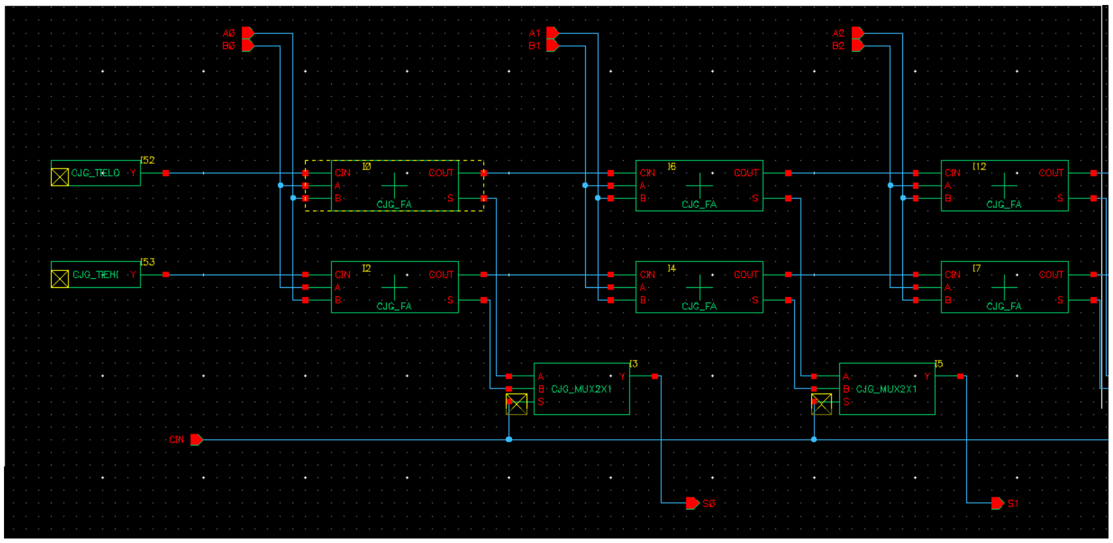
Name:_Connor Goldberg___

**Schematic:**

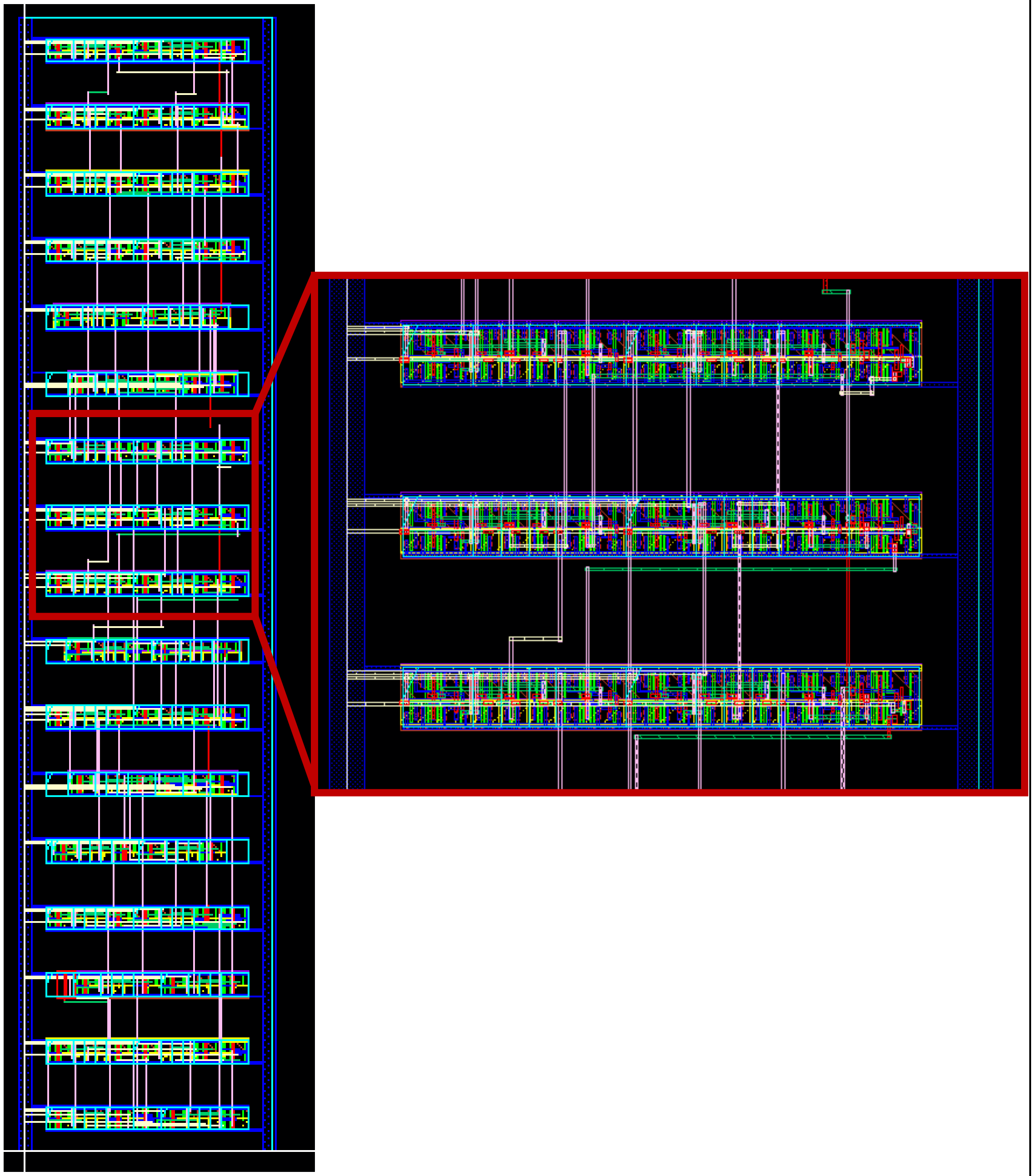
Full:



Zoomed:



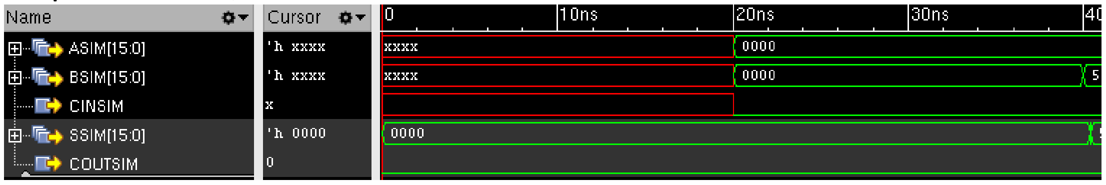
**Layout:**



Name: Connor Goldberg

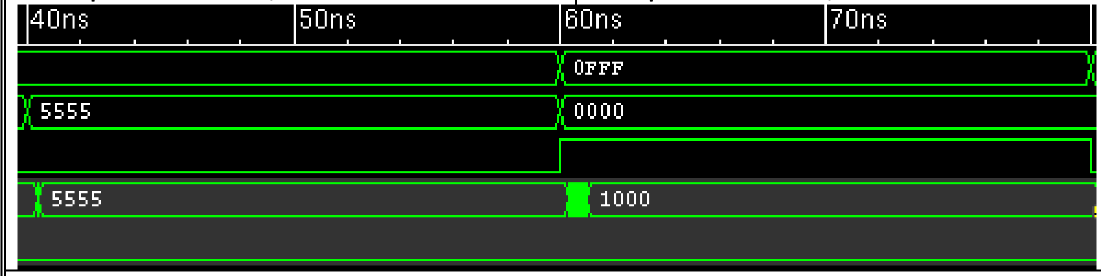
**Functional Simulation Waveforms:**

Input: 0x0000 + 0x0000, Cin = 0
Output: 0x0000, Cout = 0



Input: 0x0000 + 0x5555, Cin = 0
Output: 0x5555, Cout = 0

Input: 0x0FFF + 0x0000, Cin=1
Output: 0x1000, Cout = 0



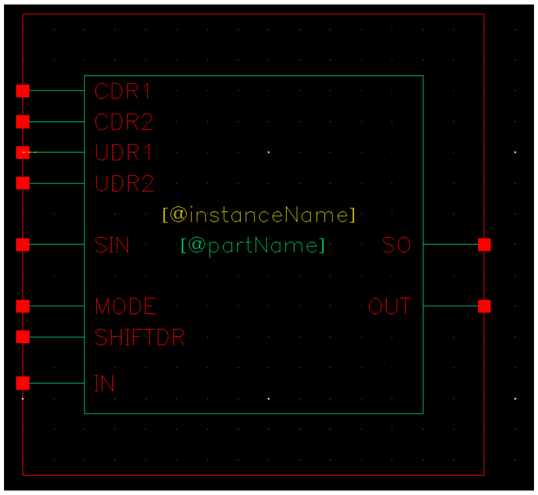
**Comments/Notes:**

EE520 | EE620 Cell Overview
Page 1 of 2
1 November 2015

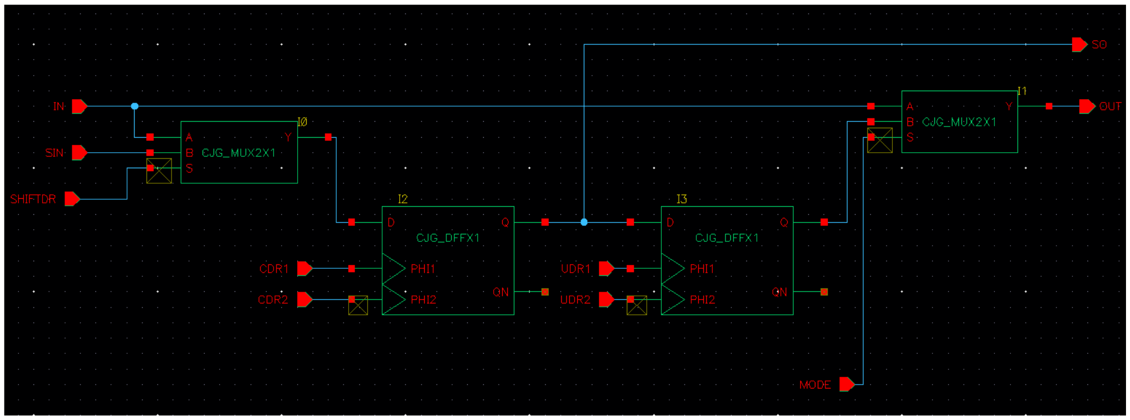
Name:_Connor Goldberg__

| | |
|---|---|
| **Library Name:** | **cjg_lib** |
| **Cell Name:** | **CJG_BSC** |

