

# Simplified DES CMOS Chip

Tommy Keating and Will Hamic

Spring 2021

## 1 Block Diagram

### 1.1 Chip Input and Output

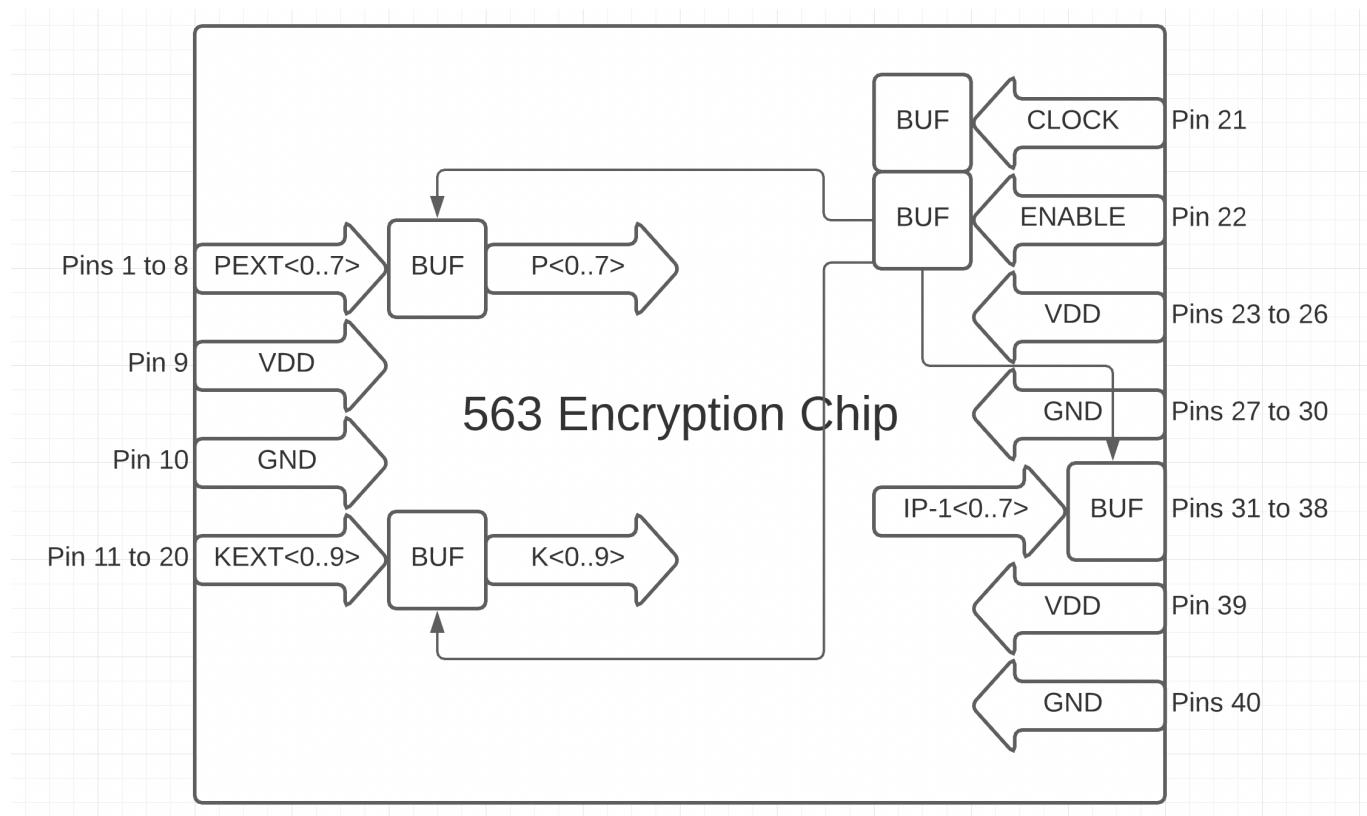


Figure 1: Chip Input and Output

## 1.2 Main Logic

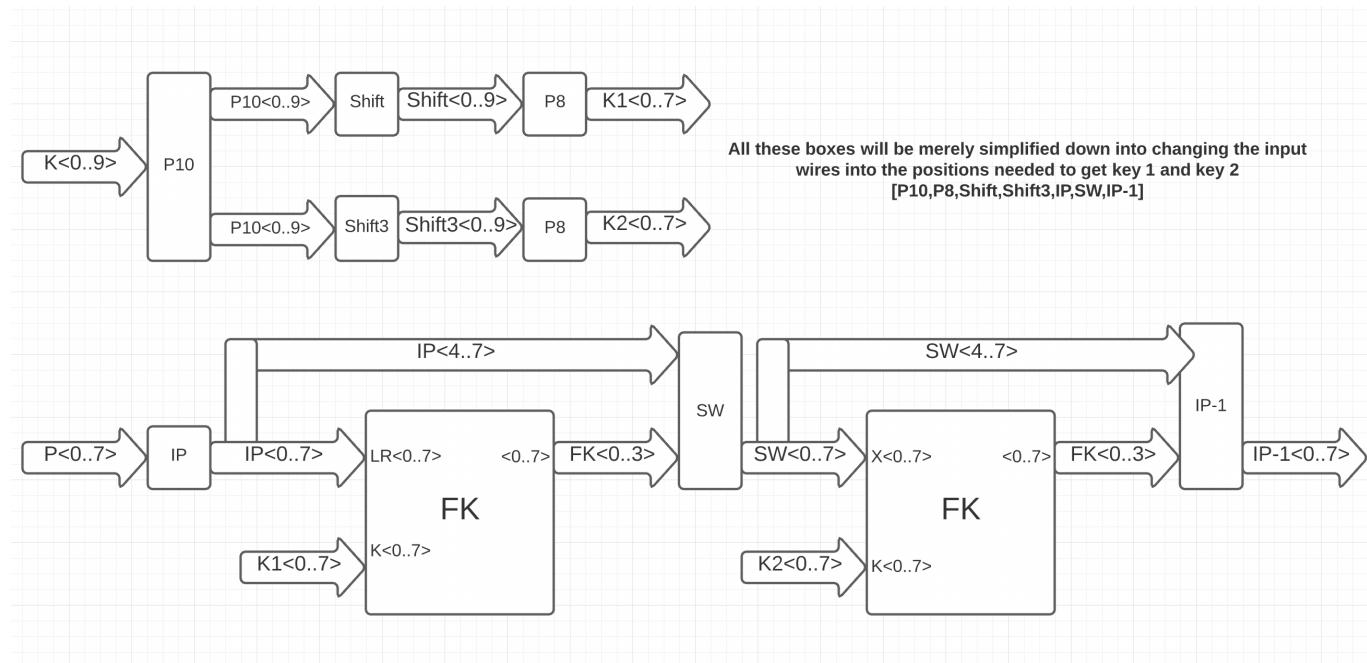


Figure 2: Main Level Block Diagram

### 1.3 FK Block

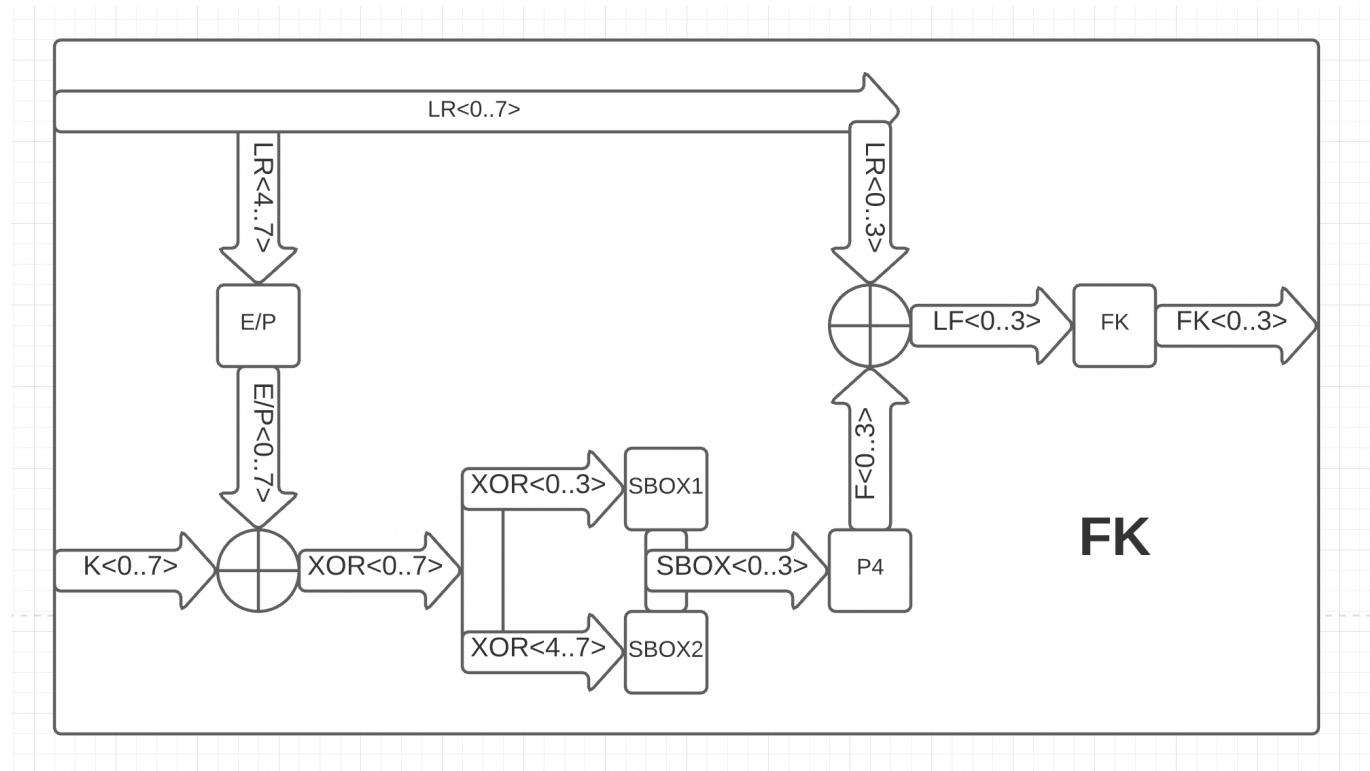


Figure 3: FK Block Diagram

## 1.4 Sboxes

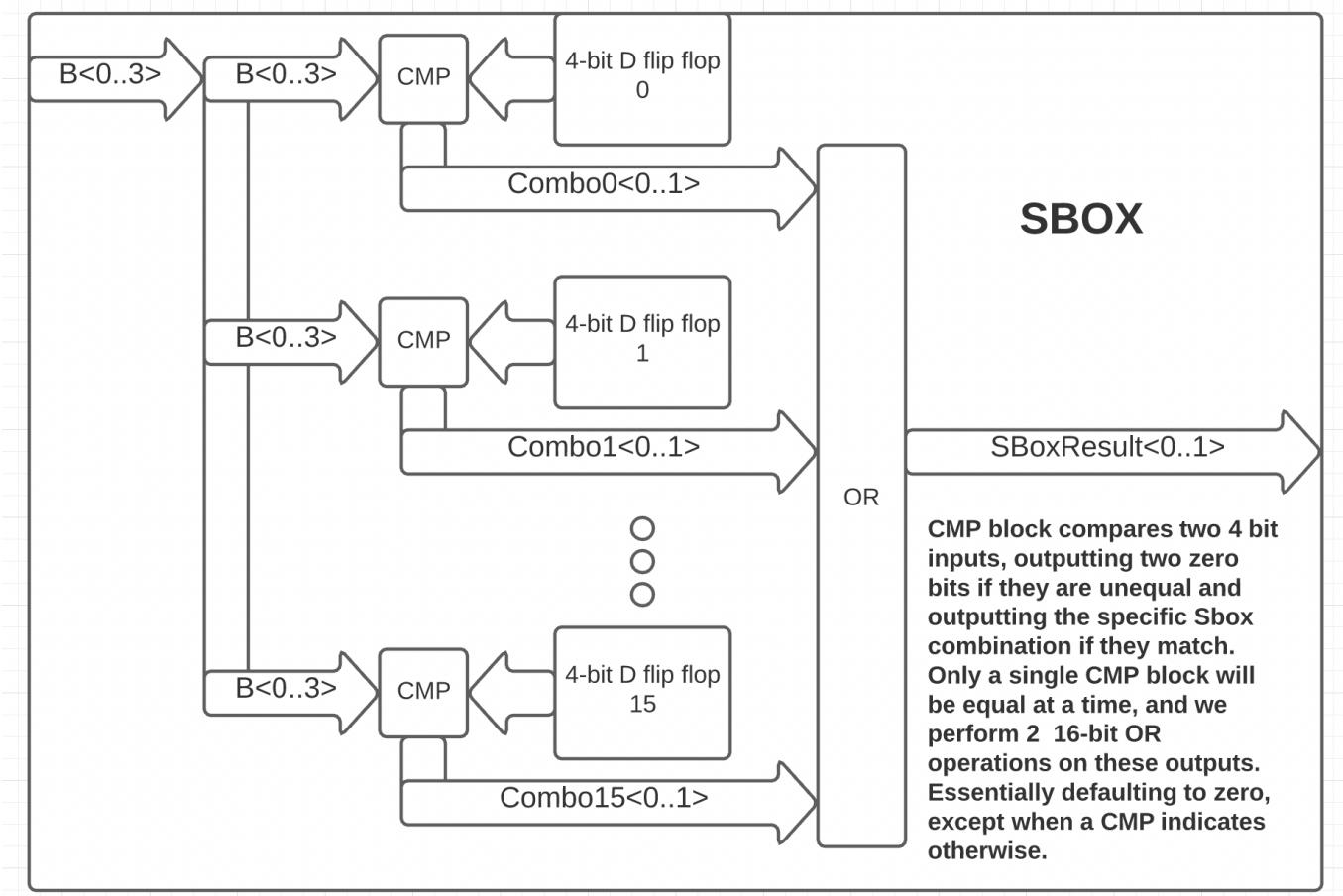


Figure 4: General Sbox Diagram

## 2 Chip Schematic

### 2.1 DES Chip

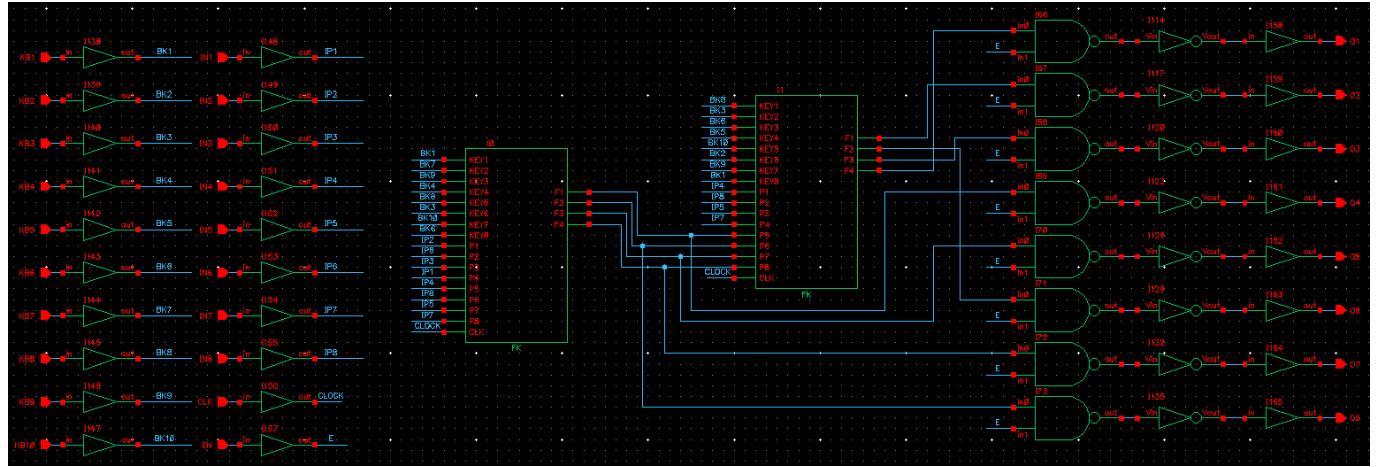


Figure 5: Full Chip Schematic

### 2.2 FK

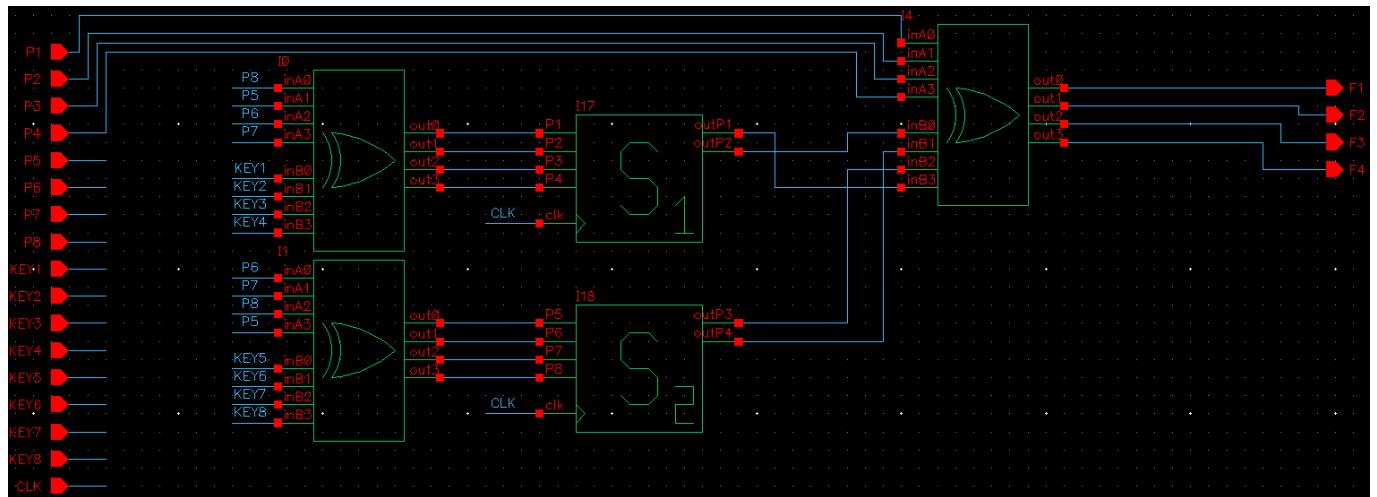


Figure 6: FK Block Schematic

## 2.3 Sboxes

### 2.3.1 SBox #1

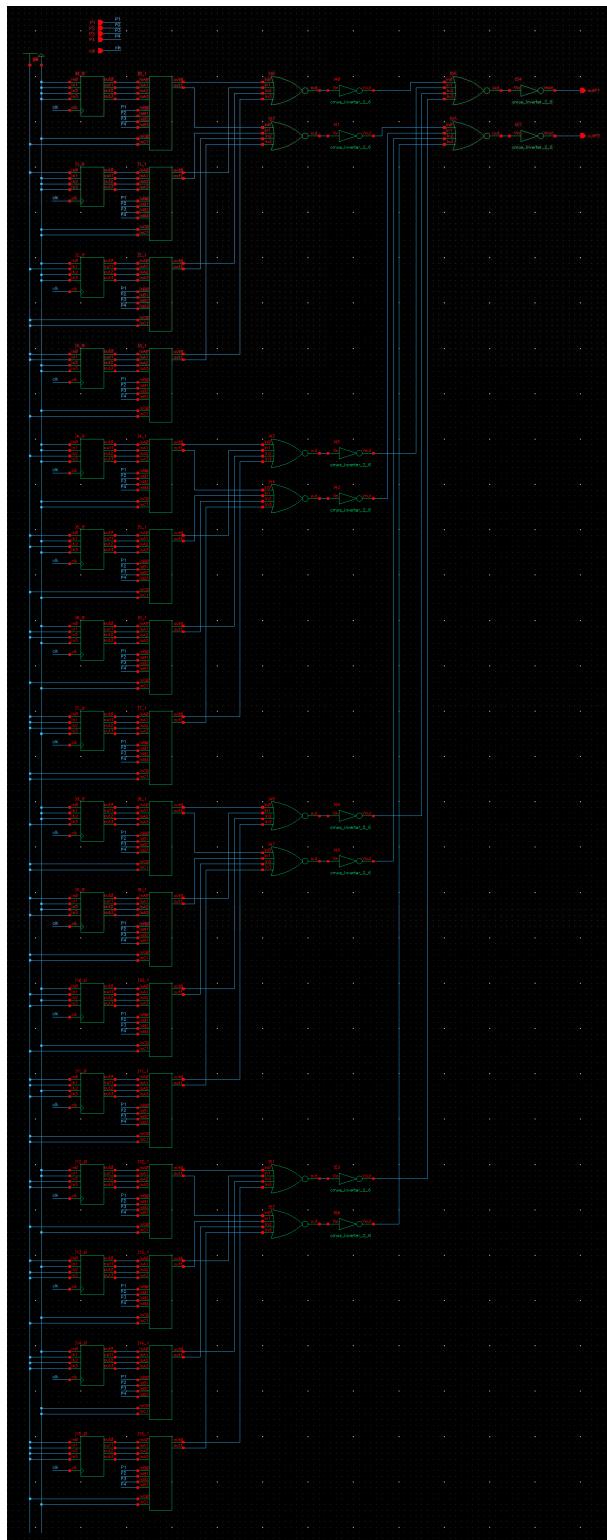


Figure 7: Sbox #1 Schematic

### 2.3.2 SBox #2

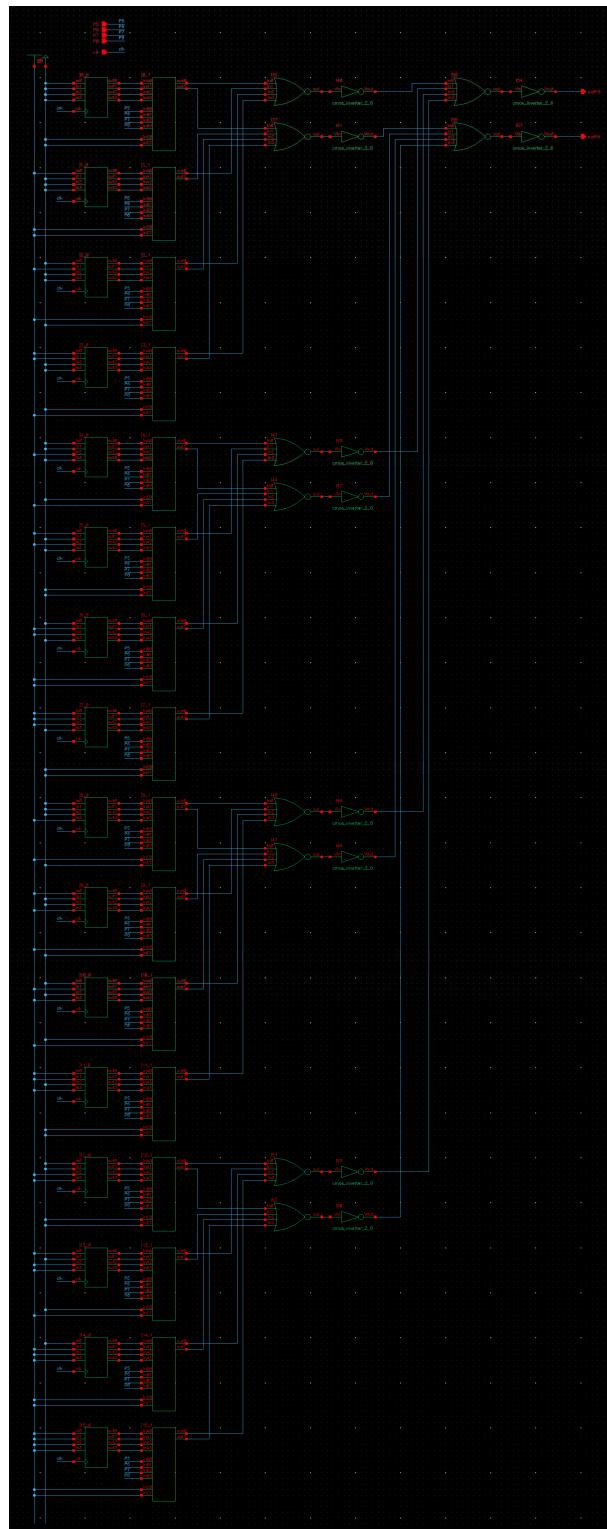


Figure 8: Sbox #2 Schematic

## 2.4 Compare

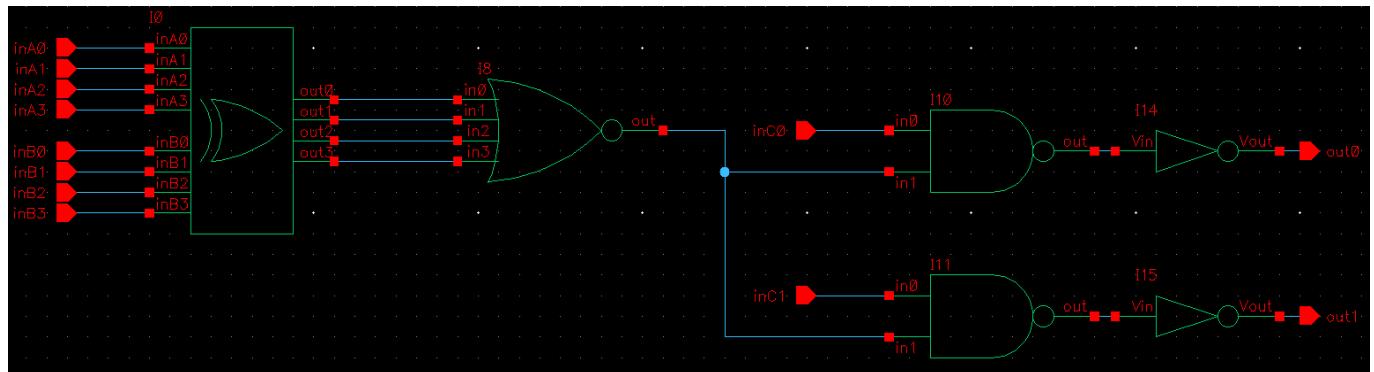


Figure 9: CMP Schematic

## 2.5 D-flip-flop

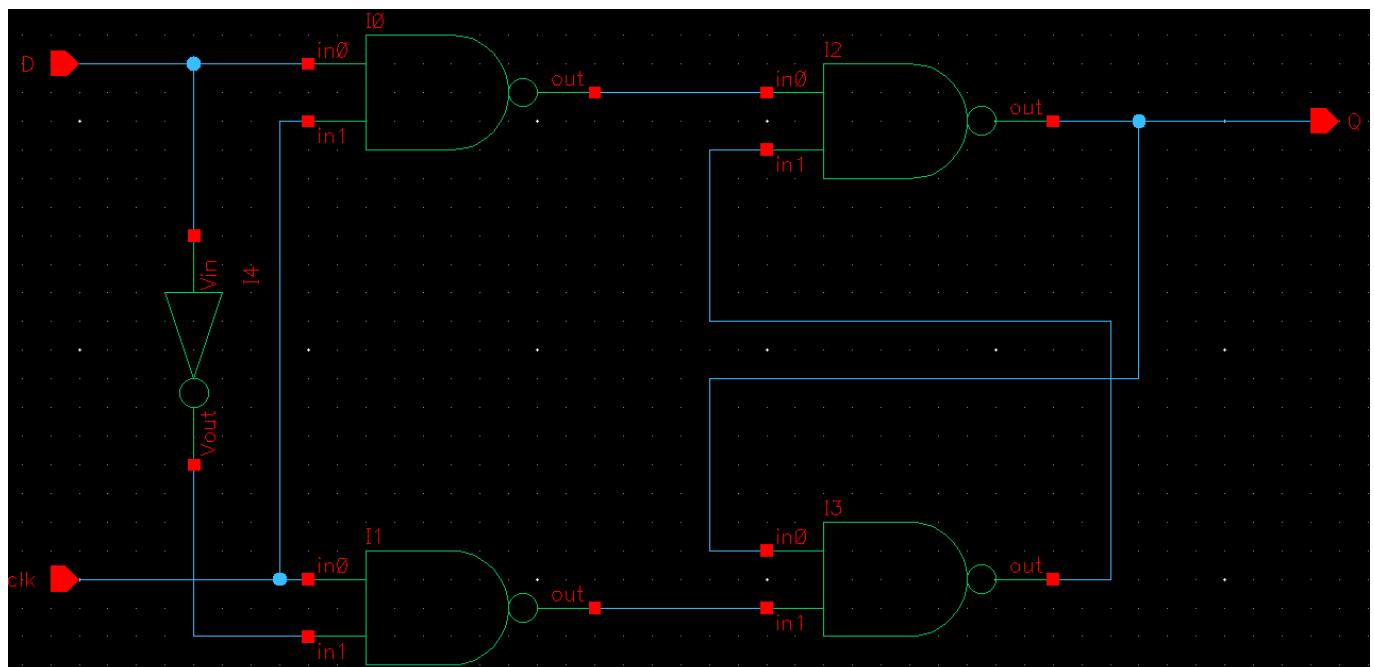


Figure 10: D Flip-Flop Schematic

## 2.6 4 D-flip-flop

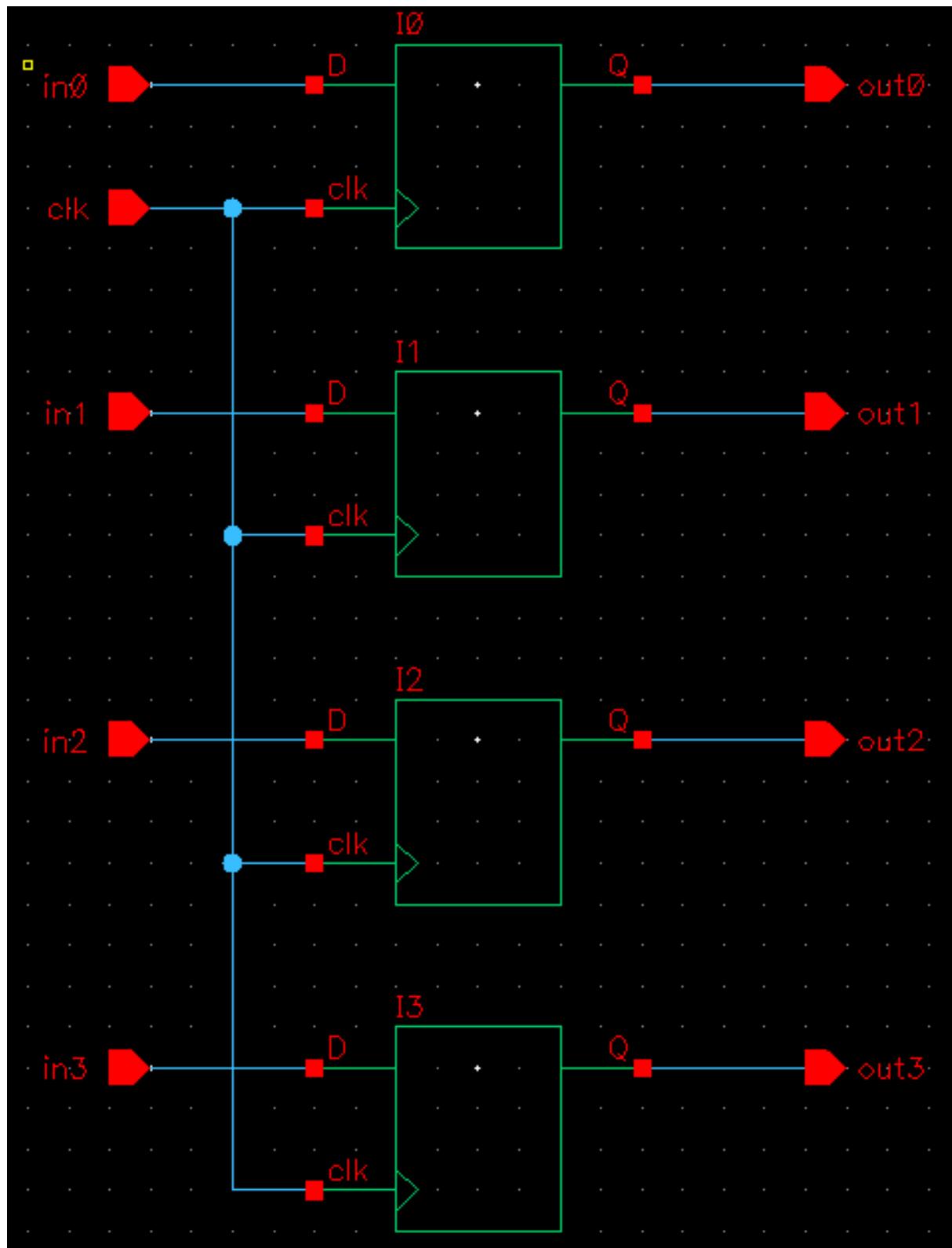


Figure 11: 4 D Flip-Flop Schematic

## 2.7 XOR 4

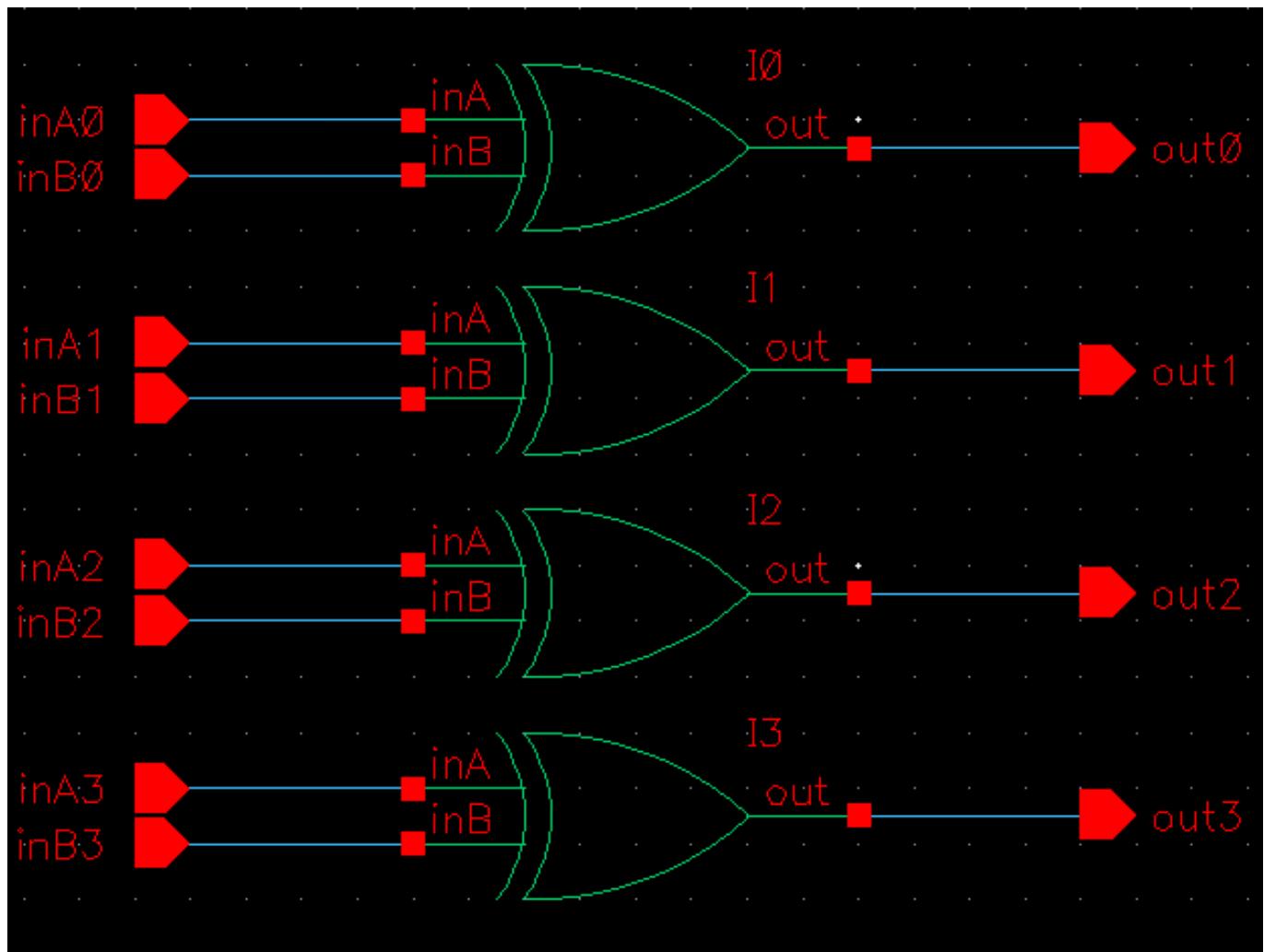


Figure 12: XOR 4 Schematic

## 2.8 XOR

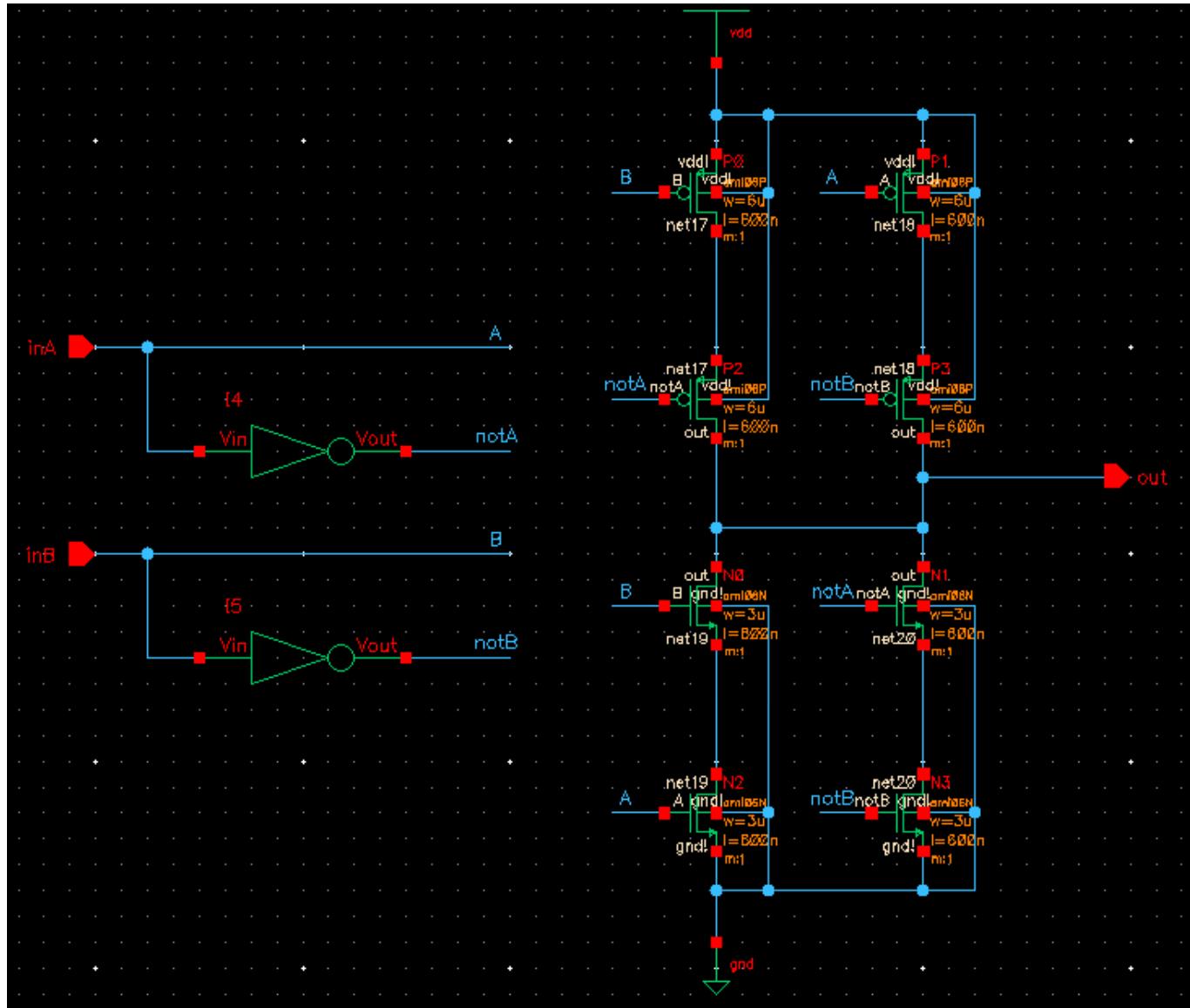


Figure 13: XOR Schematic

## 2.9 NAND

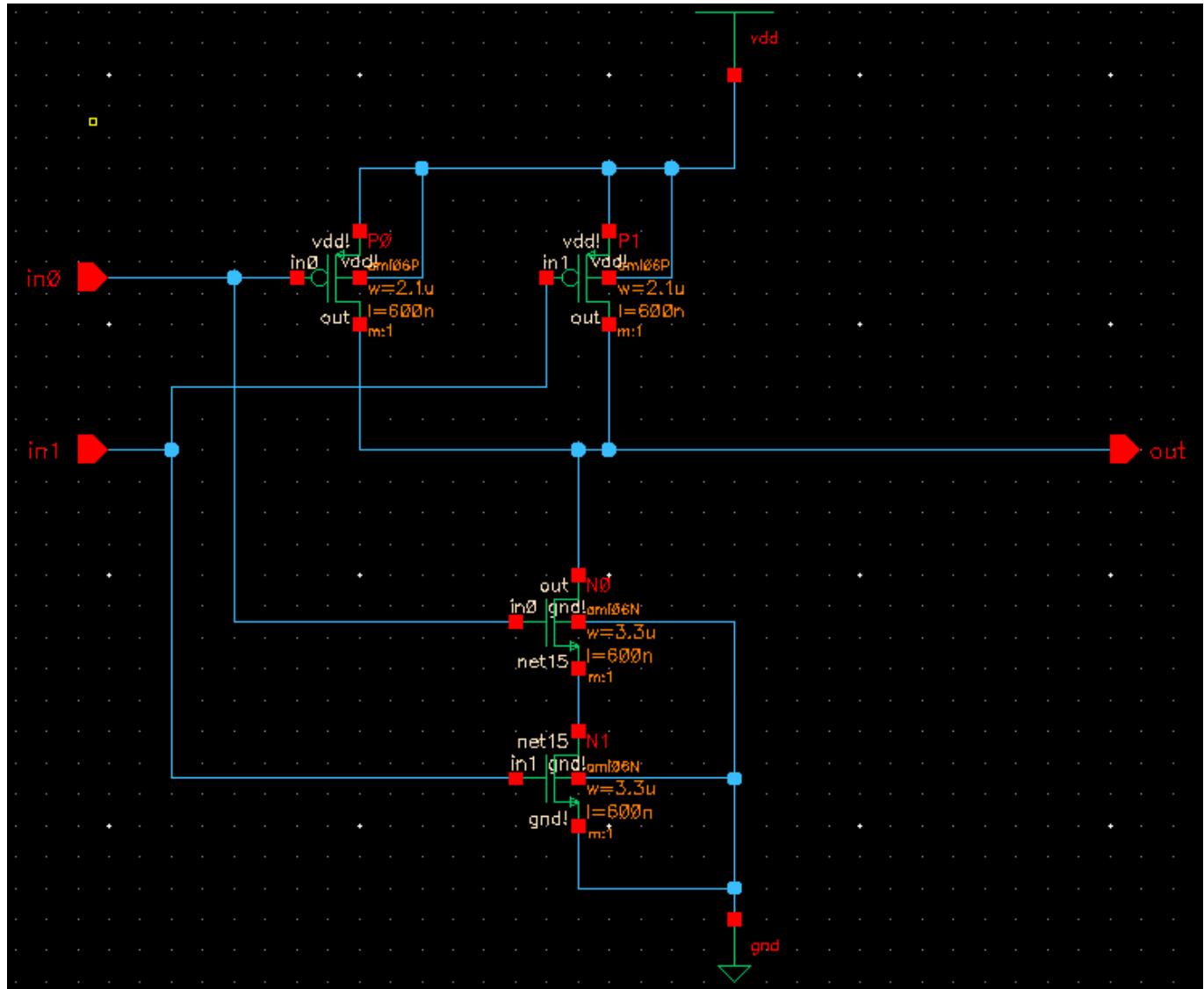


Figure 14: NAND Schematic

