

# **Super High-Speed Transmission Gate Based FPGA Architecture**

Dongyuan He

July 29, 2020

ECE 6130 Advanced VLSI Systems

Summer 2020

## Table of Contents

|       |                                   |    |
|-------|-----------------------------------|----|
| I.    | Introduction.....                 | 4  |
| II.   | CLB Architecture Design .....     | 5  |
| II.1  | Input Mux: .....                  | 8  |
| II.2  | LUT Mux: .....                    | 10 |
| II.3  | BLE Mux: .....                    | 12 |
| II.4  | Flip Flop:.....                   | 14 |
| II.5  | Bypass Clock Enable Block:.....   | 16 |
| II.6  | CLB Enable Block: .....           | 17 |
| II.7  | LUT Inverter:.....                | 18 |
| III.  | C Block Architecture Design.....  | 19 |
| III.1 | Demux:.....                       | 22 |
| III.2 | C Block Mux:.....                 | 23 |
| III.3 | C Block Inverter:.....            | 24 |
| IV.   | S Block Architecture Design ..... | 25 |
| IV.1  | S Block mux:.....                 | 30 |
| IV.2  | Tristate buffer: .....            | 31 |
| V.    | FPGA Architecture Design.....     | 32 |
| V.1   | Clock.....                        | 33 |
| V.2   | Power .....                       | 34 |
| VI.   | FPGA Bitstream Design .....       | 35 |
| VI.1  | Bitstream Flip Flop: .....        | 37 |
| VI.2  | Program Clock Enable Block: ..... | 38 |

|   |    |
|---|----|
| VII. CLB Full Layout.....               | 39 |
| VII.1 BLE:.....                         | 39 |
| VII.2 CLB:.....                         | 40 |
| VII.3 Parasitic Extraction Errors ..... | 45 |
| VIII. Logical Effort Comparison .....   | 46 |
| IX. Future Work.....                    | 48 |
| X. Conclusion .....                     | 50 |
| XI. References.....                     | 50 |

## I. Introduction

The contents of this report focus on the design and implementation of a scalable, fine-grained Field Programmable Gate Array (FPGA). The FPGA design will use the FreePDK process design kit developed by NCSU for the 45nm technology node.

This report begins by giving an overview of the three “building” blocks of the FPGA and the steps necessary to begin designing and laying out each of these blocks. A detailed description of each block is provided in the CLB, C Block, and S Block Architecture Design sections. These sections give an overall functional block diagram and floorplans of each block, in addition to detailed subsections of all the individual cells found in each block. In the next section, a detailed description of the overall FPGA layout is provided in FPGA Architecture Design. This section also contains information about clock distribution and power and ground distribution. This is followed by another section detailing the bitstream design. Afterwards, the CLB Full Layout section mainly focuses on the layout effort of the entire CLB including bitstream elements. Finally, a Future Work section details the work needed to complete the C Block and S Block layouts. Additionally, it provides insight on some improvements that can be made to specific elements of the FPGA that can help decrease propagation delay, decrease power consumption, or decrease area.

The FPGA is a reconfigurable logic chip that comprises of programmable logic blocks and programmable interconnects. The programmable logic blocks allow implementation of arbitrary Boolean equations and the programmable interconnects form the connections between these programmable logic blocks. This design will implement an island architecture using three different block types: Configurable Logic Block (CLB), Switch Block (S Block), and Connect Block (C Block). CLBs are configurable logic blocks which contain logic elements that can be programmed. C Blocks provide the connection between adjacent CLBs and S Blocks. S Blocks provide the “highway” that allows signals to pass from any C Block in the FGPA to any other C Block in the FPGA. A simplified island-style floorplan of the FPGA is given in Figure 1.

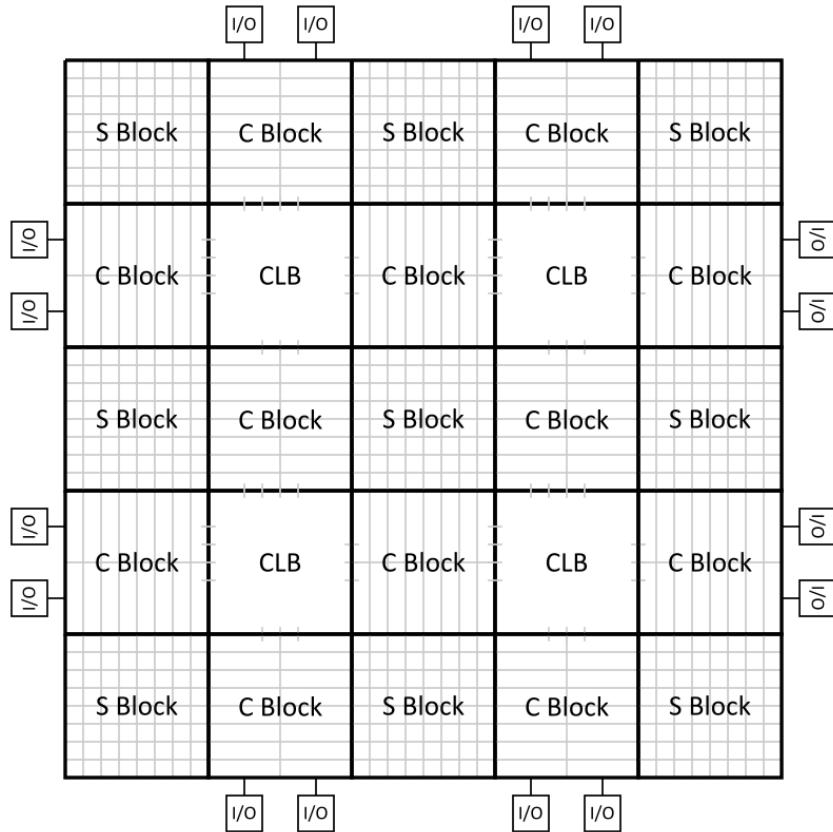


Figure 1. Example island-style FPGA with CLBs, C Blocks, and S Blocks (modified from Figure 1 in [1]).

A single FPGA can contain tens of thousands of these three blocks. Thus, the CLB, C Block, and S Block layouts should be designed in a way that each block can tile adjacently next to each other. Minimal effort should be required for routing at the very top-level hierarchy of the FPGA. Most of the layout effort should be performed on creating the individual blocks so that multiple instances of each block can be seamlessly placed next to each other. To minimize wasted layout effort, a floorplan of each block was created before laying out the cells. Furthermore, stick diagrams and transistor schematics of the individual complex gates found in each block (multiplexors, flip flops, inverters, etc.) were created before performing the layout of each gate. This allowed for easy modification and comparison of gate designs.

## II. CLB Architecture Design

The CLB is the configurable logic block (essentially the “brains”) of the FPGA. Simple functions can be implemented through the 4 Look Up Table Muxes in a single CLB, and much more complex functions can be implemented through the use of multiple CLBs. Each CLB has the option to either operate as a sequential logic block or a combinational logic block through 2:1 multiplexors at the LUT Muxes’ output.

The CLB contains 4 major components: the Input Mux, the LUT Mux, the BLE Mux, the Flip Flop, and the LUT inverter. Overall, the CLB will contain 16 Input Muxes, 4 LUT Muxes, 4 BLE Muxes, 4 master-slave Flip Flop pairs, and 16 LUT inverters. The Input Muxes each require 4 bits for the select lines. The LUT Muxes each require 16 bits for the LUT inputs. The BLE Muxes each require 1 bit for the select line. In total, the CLB will require 132 bits from the programming bitstream.

The NCSU FreePDK doesn’t allow transistors to be minimally sized 45nm in length to 45nm in width. The minimum allowable width is 90nm, twice the transistor length. For this reason, a minimum sized inverter with equal rise and fall time would have a 90nm wide NMOS and a 180nm wide PMOS.

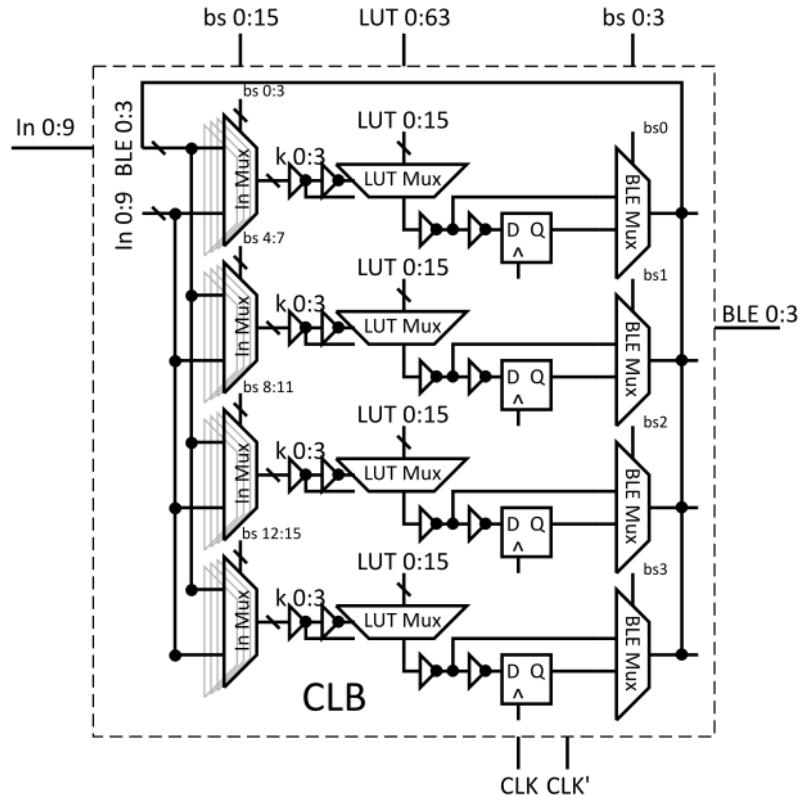


Figure 2. Functional block diagram of the CLB.

The CLB has 17 total input pins and 5 total output pins. The pins, their functionality, and active conditions are shown in Figure 3. The CLB Full Layout section gives a detailed view of the location, pitch, and metal layer of each pin. Figure 68 can be referred to for the full floorplan of the CLB with the bitstream included.

| CLB Pins List |        |   |
|---------------|--------|---|
| Signal        | I/O    | Description                               |
| phi1          | Input  | Global Clock Signal 1                     |
| phi2          | Input  | Global Clock Signal 2                     |
| bs_in         | Input  | Bitstream Input                           |
| prog_En       | Input  | Program Clock Enable Signal (Active HIGH) |
| bypass_En     | Input  | Bypass Clock Enable Signal (Active HIGH)  |
| CLB_En        | Input  | CLB Enable Signal (Active LOW)            |
| Rst           | Input  | Clock Reset Signal (Active LOW)           |
| In0-In9       | Input  | Input Signals for the Input Muxes         |
| bs_o          | Output | Bitstream Output                          |
| Out0-Out3     | Output | Output Signals from the BLE Muxes         |

Figure 3. List of input and output pins for the CLB.

An HSpice simulation was performed to estimate the worst case propagation delay through the CLB. The path of the delay travels from the input of the Input Mux to the output of the BLE Mux. The rise time was 547.43 picoseconds. The fall time was 371.87 picoseconds. The HSpice simulation used to calculate this delay is the same used to calculate the critical path delay in Figure 76.

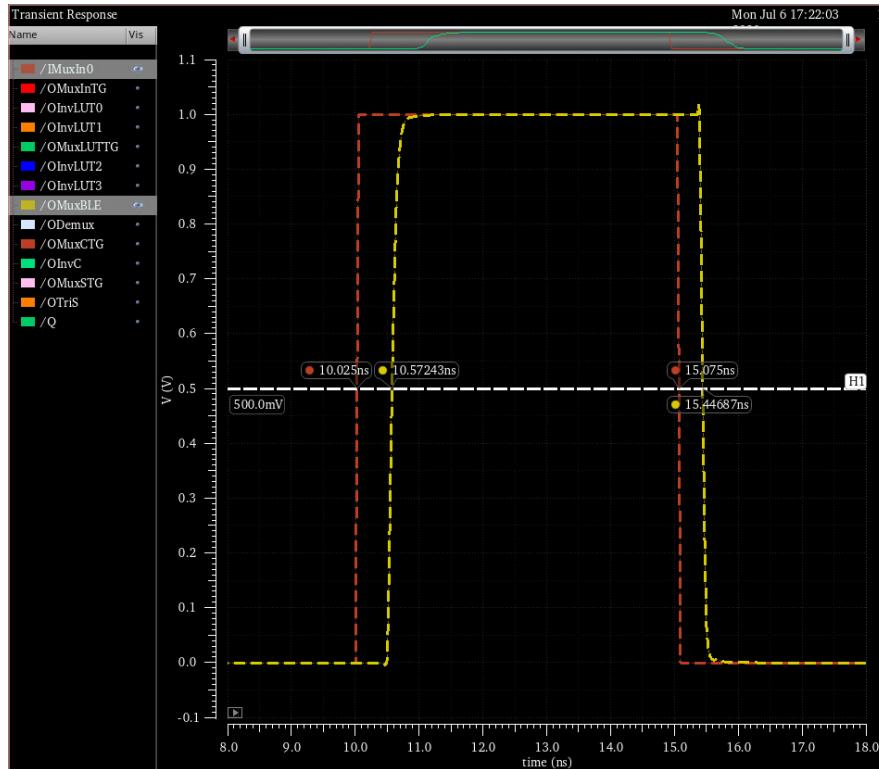
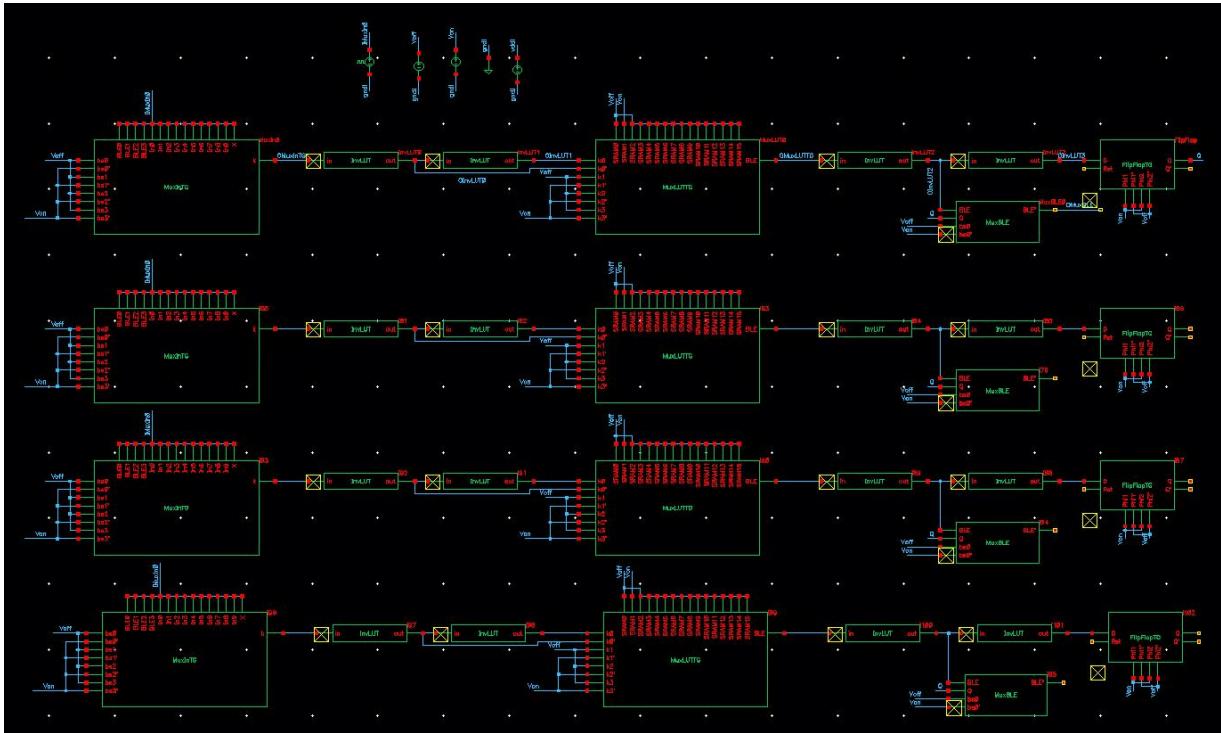
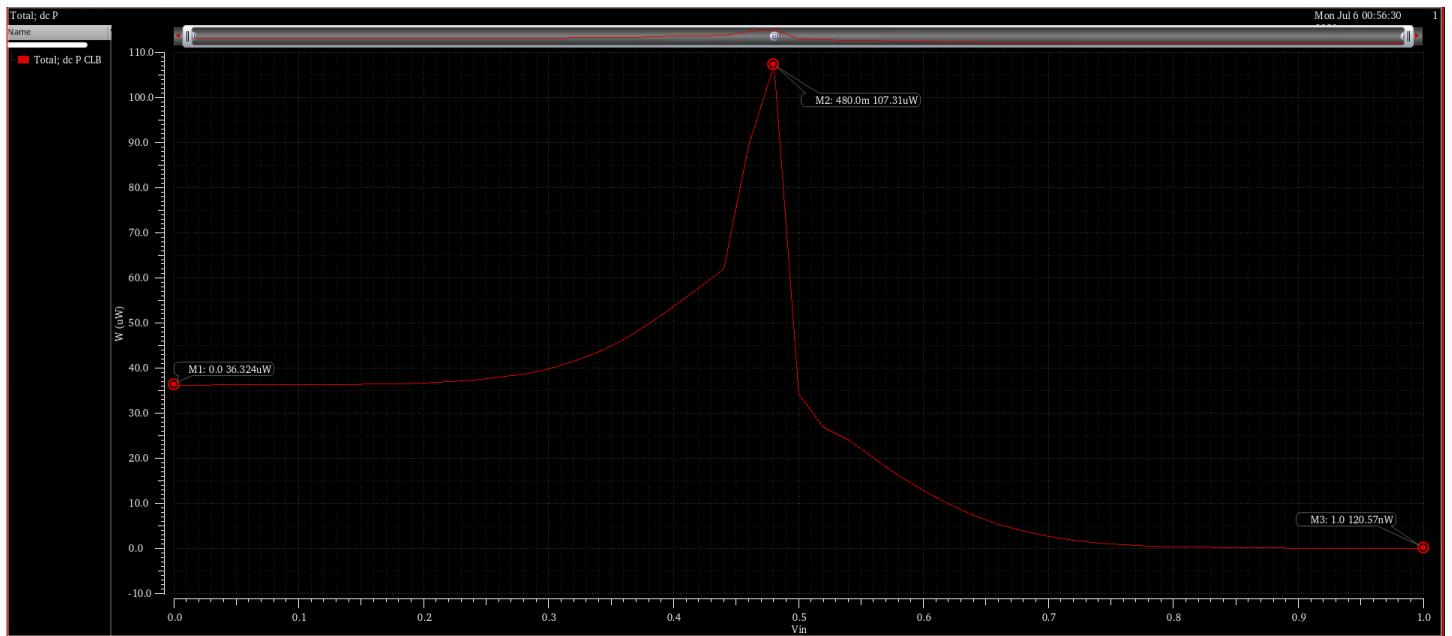


Figure 4. Overall CLB delay. HSpice delay simulation from the input of the Input Mux to the output of the BLE Mux. Rise time of 547.43 picoseconds. Fall time of 371.87 picoseconds.

An HSpice simulation was also performed to estimate the total dynamic and leakage power consumed in the CLB. A full schematic with all cells included was drawn to more accurately model the block. The leakage power through the CLB when passing a LOW was 36.324 uW. The leakage power through the CLB when passing a HIGH was 0.12057 uW. The dynamic power through the CLB when switching values was 107.31 uW.



*Figure 5. HSpice schematic used for power simulation of the entire CLB.*



*Figure 6. HSpice power simulation of the entire CLB. Marker M1 shows leakage power of 36.324 uW through the CLB when passing a LOW. Marker M2 shows dynamic power of 107.31 uW through the CLB when switching values. Marker M3 shows leakage power of 0.12057 uW through the CLB when passing a HIGH.*

## II.1 Input Mux:

The Input Mux takes 10 inputs from the C block in addition to 4 feedback inputs from the 4 BLEs in the CLB, making a total of 14 inputs. The CLB contains 16 total Input Muxes. One of these inputs will be selected with a 4 bit select line, configured during startup using a shift register to store each value of the bitstream. The output becomes a single select bit of the LUT Mux.

A 2 stage transmission gate 4:1 mux will be used to implement the Input Mux. The entire transmission gate 14:1 mux would comprise of 16 transmission gates for the bs0 select line, 8 transmission gates for the bs1 select line, 4 transmission gates for the bs2 select line, and 2 transmission gates for the bs3 select line. This makes a total of 60 transistors. To go from the input to the output of the mux, a signal has to pass through 4 transistors in series.

This design can be made to operate even faster by increasing the size of the transistors. This incurs an increase in power, but this design already have significantly less power consumption, both dynamic power and leakage power, than a regular CMOS design. By not having a vdd or a gnd connection, there is much less current passing through each transistor, thus less power consumption. The caveat is a decrease in output driving capacity of both these designs, due to the voltage drops at each gate. Furthermore, after passing a signal through 4 transistors with a voltage drop at each, a buffer (two inverters) will have to be used at the output to restore the signal.

The first stage contains four instances of a 4:1 mux. All NMOS' in this stage are VTL and 150nm in width. All PMOS' in this stage are VTL and 200nm in width. The 4 stages will be placed side by side in the layout.

The second stage conatins one instance of a 4:1 mux. All NMOS' in this stage are VTL and 150nm in width. All PMOS' in this stage are VTL and 200nm in width. This stage will be placed after the first stages as shown in the stick diagram in Figure 8.

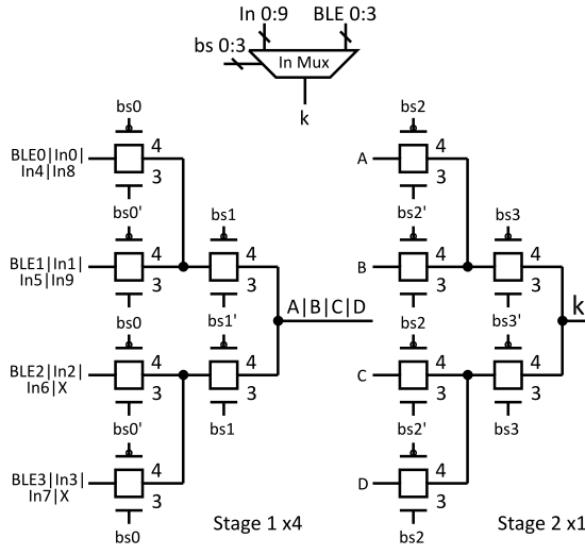


Figure 7. Transistor schematic of CLB Input Mux with sizing.

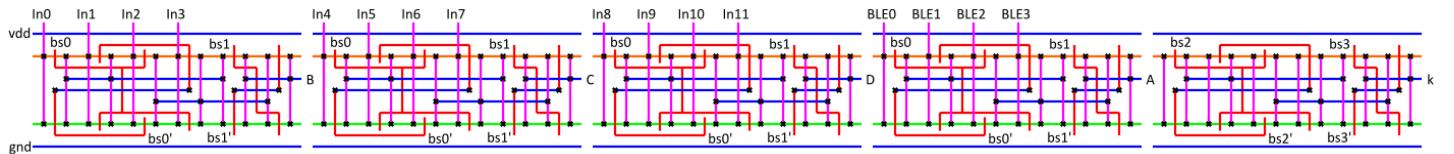


Figure 8. Stick diagram of the CLB Input Mux. Estimated footprint is 3um x 13um.

An HSpice simulation was performed to estimate the propagation delay through the Input Mux. The rise time was 227.9 picoseconds. The fall time was 154.57 picoseconds.

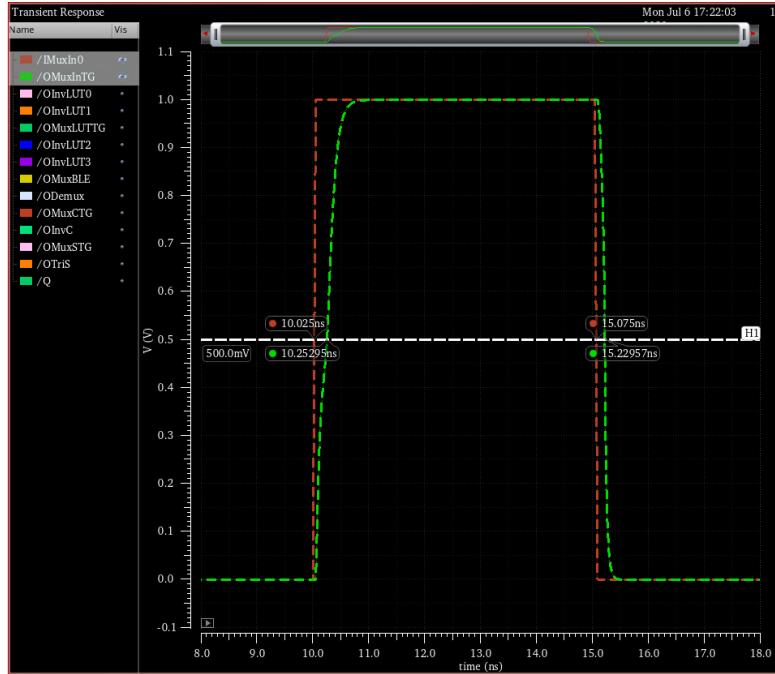


Figure 9. HSpice delay simulation of the CLB Input Mux. Rise time of 227.9 ps. Fall time of 154.57 ps.