**Symbol:**

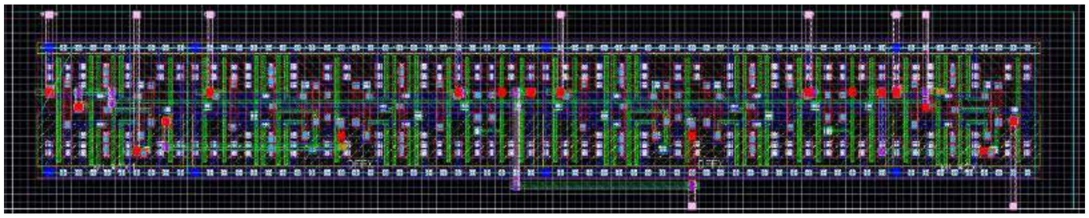


**Schematic:**

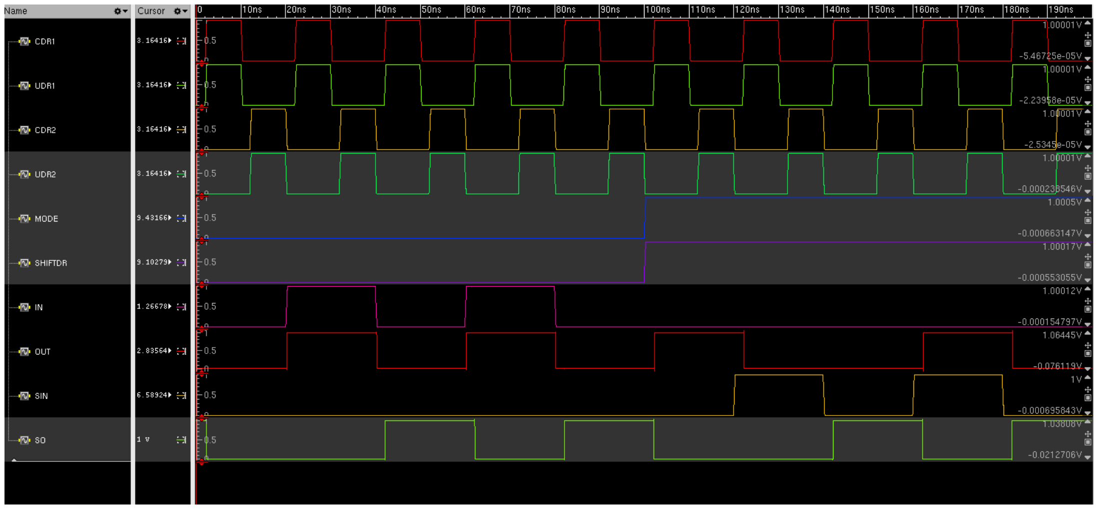


Name: Connor Goldberg

## Layout:



## Functional Simulation Waveforms:



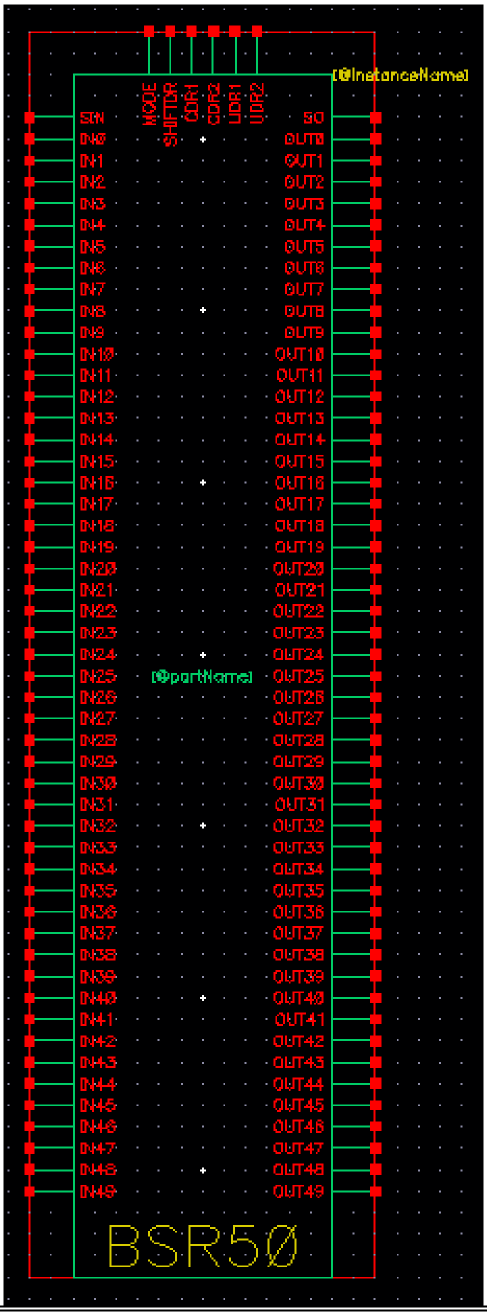
## Comments/Notes:

The simulation consisted of constantly clocking CDR and UDR. Setting MODE and SHIFTDR to 0 first allowed the testing of normal mode as well the capture mode being reflected at the shift out (SO) after a clock cycle. Then MODE and SHIFTDR were driven to 1. This tested that the value of the shift in (SIN) would be reflected at SO after a clock cycle, and in addition the value would be clocked to the output after an additional clock cycle.

Name: Connor Goldberg

| | |
|---|---|
| **Library Name:** | **cjg_lib** |
| **Cell Name:** | **CJG_BSR50** |