## 2.10 NOR4

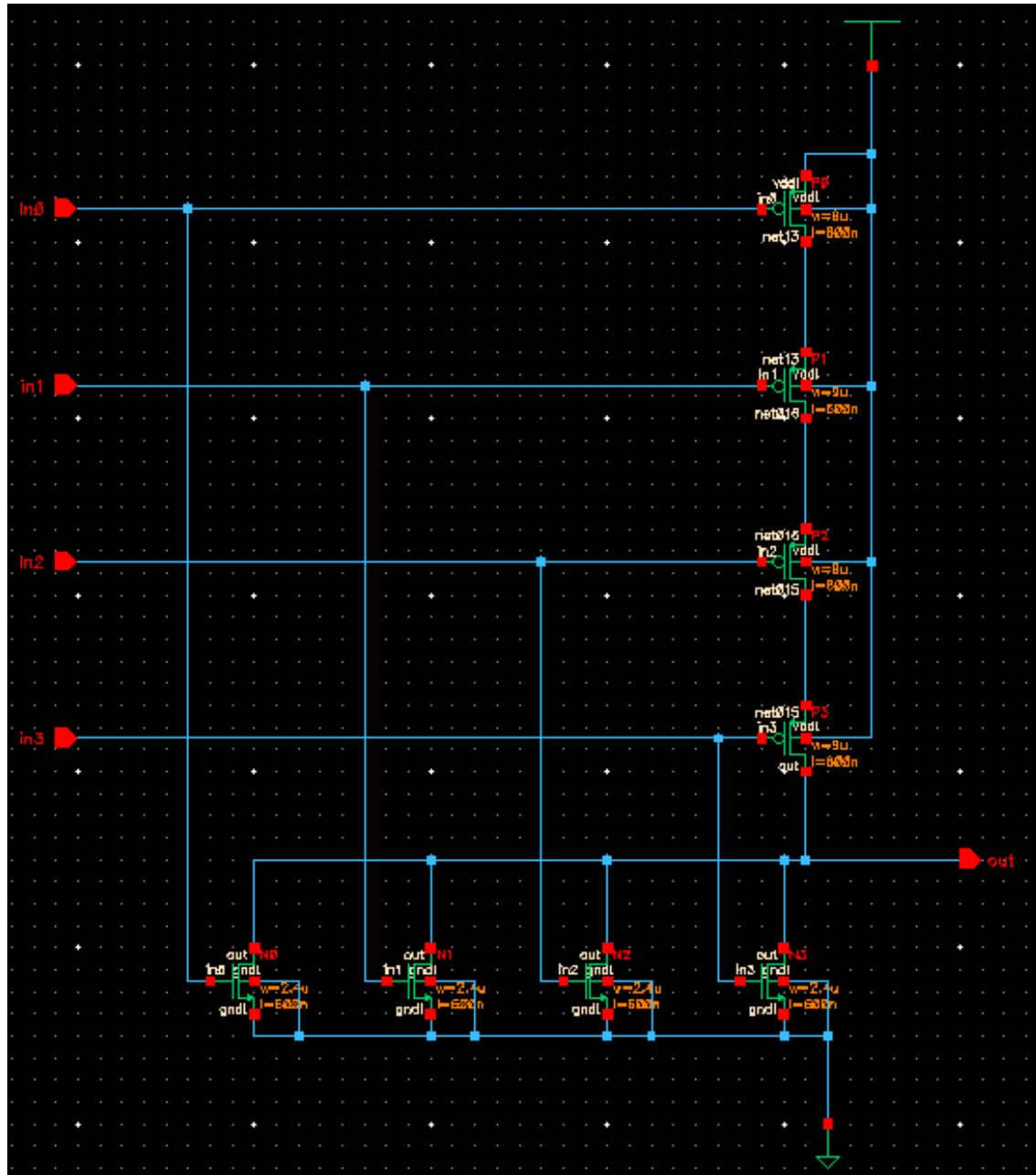


Figure 15: NOR 4 Schematic

## 2.11 Buffer

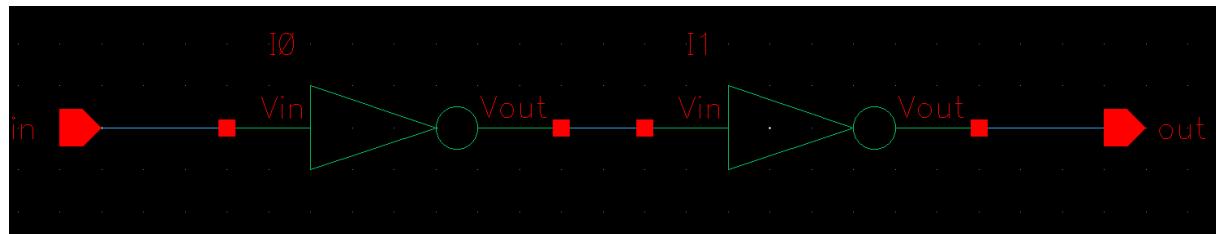


Figure 16: Buffer Schematic

## 2.12 Inverter

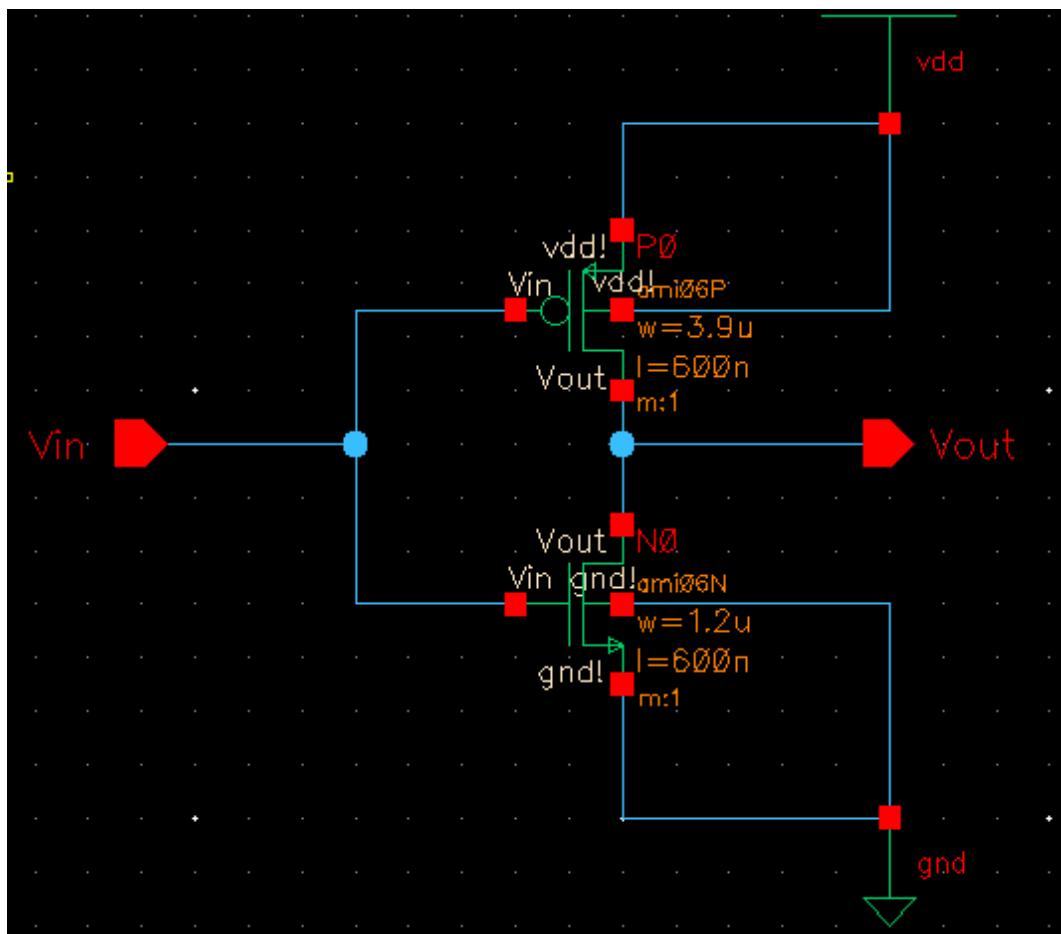


Figure 17: Inverter Schematic

### 3 Chip Layout

#### 3.1 DES Chip

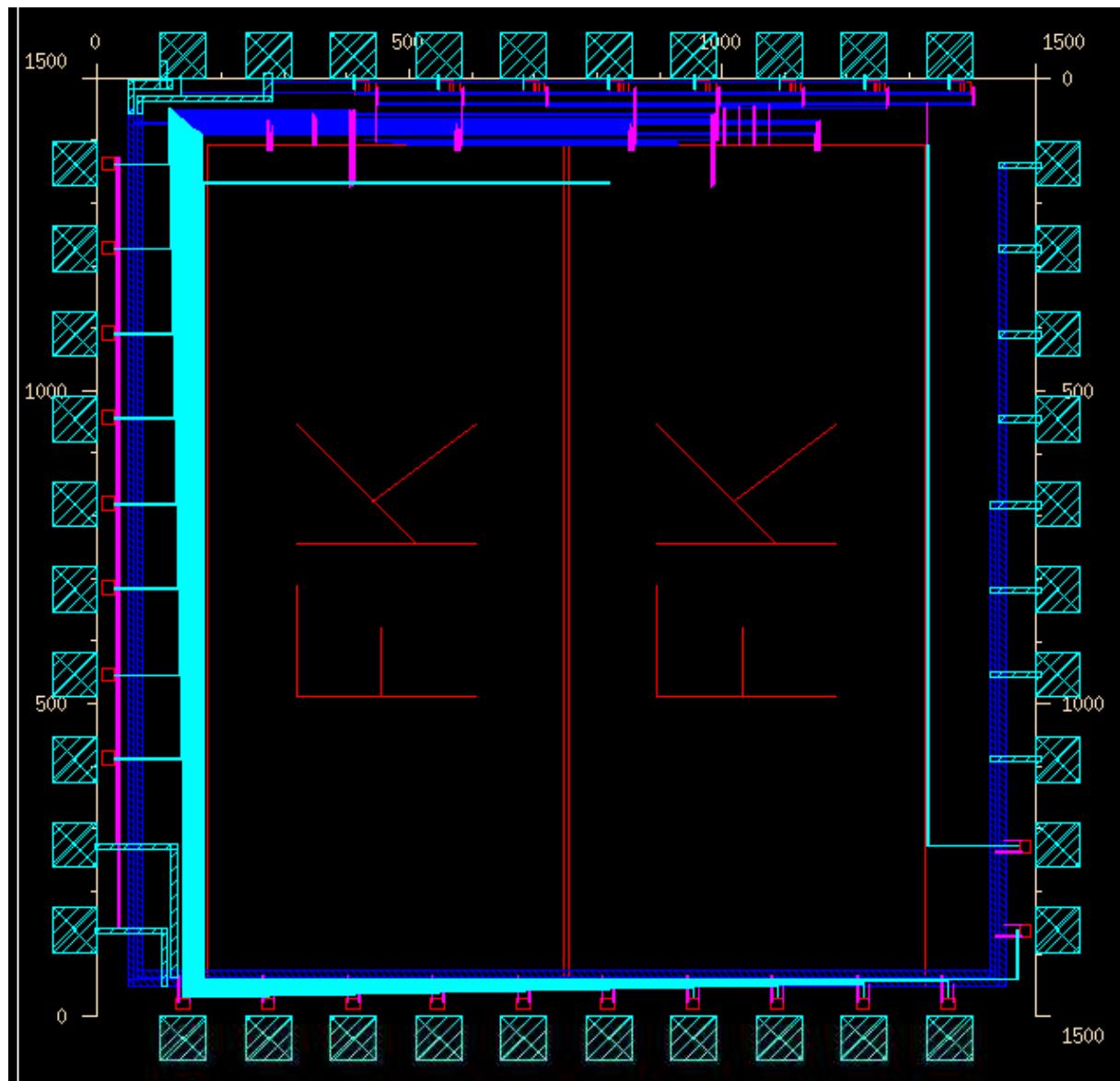


Figure 18: Chip Block Layout

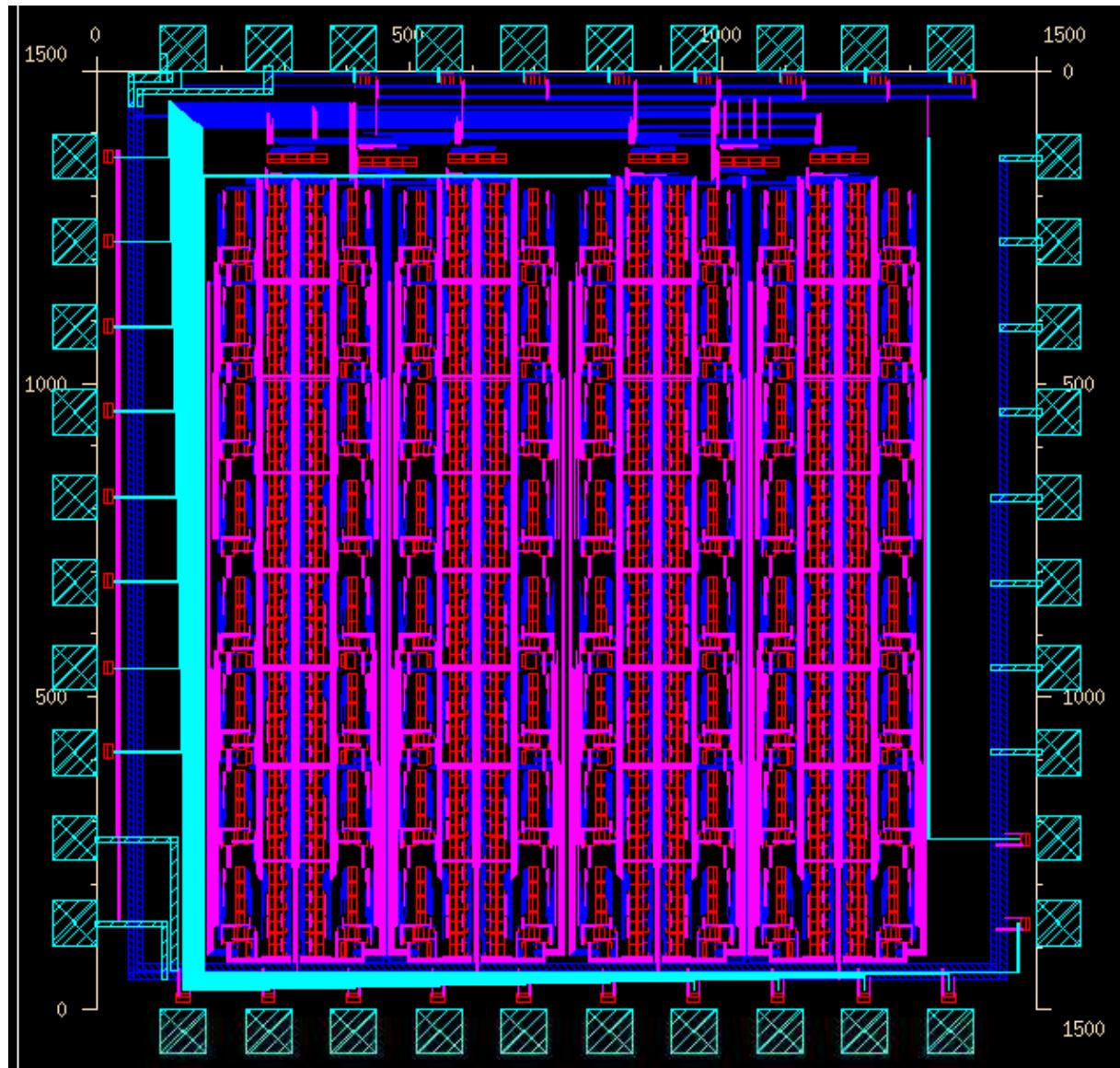


Figure 19: Chip Full Layout

### 3.2 FK

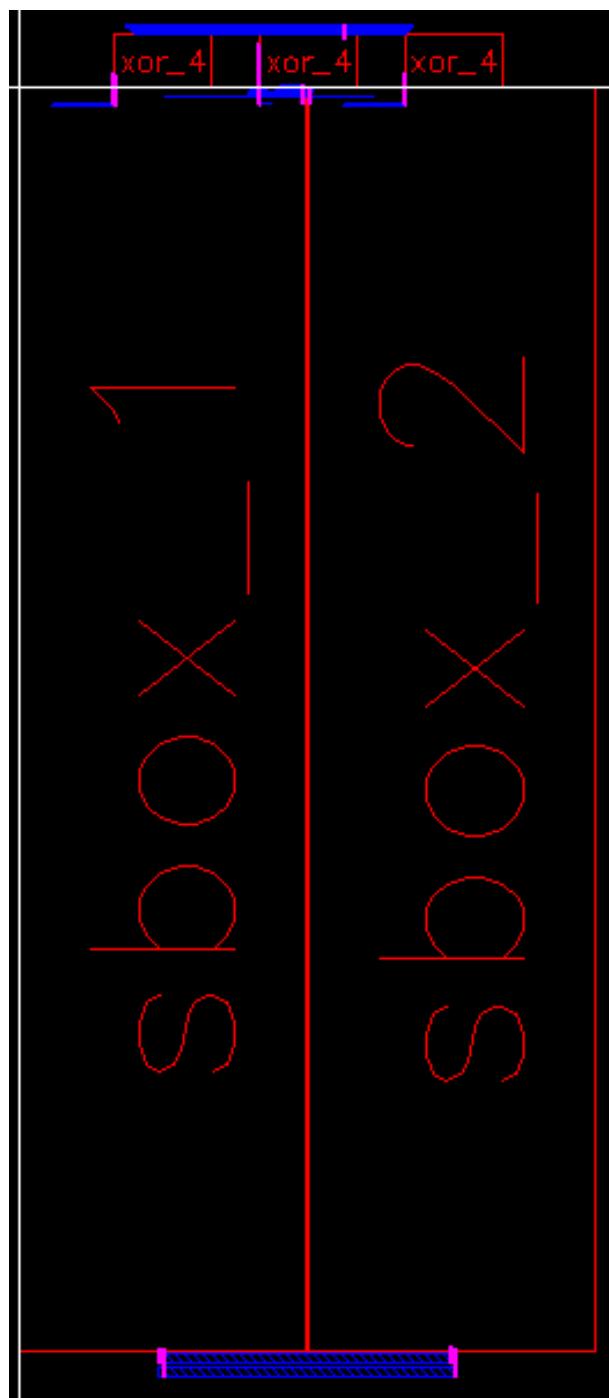


Figure 20: FK Block Layout

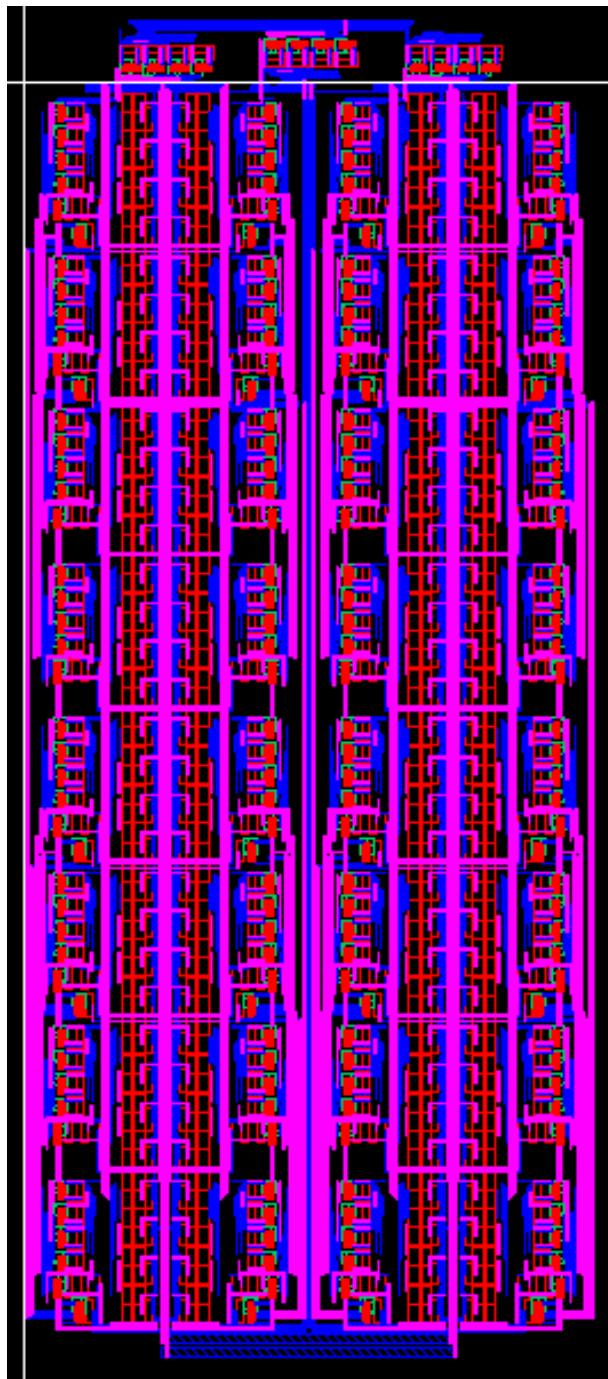


Figure 21: FK Full Layout

### 3.3 Sboxes

#### 3.3.1 SBox #1

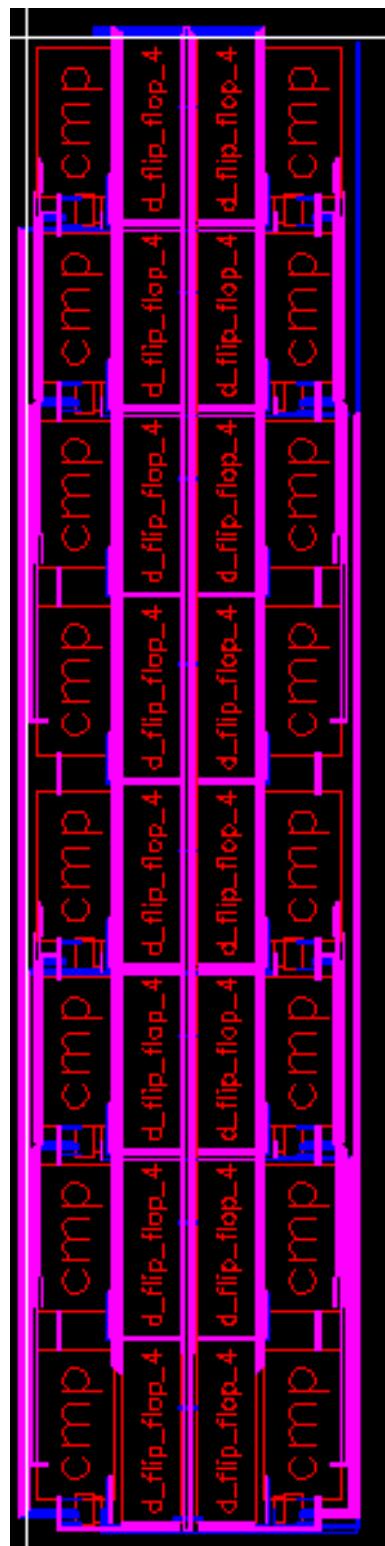


Figure 22: SBox #1 Block Layout

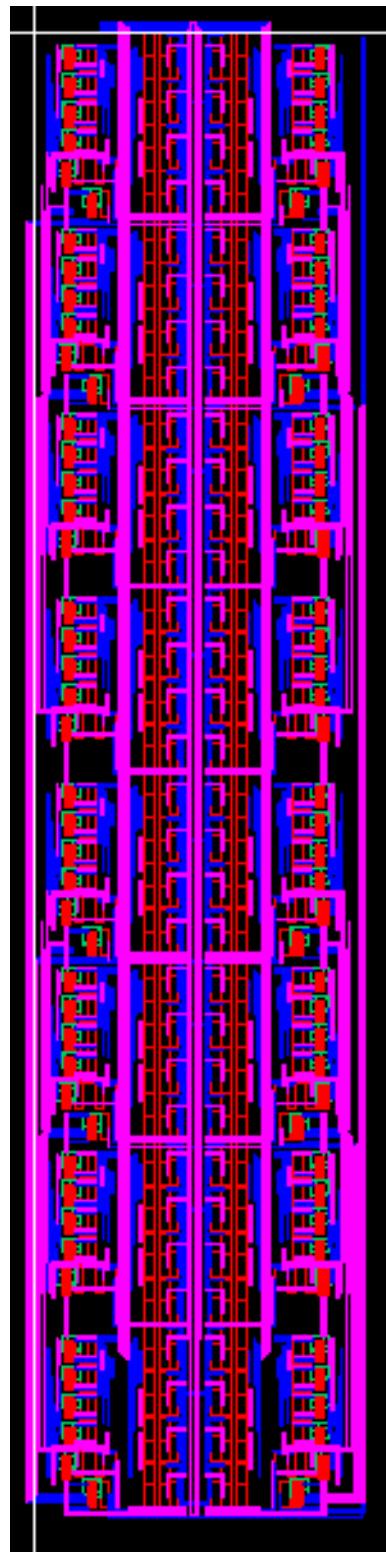


Figure 23: SBox #1 Full Layout

### 3.3.2 SBox #2

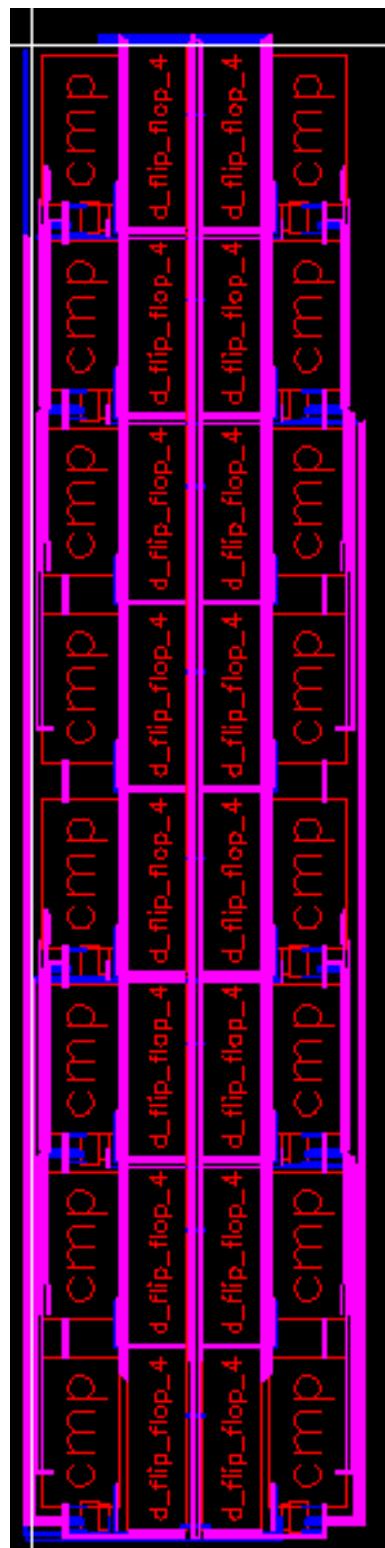


Figure 24: SBox #2 Block Layout

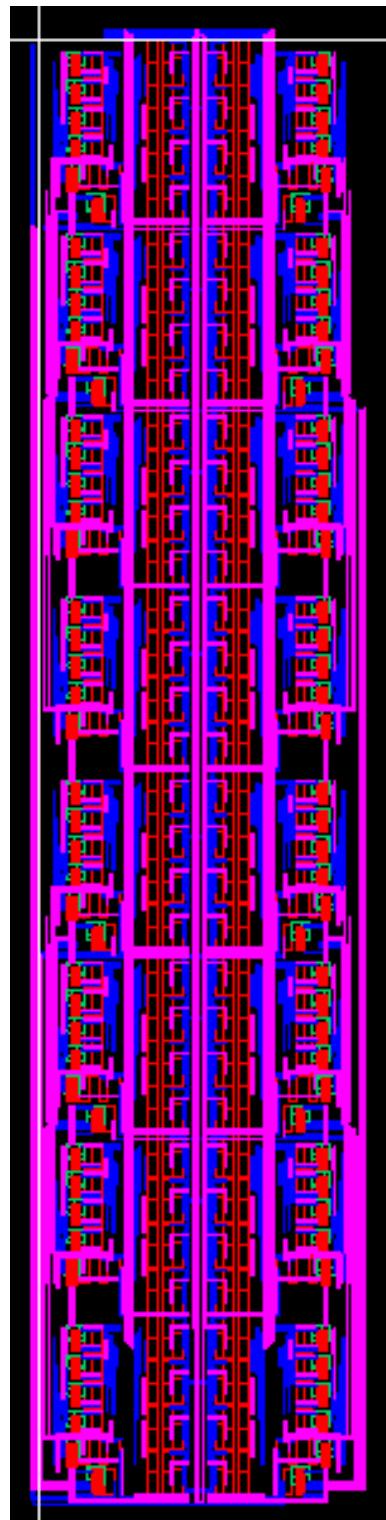


Figure 25: SBox #2 Full Layout

### 3.3.3 Zoom in

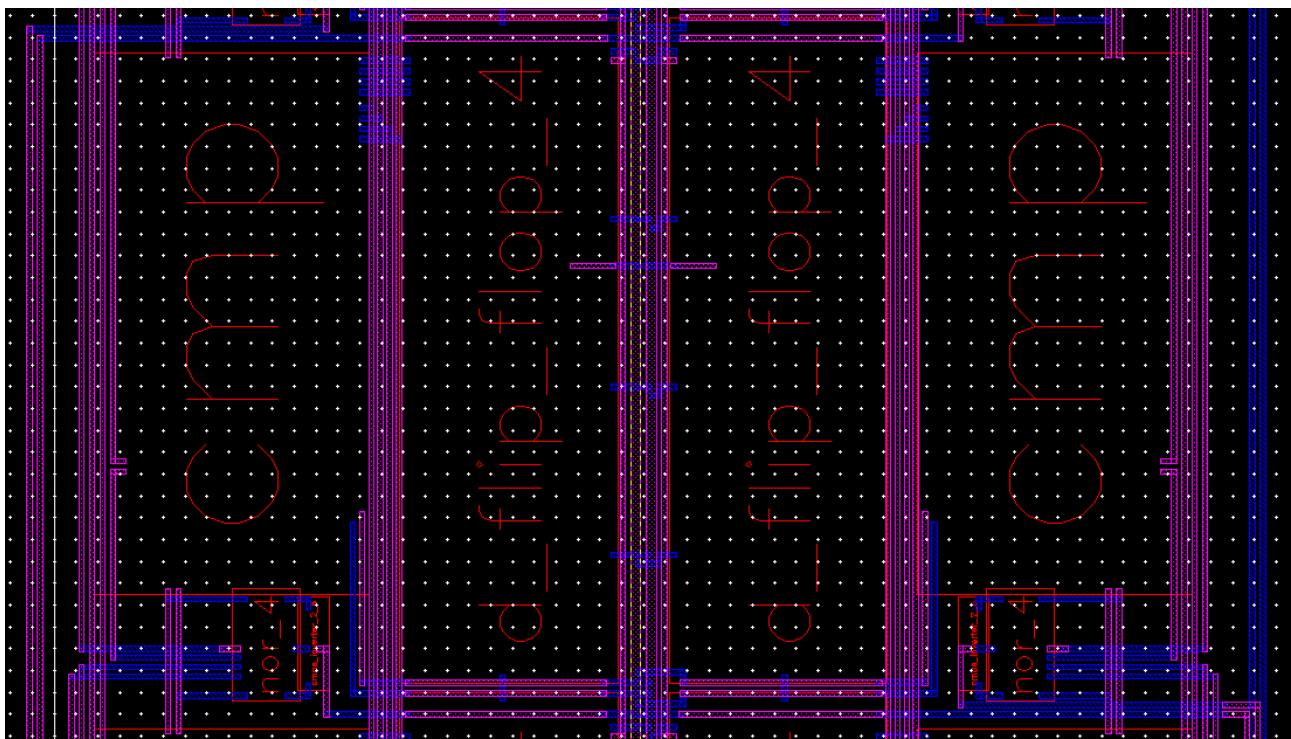


Figure 26: SBox Zoom Block Layout

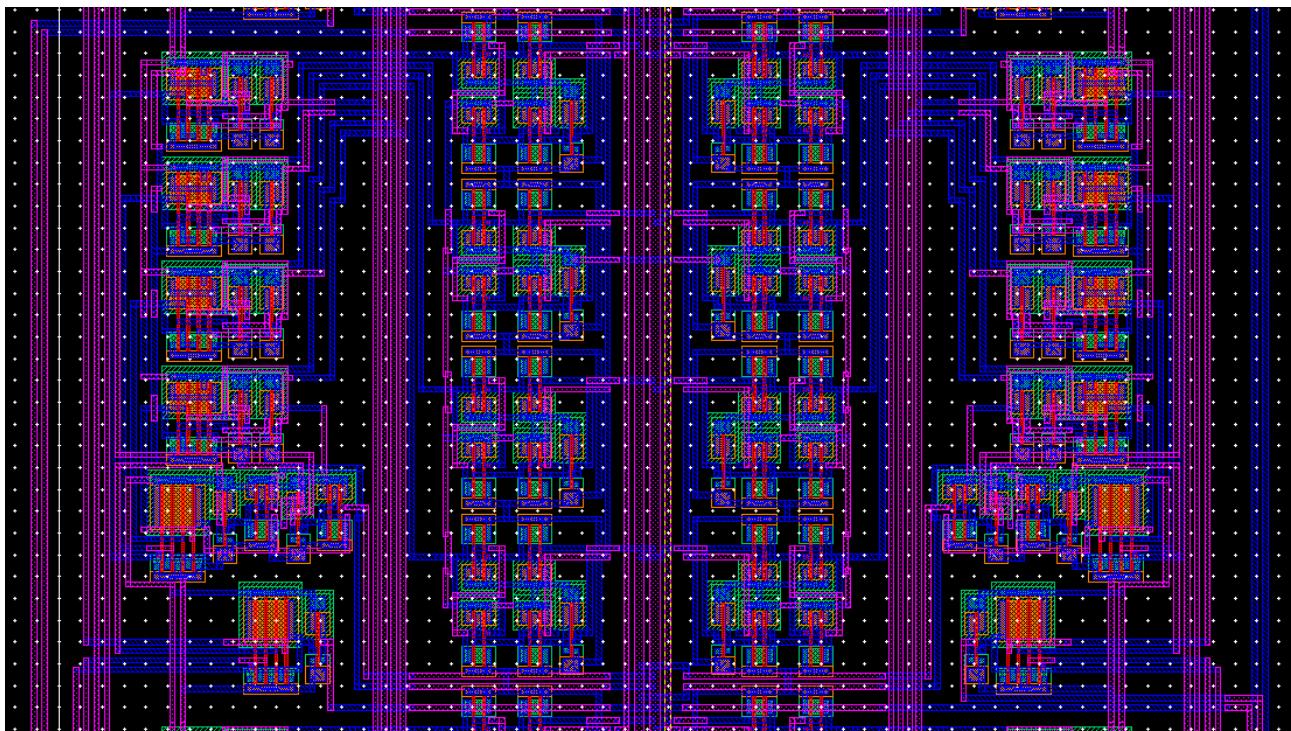


Figure 27: SBox Zoom Full Layout

### 3.4 Compare

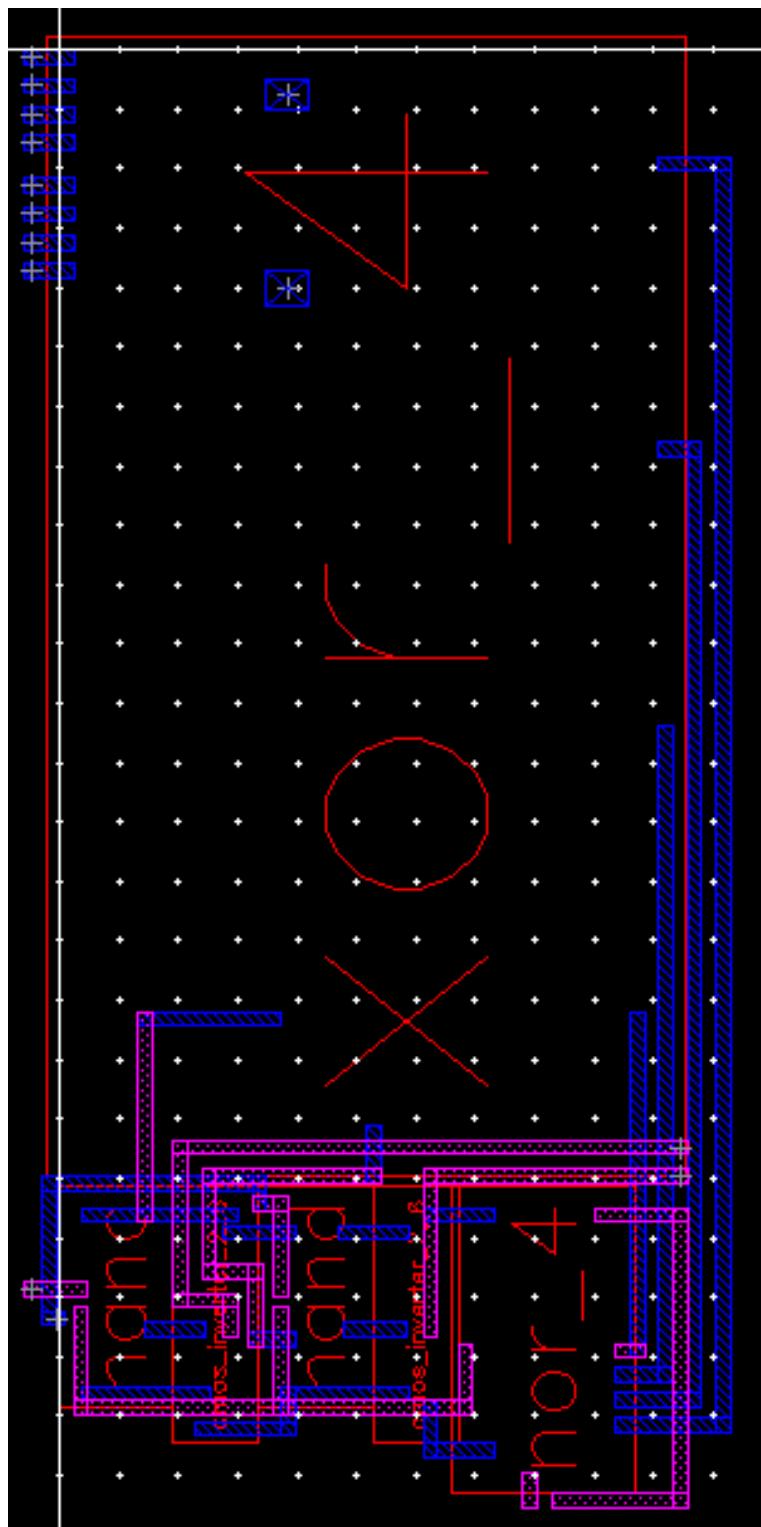


Figure 28: CMP Block Layout

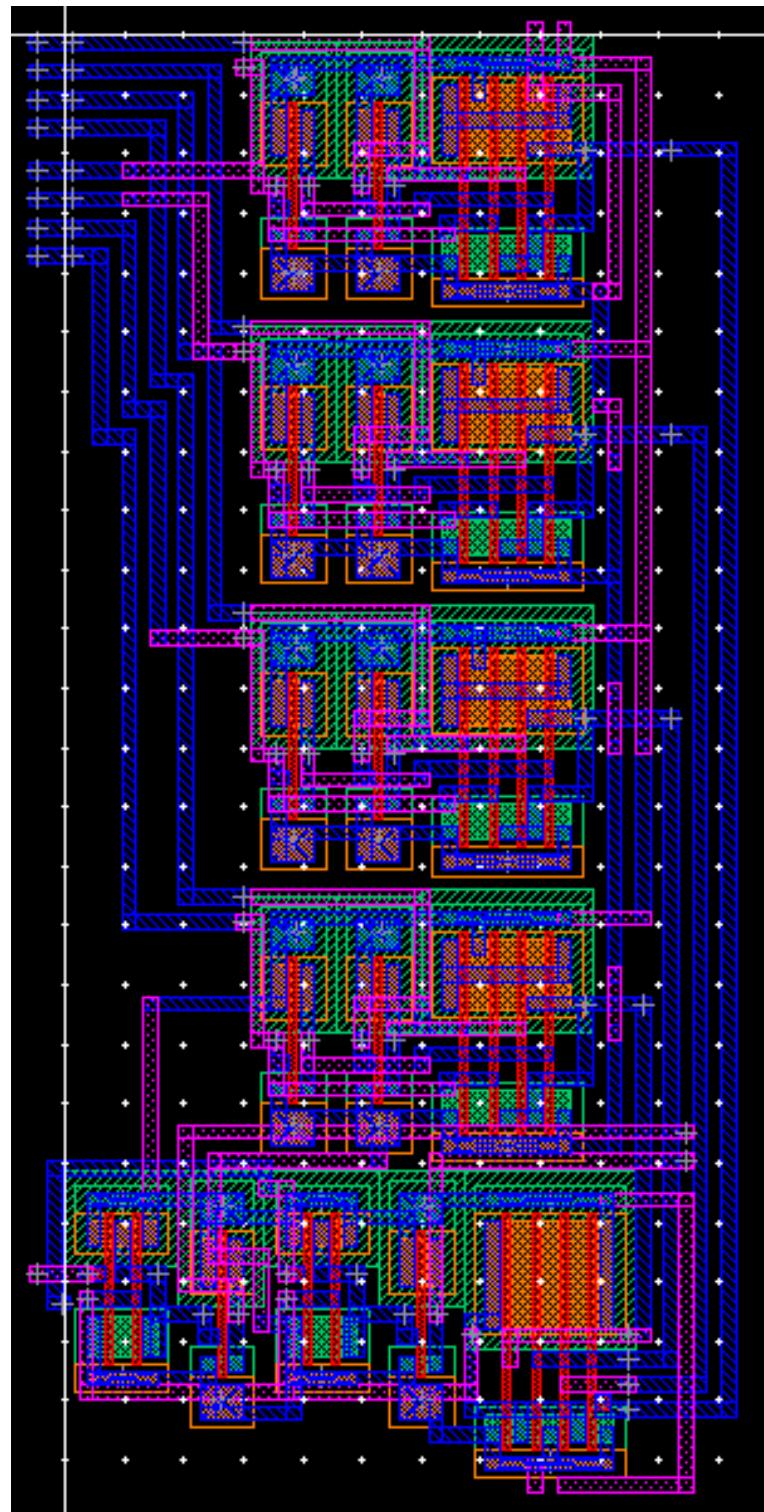


Figure 29: CMP Full Layout

### 3.5 D-flip-flop

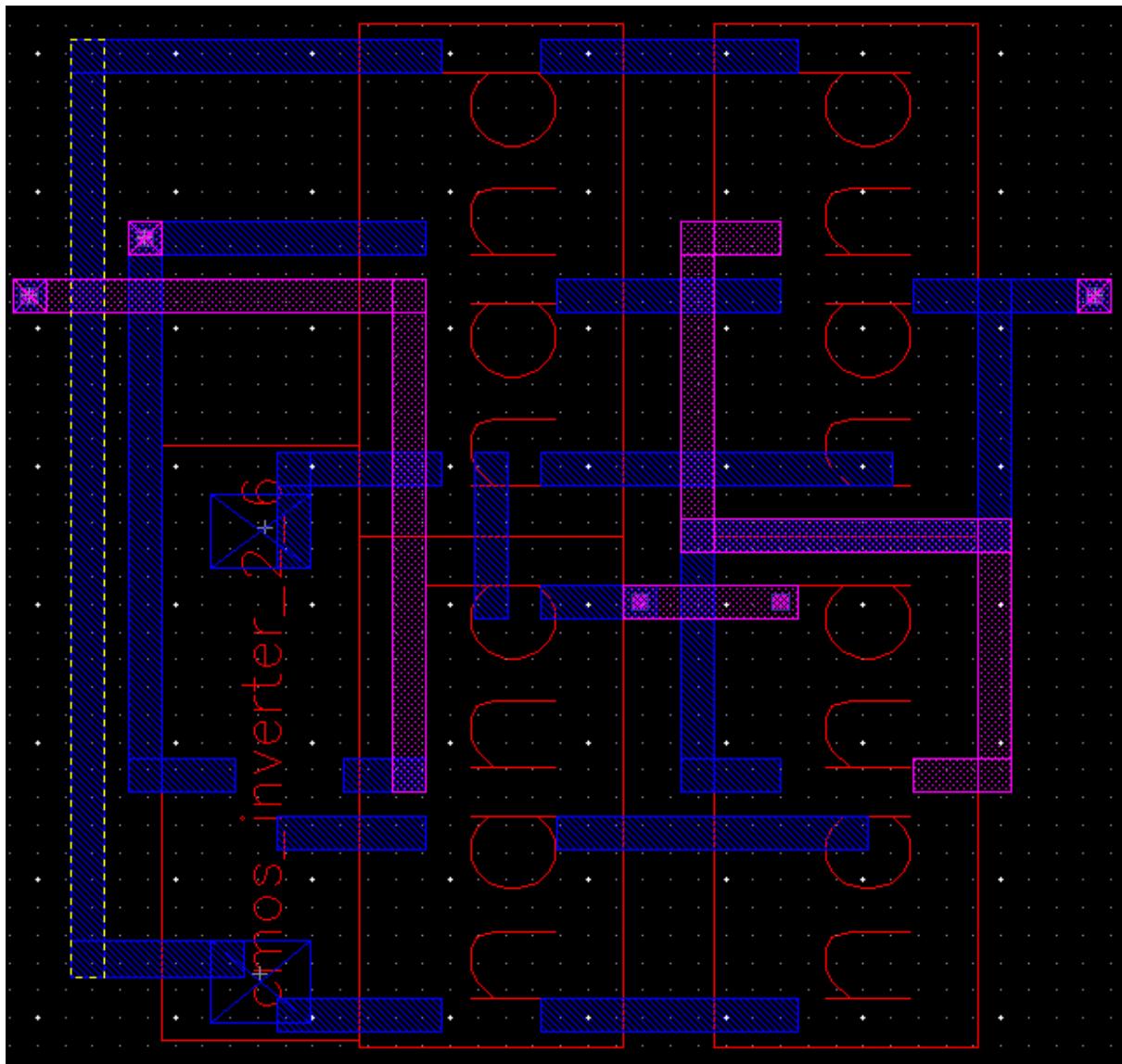


Figure 30: D-flip-flop Block Layout