The layout was performed in two hierarchy levels. Hierarchy level 1 contains the layouts of the 2 mux stages. Hierarchy 0 connects the two stages together. The cell dimensions of the Input Mux is 2.25um x 12.575um. DRC and LVS passed on this layout.

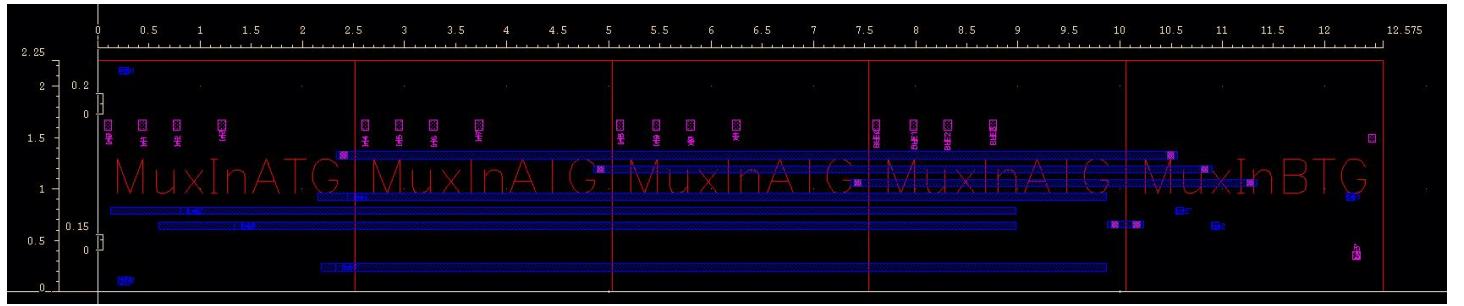


Figure 10. Layout of the CLB Input Mux. Cell dimensions are 2.25um x 12.575um. Hierarchy level 0.

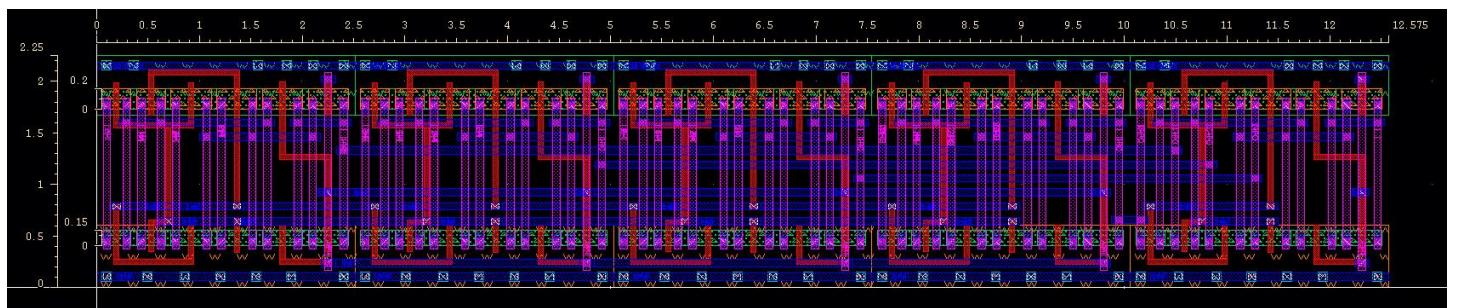


Figure 11. Layout of the CLB Input Mux. Cell dimensions are 2.25um x 12.575um. Hierarchy level 1.

## II.2 LUT Mux:

The LUT Mux takes 16 inputs from the bitstream for the Look-up Table bits, configured during startup using a shift register to store each value of the bitstream. The output of 4 of the Input Muxes feed into the 4 bit select line of the LUT Mux. Because this LUT Mux has 4 select lines, it can implement any logical function of 4 variables. The CLB contains 4 total LUT Muxes.

The same 2 stage transmission gate 4:1 mux used to implement the Input Mux will be used to implement the LUT Mux.

The transistors in the LUT Mux are sized slightly larger than the ones in the Input Mux to decrease delay without significantly increasing area or power consumption.

The first stage contains four instances of a 4:2 mux. All NMOS' in this stage are VTL and 200nm in width. All PMOS' in this stage are VTL and 250nm in width. The 4 stages will be placed side by side in the layout.

The second stage contains one instance of a 4:2 mux. All NMOS' in this stage are VTL and 200nm in width. All PMOS' in this stage are VTL and 250nm in width. This stage will be placed after the first stages as shown in the stick diagram in Figure 13.

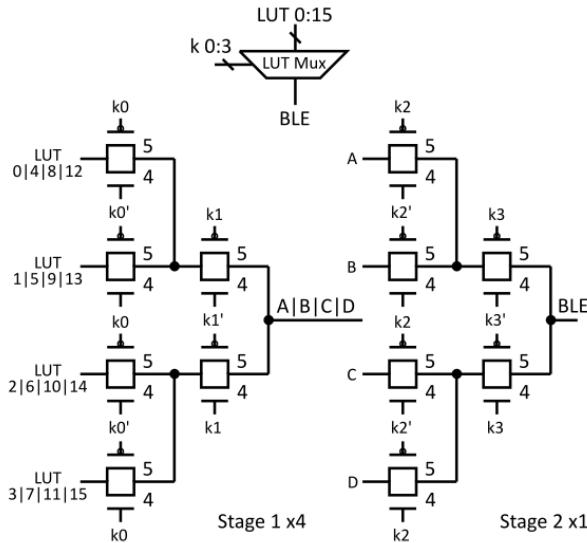


Figure 12. Transistor schematic of LUT Mux with sizing.

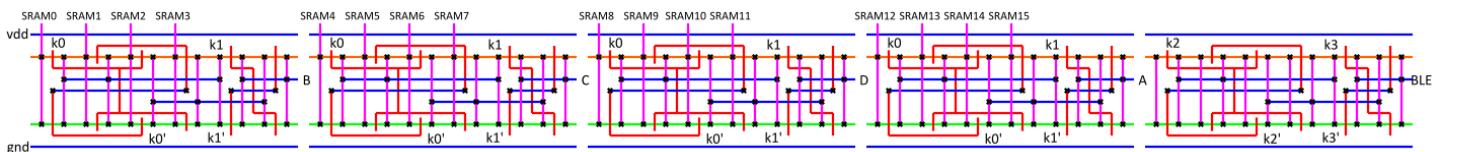


Figure 13. Stick diagram of the LUT Mux. Estimated footprint is 3um x 13um.

An HSpice simulation was performed to estimate the propagation delay through the Input Mux. The rise time was 82.31 picoseconds. The fall time was 24.23 picoseconds.

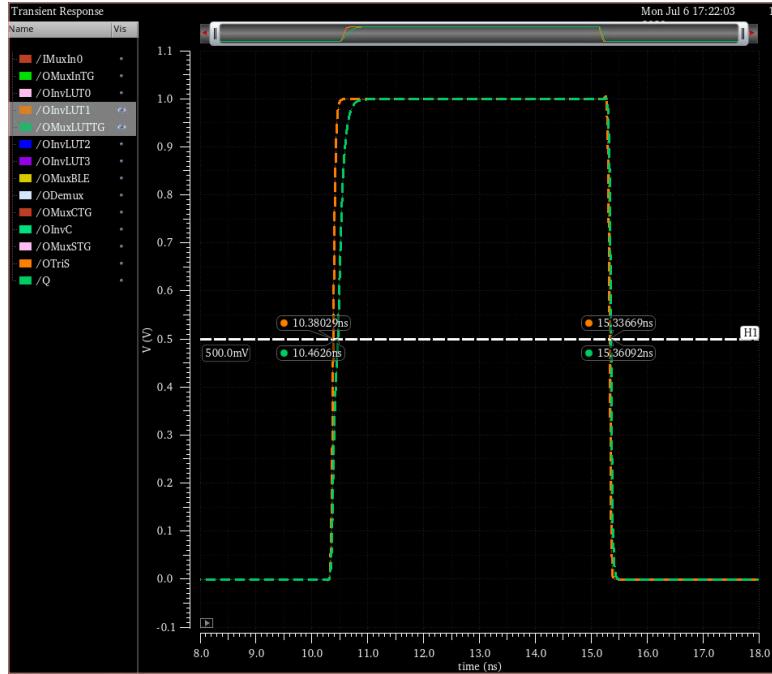


Figure 14. HSpice delay simulation of the CLB LUT Mux. Rise time of 82.31 ps. Fall time of 24.23 ps.

The layout was performed in two hierarchy levels. Hierarchy level 1 contains the layouts of the 2 mux stages. Hierarchy 0 connects the two stages together. The cell dimensions of the Input Mux is 2.35um x 12.575um. DRC and LVS passed on this layout.

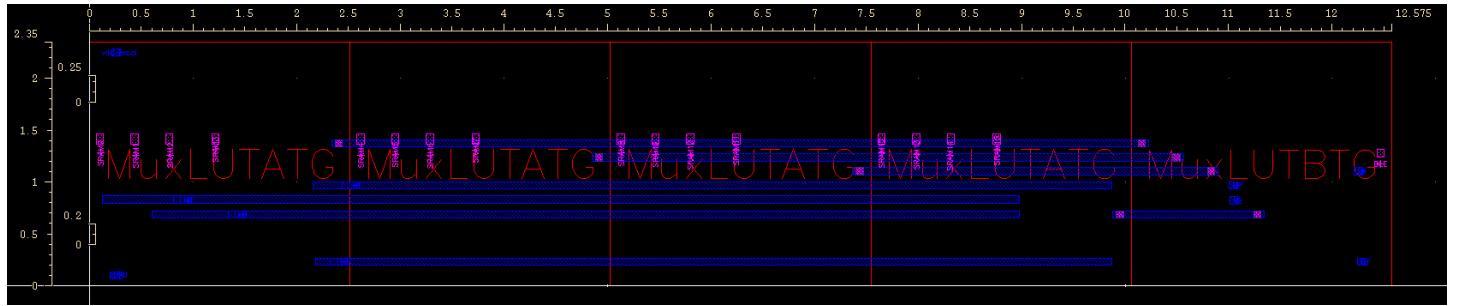


Figure 15. Layout of the CLB LUT Mux. Cell dimensions are 2.35um x 12.575um. Hierarchy level 0.

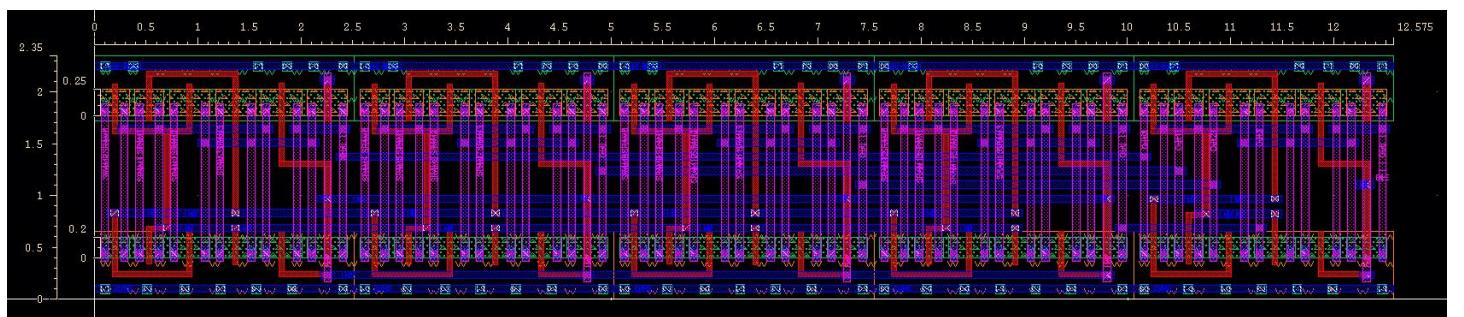


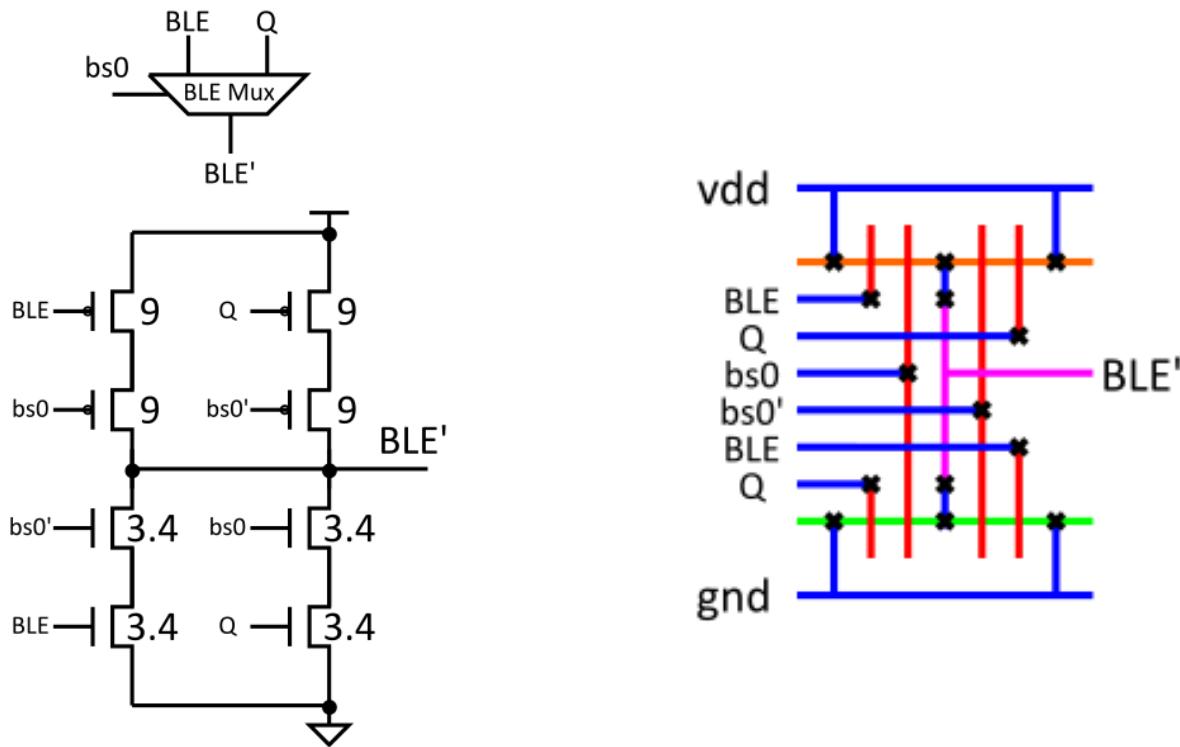
Figure 16. Layout of the CLB LUT Mux. Cell dimensions are 2.35um x 12.575um. Hierarchy level 1.

### II.3 BLE Mux:

The BLE Mux takes an input from the LUT Mux and an input from the Flip Flop. The CLB contains 4 total BLE Muxes. One of these inputs will be selected with a 1 bit select line, configured during startup using a shift register to store each value of the bitstream. This determines whether the BLE is using sequential or combinational logic. The output is fed back into all 16 of the Input Muxes in the CLB and becomes the input to the demux in the C Block.

The BLE Mux drives 22 different gates at the output of my design. If the mux had a transmission gate design or a pass gate design, the signal would become degraded. Furthermore, without additional inverter stages, using a transmission gate design for the BLE Mux would put 6 transmission gates in series, leading to further signal degradation. For these reasons, an inverting CMOS design will be used to implement this 2:1 mux, at the cost of slightly higher power consumption.

All PMOS' in this cell are VTL and 450nm in width. All NMOS' in this cell are VTL and 170nm in width.



*Figure 17. Transistor schematic of inverting CMOS BLE Mux with sizing (left figure). Stick diagram of the CLB BLE Mux (right figure). Estimated footprint is 3um x 1um.*

An HSpice simulation was performed to estimate the propagation delay through the BLE Mux. The rise time was 80.89 picoseconds. The fall time was 57.69 picoseconds.

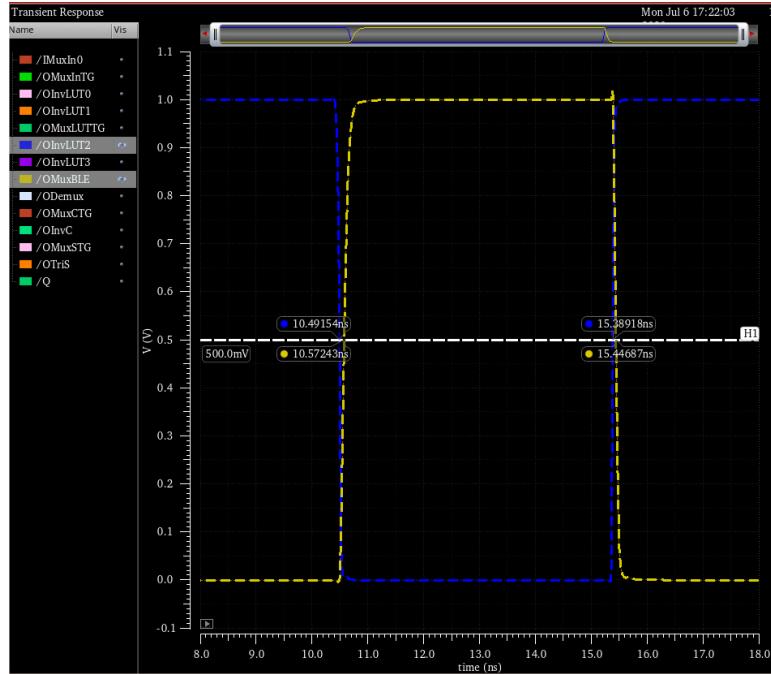


Figure 18. HSpice delay simulation of the CLB BLE Mux. Rise time of 80.89 ps. Fall time of 57.69 ps.

The cell dimensions of the BLE Mux layout is 1.855um x 1.07um. DRC and LVS passed on this layout.

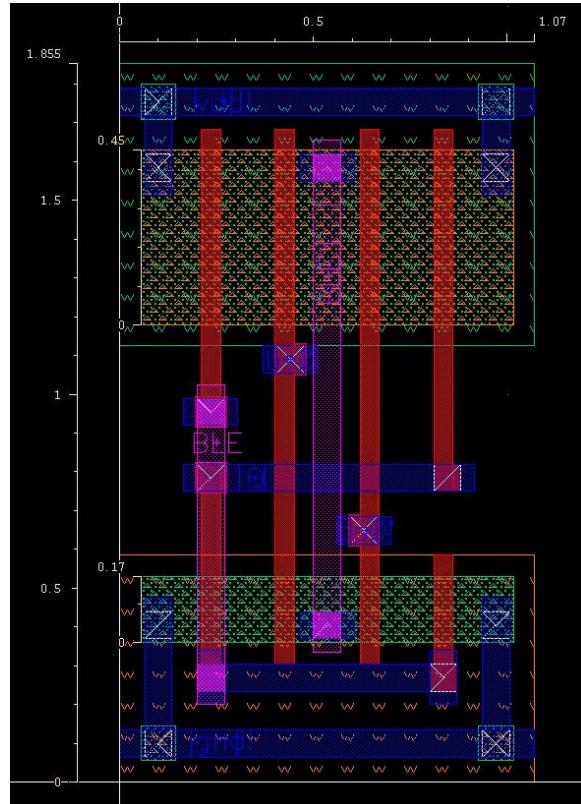


Figure 19. Layout of the CLB BLE Mux. Cell dimensions are 1.855um x 1.07um.

## II.4 Flip Flop:

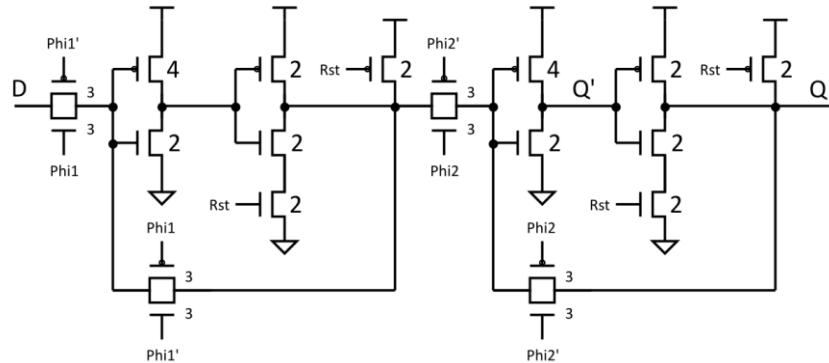
The Flip Flop takes a data input from the LUT Mux, a global clock signal, and a global active LOW reset signal. The output is a latched value of D (Q) and its complement (Q'). The CLB contains 4 total master-slave Flip Flop pairs.

A modified transmission gate based design will be used to implement the Flip Flop. The transmission gate design consists of four transmission gates, two inverters, and two NAND gates with a reset input. This makes a total of 20 transistors for a master-slave Flip Flop pair.

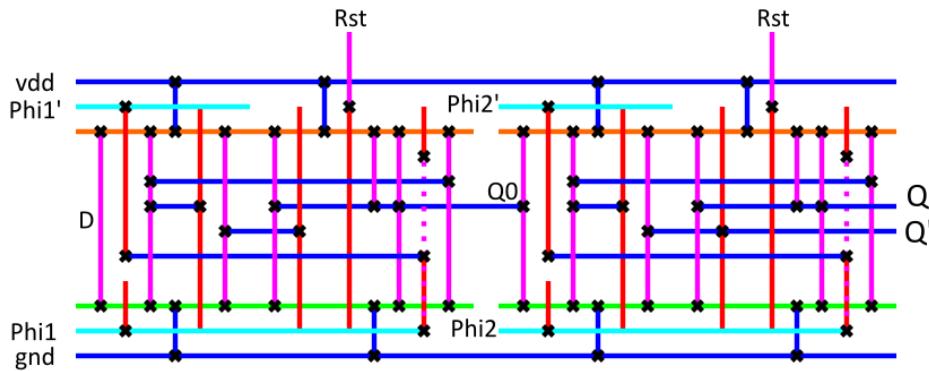
In the modified transmission gate design both clock and clock complemented are needed to correctly toggle the two transmission gates on or off. This poses a challenge for generating the complemented clock to have as little skew as possible with respect to the uncomplemented clock. If, for example, the complemented clock is slightly slower than the uncomplemented clock, during the set stage of the Flip Flop, a D value of '1' will pass through to the inverter faster than a D value of '0'. However, the delay through the Flip Flop isn't too big of an issue since it wouldn't be a part of the critical path.

This same Flip Flop design can be used to create the shift register for the programming bitstream. An active LOW Rst signal can be sent to force all Q outputs to HIGH and force all Q' outputs to LOW. This is used to avoid metastability during power on and off.

All transmission gate transistors are 150nm in width. All NAND gate transistors are 100nm in width. The PMOS' in the inverter are all 200nm in width. The NMOS' in the inverter are all 100nm in width. All transistors are VTL in this cell.



*Figure 20. Transistor schematic of the CLB master-slave Flip Flop pair with sizing (modified design from Figure 2 in [2]).*



*Figure 21. Stick diagram of the CLB Flip Flop. Estimated footprint is 3um x 4um.*

An HSpice simulation was performed to estimate the propagation delay through the CLB Flip Flop. The rise time was 104.23 picoseconds. The fall time was 86.22 picoseconds.

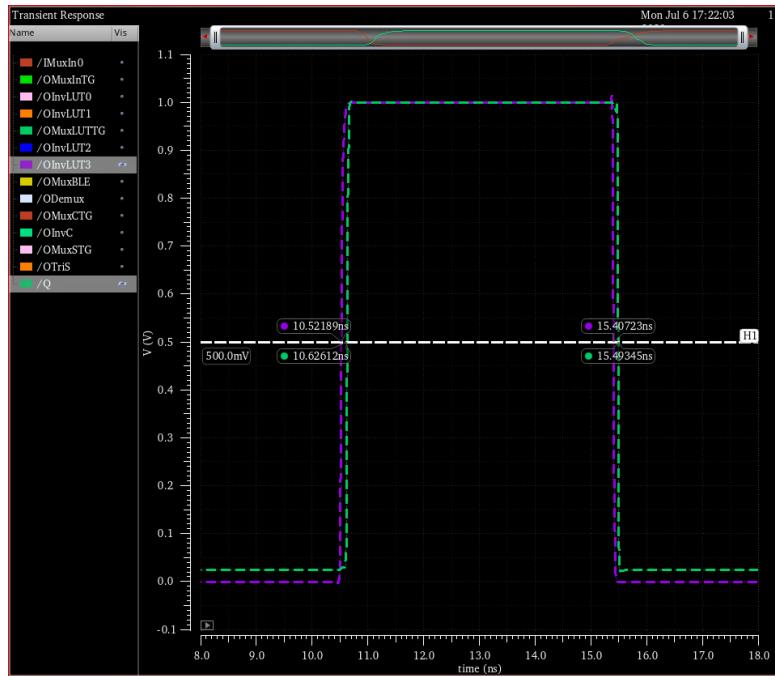


Figure 22. HSpice delay simulation of the CLB Flip Flop. Rise time of 104.23 ps. Fall time of 86.22 ps.