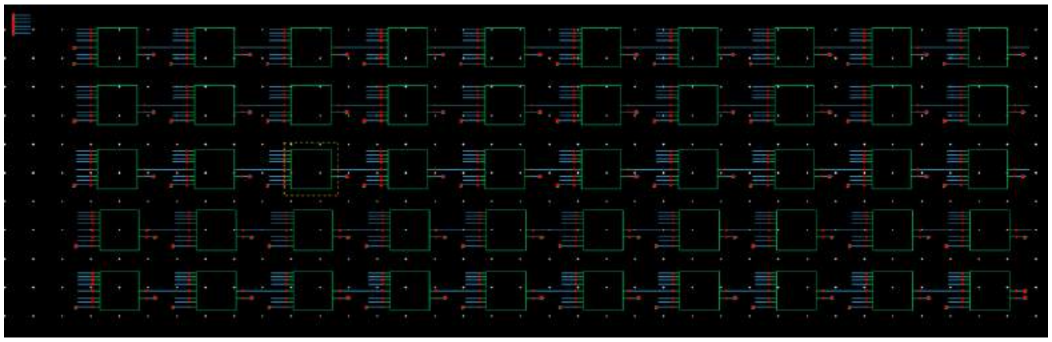
**Symbol:**



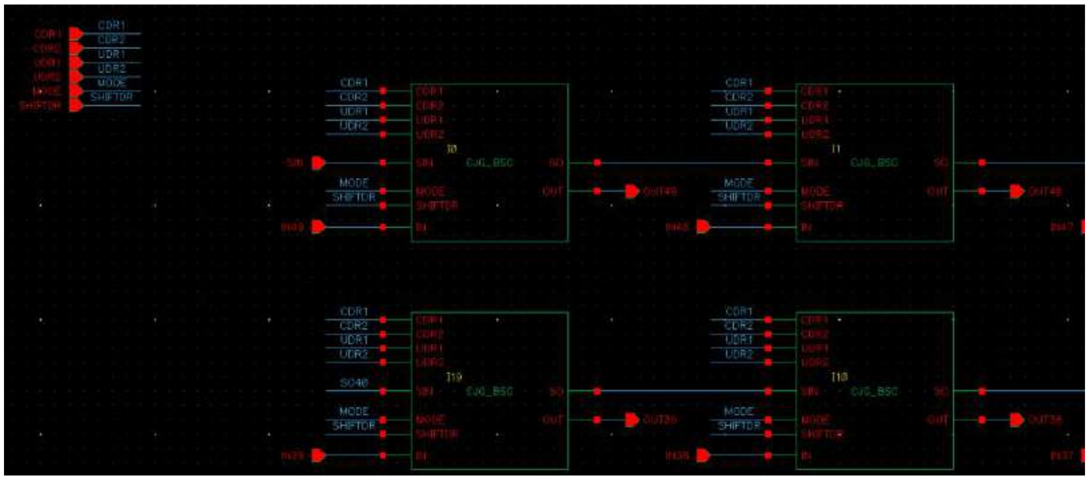
Name: Connor Goldberg

**Schematic:**

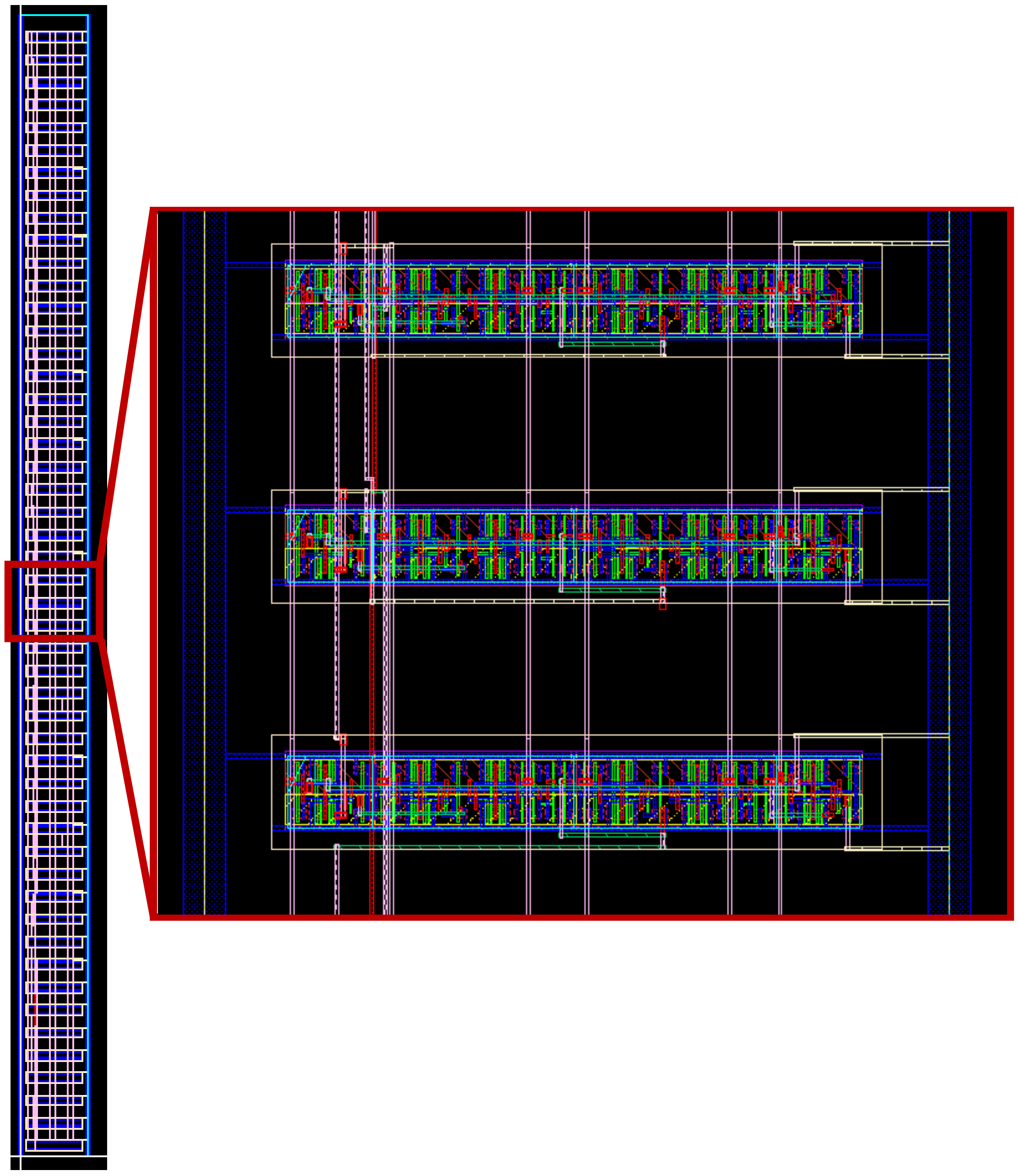
Full:



Zoomed:



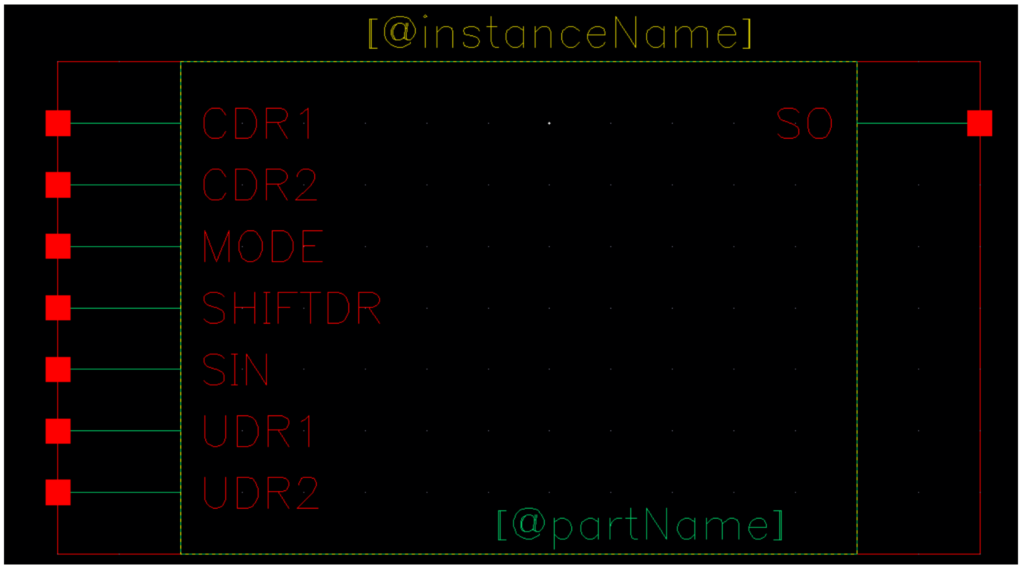
**Layout:**



Name: Connor Goldberg

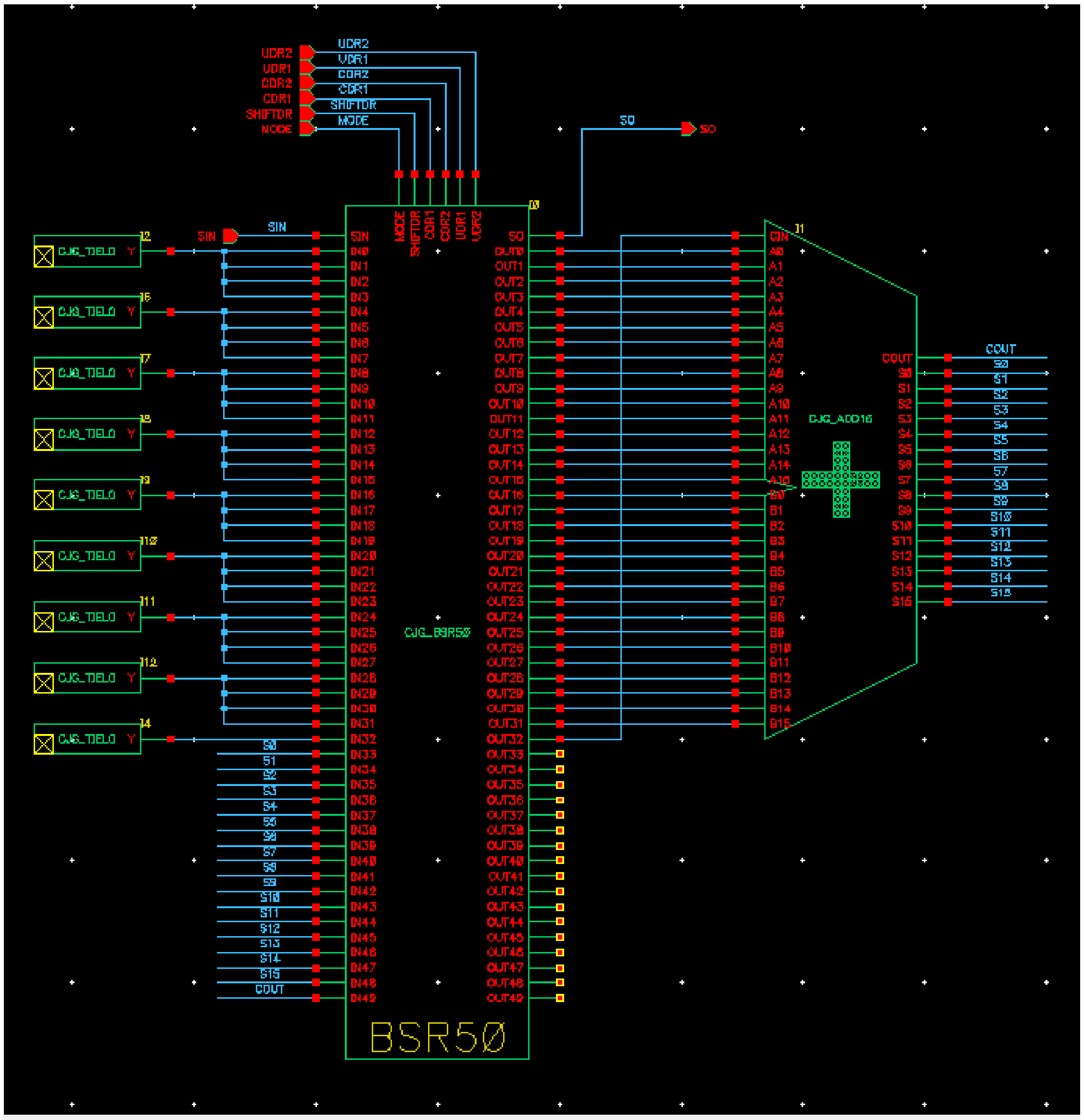
| | |
|---|---|
| **Library Name:** | **cjg_lib** |
| **Cell Name:** | **CJG_BSSUM** |