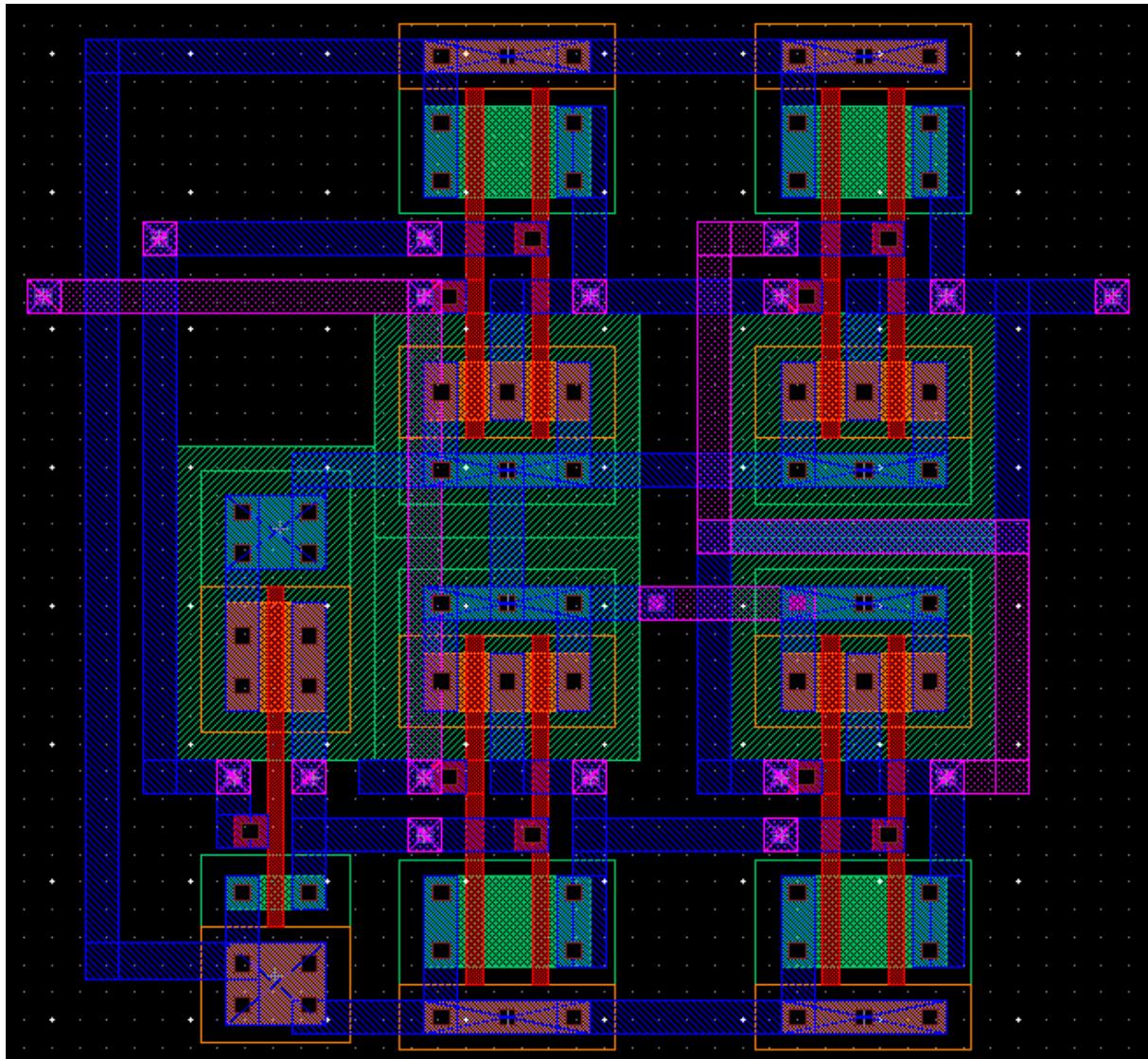


Figure 31: D-flip-flop Full Layout

### 3.6 4 D-flip-flop

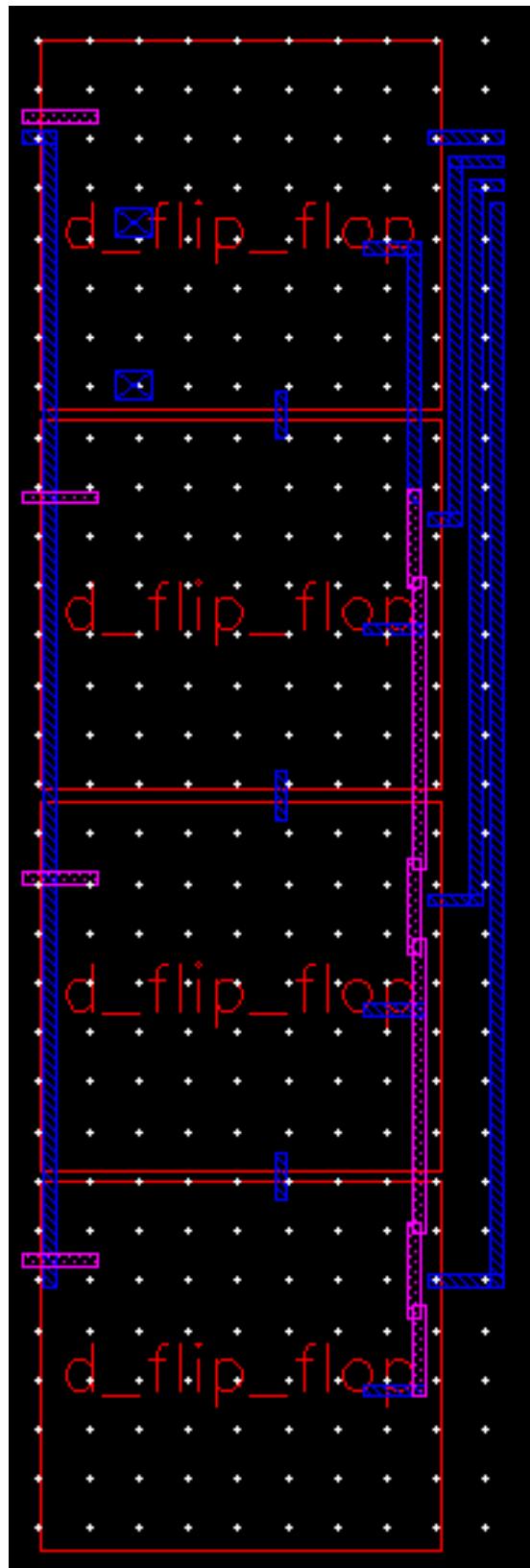


Figure 32: 4 D-flip-flop Block Layout

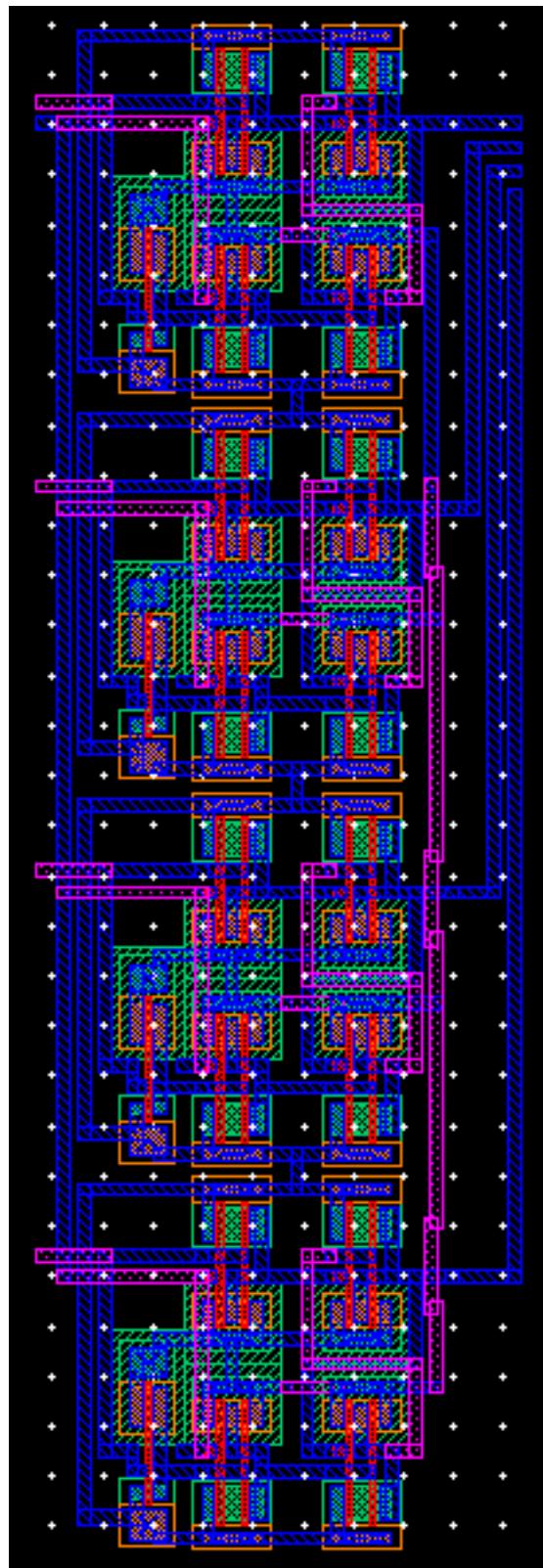


Figure 33: 4 D-flip-flop Full Layout

### 3.7 XOR4

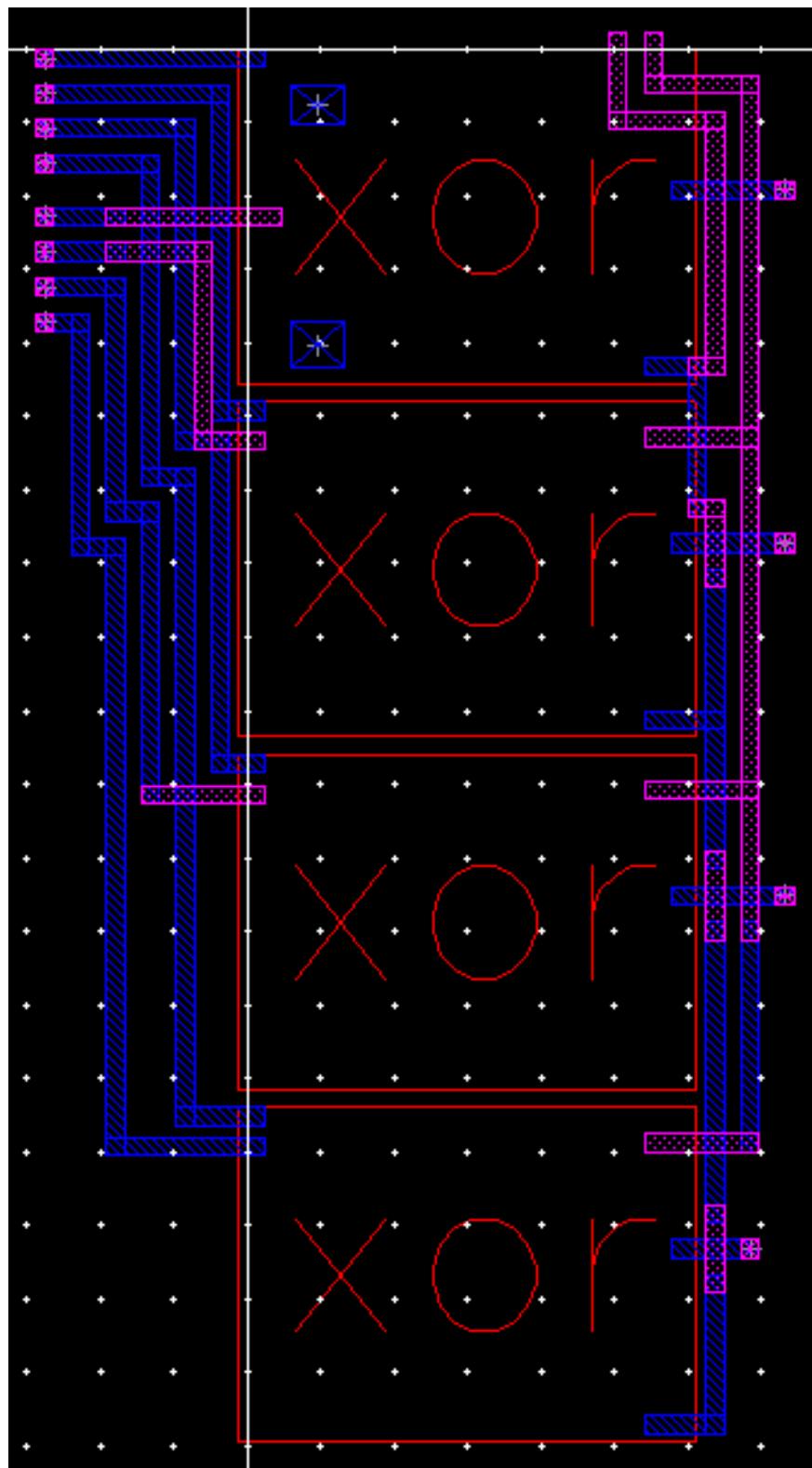


Figure 34: XOR4 Block Layout

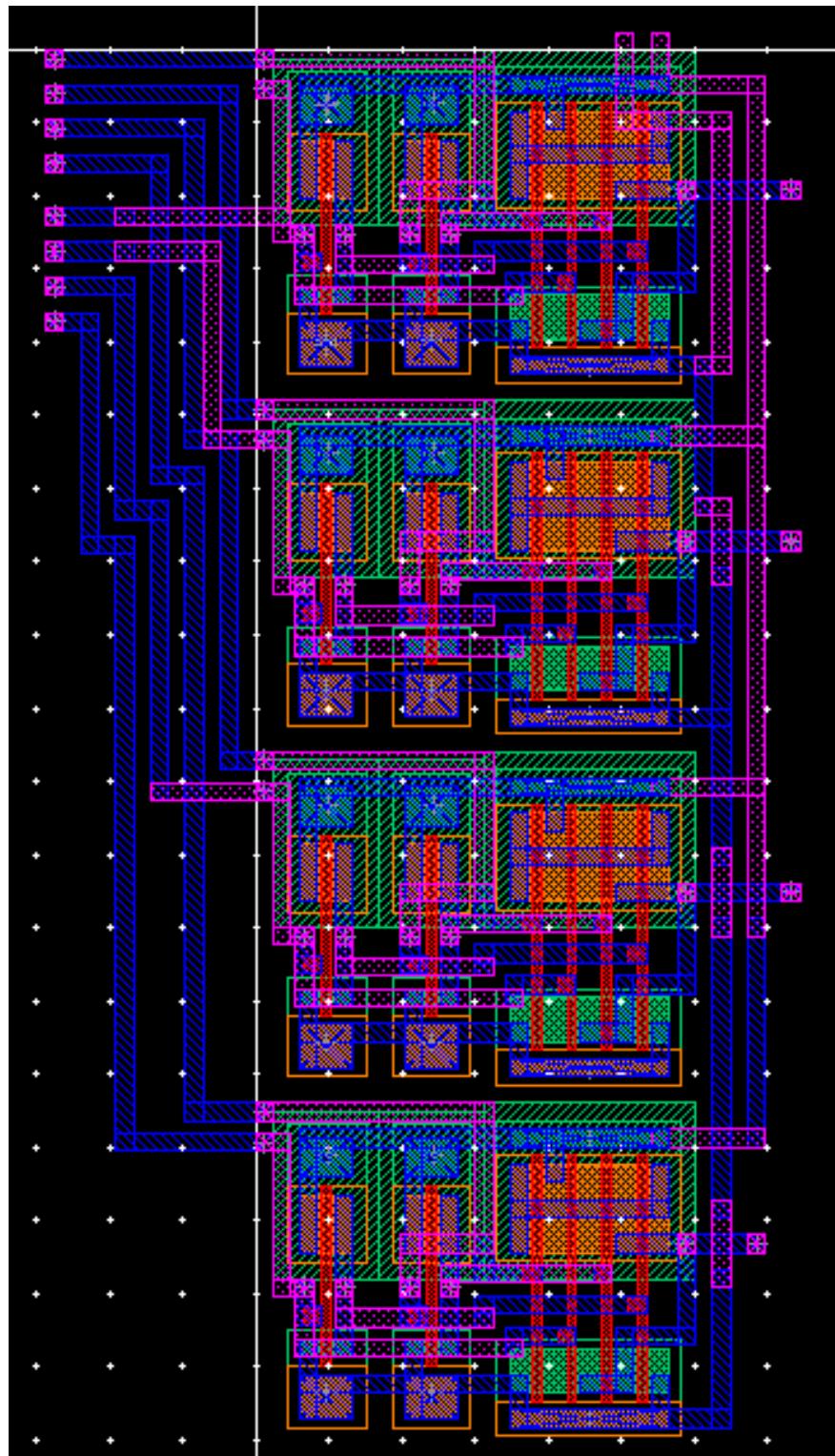


Figure 35: XOR4 Full Layout

### 3.8 XOR

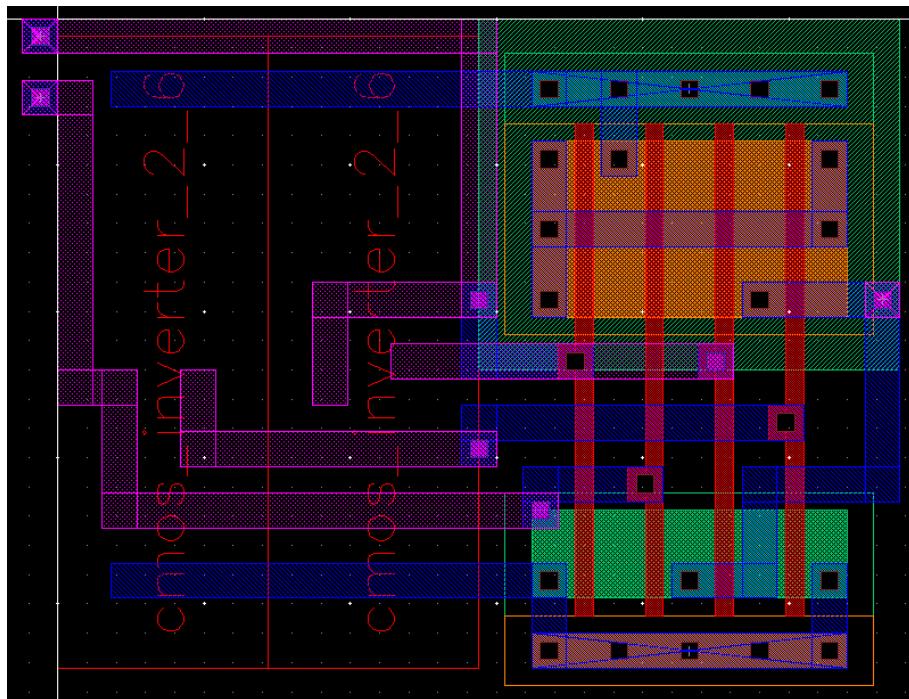


Figure 36: XOR Block Layout

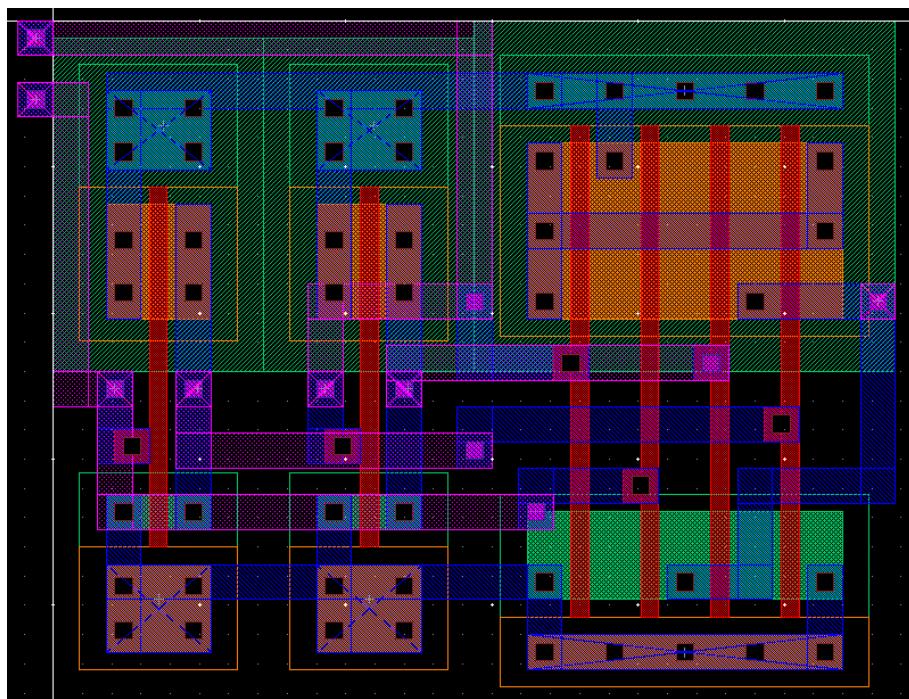


Figure 37: XOR Full Layout

### 3.9 NAND

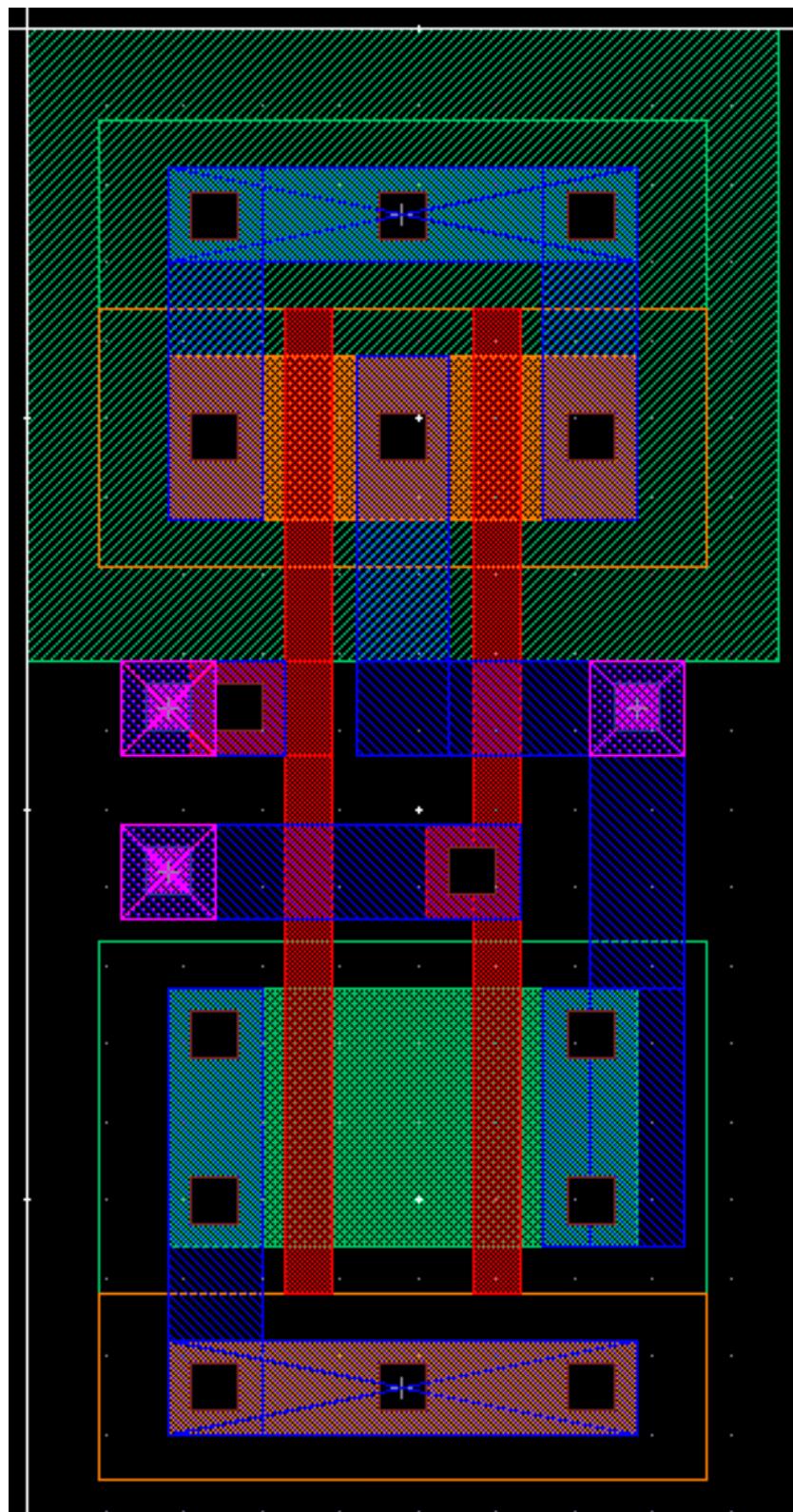


Figure 38: NAND Block Layout

### 3.10 NOR4

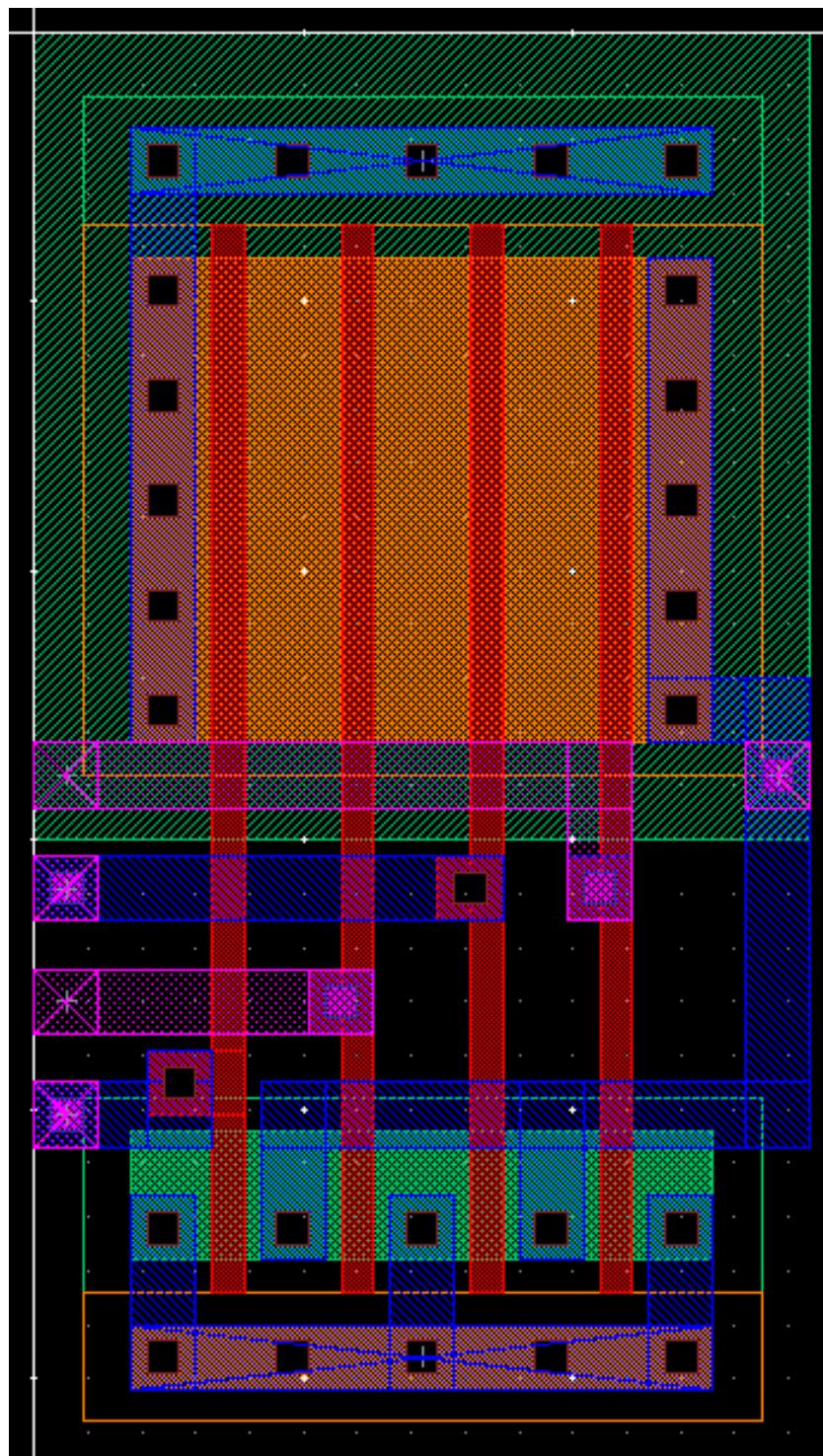


Figure 39: NAND Full Layout

### 3.11 Buffer

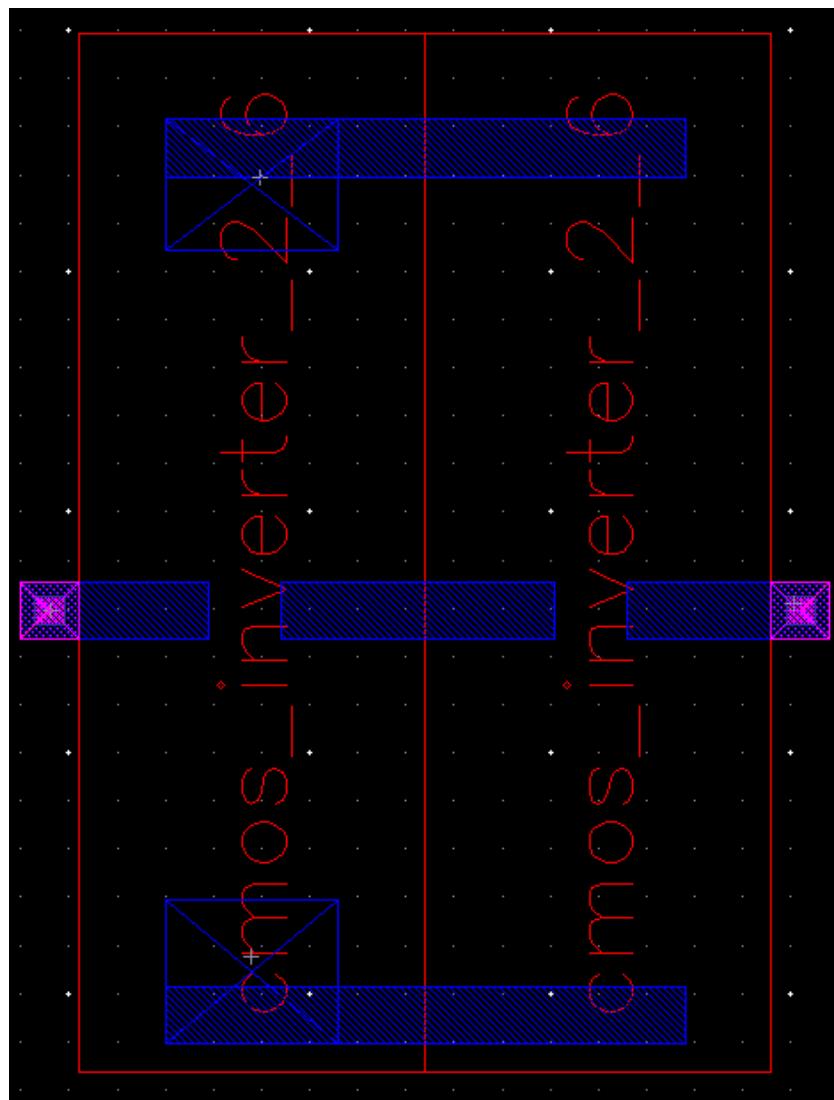


Figure 40: Buffer Block Layout

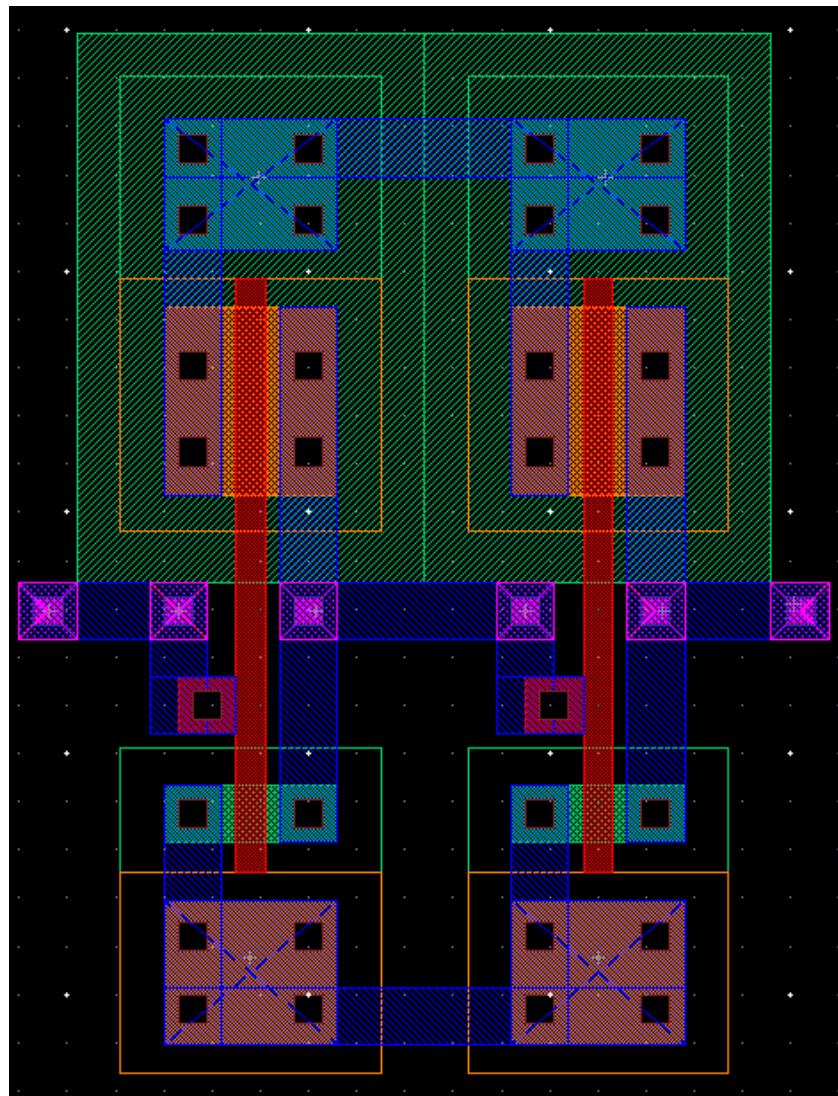


Figure 41: Buffer Full Layout

### 3.12 Inverter

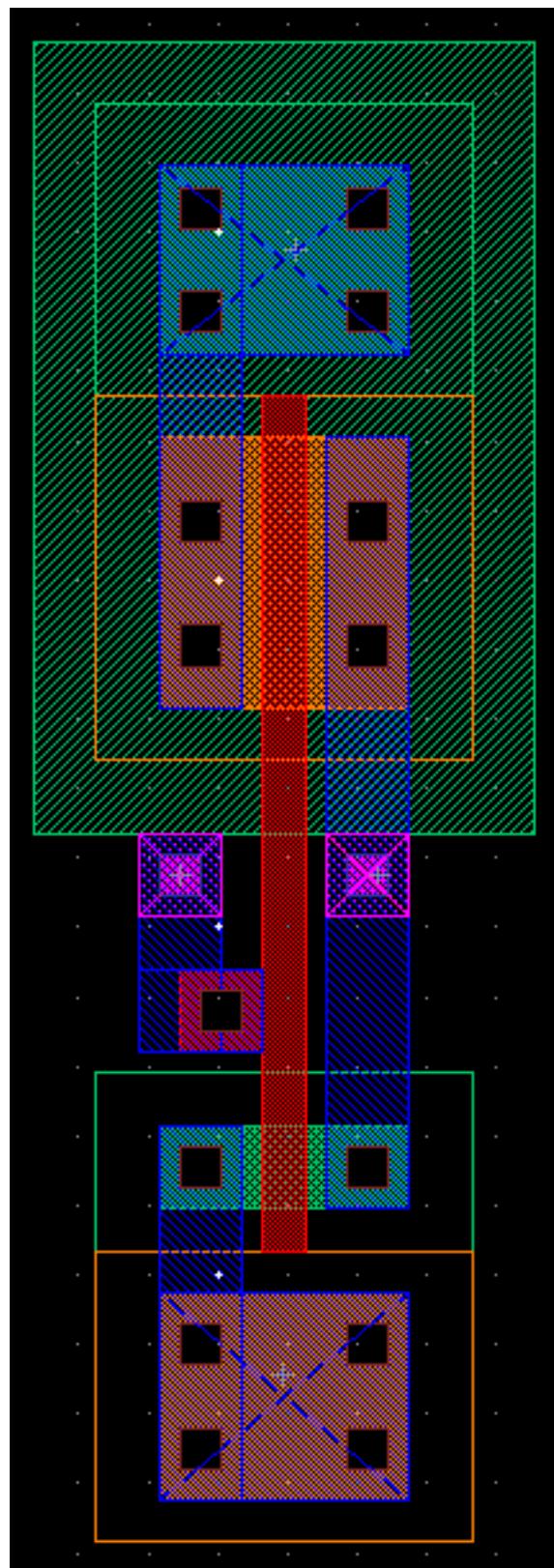


Figure 42: Inverter Full Layout

### 3.13 Layout DRC, Extraction, LVS

```
DRC started.....Mon May 10 21:48:45 2021
completed ....Mon May 10 21:48:56 2021
CPU TIME = 00:00:05 TOTAL TIME = 00:00:11
***** Summary of rule violations for cell "DESChip layout" *****
Total errors found: 0
```

Figure 43: DRC

```
Extraction started.....Mon May 10 21:49:20 2021
completed ....Mon May 10 21:49:38 2021