The cell dimensions of the CLB Flip Flop layout is 1.855um x 1.07um. DRC and LVS passed on this layout.

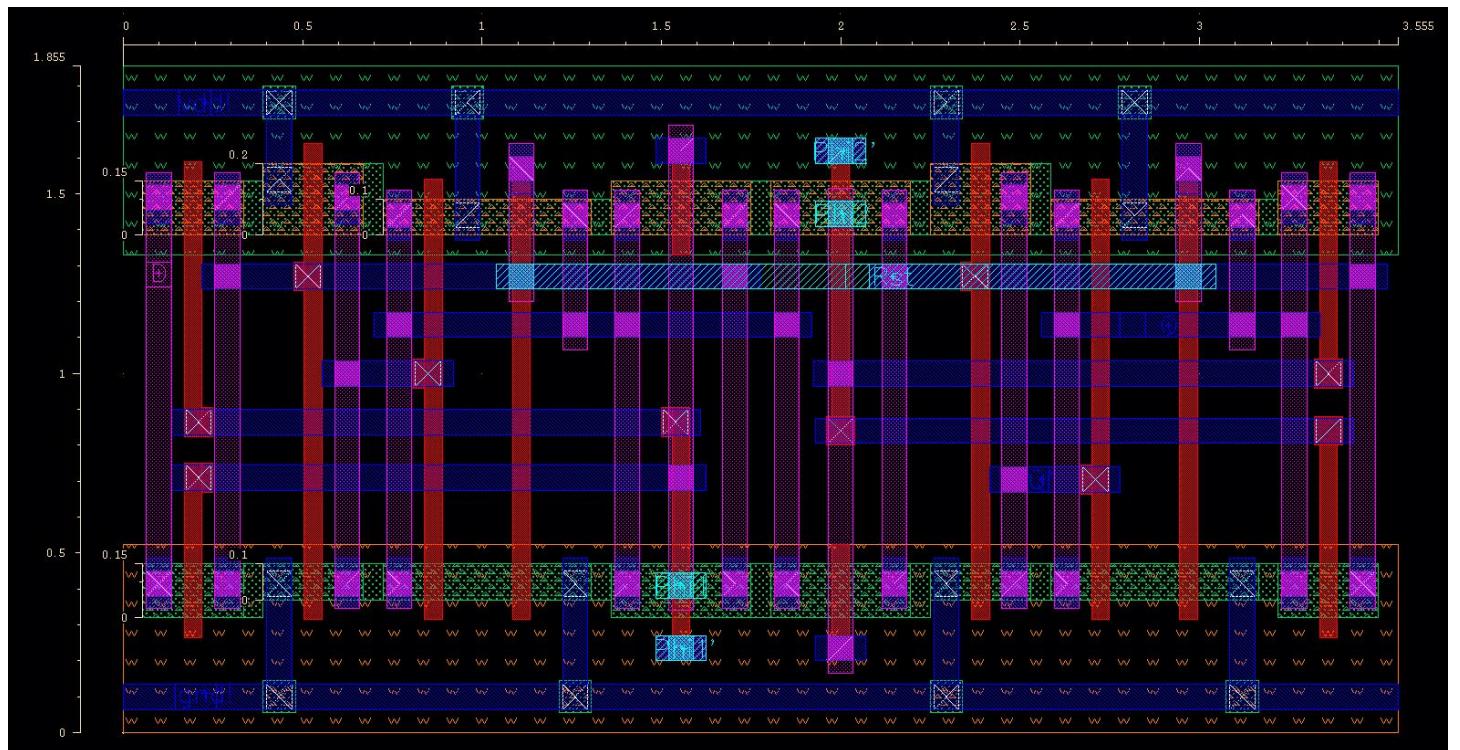
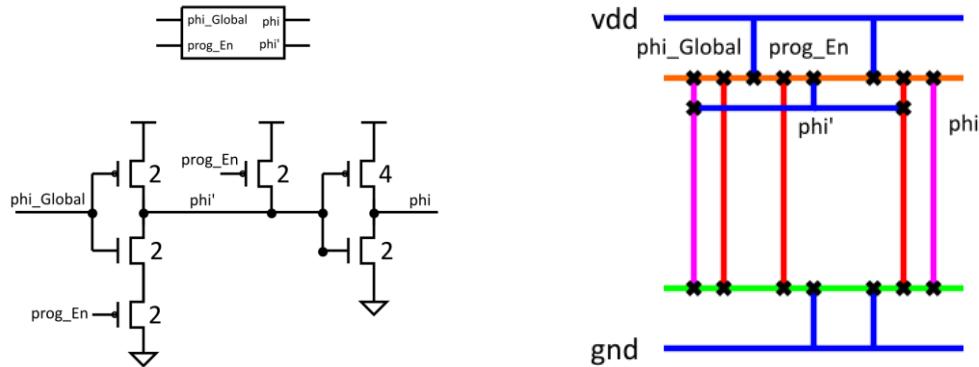


Figure 23. Layout of the CLB Flip Flop. Cell dimensions are 1.855um x 3.555um.

## II.5 Bypass Clock Enable Block:

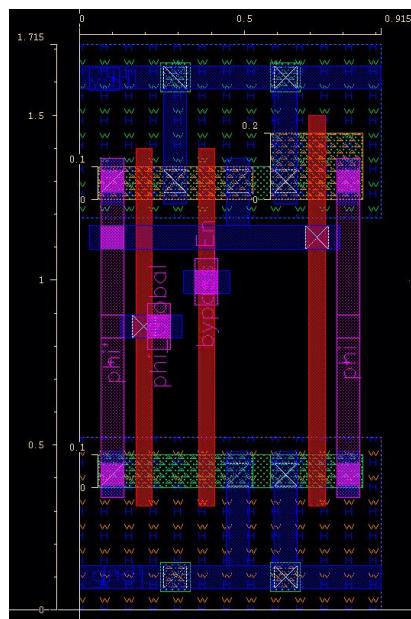
The Bypass Clock Enable Block generates a local phi and phi' signal from a global phi signal and allows clocking of the BLE signals within the CLB Flip Flops. When the Bypass Clock Enable signal is HIGH, the BLE output from the BLE Mux will pass through the D input of the CLB Flip Flop, because the generated phi and phi' signals turn on the transmission gate at the input D. When the Bypass Clock Enable signal is LOW, the transmission gate at the input D of each CLB Flip Flop turns off, causing the current values inside each CLB Flip Flop to become latched. Every CLB will have two Bypass Clock Enable Blocks to locally generate phi1, phi1', phi2, and phi2' for the 4 CLB Flip Flops.

All NAND gate transistors are 100nm in width. The PMOS' in the inverter are all 200nm in width. The NMOS' in the inverter are all 100nm in width. All transistors are VTH in this cell.



*Figure 24. Transistor schematic of the Bypass Clock Enable Block with sizing (left figure). Stick diagram of the Bypass Clock Enable Block (right figure). Estimated floorplan is 3um x 1um (modified design from Figure 2 in [2]).*

The cell dimensions of the Bypass Clock Enable Block layout is 1.715um x 0.915um. DRC and LVS passed on this layout.



*Figure 25. Layout of the Bypass Clock Enable Block. Cell dimensions are 1.715um x 0.915um.*

## II.6 CLB Enable Block:

The CLB Enable Block allows the CLB to disconnect from the vdd supply when it's not in use, significantly reducing leakage power from an unutilized CLB. When the CLB\_En signal is LOW, the global vdd supply will pass through the source of a PFET and a local vdd will be generated at the drain. When the CLB\_En signal is HIGH, the global vdd supply won't be allowed to pass through the PFET. There will be one CLB Enable Block for every vdd line needed by the CLB. Only cells within the CLB will have a local vdd generated this way. All other cells will connect to the global vdd supply. The CLB will need 7 local vdd lines, or 7 total CLB enable Blocks.

The single PFET is VTH and 500nm in width.

The cell dimensions of the CLB Enable Block is 0.905um x 0.39um. DRC and LVS passed on this layout.

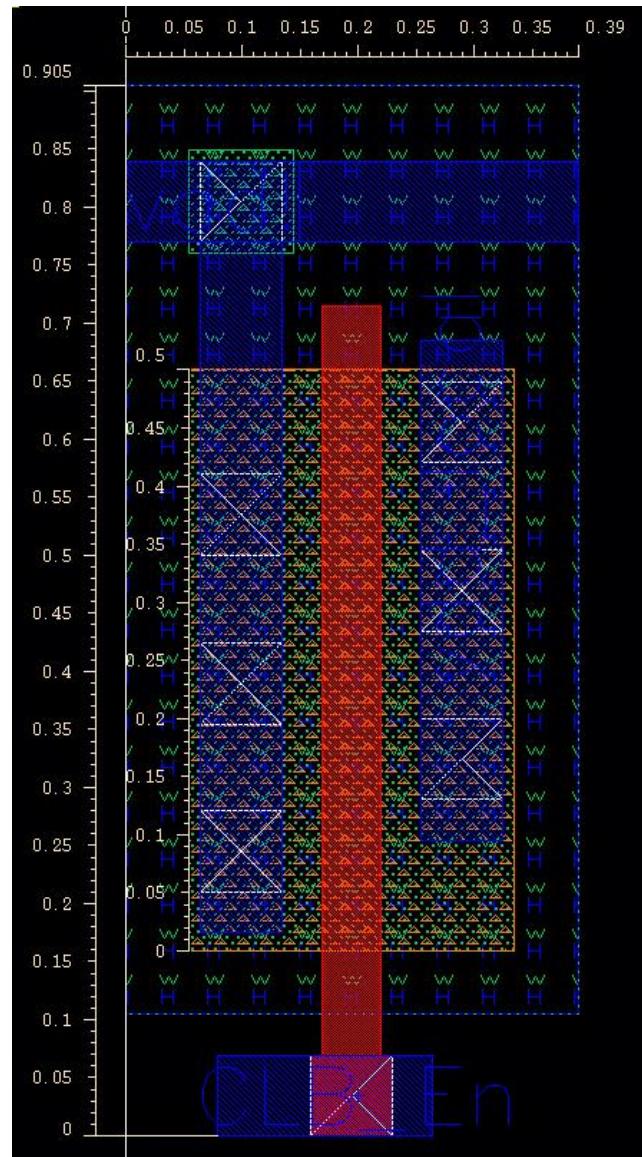


Figure 26. Layout of the CLB Enable Block. Cell dimensions are 0.905um x 0.39um.

## II.7 LUT Inverter:

The LUT inverter is used to restore the signal at the LUT Mux select line input and the LUT Mux output. 2 inverters are used to create a buffer. The CLB contains 40 total LUT inverters (20 buffers).

The NMOS in the LUT inverter is VTL and 100nm in width. The PMOS in the LUT inverter is VTL and 210nm in width. This gives roughly equal rise and fall time.

The cell dimensions of the CLB Enable Block is 2.25um x 1.16um. DRC and LVS passed on this layout.

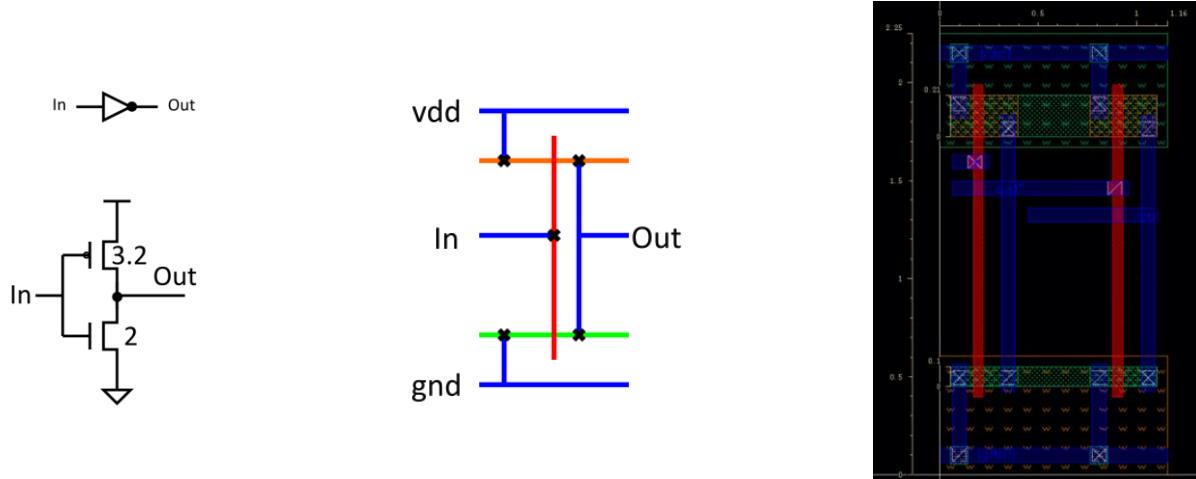


Figure 27. Transistor schematic of the CLB LUT inverter with sizing (left figure). Stick diagram of the CLB LUT inverter (middle figure). Estimated footprint is 3um x 1um. Layout of the CLB LUT inverter pair (right figure). Cell dimensions are 2.25um x 1.16um.

An HSpice simulation was performed to estimate the propagation delay through the CLB Flip Flop. The rise time was 60.93 picoseconds. The fall time was 70.87 picoseconds.

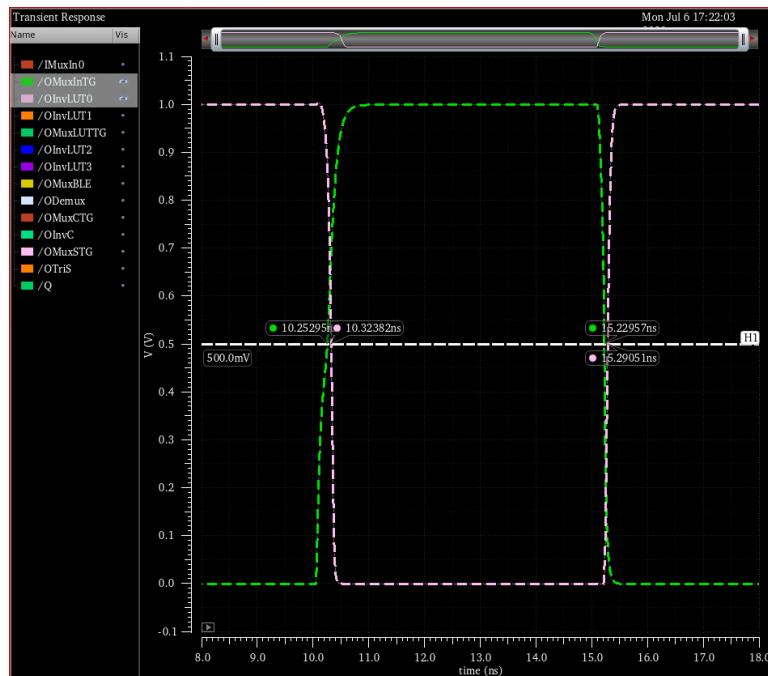


Figure 28. HSpice delay simulation of the CLB LUT inverter. Rise time of 60.93 ps. Fall time of 70.87 ps.

### III. C Block Architecture Design

The C Block forms a routing junction between the CLBs and the S Blocks. Outputs from the CLB are placed on specific tracks as determined by the demuxes, and inputs are selected from the S Block as determined by the C Block Muxes.

The C Block contains 3 major components: the demux, the C Block Mux, and the C Block inverter. Overall, the C Block will contain 2 demuxes, 5 C Block Muxes, and 5 C Block inverters. The demuxes each require 6 bits for the select lines. The C Block Muxes each require 4 bits for the select lines. In total, the C Block will need 32 bits from the programming bitstream.

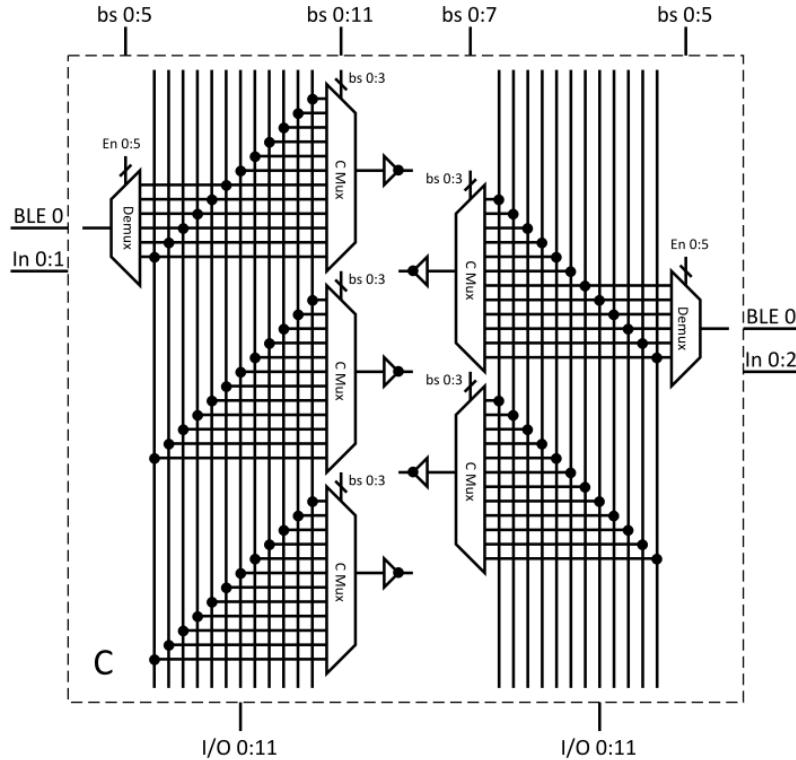


Figure 29. Functional block diagram of the C Block.

The C Block takes an input from the CLB on one side and an input from the CLB on the other side. The C Block gives 3 outputs to the CLB on one side and gives 2 outputs to the CLB on the other side. In the floorplan shown in Figure 30, the C Block Muxes will have their outputs connected directly into the adjacent inverters, where the outputs can feed into the CLBs on either side.

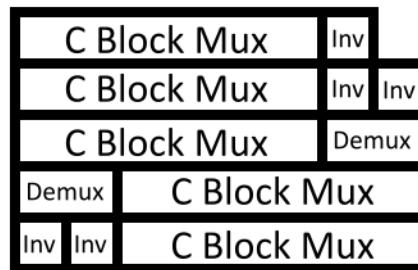


Figure 30. Floorplan of the C Block. The estimated footprint of the C Block is 11.25 um x 15.175um.

Figure 31 shows the full floorplan of the C Block with the bitstream included. The input and output pins are shown with their relative locations on the block. The color of the wire corresponds to the metal layer. A floorplan was created for CLBs connecting North to South, and another floorplan was created for CLBs connecting West to East. Both have similar designs but with different locations for the input and output pins.

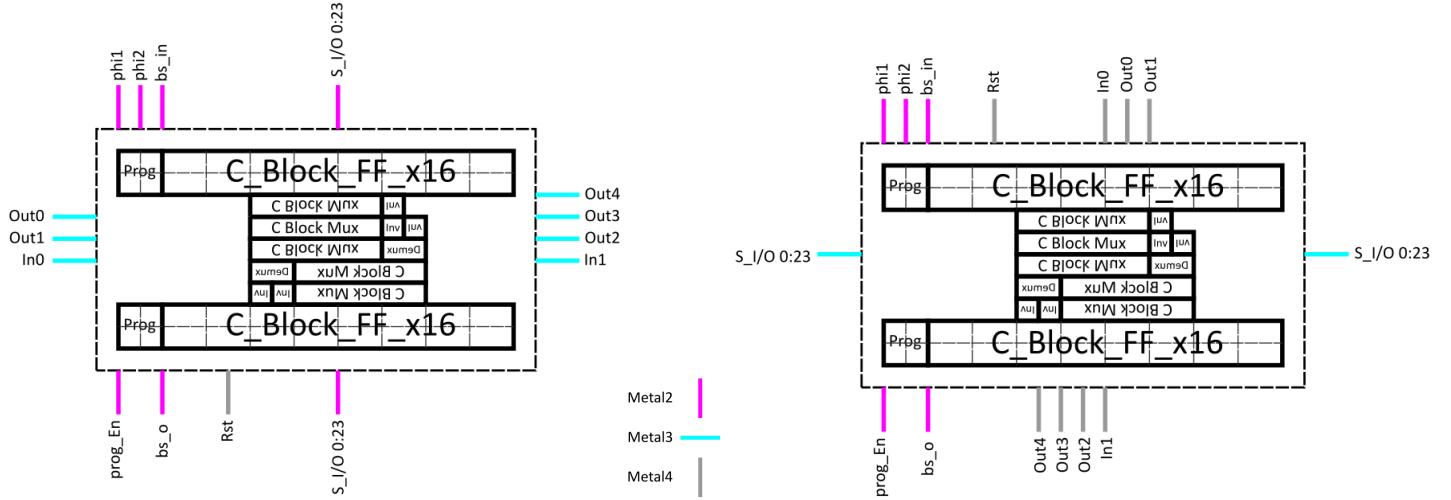


Figure 31. Floorplan of the C Block with 32 Bitstream Flip Flops included. The West-East (left) and North-South (right) C Block configurations shown. The estimated footprint of the C Block is 18.67 um x 25.84um.

An HSpice simulation was performed to estimate the worst case propagation delay through the C Block. The path of the delay travels from the input of the demux to the output of the C Block inverter. The rise time was 428.26 picoseconds. The fall time was 558.84 picoseconds. The HSpice simulation used to calculate this delay is the same used to calculate the critical path delay in Figure 76.

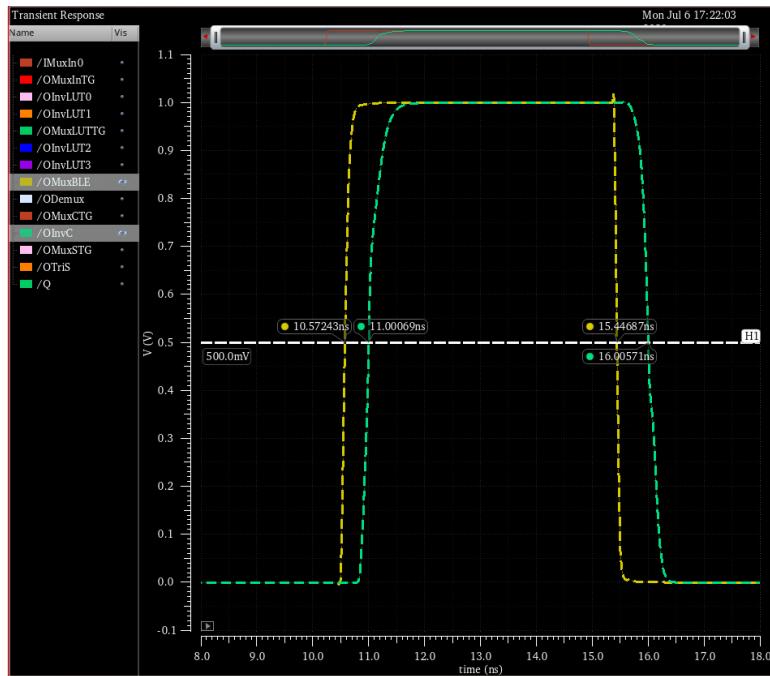


Figure 32. Overall C Block delay. HSpice delay simulation from the input of the tristate demux to the output of the C Block inverter. Rise time of 428.26 picoseconds. Fall time of 558.84 picoseconds.

An HSpice simulation was also performed to estimate the total dynamic and leakage power consumed in the C Block. A full schematic with all cells included was drawn to more accurately model the block. The leakage power through the C Block when passing a LOW was 0.11264 uW. The leakage power through the CLB when passing a HIGH was 0.03358 uW. The dynamic power through the CLB when switching values was 117.34 uW.