**Symbol:**



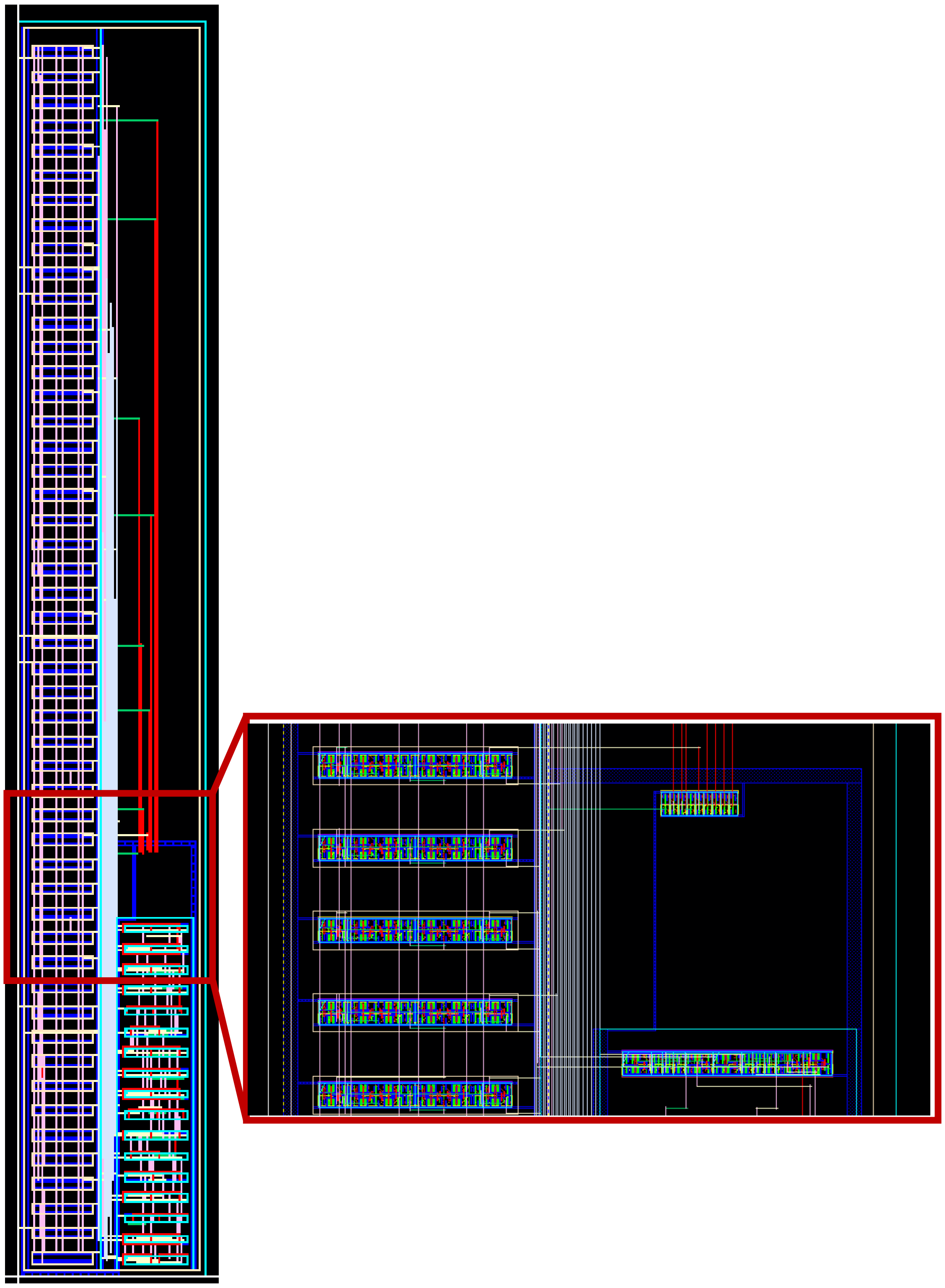
Name: _Connor Goldberg__

**Schematic:**



Name: Connor Goldberg

## Layout:

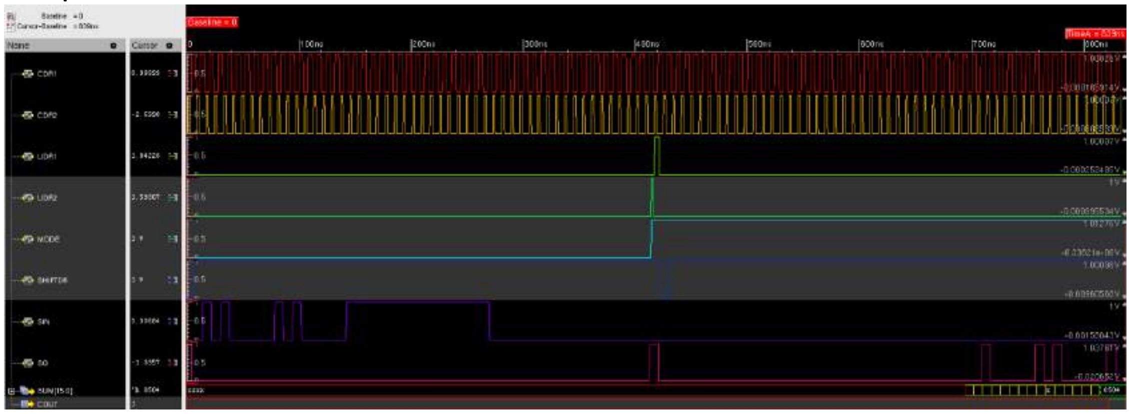


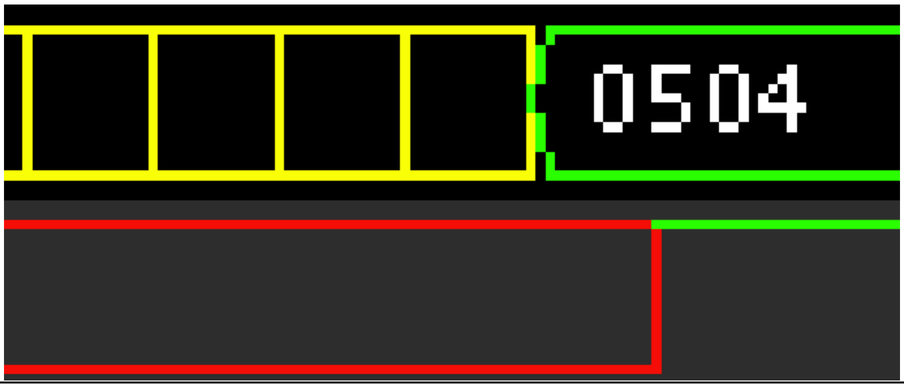
Name: Connor Goldberg

## Functional Simulation Waveforms:

Input: 0x0505 + 0xFFFF, Cin = 0

Output:





## Comments/Notes:

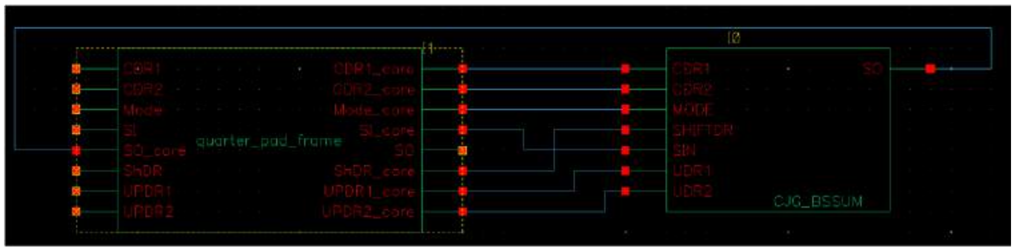
This cell was tested by scanning a large input vector that contained the all of the inputs for that ADD16 (A, B, Cin), into the Shift in input to the cell. These values were shifted into place, then sent to the adder. The adder had its output wired to some of the inputs of this register. The output was then captured by the register and shifted out. This result was the correct addition of the two inputs and the carry-in.

EE520 | EE620 Cell Overview
Page 1 of 1
1 November 2015

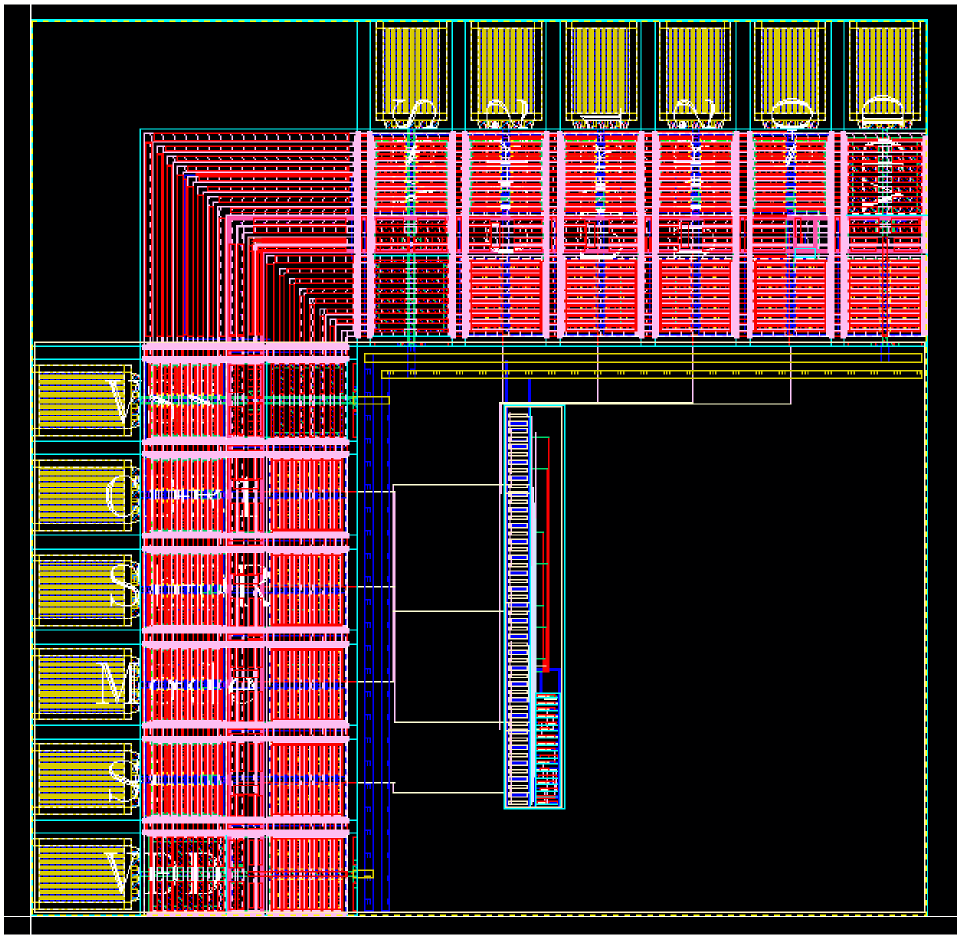
Name: Connor Goldberg

| | |
|---|---|
| **Library Name:** | **cjg_lib** |
| **Cell Name:** | **CJG_BSTEST** |