CPU TIME = 00:00:09 TOTAL TIME = 00:00:18
***** Summary of rule violations for cell "DESChip layout" *****
Total errors found: 0
```

```
saving rep 563Final/DESChip/extracted
Getting layout property bagGetting layout property bag
```

Figure 44: Extraction

```
saving rep 563Final/DESChip/extracted
Getting layout property bagGetting layout property bag
Getting layout property bagGetting layout property bagLVS job is now started...
The LVS job has completed. The net-lists match.
```

```
Run Directory: /home/warehouse/tommykeating/LVS
```

Figure 45: LVS

### 3.14 Layout Description

Some of the components of our layout design come from homework assignments. The inverter and NAND gate are taken from our homework implementation. They are examples of basic building blocks used to complete our chip design. The other basic blocks used to implement our design are the D-Flip-Flop, the XOR, and the 4-input NOR. The inverter just changes the signal from a 1 to a 0 and vice versa. The NAND gate performs a NAND logic operation on its 2 bit input. D-Flip-Flop uses the clock to latch in a value via 4 NAND gates. XOR does an exclusive or operation on its 2 bit input.

These blocks are used to create some intermediate blocks such as the 4 D-Flip-Flop, the 4 input XOR, the buffer, and the cmp. The 4 D Flip-Flop is 4 of our D-Flip-Flops that all share a connection to the clock and have 1 input