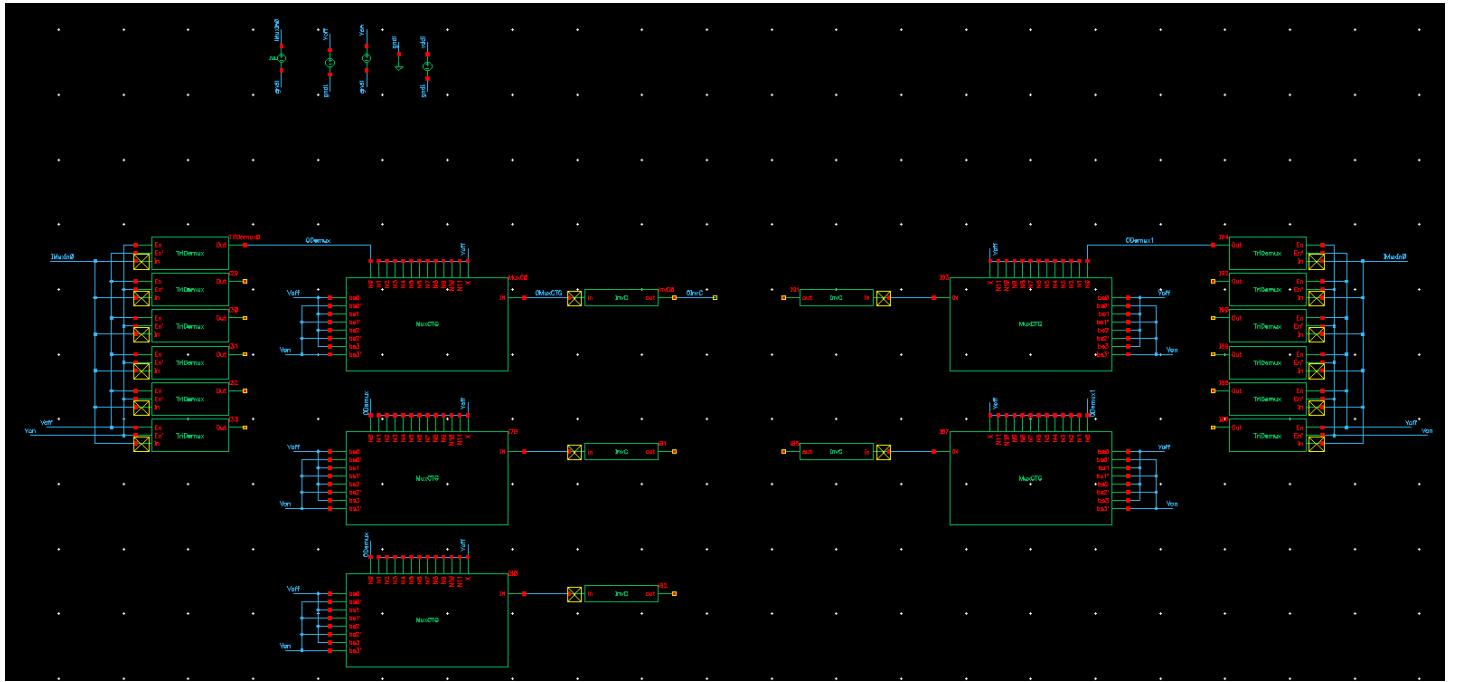


Figure 33. HSpice schematic for power simulation of the entire C Block.

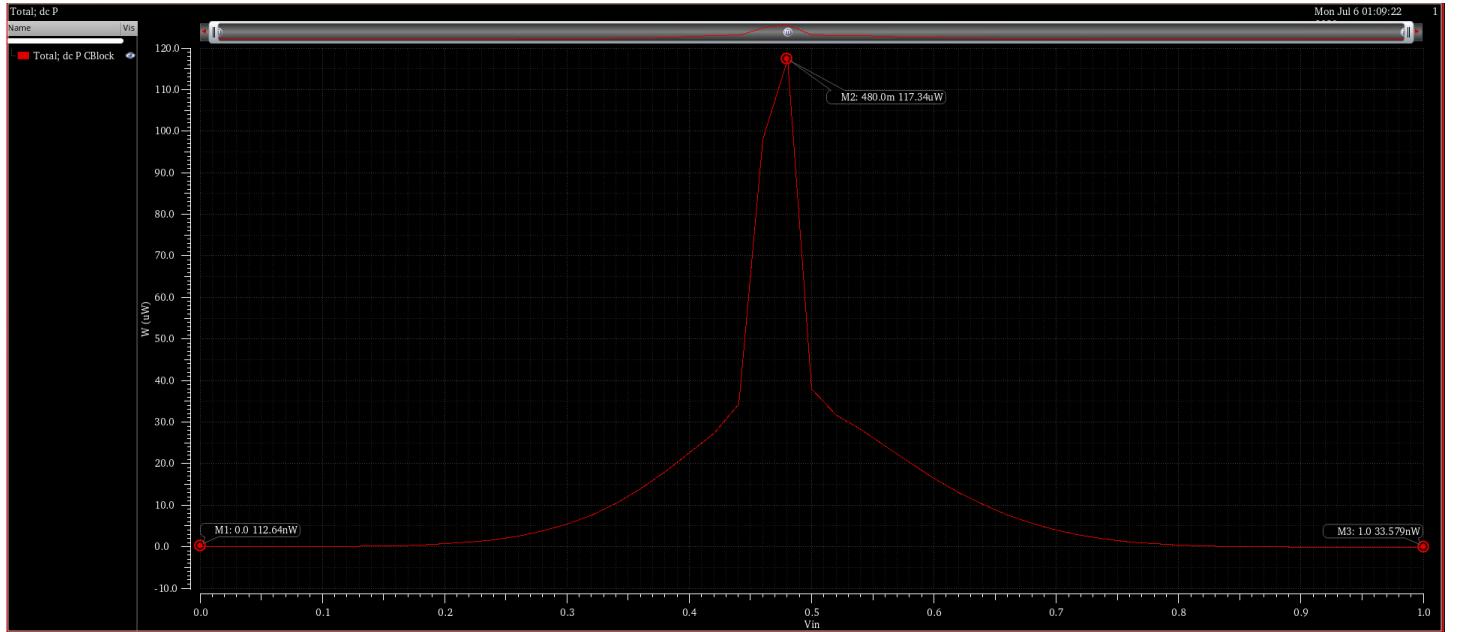


Figure 34. HSpice power simulation of the entire C Block. Marker M1 shows leakage power of 0.11264 uW through the C Block when passing a LOW. Marker M2 shows dynamic power of 117.34 uW through the C Block when switching values. Marker M3 shows leakage power of 0.03358 uW through the C Block when passing a HIGH.

### III.1 Demux:

The demux takes an input from the BLE Mux in the CLB. This input will be placed on one of 6 different tracks as determined by 6 different enable bits, configured during startup using a shift register to store each value of the bitstream. The 5 tracks that aren't selected will remain in a high impedance state, so as to not interfere with outputs from the connected S Blocks. The C Block contains 2 total 1:6 demuxes.

A tristate buffer design will be used to implement the demux. The tristate buffer design consists of 6 tristate buffers. This is a total of 24 transistors. The signal passes through only one gate for this design. In the tristate buffer design, if the enable gates are directly adjacent to vdd and gnd, the delay will be decreased since the enable bits are configured and set during startup.

All NMOS' in the demux are VTL and 300nm in width. All PMOS' in demux are VTL and 700nm in width. These transistors were sized to be larger because this cell has a larger propagation delay than the other cells.

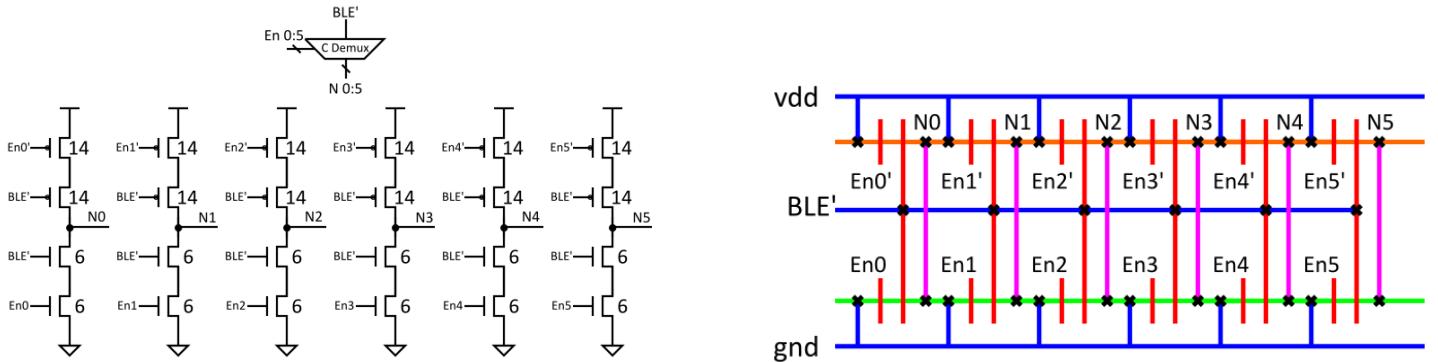


Figure 35. Transistor schematic of the C Block demux (left figure) with sizing. Stick diagram of the C Block demux (right figure). Estimated footprint is 3um x 2.6um.

An HSpice simulation was performed to estimate the propagation delay through the C Block demux. The rise time was 200.94 picoseconds. The fall time was 110.48 picoseconds.

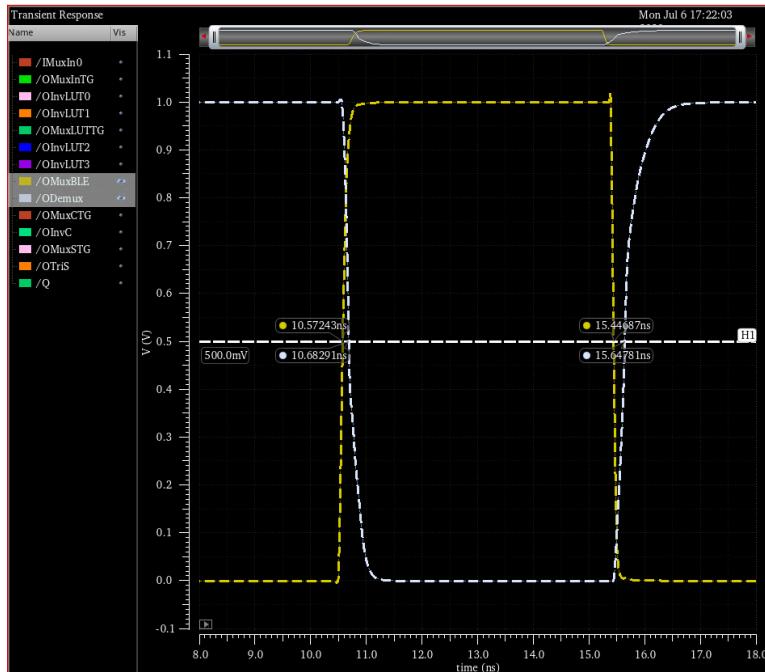


Figure 36. HSpice delay simulation of the C Block demux. Rise time of 200.94 ps. Fall time of 110.48 ps.

### III.2 C Block Mux:

The C Block Mux takes 1 input from the demux output, and 12 inputs from the adjacent S Blocks. This makes a total of 12 inputs. Note that the demux output is connected to one of the S Block inputs/outputs. One of these inputs will be selected with a 4 bit select line, configured during startup using a shift register to store each value of the bitstream. The output of the C Block Mux will feed into 16 Input Muxes. The C Block contains 5 total C Block Muxes. The same 2 stage transmission gate 4:1 mux used to implement the Input Mux will be used to implement the C Block Mux.

The first stage contains four instances of a 4:2 mux. All NMOS' in this stage are VTL and 200nm in width. All PMOS' in this stage are VTL and 250nm in width. The 4 stages will be placed side by side in the layout.

The second stage contains one instance of a 4:2 mux. All NMOS' in this stage are VTL and 200nm in width. All PMOS' in this stage are VTL and 250nm in width. This stage will be placed after the first stages as shown in the stick diagram in Figure 38.

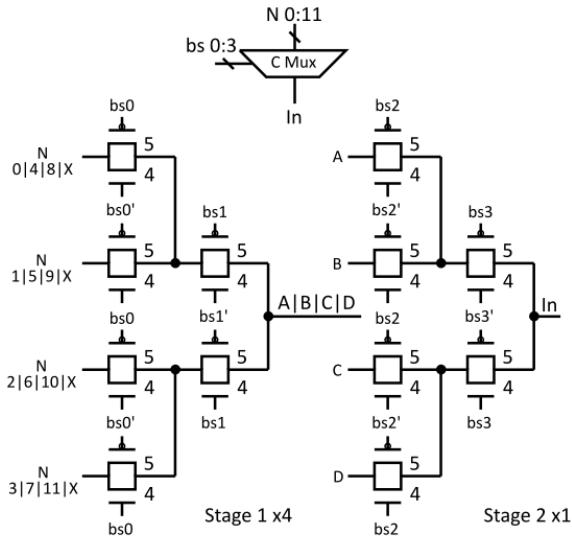


Figure 37. Transistor schematic of the C Block Mux with sizing.

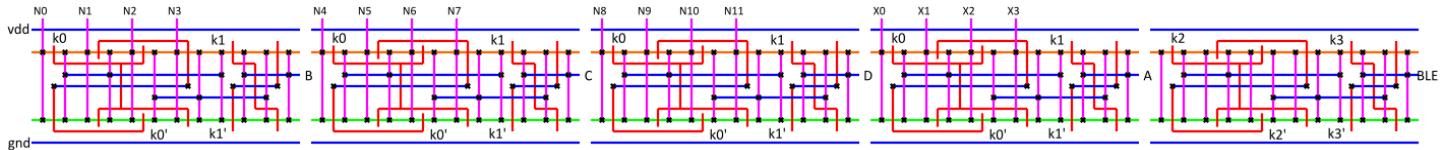


Figure 38. Stick diagram of the C Block Mux. Estimated footprint is 3um x 13um.

An HSpice simulation was performed to estimate the propagation delay through the C Block Mux. The rise time was 184.46 picoseconds. The fall time was 147.9 picoseconds.

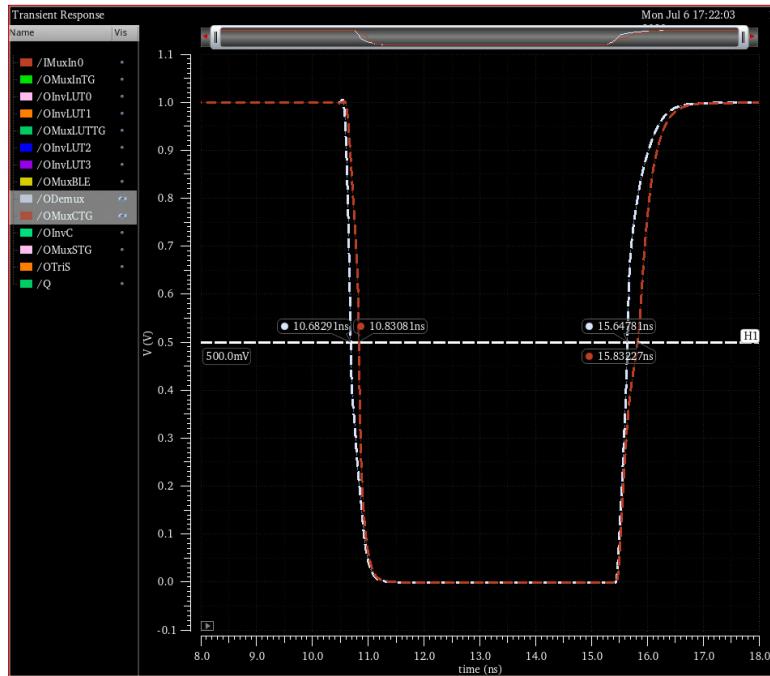


Figure 39. HSpice delay simulation of the C Block Mux. Rise time of 184.46 ps. Fall time of 147.9 ps.

### III.3 C Block Inverter:

The C Block inverter is used to restore the signal at the C Block Mux output. C Block contains 5 total C Block Inverters.

The NMOS in the C Block Inverter is VTL and 300nm in width. The PMOS in the C Block inverter is VTL and 700nm in width. These transistors were sized to be larger because this cell has a larger propagation delay than the other cells.

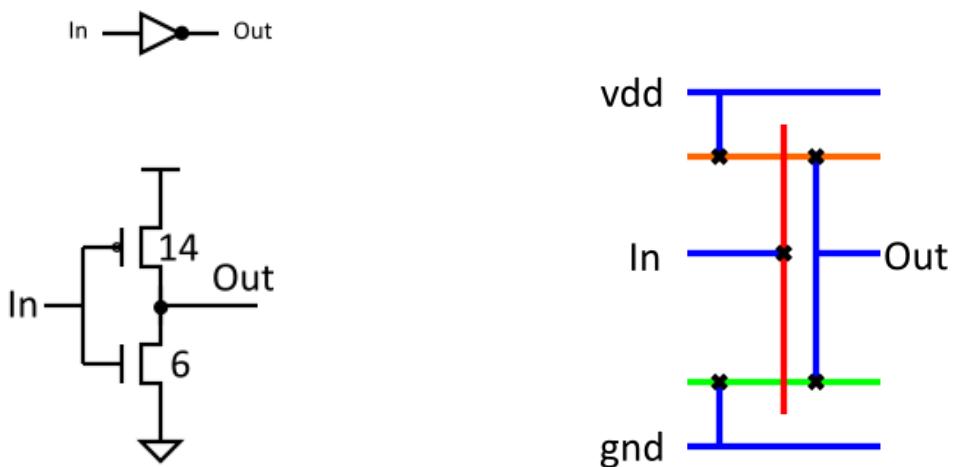


Figure 40. Transistor schematic of the C Block inverter (left figure) with sizing. Stick diagram of cell (right figure). Estimated footprint is 3um x 0.5um.

An HSpice simulation was performed to estimate the propagation delay through the C Block inverter. The rise time was 169.88 picoseconds. The fall time was 173.44 picoseconds.

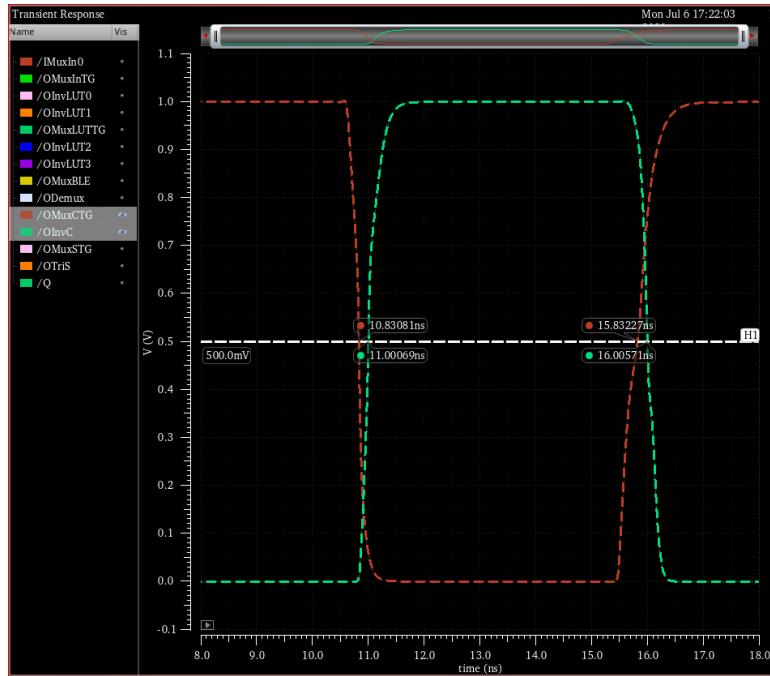


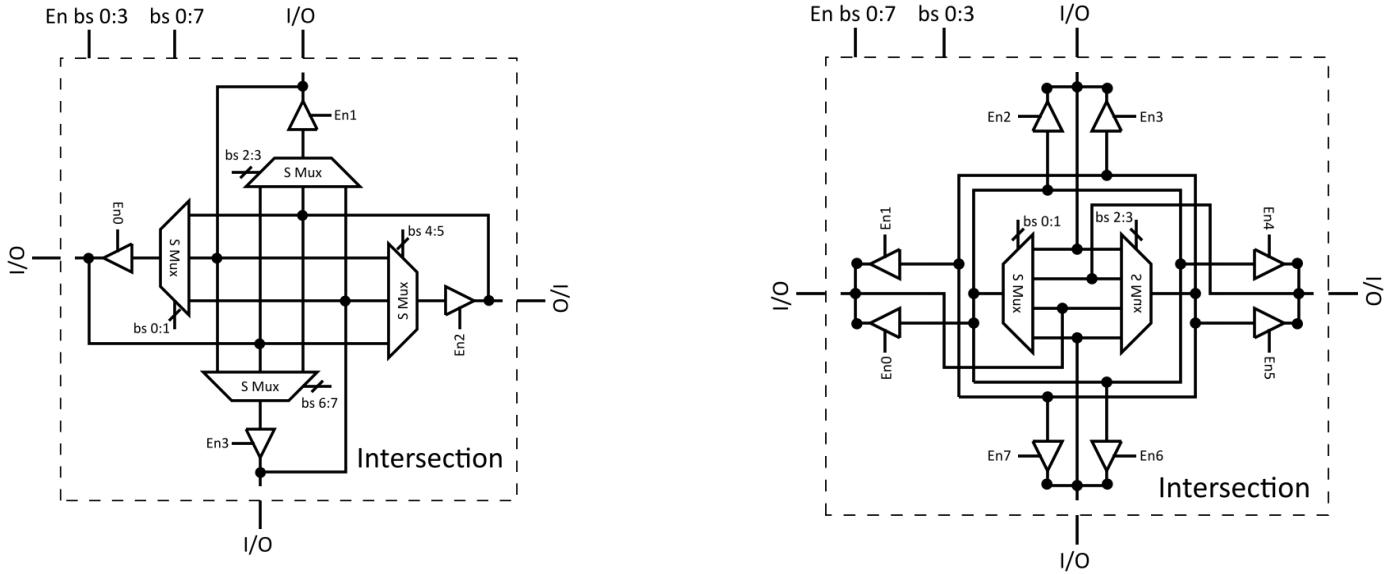
Figure 41. HSpice delay simulation of the C Block inverter. Rise time of 169.88 ps. Fall time of 173.44 ps.

#### IV. S Block Architecture Design

With the original design, the S Block contains 2 major components: the S Block mux, and the tristate buffer. Overall, the S Block will contain 96 S Block muxes, and 96 tristate buffers. Each of the S Block Muxes require 2 bits for the select lines. Each of the tristate buffers require a single bit for the enable bit. In total, the S Block will need 288 bits from the bitstream if using the original design.

Area and power is a large concern in this block because of how many cells are required to support 24 vertical tracks and 24 horizontal tracks. The designs chosen the S Block mux and the tristate buffer were made to minimize both area and power, without having a large impact on delay.

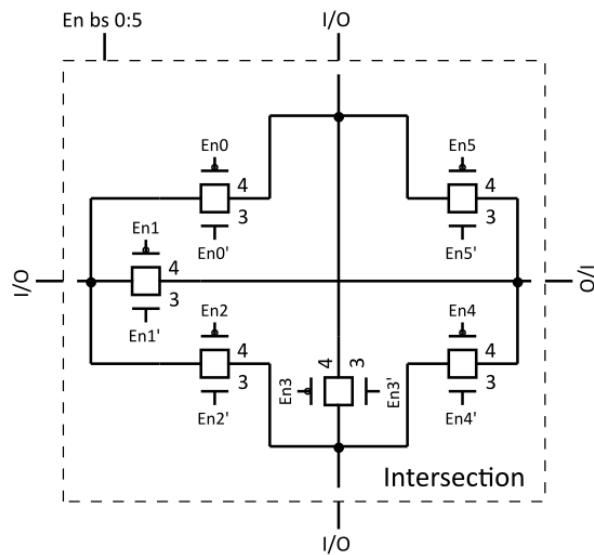
A modified version of the intersection cell with a similar design was created to minimize area, shown in the right figure of Figure 42. The original intersection cell contains 4 tristate buffers and 4 4:1 muxes. The modified intersection cell contains 8 tristate buffers and 2 4:1 muxes. Since the 4:1 mux has an approximate width of 2.5um and the tristate buffer has an approximate width of 0.5um, the modified intersection cell would use approximately 3um less in width, keeping height the same. This would be at the cost of doubling the power consumption of the tristate buffers in the S Block.