**Schematic:**



**Layout:**

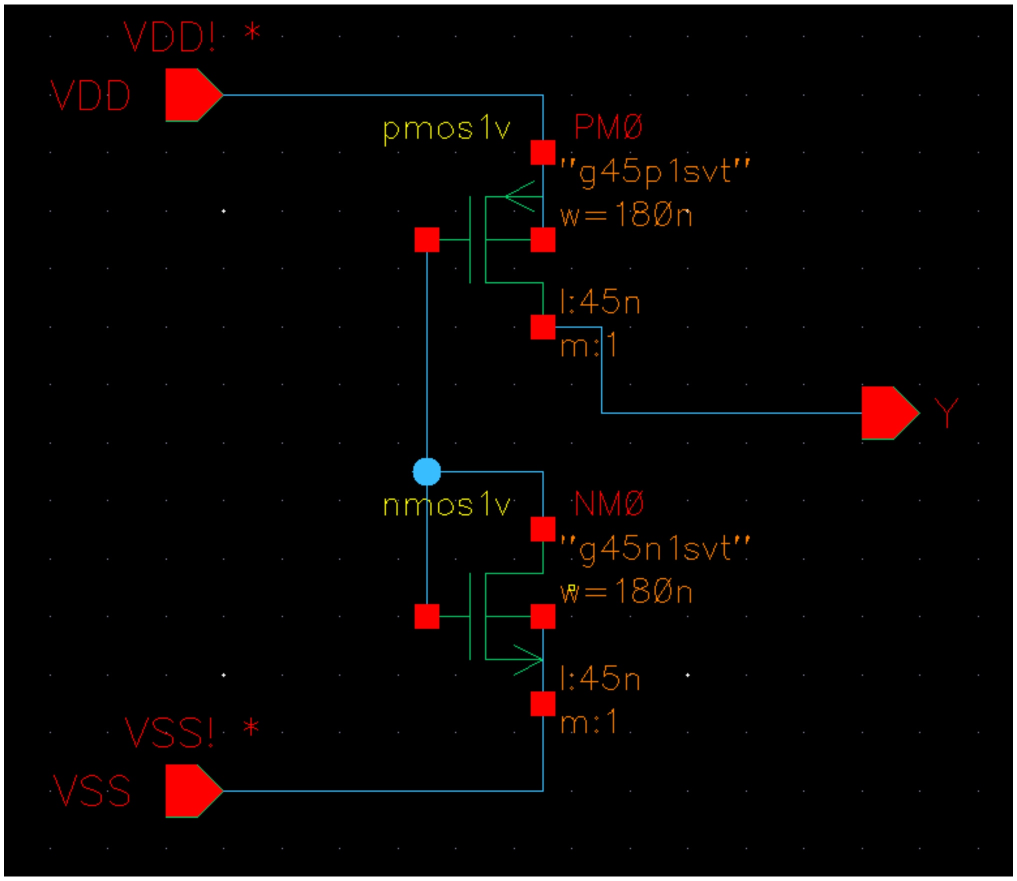


1 November 2015

Name:_Connor Goldberg__

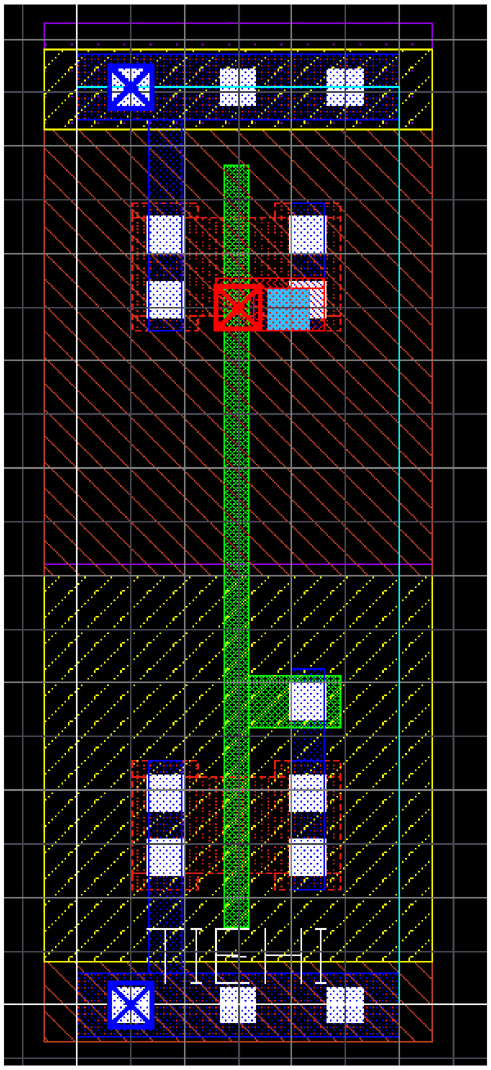
| | |
|---|---|
| **Library Name:** | **cjg_lib** |
| **Cell Name:** | **CJG_TIEHI** |