of the 4 D-Flip-Flop associated with one D-Flip-Flop. The 4 input XOR takes in 8 values, 4 bits associated with one input and 4 associated with the other. They are then routed to 4 XORs inside the block that compared the equivalent bit of each input. The buffer is simply 2 connected inverters that we use for protecting input and output to and from the pins of the chip. The compare block (CMP) is a core element of the Sboxes. The block take two 4 bit inputs (A and B) and a 2 bit input (C). The block compares the inputs A and B, and outputs two zeros if they are not equal. If the 4 bit values A and B match, the values input to C are passed through to the output.

The main 3 blocks of our design, not counting the design as a whole, are FK, SBox # 1, and SBox #2. At the top most level only 2 FK blocks are used because within these blocks the majority of the work occurs. Key 1 and Key 2 are generated via routing the necessary bits of the 10-bit key into FK 1 and FK 2 respectively. The input also experiences its transformation via IP through routing at this level. The output of each FK is routed to complete the work of  $IP^{-1}$  and each bit goes through a NAND gate, an inverter and a buffer before being output through a pin. The NAND gate connects to EN so that its value can be used to determine if output will be allowed or if only 0s will be output. The inverter makes sure the bit is flipped to be correct and the buffer protects the output from being changed. The EN signal, clock, and all input and key bits go through a buffer when entering the chip to protect their values before connecting to either FK block. Between the two FK blocks, the SW operation is done through the routing of wires.

At the FK level, The generated key and transformed input are each fed into a 4 input XOR to be prepared to go into each SBox. Bits 1 to 4 of the key go into the first XOR that feeds SBox #1 and Bits 5 to 8 of the key go into the other XOR that feeds SBox #2. Bits 5 to 8 of the input into FK are transformed through routing according to E/P with the first 4 bits of E/P going into the first XOR and the last 4 going into the other XOR. The output of the SBoxes are combined and transformed through routing according to P4 before entering a final 4 input XOR that compares P4 with the first 4 bits of the input in FK.

Both SBox #1 and SBox #2 are identical other than the 2 wires that connect to inC0 and inC1 for the 16 different cmps associated with the 16 unique inputs to the Sboxes. These 2 inputs connect to either vdd or gnd to act as 1 or 0 respectively in order to be the contents stored in the SBoxes as determined by the handout. The 16 4 D-Flip-Flop are connected to the clock that enters the SBox as well as vdd or gnd to stand in for all 16 possible SBox