*Figure 42. The left figure shows the architecture of the original intersection cell. The right figure shows the architecture of the modified intersection cell.*

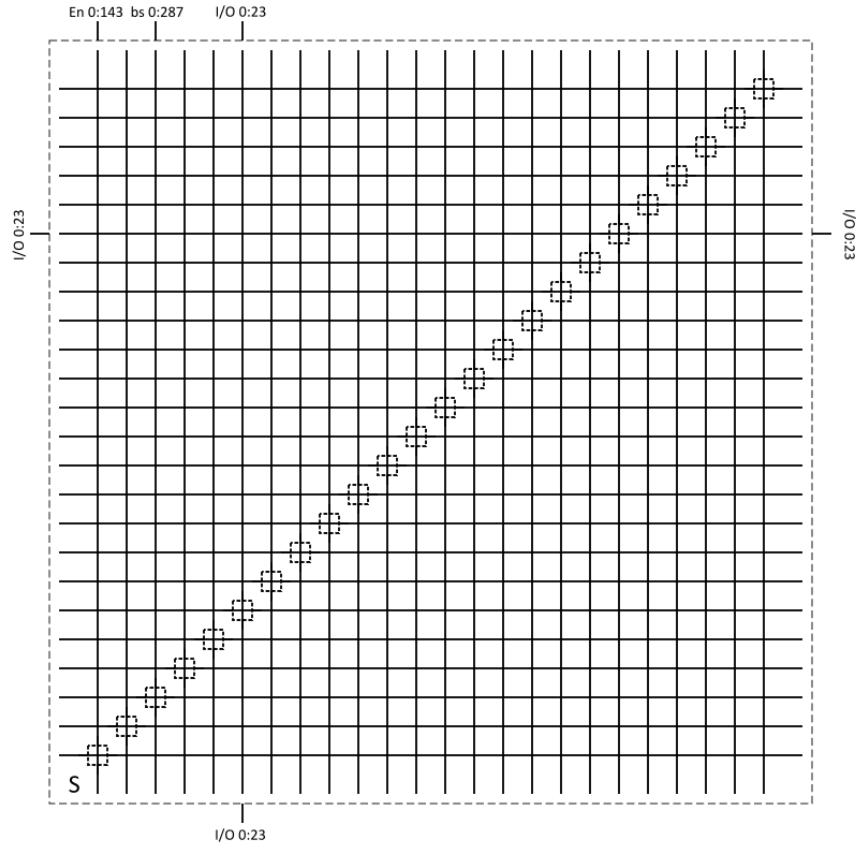
Even with the reduced area of the modified intersection cell, the Bitstream Flip Flops will still take up an estimated footprint of  $61.74\text{um} \times 28.44\text{um}$  if arranged in a  $36 \times 8$  grid. To decrease the area that the Bitstream Flip Flops take up, a different intersection cell design that operates with fewer bits from the bitstream needs to be created. In an intersection, there are only 6 paths a signal can arrive and leave from: N – S, N – W, N – E, W – E, W – S, E – S. This means that a transmission gate could be placed at each path either to allow signals to pass through the path, or to block signals with a high impedance state.

This intersection design would contain 6 transmission gates and only requires 6 bits to program. This decreases the total number of Bitstream Flip Flops needed in the S Block by half to 144. Even the transmission gate design itself would take smaller area than the original intersection design. A buffer (inverter pair) could be placed at the outputs of the intersection cell to restore the signal before passing on to the C Block Mux. The intersection cell shown in Figure 43 will be used for the S Block.



*Figure 43. Architecture of the transmission gate intersection cell.*

The S Block needs to be able to switch 24 different horizontal tracks and 24 different vertical tracks. The simplest way to implement this would be to place an intersection cell at each unique track crossing as shown in Figure 44.



*Figure 44. Functional block diagram of the S Block. Each of the dotted squares on the right figure represents an instance of the intersection.*

The S Block needs a total of 24 intersection cells. These cells can be grouped to form a grid of 3x8 cells. This creates an S Block that is similar in width to the CLB and C Block.

|      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|
| Int. |
| Int. |
| Int. |

*Figure 45. Floorplan of the S Block. Each intersection block contains 6 transmission gates and 4 buffers and the estimated footprint is 4um x 3um. The estimated footprint of the entire S Block is 12um x 24um.*

Figure 46 shows the full floorplan of the S Block with the bitstream included. The input and output pins are shown with their relative locations on the block. The color of the wire corresponds to the metal layer.

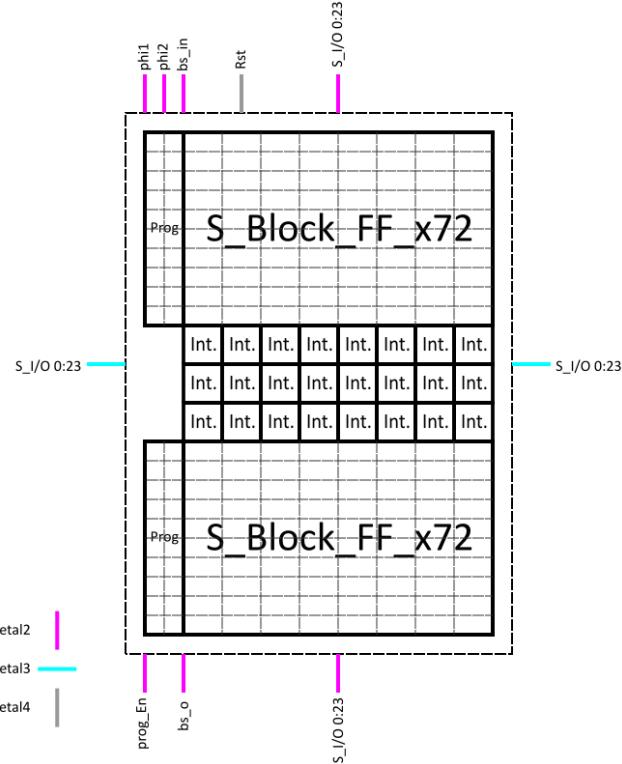


Figure 46. Floorplan of the S Block with 144 Bitstream Flip Flops included. The estimated footprint of the entire S Block is 42.87 um x 28.44um.

An HSpice simulation was performed to estimate the worst case propagation delay through the S Block. The simulation models the unmodified intersection cell shown in Figure 42. The path of the delay travels from the input of the S Block Mux to the output of the tristate buffer. The rise time was 351.63 picoseconds. The fall time was 365.83 picoseconds.

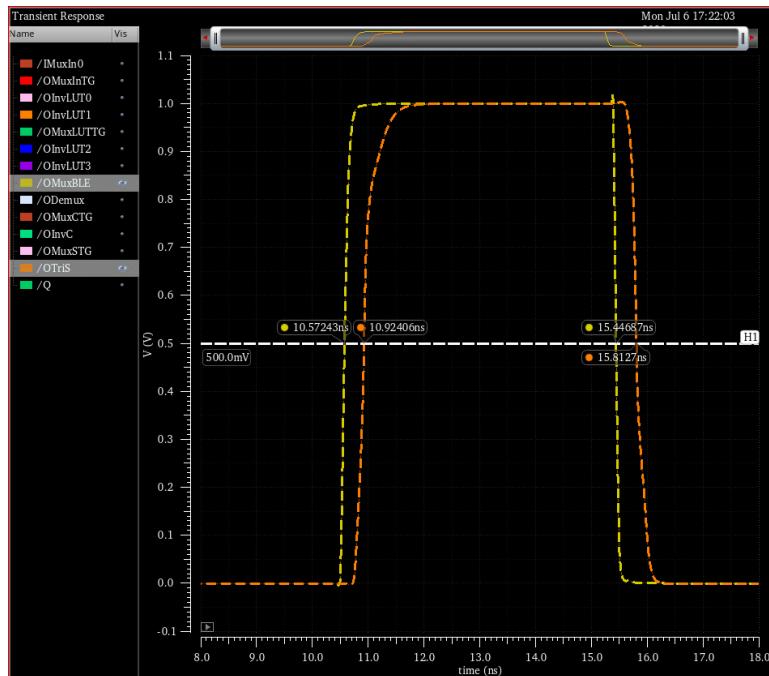


Figure 47. Overall S Block delay. HSpice delay simulation from the input of the S Block Mux to the output of the tristate buffer. Rise time of 351.63 picoseconds. Fall time of 365.83 picoseconds.

An HSpice simulation was also performed to estimate the total dynamic and leakage power consumed in the S Block. The simulation uses models for the unmodified intersection cell shown in Figure 42. A full schematic with all cells included was drawn to more accurately model the block. The leakage power through the S Block when passing a LOW was 152.06 uW. The leakage power through the CLB when passing a HIGH was 152.06 uW. The dynamic power through the CLB when switching values was 156.68 uW.

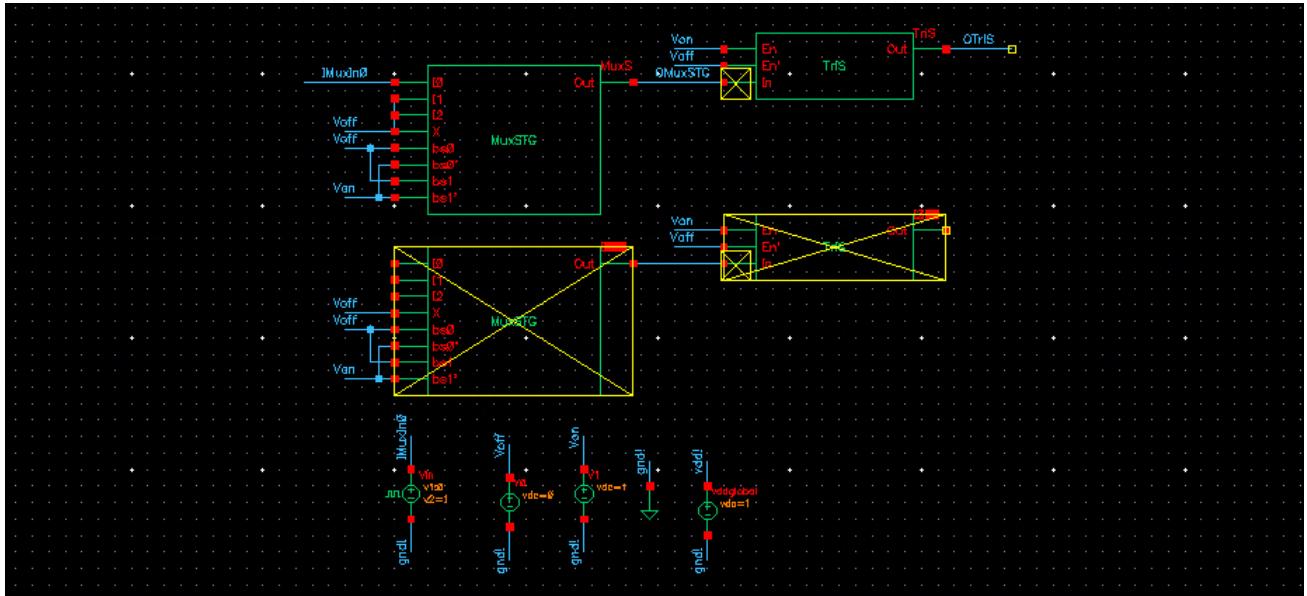


Figure 48. HSpice schematic for power simulation of the entire S Block. The schematic contains 144 instances of the S Block Mux cell and 144 instances of the tristate buffer; they're overlayed on top of each other.

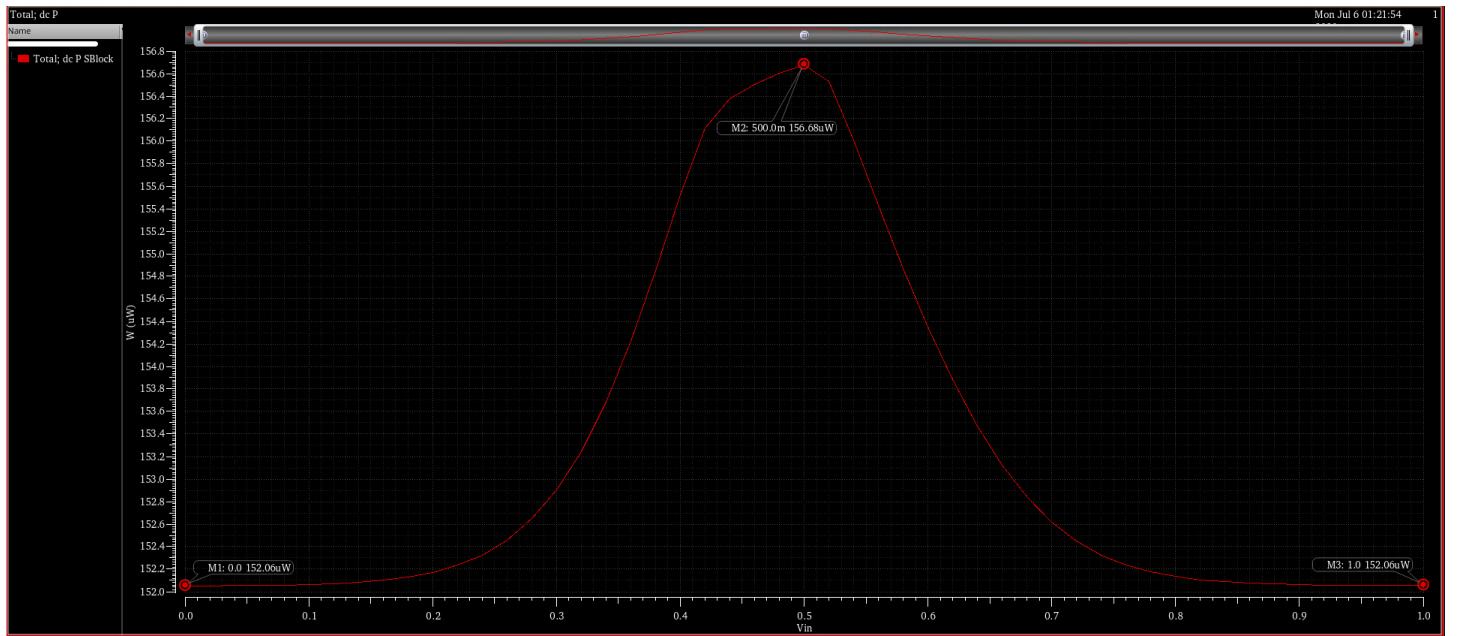
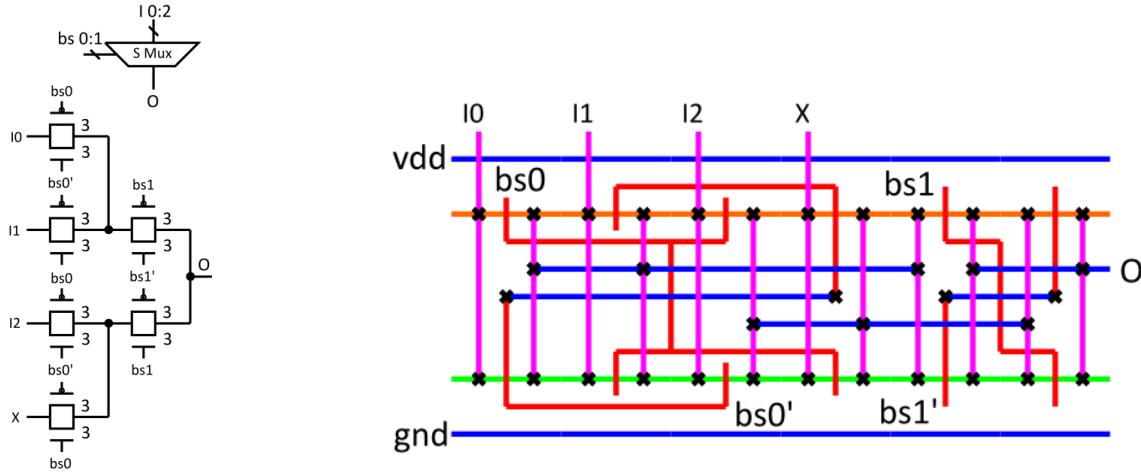


Figure 49. HSpice power simulation of the entire S Block. Marker M1 shows leakage power of 152.06 uW through the S Block when passing a LOW. Marker M2 shows dynamic power of 156.68 uW through the S Block when switching values. Marker M3 shows leakage power of 152.06 uW through the S Block when passing a HIGH.

#### IV.1 S Block mux:

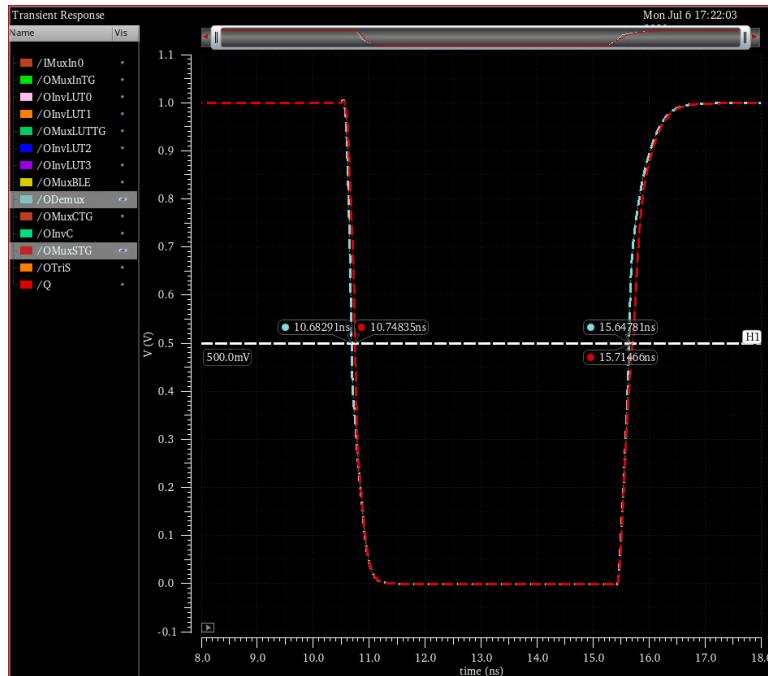
The S Block mux takes in 3 inputs from the C Block. One of these inputs will be selected with a 2 bit select line, configured during startup using a shift register to store each value of the bitstream. The output of the S Block mux will feed into a tristate buffer. For the same reasons as described for the Input Mux, a transmission gate 4:1 mux design will be used to implement the S Block mux. The S Block contains a total of 144 S Block muxes.

All NMOS' are VTL and 150nm in width. All PMOS' are VTL and 150nm in width.



*Figure 50. Transistor schematic of the S Block mux (left figure) with sizing. Stick diagram of the S Block mux (right figure). Estimated footprint is 2um x 2.6um.*

An HSpice simulation was performed to estimate the propagation delay through the S Block Mux. The rise time was 66.85 picoseconds. The fall time was 65.44 picoseconds.



*Figure 51. HSpice delay simulation of the S Block mux. Rise time of 66.85 ps. Fall time of 65.44 ps.*

## IV.2 Tristate buffer:

The tristate buffer takes a single input from the S Block mux. This input can be allowed to pass through with a 1 bit enable signal, configured during startup using a shift register to store each value of the bitstream. The output of the tristate buffer will feed into the C Block Muxes. The tristate buffer allows the connected line at the output to either be an input or an output. The S Block contains a total of 144 tristate buffers,

All NMOS' are VTL and 100nm in width. All PMOS' are VTL and 200nm in width.

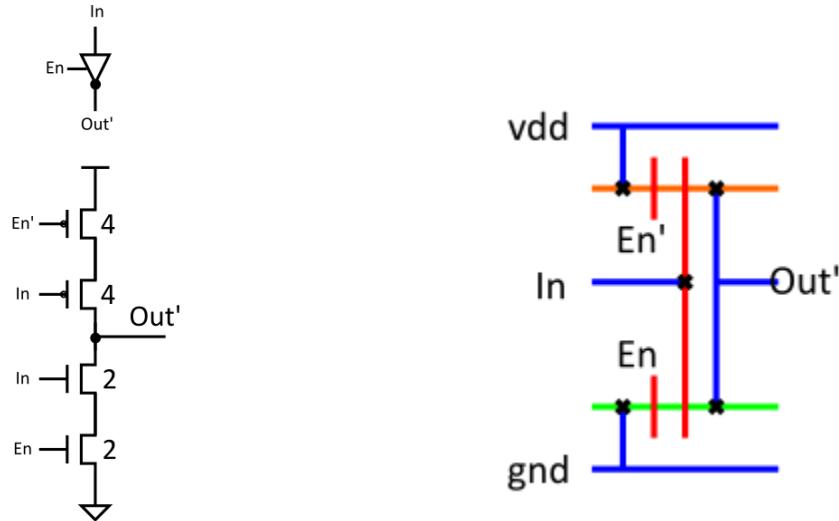


Figure 52. Transistor schematic of the S Block tristate buffer (left figure) with sizing. Stick diagram of the S Block tristate buffer (right figure). Estimated footprint is 2um x 0.5um.

An HSpice simulation was performed to estimate the propagation delay through the tristate buffer. The rise time was 175.71 picoseconds. The fall time was 98.04 picoseconds.

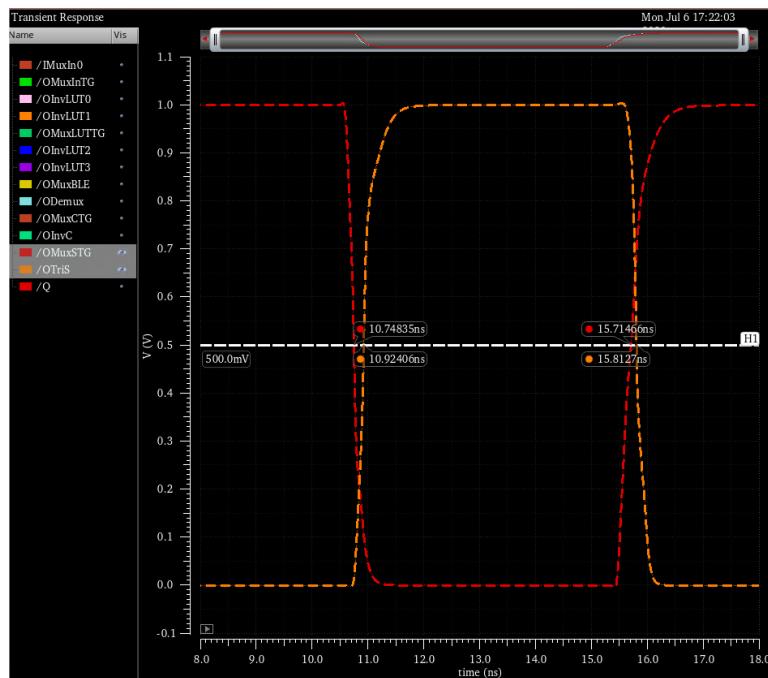


Figure 53. HSpice delay simulation of the S Block tristate buffer . Rise time of 175.71 ps. Fall time of 98.04 ps.

## V. FPGA Architecture Design

As mentioned before, the CLB, C Block, and S Block should be designed in a way that allows seamless tiling with each other. Therefore, the floorplans were created so that the pins of each block roughly aligned with each other. Figure 54 shows the CLB floorplan, the two C Block floorplans, and the S Block floorplan side by side. The West and East input and output pins of the CLB align exactly with the West-East C Block configuration. The North and South input and output pins of the CLB align exactly with the North-South C Block configuration. The input/output pins of the S Block also align exactly with the input/output pins of the two C Block configurations.