**Schematic:**

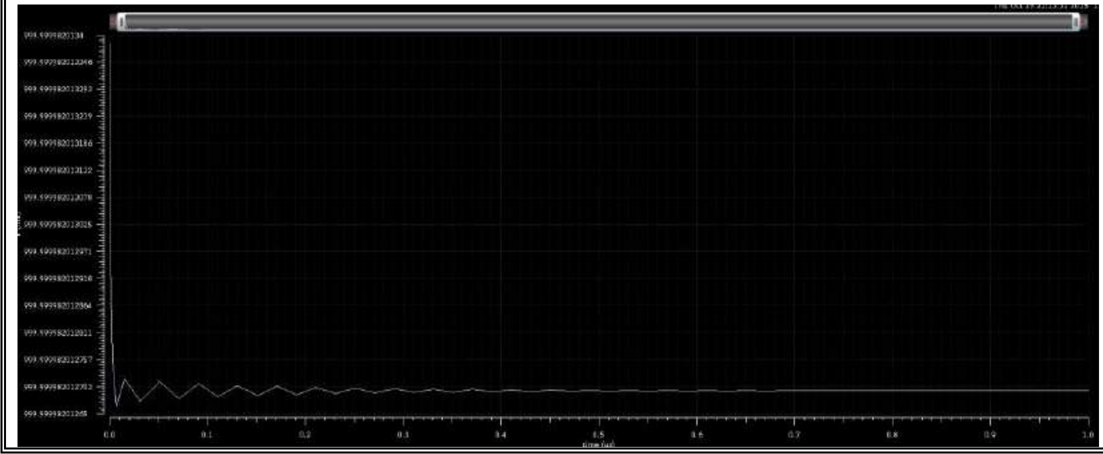


Name: Connor Goldberg

**Layout:**



**Waveform:**

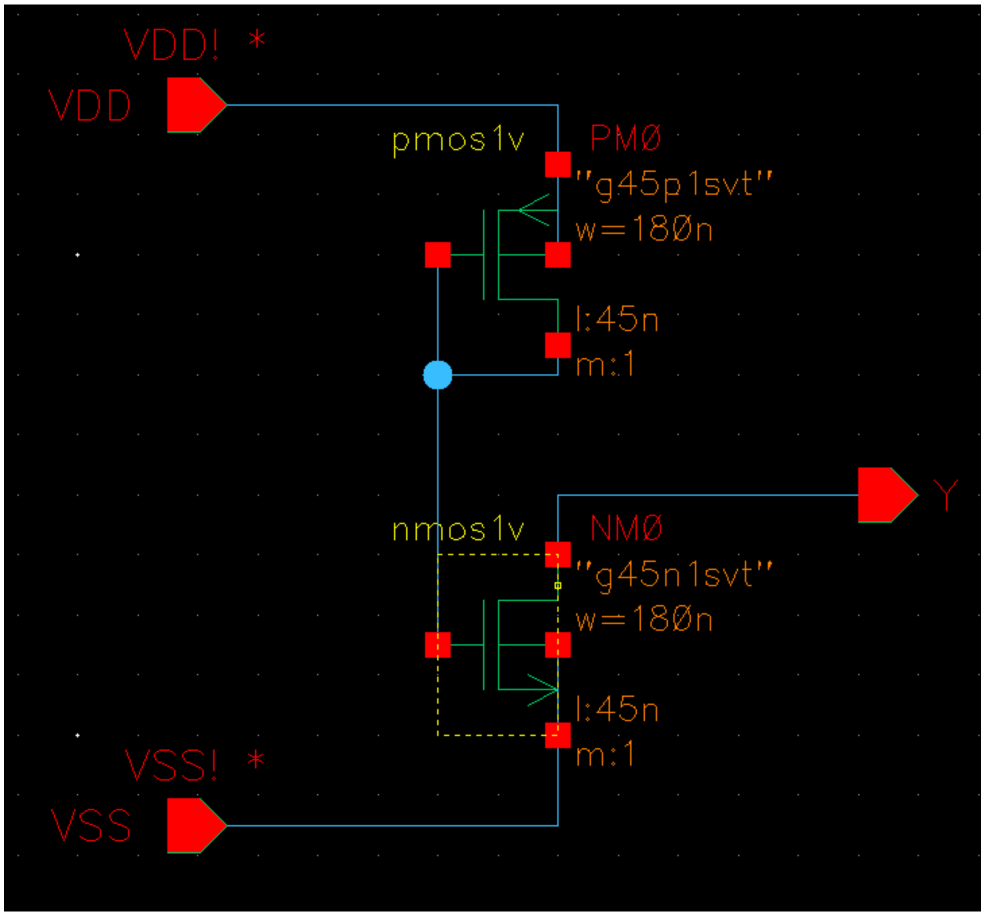

999.9999820134
999.99998201334б
999.99998201329З
999.99998201323Э
999.999982013186
999.999982013132
999.999982013078
999.999982013025
999.999982012971
999.999982012918
999.999982012864
999.999982012811
999.999982012757
999.999982012703
999.99998201265
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0
time (us)

1 November 2015

Name: Connor Goldberg

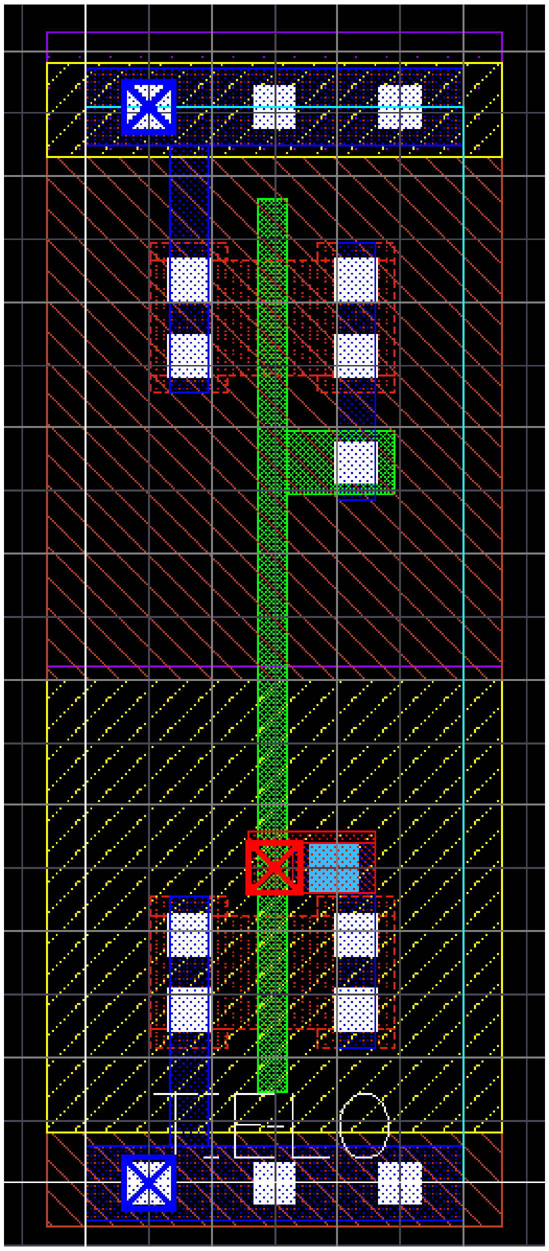
| | |
|---|---|
| **Library Name:** | **cjg_lib** |
| **Cell Name:** | **CJG_TIELO** |

**Schematic:**



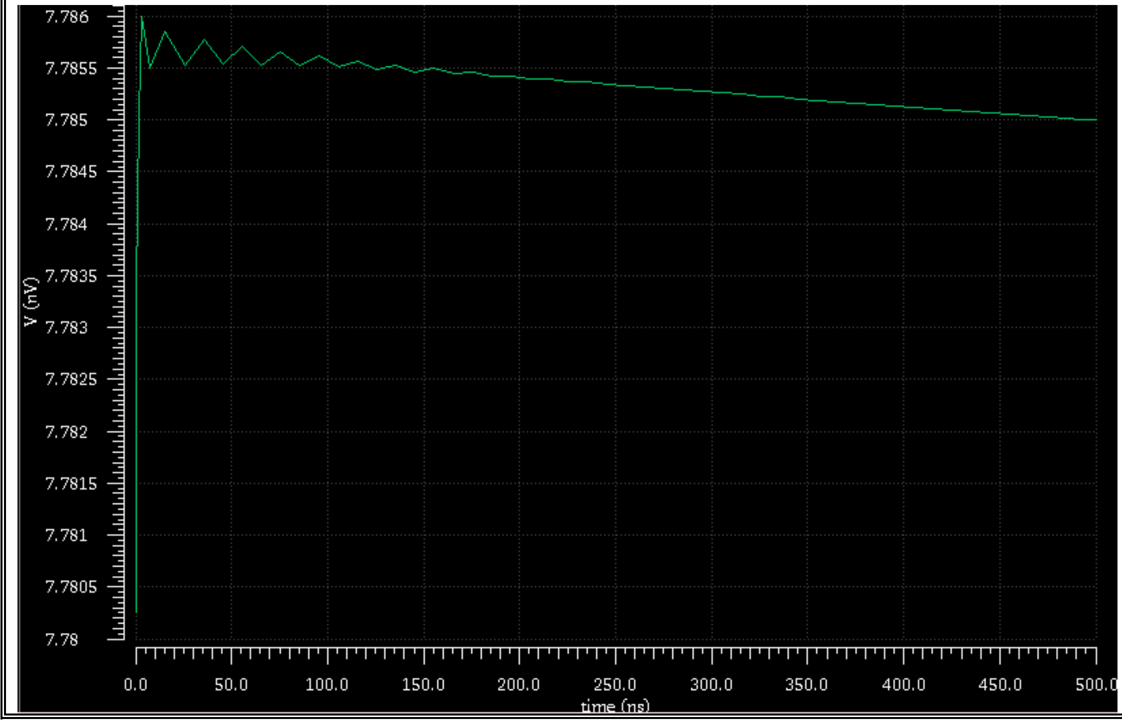
Name: Connor Goldberg

**Layout:**



Name: Connor Goldberg

**Waveform:**


V (nV)
time (ns)