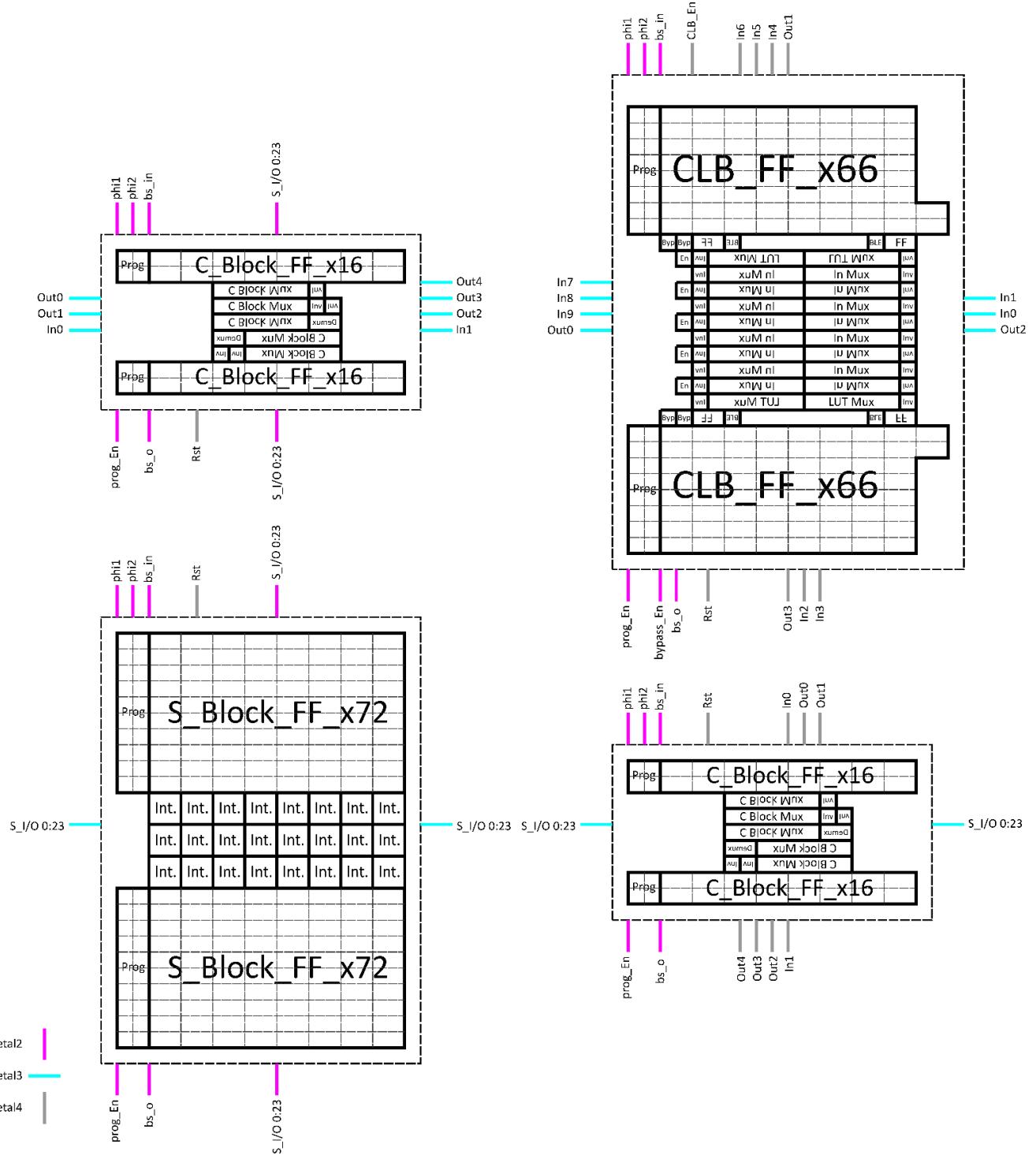


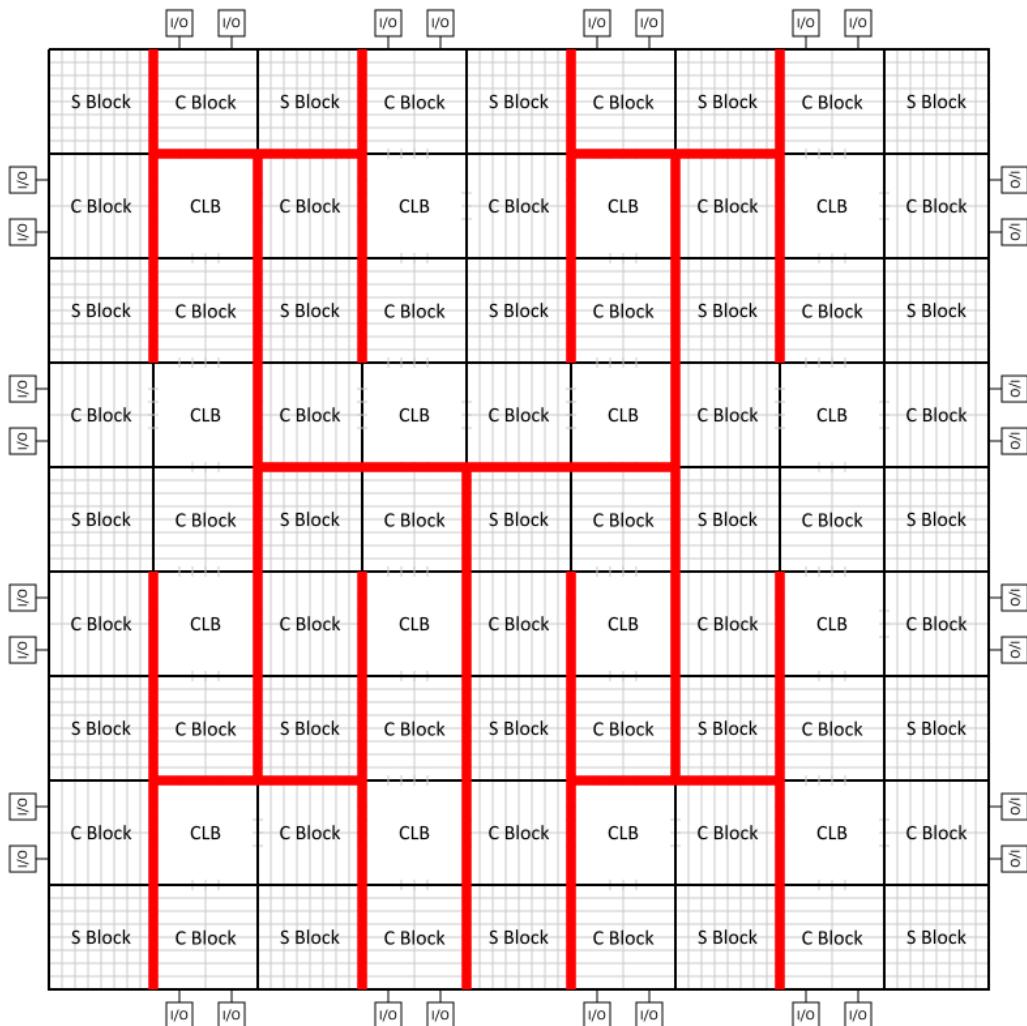
Figure 54. Final floorplan design of the CLB, 2 C Block configurations, and the S Block tiled side by side.

## V.1 Clock

This FPGA design will require two global clock signals, phi1 and phi2, for the Bitstream Flip Flops and the BLE Flip Flops. From the global clock signals, local phi1, phi1', phi2, and phi2' signals will be generated for every row of Bitstream Flip Flops in the CLB, C Block, and S Block. This is accomplished by using two Program Clock Enable Blocks. Similarly, local phi1, phi1', phi2, and phi2' signals will be generated for every row of BLE Flip Flops within a CLB. This is accomplished by sending the global phi1 and phi2 through two Bypass Clock Enable Blocks.

The global clock distribution is important for ensuring that the clock signal reaches each block with as little skew as possible. The goal of this project wasn't focused on clock design or clock generation, so detailed clock analysis and architecture was not performed. As used in most modern FPGA designs, an H bar clock tree distribution was implemented in this design. Due to the symmetry and equidistant branching of this design, the clock delay reaching the ends of each branch should have minimal variation.

To accommodate the clock tree, a routing channel will be left between each block. The routing channel will be approximately the widths of 2 Program Clock Enable Blocks as they will extrude out from each layouts of each series of Bitstream Flip Flops. This routing channel will also contain the power and ground distributions on a lower metal layer.

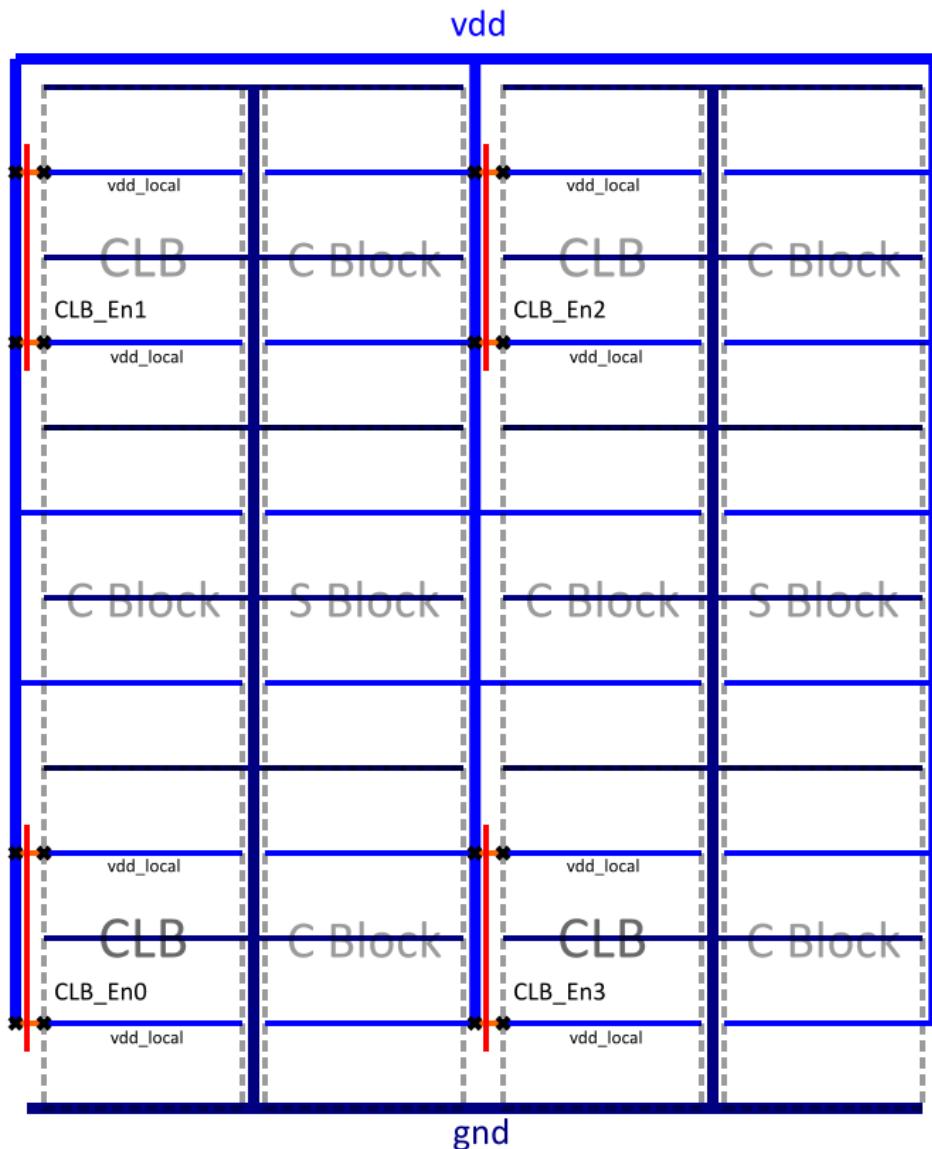


*Figure 55. H bar clock tree distribution running along the routing channel between each block (modified from Figure 11 in [1]).*

## V.2 Power

This FPGA design uses a single power domain of 1.0V. The power and ground distribution will branch out in interweaving fingers. For each CLB, CLB Enable Blocks will generate local power lines from the main power branches. If a CLB is unutilized, the local power lines for that specific CLB won't be generated, leaving the entire block unpowered. More detail on this block is given in its respective section further on in the report.

Figure 4 shows the power and ground distribution design, which can be extended arbitrarily in the vertical direction or the horizontal direction by adding more branches. Both power and ground will be carried with metal 1. White space between each block is minimized to a maximum of two metal 1 spacing distances and whatever the width of the power and ground wires are. For instance, if the widths of the power and ground lines are 100nm, the maximum spacing between each block would be 240nm. This design provides a power line to each column of blocks, which allows each of the CLBs to generate its own local vdd without affecting any other block. This distribution will run along the routing channel between each block.



*Figure 56. Power and ground distribution design. The main power and ground lines (bolded lines) will run along the routing channel between each block. Power is shown in lighter blue and ground is shown in darker blue for visual differentiation.*

## VI. FPGA Bitstream Design

The configurable bitstream in an FPGA is what makes the FPGA programmable. The individual bits latched within the bitstream become the enable signals, the select lines, and the Look Up Table values for the CLBs, C Blocks, and S Blocks in the FPGA. In my architecture, the bitstream is implemented as a shift register composed of flip flops connected in series. Each flip flop will store a single bit of the full bitstream. The bitstream is entirely reprogrammable and is only passed through the FPGA once during startup. The values of the bitstream remain latched until the FPGA is switched off, or until the active LOW Rst signal is sent to the Bitstream Flip Flops.

A single CLB requires 132 bits total from the programming bitstream. A single C Block requires 32 bits total from the programming bitstream. A single S Block requires 432 bits total from the programming bitstream. The placement of the Bitstream Flip Flops will make the bitstream “snake” along the top and bottom edges of each block, providing each block with localized access to the number of bits it needs. Due to the large number of bits required by each block in the FPGA, the overall bitstream will take up the majority of the FPGA footprint.

For the CLB, I split the bitstream in half to “sandwich” the top and bottom edges of the CLB. Because my CLB design is symmetrical vertically and horizontally, a lot of layout effort was saved because only half of the Bitstream Flip Flops had to be routed to the CLB and the other half could just be mirrored. Each bitstream half consists of 66 Bitstream Flip Flops. The Bitstream Flip Flops form an 8x8 grid, with two additional instances protruding from one end. The width of 8 Bitstream Flip Flops was nearly identical to the width of the CLB, motivating the 8x8 grid design decision. The empty space produced by the 2 protruding Bitstream Flip Flops could eventually be filled with more Bitstream Flip Flops from the C Block or the S Block. The bitstream input pin and the bitstream output pin do happen to be on the same side of this cell design, however, by adding an extra row of Bitstream Flip Flops to create an odd number of flip flop rows, the bitstream output would exit the opposite side from the bitstream input. Extra routing may also be added to move the bitstream output.

Each row of Bitstream Flip Flops has its two Program Clock Enable Block to generate local phi1, phi1', phi2, and phi2' signals from a global phi1 and a global phi2 clock signal. The Program Clock Enable Block will also determine whether the Bitstream Flip Flops are in the bitstream programming state or the latching state.

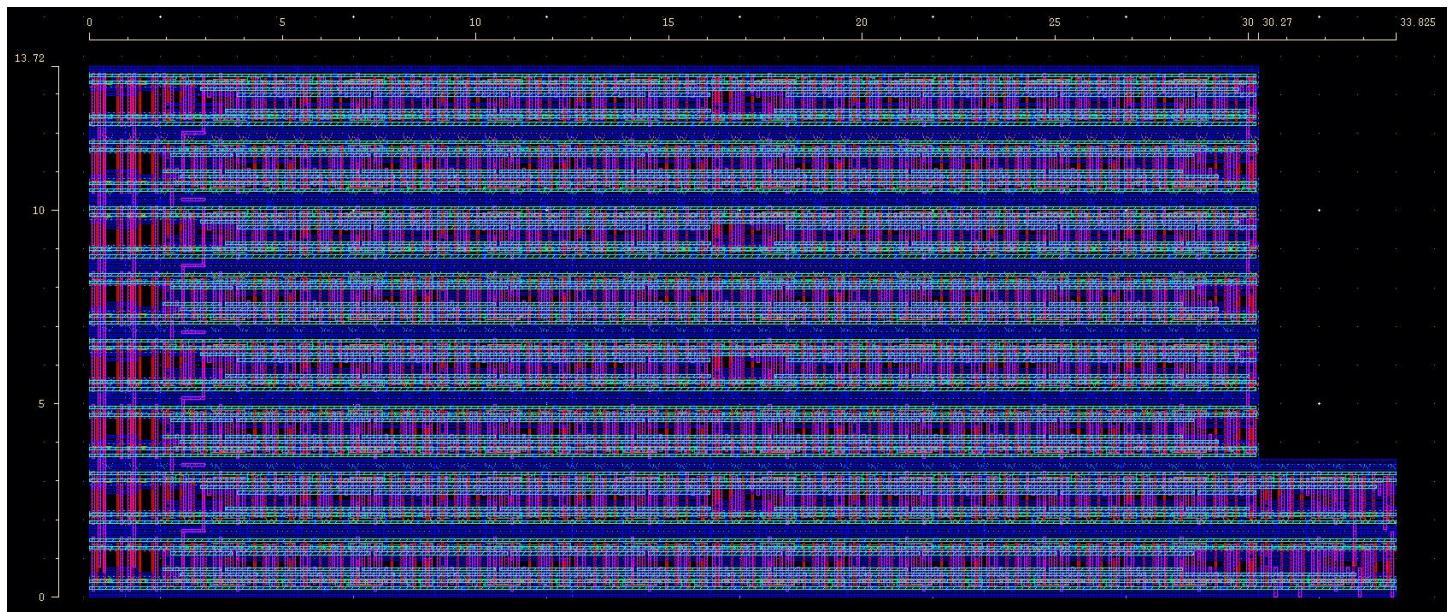


Figure 57. Layout of half of the bitstream for the CLB. This block contains 66 Bitstream Flip Flops. Cell dimensions are 13.72um x 33.825um. Hierarchy level 1.

The path of the bitstream in this block flows from the input D of the Northeast Bitstream Flip Flop through the row of 8 Bitstream Flip Flops, then down to the next column continuing in the reverse direction. It follows this path until it exits the last Bitstream Flip Flop. As mentioned previously, an even number of rows will make the bitstream exit the same side as the input. This can be seen in Figure 58.

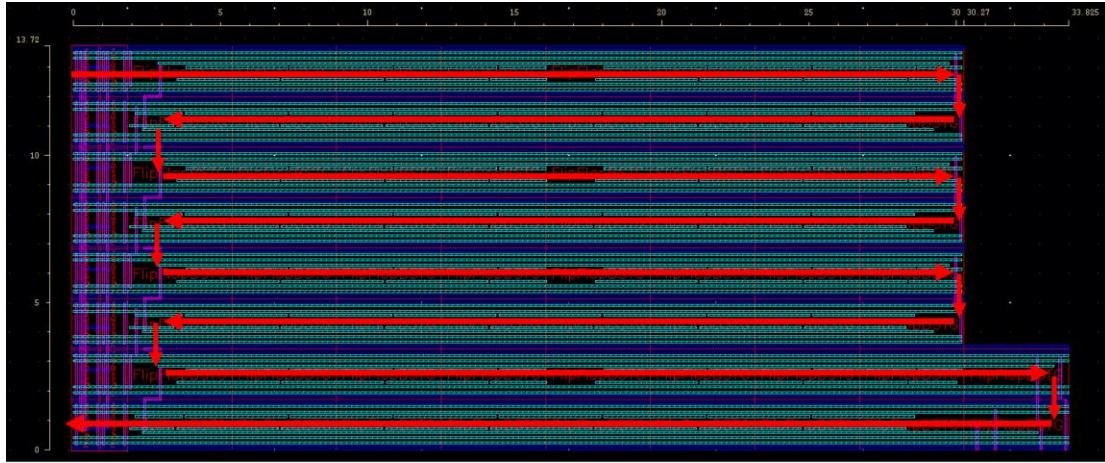


Figure 58. Layout of half of the bitstream for the CLB. This block contains 66 Bitstream Flip Flops. Cell dimensions are 13.72um x 33.825um. Hierarchy level 0. The bitstream direction is indicated with red arrows.

The entire bitstream path for the CLB is shown in Figure 59. The corresponding cell input that uses each bit is listed underneath the name given for the bitstream bit. For example, the first Bitstream Flip Flop, bs0, has its output routed to the k3 select line of an Input Mux in the CLB, In6\_k3. In6 is the 7<sup>th</sup> Input Mux instance in the CLB. Another example, the last Bitstream Flip Flop, bs127, has its output routed to the 15<sup>th</sup> input line of a LUT Mux in the CLB, LUT3\_14. LUT3 is the 4<sup>th</sup> LUT Mux instance in the CLB.

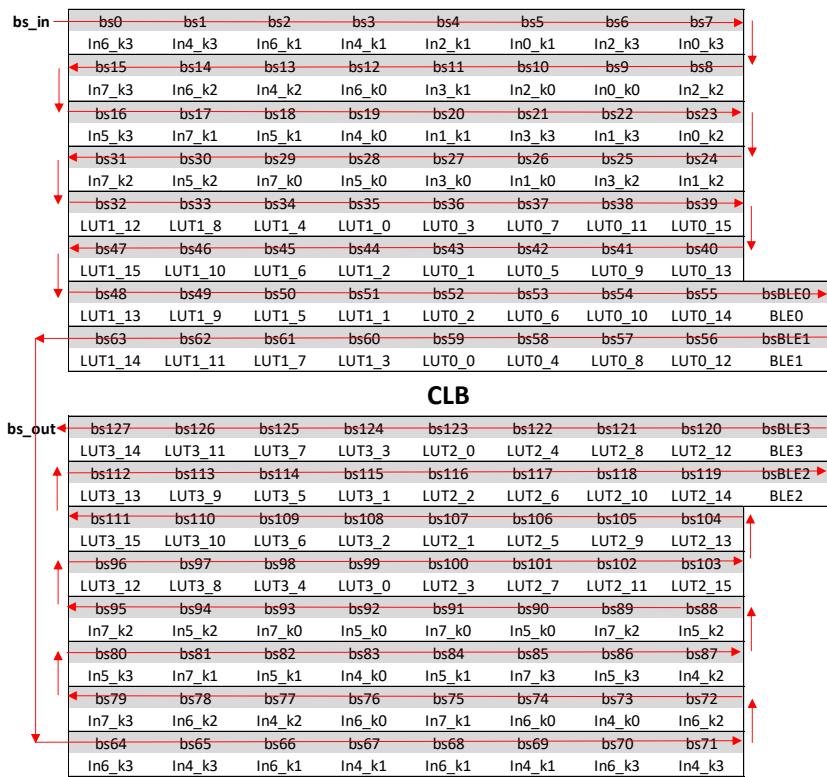


Figure 59. Diagram and mapping of the full CLB bitstream. There are 132 bits total. Each bit is shown with its corresponding connection in the CLB. The bitstream direction is indicated with red arrows.

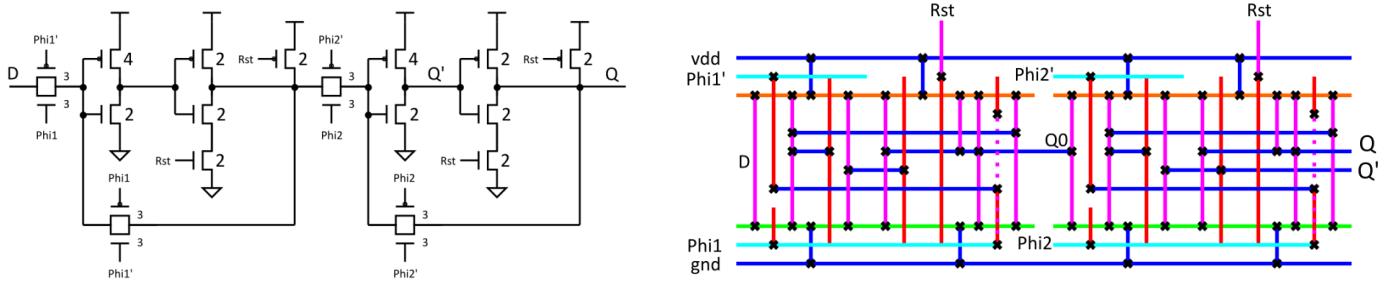
## VI.1 Bitstream Flip Flop:

The Bitstream Flip Flop latches values assigned from the bitstream and uses those value to “program” the select bits, LUT bits, and enable bits in the entire FPGA design. The same design used for the CLB Flip Flops is used for the Bitstream Flip Flops. Please view the Flip Flop section under CLB Architecture Design for explanation of design.

The Flip Flops in the CLB use low threshold turn on voltage (VTL) transistors to ensure the fastest signal propagation out of the three types of transistors available to us. The caveat to the VTL transistors, is that they have the highest on current since they turn on at a significantly lower voltage, which leads them to consume a lot power. On the other hand, VTH transistors have the lowest power consumption but the highest propagation delay. Since the programming bitstream only needs to be initialized once at startup, the Flip Flops used in the shift register don’t need to have the fastest signal propagation. The shift register will also contain the greatest number of active cells, so power would be of significant concern. For this reason, the Flip Flops in the shift register will only use VTH transistors which will decrease the overall leakage power across the entire design.

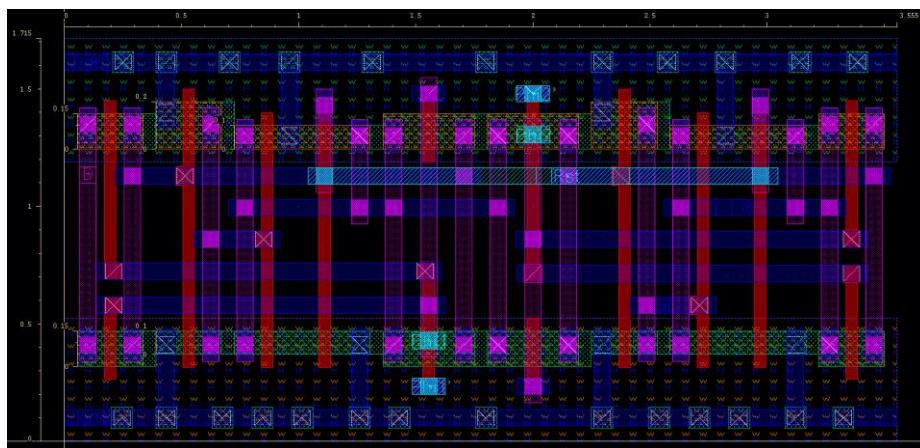
An active LOW Rst signal can be sent to force all Q outputs to HIGH and force all Q’ outputs to LOW. This is used to avoid metastability during power on and off.

All transmission gate transistors are 150nm in width. All NAND gate transistors are 100nm in width. The PMOS’ in the inverter are all 200nm in width. The NMOS’ in the inverter are all 100nm in width. All transistors are VTH in this cell.



*Figure 60. Transistor schematic of the Bitstream master-slave Flip Flop pair with sizing (left figure). Stick diagram of the Bitstream master-slave Flip Flop pair (right figure). Estimated footprint is 3um x 4um (modified design from Figure 2 in [2]).*

The cell dimensions of the Bitstream Flip Flop layout is 1.715um x 3.555um. DRC and LVS passed on this layout.

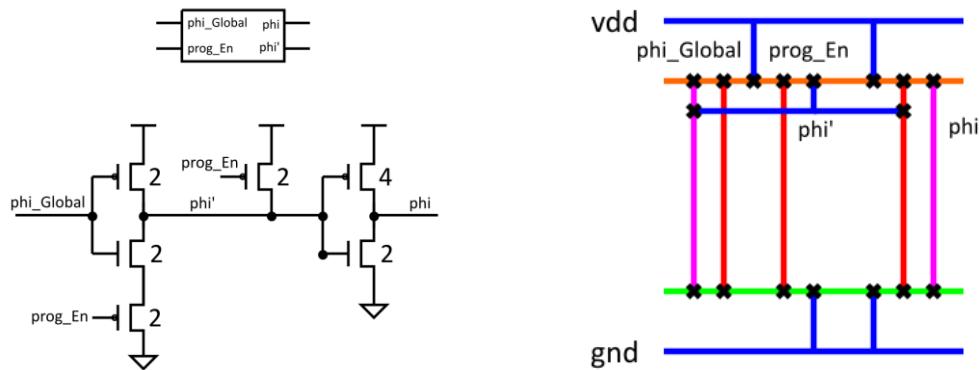


*Figure 61. Layout of the Bitstream Flip Flop. Cell dimensions are 1.715um x 3.555um.*

## VI.2 Program Clock Enable Block:

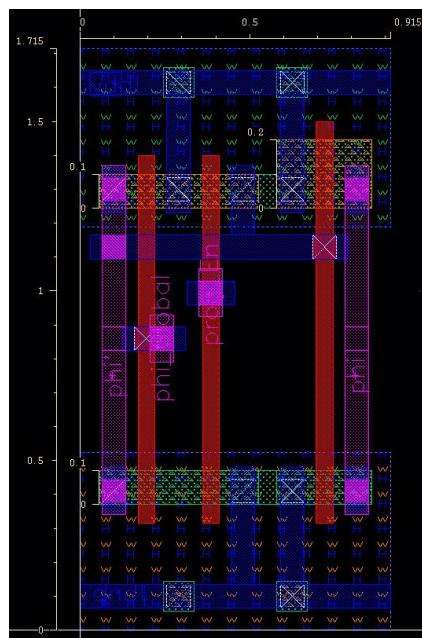
The Program Clock Enable Block generates a local phi and phi' signal from a global phi signal and allows the bitstream to pass serially through each Bitstream Flip Flop. When the Program Enable signal is HIGH, the Q output of each Bitstream Flip Flop connects to the D input of each sequential Bitstream Flip Flop because the generated phi and phi' signals turn on the transmission gate at the input D. When the Program Enable signal is LOW, the transmission gate at the input D of each Bitstream Flip Flop turns off, causing the current values inside each Bitstream Flip Flop to become latched. Every contiguous block of Bitstream Flip Flops will have two Program Clock Enable Blocks to locally generate phi1, phi1', phi2, and phi2' for the Bitstream Flip Flops.

All NAND gate transistors are 100nm in width. The PMOS' in the inverter are all 200nm in width. The NMOS' in the inverter are all 100nm in width. All transistors are VTH in this cell.



*Figure 62. Transistor schematic of the Program Clock Enable Block with sizing (left figure). Stick diagram of the Program Clock Enable Block (right figure). Estimated floorplan is 3um x 1um (modified design from Figure 2 in [2]).*

The cell dimensions of the Program Clock Enable layout is 1.715um x 0.915um. DRC and LVS passed on this layout.



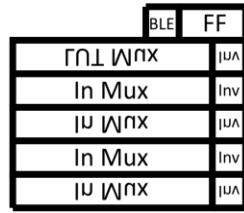
*Figure 63. Layout of the Program Clock Enable Block. Cell dimensions are 1.715um x 0.915um.*

## VII. CLB Full Layout

A full description of the CLB is given in the CLB Architecture Design section. This section details the layout of the entire CLB.

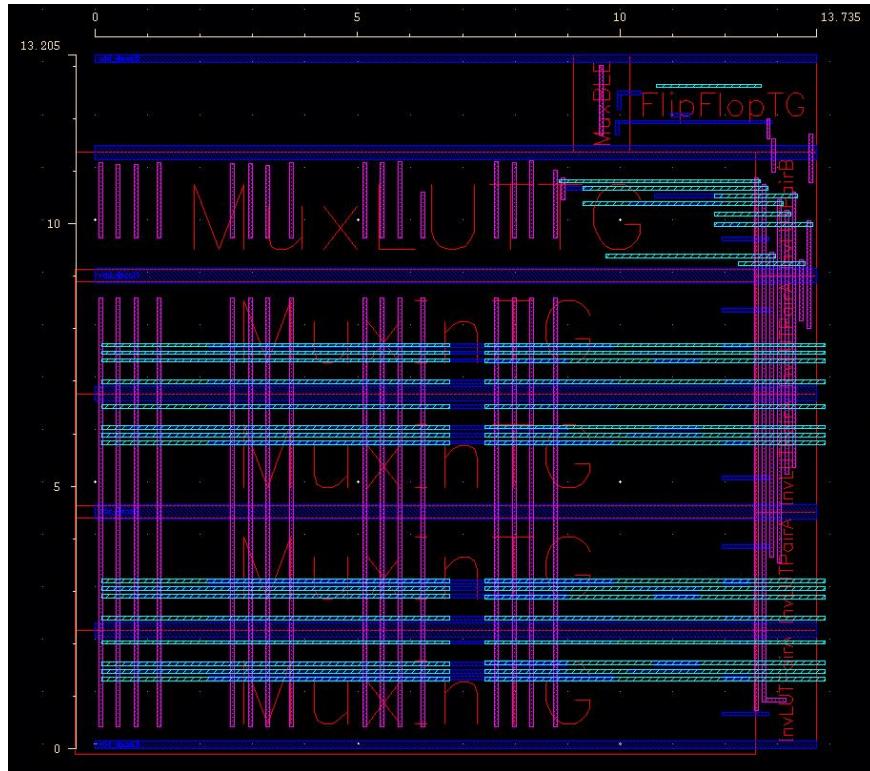
### VII.1 BLE:

The CLB can be split into multiple identical basic logic elements. In this specific CLB, there are 4 BLEs. A BLE implements the function given by a single LUT Mux and the output is either registered or unregistered by a flip flop that can be bypassed through the BLE Mux. The BLE contains 4 Input Muxes, 1 LUT Mux, 1 BLE Mux, 1 flip flop, and 5 buffers (inverter pairs). Since each BLE is identical within the CLB, only a single layout of the BLE was created. Figure 64 shows the floorplan of a single BLE.



*Figure 64. Floorplan of a single BLE cell in the CLB. The estimated footprint of the BLE is 14.96um x 13.735um.*

The layout of the BLE was performed in three hierarchy levels. Hierarchy level 1 contains the layouts of the 4 Input Muxes, LUT Mux, 4 buffers, BLE Mux, and CLB Flip Flop. Hierarchy level 0 contains all the routing needed to connect the cells shown in Figure 65. The cell dimensions of the BLE is 13.205um x 13.735um. DRC and LVS passed on this layout.



*Figure 65. Layout of the BLE block. Cell dimensions are 13.205um x 13.735um. Hierarchy level 0.*

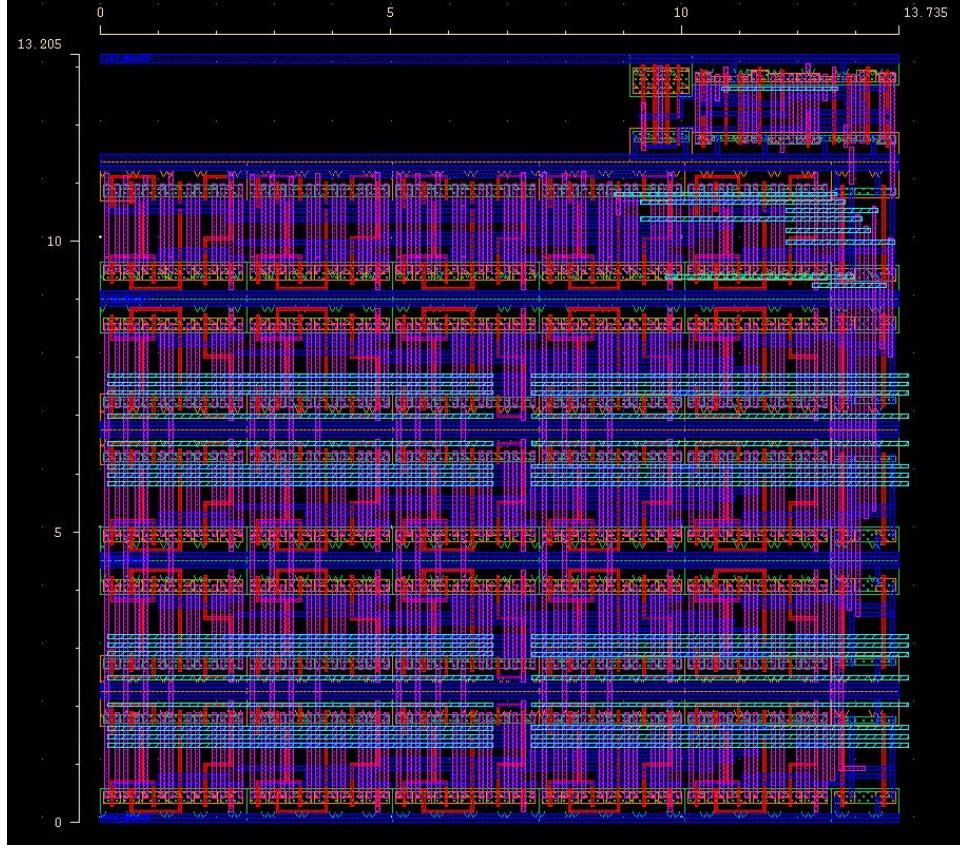


Figure 66. Layout of the BLE block. Cell dimensions are 13.1um x 13.735um. Hierarchy level 2.

## VII.2 CLB:

Since all 16 of the Input Muxes have the same inputs, I created my CLB floorplan around the need for connecting all 16 inputs together. The resulting floorplan, Figure 67, vertically tiles the Input Muxes together so minimal effort is needed to connect the all of the inputs together. The buffers (inverter pairs) are added at the output end of the Input Muxes and will connect upwards to the select lines of the LUT Mux. The output of the LUTs will feed into the BLEs and FFs located at the corners of the design.

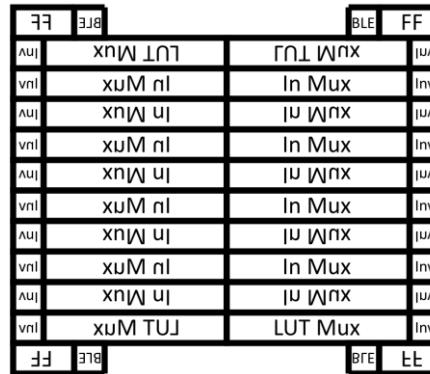
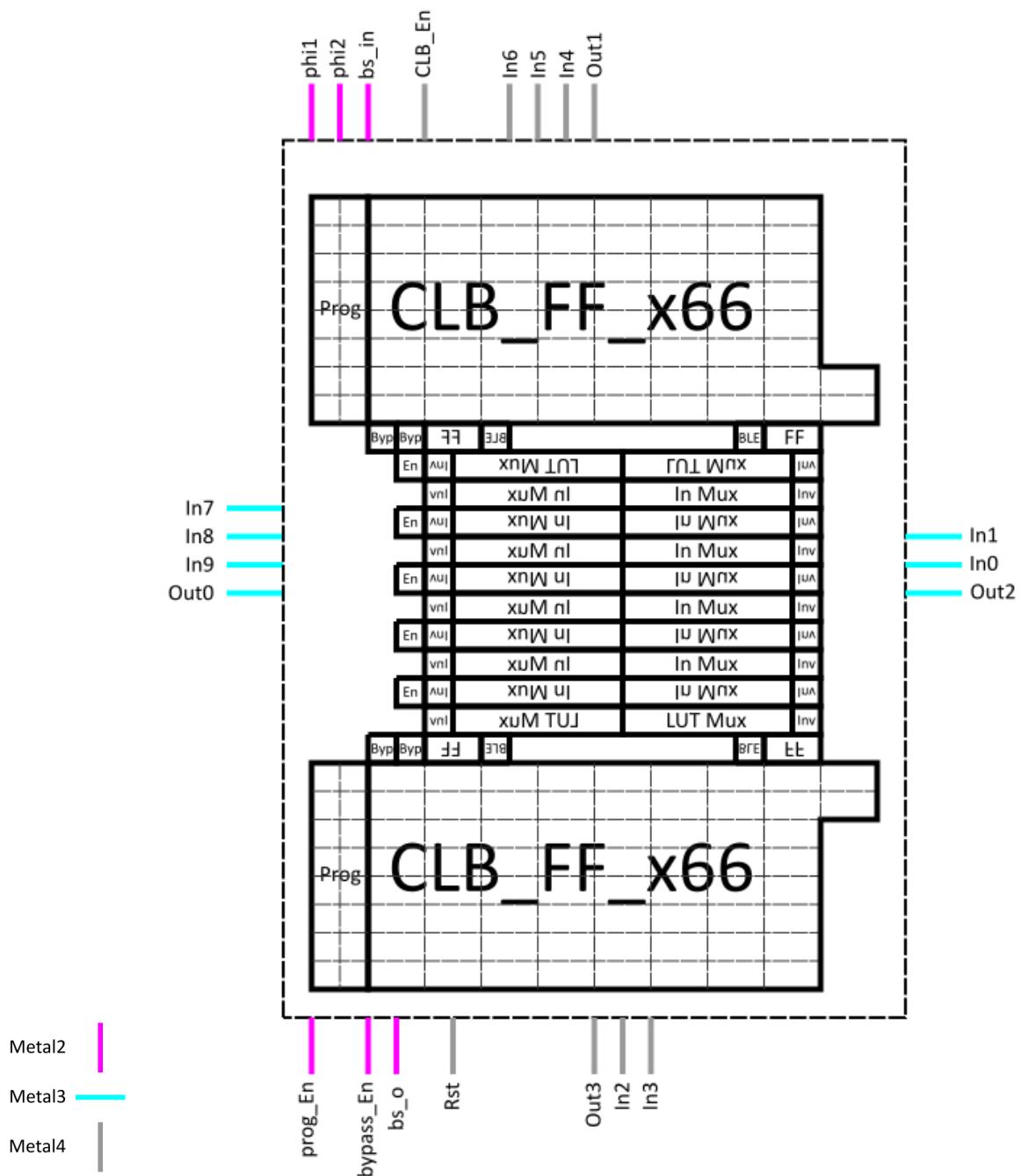


Figure 67. Floorplan of the entire CLB. The CLB contains 4 instances of the the BLE, flipped vertically or horizontally. The estimated footprint of the CLB is 29.92um x 27.47um.

Figure 68 shows the full floorplan of the CLB with the bitstream included. The input and output pins are shown with their relative locations on the block. The color of the wire corresponds to the metal layer. The relative sizes of each cell in the floorplan don't perfectly match the sizes of the cells in the actual layout.



*Figure 68. Floorplan of the entire CLB with 132 Bitstream Flip Flops and all pins included. Estimated floorplan of the CLB with Bitstream Flip Flops is 52.18um x 27.47um.*

At first, I created a layout of the CLB by itself without the Bitstream Flip Flops. The layout is shown in Figure 69. This was a mistake because having a full CLB layout in this hierarchy without any Bitstream Flip Flops meant that all 132 Bitstream Flip Flops had to be individually routed in the hierarchy above this one. That would have been a lot of wasted layout effort, so I decided to recreate the CLB in two identical halves with the Bitstream Flip Flops.

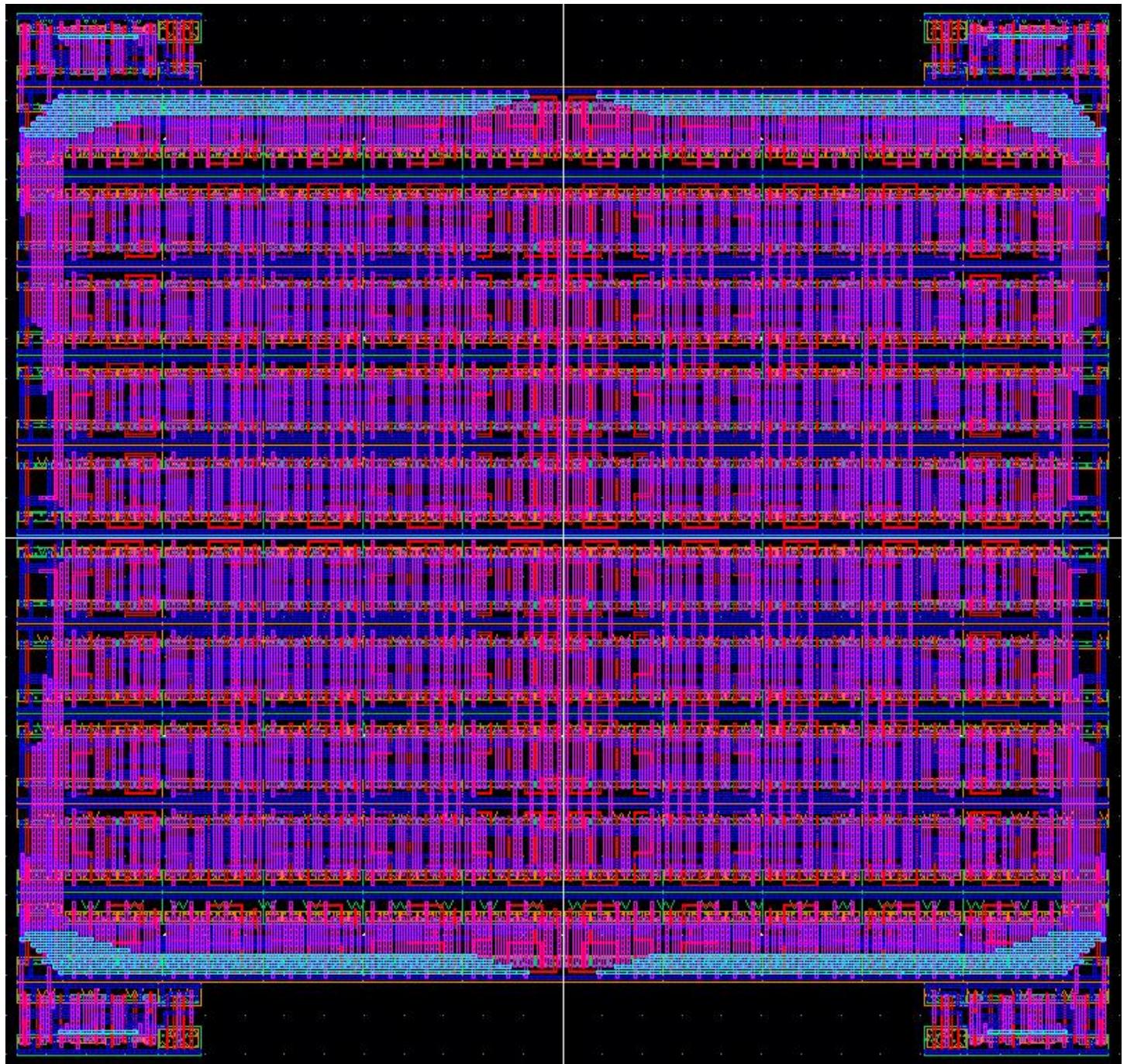
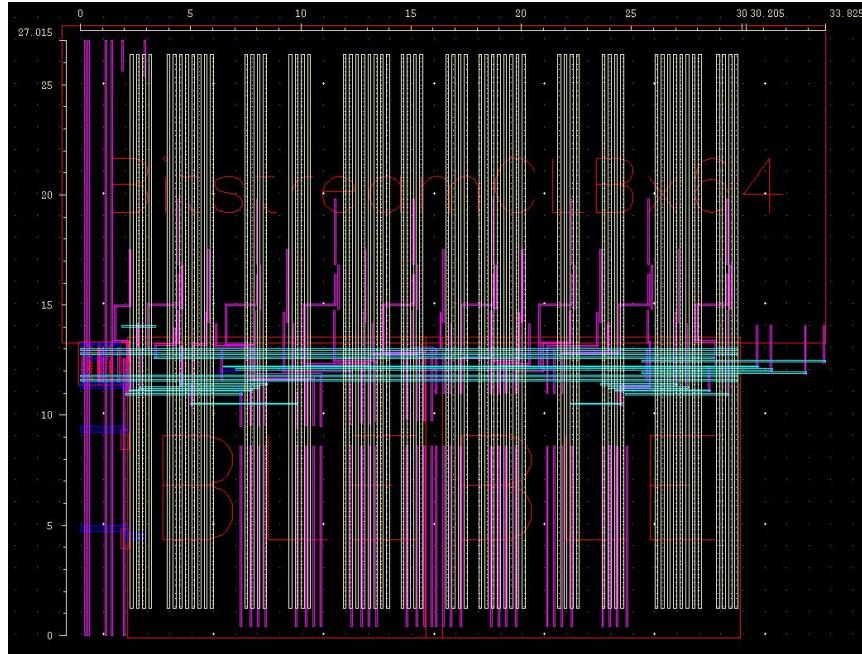


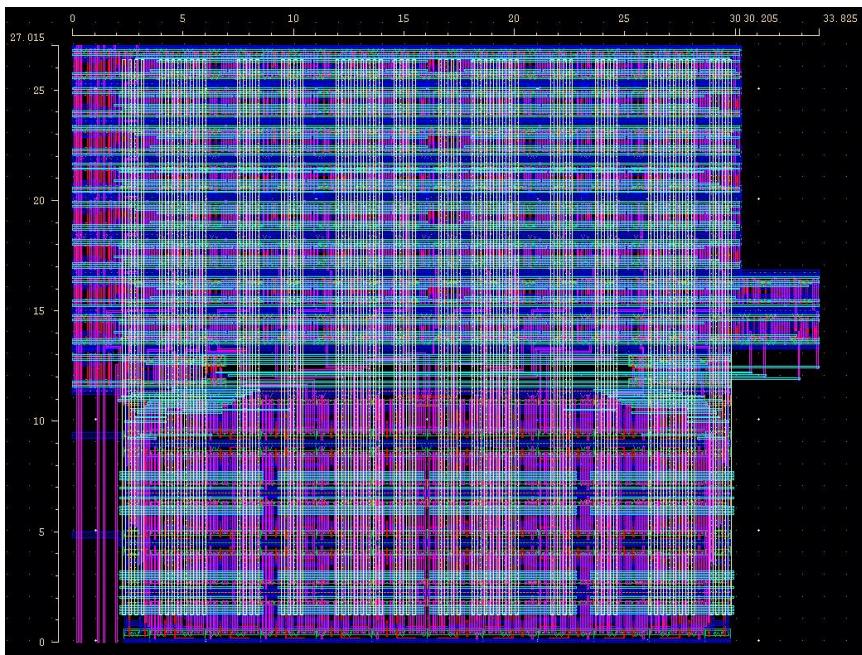
Figure 69. Unused layout of the CLB block. Cell dimensions are 26.2um x 27.47um. Hierarchy level 2.

By splitting the CLB up into a top half and a bottom half, I only had to spend time on manually routing half of the Bitstream Flip Flops (66 total). The half CLB design consists of 2 BLEs, 3 CLB Enable Blocks, and the 66 flip flop bitstream block. A lot of effort was spent on accurately routing the outputs of each Bitstream Flip Flop to the correct Input Mux select lines and the correct LUT Mux inputs. It was also crucial to create this half in a way that can be abutted seamlessly with the bottom half (a mirror version of this block).

The footprint of this block, and the full CLB layout, turned out to be a bit larger than expected. I didn't account for the 4 Bitstream Flip Flops that were needed for the BLE Muxes, which added 3.555um to the total width. I also didn't account for the width of the Program Clock Enable Blocks, which added 1.83um to the total width. The layout of the CLB half block is shown in Figure 70 and Figure 71. DRC and LVS passed on this layout.



*Figure 70. Layout of the CLB half block. Cell dimensions are 27.015um x 33.825um. Hierarchy level 0.*



*Figure 71. Layout of the CLB half block. Cell dimensions are 27.015um x 33.825um. Hierarchy level 3.*

The layout of the entire CLB was performed in four hierarchy levels. Hierarchy level 1 contains the layouts of the CLB half block. Hierarchy level 0 contains all the routing needed to connect the two halves together in Figure 72. The cell dimensions of the CLB is 54.03um x 30.27um. DRC and LVS passed on this layout. The images of the full CLB layout are too large to properly view because Word compresses them.

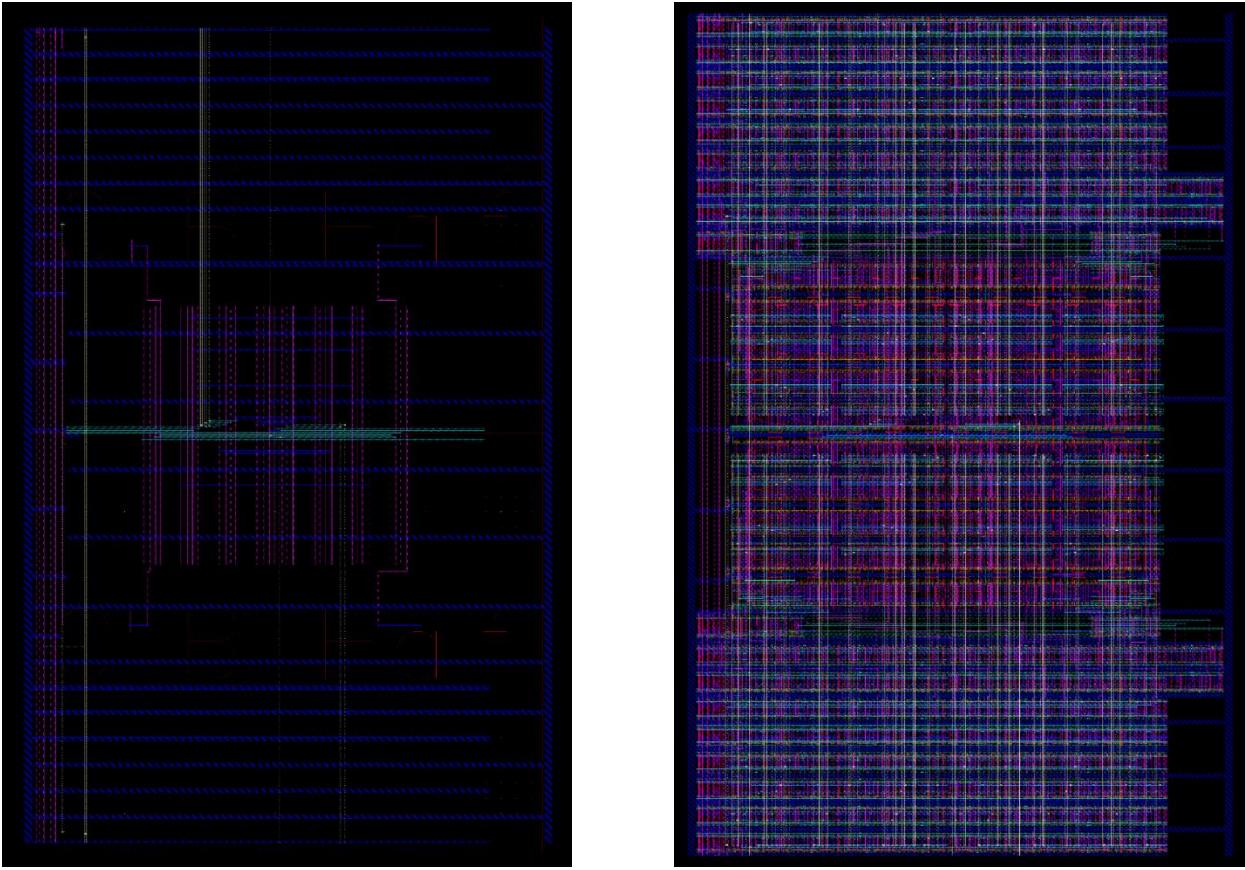


Figure 72. Layout of the full CLB block. Cell dimensions are 54.03um x 30.27um. Hierarchy level 0 (left figure). Hierarchy level 4 (right figure).

From the full CLB layout, the distances of each input and output pin was measured and shown in Figure 73. The color of the pins indicate which metal layer they are on.

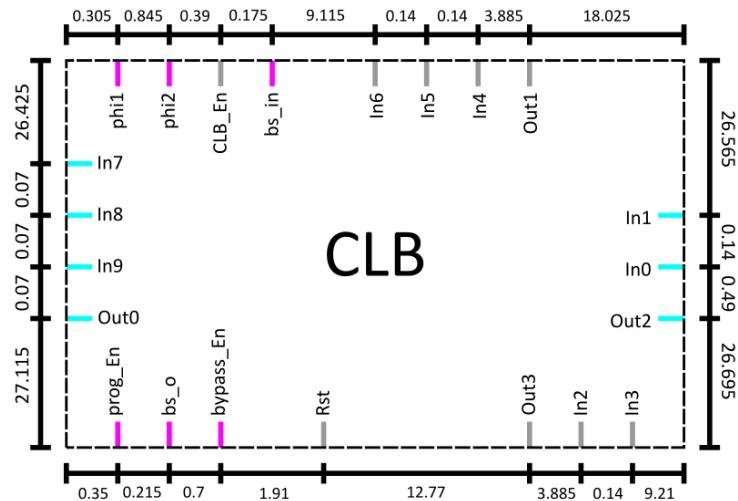


Figure 73. Full CLB block with all input and output pins. Distances of each pin measured in um.

### VII.3 Parasitic Extraction Errors

After the full layout of the CLB was completed, Parasitic Extraction (PEX) was run to extract a more extensive netlist of the design with all the parasitic capacitance and resistance contributions from the different metal traces, vias, etc. With this updated netlist, an accurate delay simulation of the critical path could be generated of the current layout.

After running the following commands for PEX, a schematic of all extracted capacitances and resistances should appear in a separate window, however a fatal Error message is encountered everytime the program tries to show the schematic. In the log file it shows that it tries to open a shared library that doesn't exist on that path anymore.

Calibre -> Run PEX

NCSU techfile xRC ruleset used for PEX.

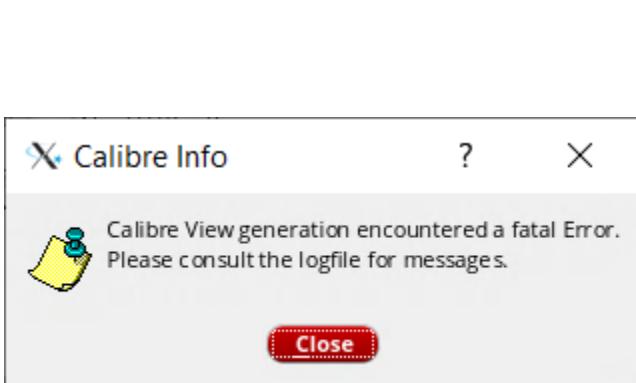
Inputs:

Layout: Export from layout viewer. Using CLB.calibre.db file.

Netlist: Export from schematic viewer. Using CLB.src.net file.

Outputs:

CALIBREVIEW format. Output file is CLB.pex.netlist.



```
[INFO: cds.lib has been converted to lib.defs /var/tmp/161167_result.lib.defs
RUNNING PEX back-annotation_2017.3_29.23 Fri Sep 1 13:35:52 PDT 2017
// Calibre FDI v2017.3_29.23 Fri Sep 1 13:35:52 PDT 2017
//
// Copyright Mentor Graphics Corporation 2008-2017
// All Rights Reserved.
// THIS WORK CONTAINS TRADE SECRET AND PROPRIETARY INFORMATION
// WHICH IS THE PROPERTY OF MENTOR GRAPHICS CORPORATION
// OR ITS LICENSORS AND IS SUBJECT TO LICENSE TERMS.
//
// Mentor Graphics software executing under x86-64 Linux
// 64 bit virtual addressing enabled
//
// This software is in pre-production form and is considered to be
// beta code that is subject to the terms of the current Mentor
// Graphics End-User License Agreement or your signed agreement
// with Mentor Graphics that contains beta terms, whichever applies.
//
// Running on Linux ece-linlabsrv01.ece.gatech.edu 3.10.0-1127.10.1.el7.x86_64 #1 SMP Tue May 26 15:0
5:43 EDT 2020 x86_64
INFO: Parsing command line arguments...
INFO: License checked successfully for calibreqdb.
INFO: Parsing schema files...
INFO: Executing back annotation...
ERROR: Could not access LibDef plug-in for calibDefSystem: #4: Shared Library Not Found: Error loading
library 'libddbase_sh.so', '/tools/cadence/ic617new/tools/lib/64bit/libddbase_sh.so': undefined symbol:
_ZN80aCommon11FactoryBase11getRefCountEv
```

Figure 74. Fatal Error message when trying to open the extracted schematic (left figure). Error shared library message in the log file (right figure).

At the end of PEX, a netlist was successfully generated, however I couldn't figure out how to convert the netlist into a format that HSPICE could read. There were also 7 "PEX RESISTANCE LUMPED is obsolete" warnings that came up which seemed to cause none of the resistances to be extracted.

## VIII. Logical Effort Comparison

Preliminary logical effort calculations were performed on the individual cells within each block to help determine sizing, number of stages, and design of each cell. Some of these designs were simulated in HSpice to compare delay time and power consumption between different implementations.

Table 1. Comparison of worst case HSpice simulated delays for each individual cell vs logical effort delay calculations for each individual cell. Both are given in terms of picoseconds.

|                     | MuxIn  | InvLUT | MuxLUT | Mux BLE | Demux  | MuxC   | InvC   | MuxS  | TriS   | Dlatch | <b>Worst Case</b> |
|---------------------|--------|--------|--------|---------|--------|--------|--------|-------|--------|--------|-------------------|
| Rise time (ps)      | 227.9  | 60.93  | 82.31  | 80.89   | 200.94 | 184.46 | 169.88 | 66.85 | 175.71 | 104.23 | <b>1020.81</b>    |
| Fall Time (ps)      | 154.57 | 70.87  | 24.23  | 57.69   | 110.48 | 147.9  | 173.44 | 65.44 | 98.04  | 86.22  |                   |
| Logical Effort (ps) | 413.84 | 46.85  | 364.65 | 85.89   | 84.33  | 499.73 | 112.44 | 31.23 | 124.93 | 187.4  | <b>1951.30</b>    |

The HSpice simulated worst case delay for the sample critical path turned out to be 930.49 picoseconds less than the delay calculated by logical effort, 1020.81 picoseconds versus 1951.30 picoseconds. Tau was measured using an inverter chain with VTL transistors and came out to be around 15.6 picoseconds. This difference is mostly due to my complete redesign of the Input Mux, the LUT Mux, and the C Block Mux into a transmission gate design. Furthermore, the parasitics and logical effort of the transmission gates couldn't be very accurately calculated so the logical effort calculations are off in that regard as well.

In the beginning, I noticed my output signal at the end of the sample critical path had a very degraded profile. The signal wasn't restored enough after passing through the C Block inverter. Since the C Block inverter connects to 16 instances of the Input Mux in the CLB, it needed a lot more driving power. To fix this issue, the size of the NMOS and PMOS in the C Block inverter were significantly increased to better restore the signal and to make it easier to drive its output load.

Table 2. Summary of the delay times and power consumption of the CLB, the C Block, and the S Block.

|                           | CLB     | C Block | S Block |
|---------------------------|---------|---------|---------|
| Rise time (ps)            | 547.43  | 428.26  | 365.83  |
| Fall Time (ps)            | 371.87  | 558.84  | 351.63  |
| Leakage Power Pass 0 (uW) | 36.324  | 0.11264 | 152.06  |
| Leakage Power Pass 1 (uW) | 0.12057 | 0.03358 | 152.06  |
| Dynamic Power (uW)        | 107.31  | 117.34  | 156.68  |

The leakage power simulated in the S Block is for the case when every single mux and tristate buffer is turned on in the S Block. The dynamic power simulated is also for the case when every single mux and tristate buffer in the S Block is switching. This is highly unlikely as some of the tracks will be switched to inputs as opposed to outputs, turning those tristate buffers off.

An HSpice simulation was performed to estimate the worst case propagation delay through the critical path shown in Figure 75 and Figure 76. The path of the delay travels from the input of the Input Mux of one CLB to the input of the Input Mux of another CLB. This critical path schematic is shown in Figure 75. The input square wave is sent through two inverters before reaching the Input Mux to more accurately simulate an input signal. The rise time was 975.69 picoseconds. The fall time was 930.71 picoseconds. The delay simulation is shown in Figure 77.

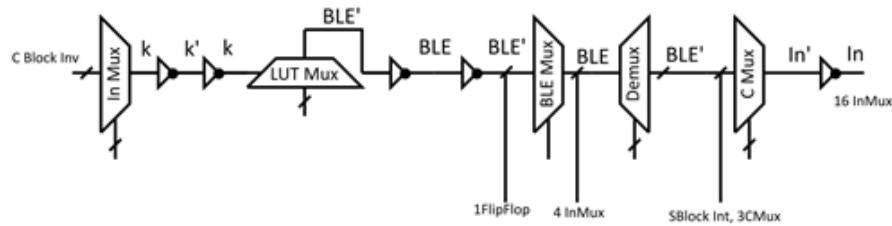


Figure 75. Sample critical path from the Input Mux of one CLB to the Input Mux of the next CLB.

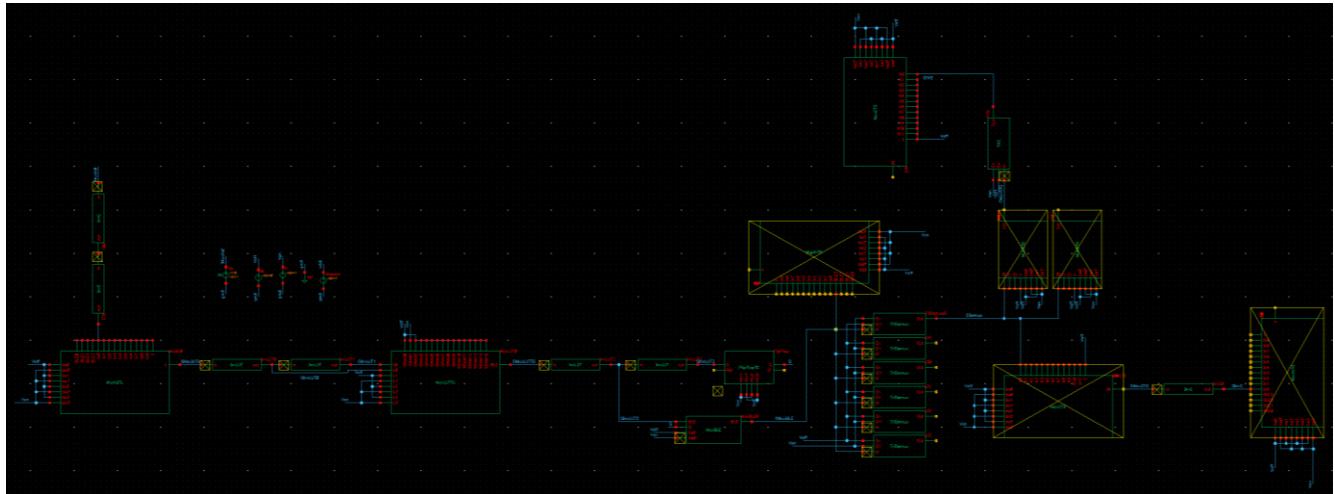


Figure 76. HSpice schematic for delay of entire sample critical path from the Input Mux of one CLB to the Input Mux of the next CLB.

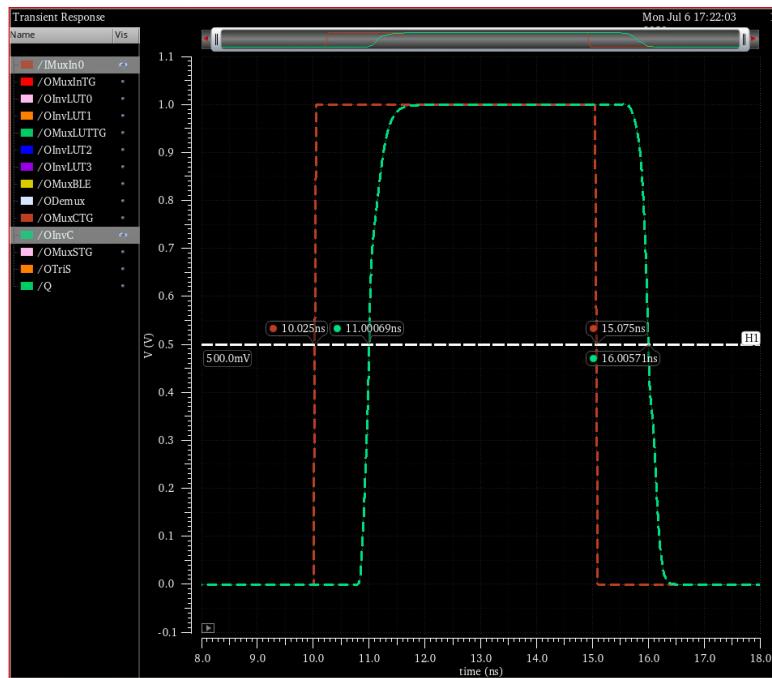


Figure 77. HSpice delay simulation of sample critical path from the Input Mux of one CLB to the Input Mux of the next CLB. Rise time of 975.69 picoseconds. Fall time of 930.71 picoseconds.

## IX. Future Work

Although a significant amount of time and effort has been spent on this project, there is still a lot of work to be done for this FPGA design. Due to the time constraints and the remote usage of Cadence tools, the C Block and S Block layouts were not complete. The input and output pins of these two blocks need to be designed to align with the input and output pins of the CLB, as shown in Figure 54. The Bitstream Flip Flops in the C Block and S Block can also be spatially optimized to fill in any white space that may be created when tiling the CLB, C Block and S Block. One such example is in the CLB layout. The 4 protruding Bitstream Flip Flops create an entire column of empty white space. This white space is highlighted in red in Figure 78. Some of the Bitstream Flip Flops needed for the C Block or the S Block can be placed in this white space to maximize area utilization.

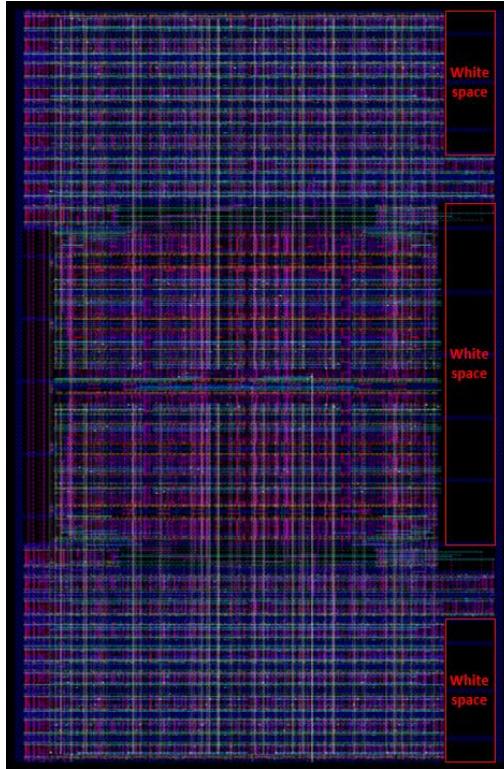
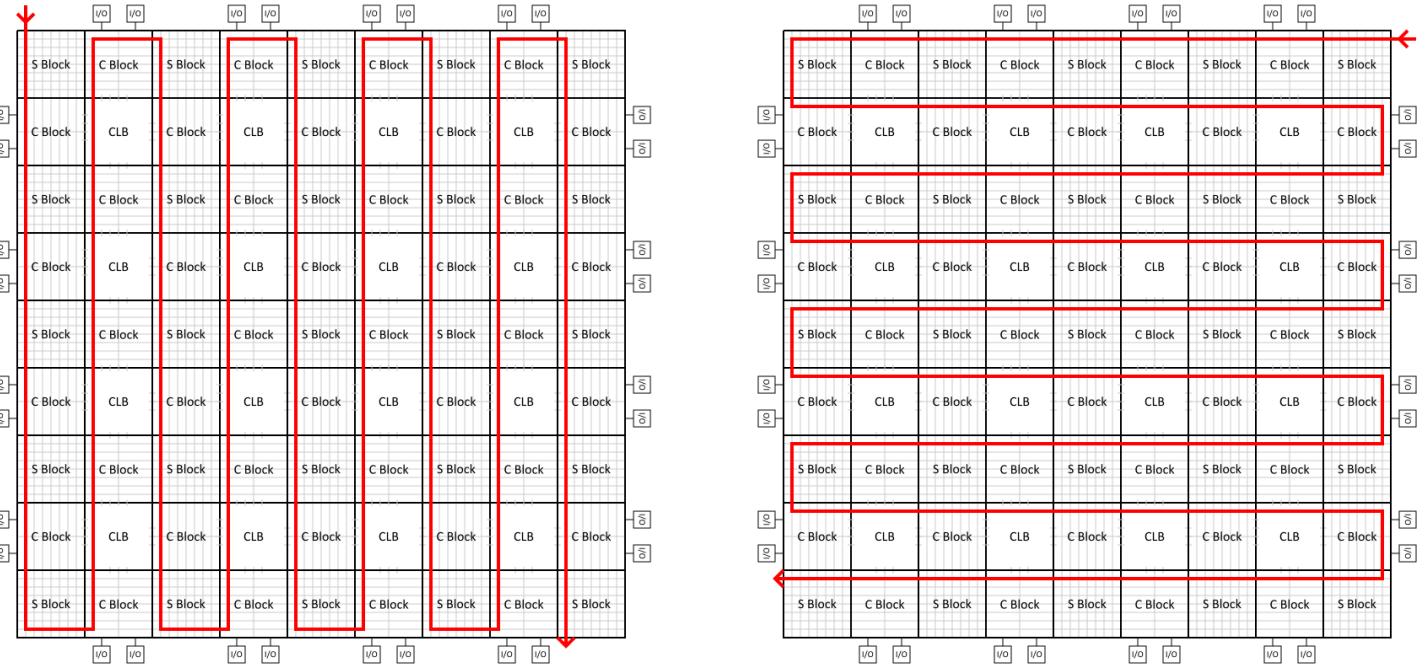


Figure 78. CLB layout with empty white space highlighted in red.

The widths of the transistors used in the Bitstream Flip Flops can be reduced. While speed isn't a large concern during the programming stage, power consumption and area still are. This means that all the PMOS and NMOS transistors in the Bitstream Flip Flop could be made minimally sized, 90nm wide as allowed by the technology file. This would slightly decrease the vertical dimension by 150nm from the current flip flop layout, which actually adds up to 2.4um saved off of the vertical dimension of the CLB layout.

Two potential paths for the overall FPGA bitstream path will need to be evaluated. With the current bitstream block design implemented in the CLB, the bitstream input and bitstream output pins are located vertically on the West side of the block. In this case, the bitstream input and output pins in the C Block and the S Block can also be made vertical on the West side to align with the CLB bitstream pins. This allows for a “vertical snake” of the bitstream path through the FPGA, shown in the left figure of Figure 79. Alternatively a “horizontal snake” could be implemented by removing the current vertical routing of the bitstream pins and replacing them with horizontal routing, shown in the right figure of Figure 79. This may save total routing length on the bitstream path.



*Figure 79. Vertical snaking of the bitstream path through the FPGA (left figure). Horizontal snaking of the bitstream path through the FPGA (right figure).*

Although it was out of the scope of this project, clock generation and analysis of clock skews are both necessary for the FPGA design. The clock skew between different rows of Bitstream Flip Flops is very important in ensuring the values of the bitstream are being latched sequentially as they are passed through. Even if a single bit didn't become latched due to errors in clock skew, the entire bitstream would be incorrect. This would cause the entire FPGA to behave differently from the intended behavior.

Further simulations on power consumption need to be performed. In particular, the S Block design used in the floorplan differs from the S Block design used for the HSpice power simulation in Figure 48. Because S Block design used in the floorplan only consists of 6 transmission gates, the block should consume much less dynamic and leakage power than what was simulated.

Finally, the integrity of the signal needs to be analyzed through a variety of paths in the FPGA. Since this super high-speed FPGA design revolves around using transmission gates in most of the muxes and complex gates, the profile of a signal will degrade as it passes through every transmission gate. This can lead to logic errors and huge increases in leakage power consumption if the signal can't properly turn on or off the transistors. Buffers need to be inserted anywhere the signal is significantly degraded.

## X. Conclusion

The Super High-Speed Transmission Gate Based FPGA Architecture final report gave an overview of the architecture of a scalable, fine-grained Field Programmable Gate Array (FPGA). The floorplans of each of the three blocks (CLB, C Block, and S Block) were given and the designs explained in detail, with emphasis placed on the CLB. The different complex cell designs used in each block were separately explained, and designs were shown in transistor schematics and stick diagrams. For the complex cell designs in the CLB, passing layouts were also shown. Afterwards, the clock distribution, power distribution, and bitstream design for the overall FPGA was discussed at length, and how each block fits within these aspects of the FPGA design. Finally, some recommendations for major improvements and future work was provided.

This final report and a full functional layout of the complete CLB were the final deliverables of the final project for ECE 6130 Advanced VLSI Systems taught by Associate Professor Vincent Mooney during the remote Summer semester of 2020 at Georgia Institute of Technology.

## XI. References

- [1] V. Mooney, "ECE 4130/6130 FPGA Project," Georgia Tech, 25 July 2020.
- [2] K. Bates and V. Mooney, "Bitstream Floorplan Guidelines," Georgia Tech, 25 July 2020.