inputs. The out of each 4 D-Flip-Flop enters a cmp block as well as the 4 SBox input bits. These bits are compared and if they match, the contents of that box are sent as the output of the SBox. To make sure only the correct output escapes an SBox all out0 bits of all 16 cmp blocks are routed to 4 different 4 input NOR gates that are then inverted. The same thing happens to the out 0 bits of cmp. After this is done, the 4 4 input NORs associated with out 0 bits go to a final 4 input NOR so all 16 out0 wires are compared and the same thing happens with out1. Before leaving the output is once again inverted to make sure it is correct.

## 4 Chip Simulation

### 4.1 Schematic Simulation

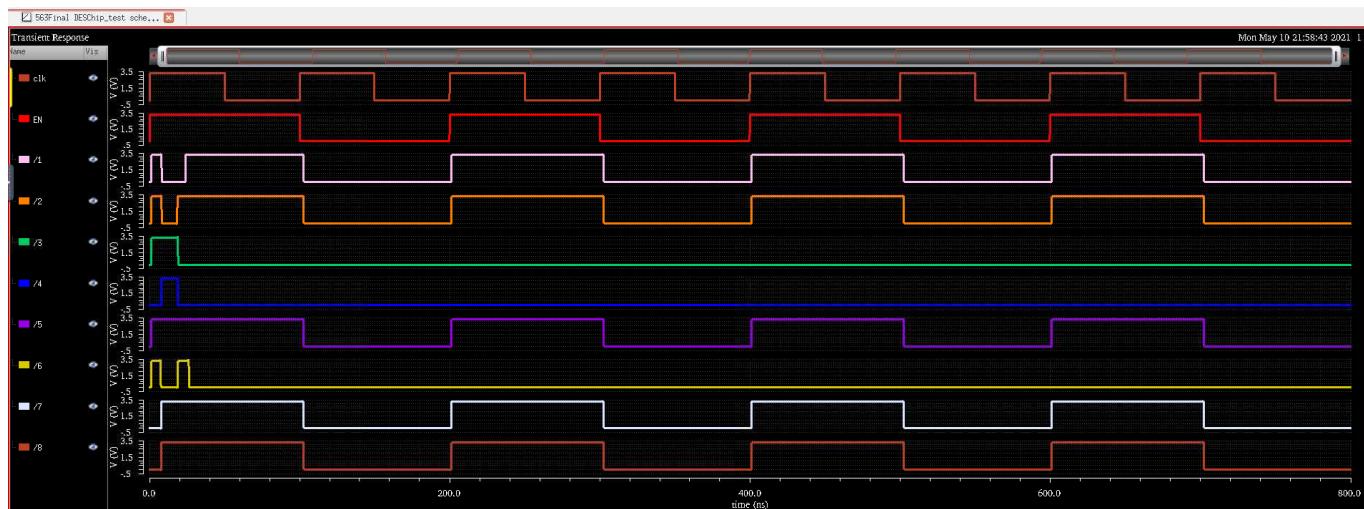


Figure 46: Schematic Sim

### 4.2 Layout Simulation

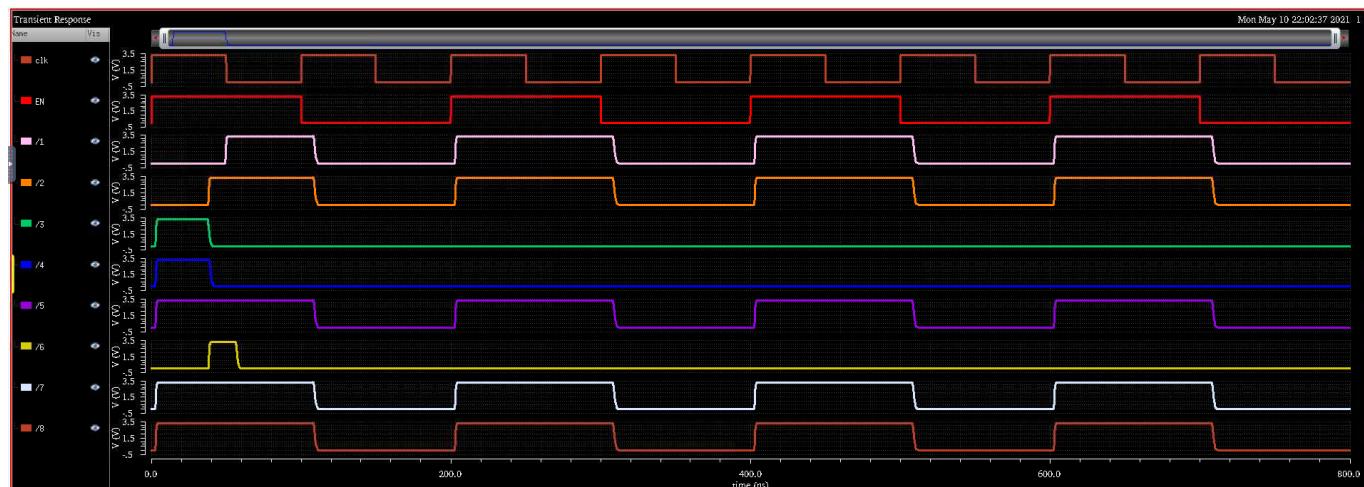


Figure 47: Schematic Sim

### 4.3 Operation Specifications

Since we intend to use this chip with a VDD of 3V that is how we tested our chip. Under these conditions the rise and fall time of the output seemed to be on average 0.8282ns and 1.87455ns respectively.

## 5 Clock Analysis

The only element being clocked in the circuit are the D-Flip-Flops in the sboxes. When the clock exceeds a certain speed, we might see failure of these devices, specifically that they might not be able to latch the input values during the high cycle. For this analysis we will leave enable high and look at the time required for all bits to settle for the design. The default behavior at 100MHz is shown in the next section.

### 5.1 10 MHZ - Default

As shown in Figure 48, at 10 MHz, all values have settled by 58.4ns. This delay is a result of the time required for values to propagate through the series of transistors making up the circuit, with the delay exacerbated by the capacitance in the circuit.



Figure 48: 10Mhz Settling Time

### 5.2 Faster Clock Speeds

We used Cadence's "Parametric Analysis" feature to evaluate the design for multiple clock speeds. We ran tests with clocks ranging from 10Mhz to 1GHz (periods of 100ns to 1ns). As the clock speed increased the D-Flip-Flops began to take longer to latch in values, resulting in a longer time for the input signal to propagate through. A selection of these speeds are shown below. By looking at the time it takes the values to settle we can make an estimate as to when failure will occur. Based on this we found the failure point was when the clock speed reached 1GHz or rather a 1ns period. This kind of failure occurs because when the clock is too fast the flip flops cannot properly latch in their values. While this was the point where total failure occurred, the other higher clock speeds still took longer for the signal to propagate through, so the chip does not work perfectly up to 1GHz.

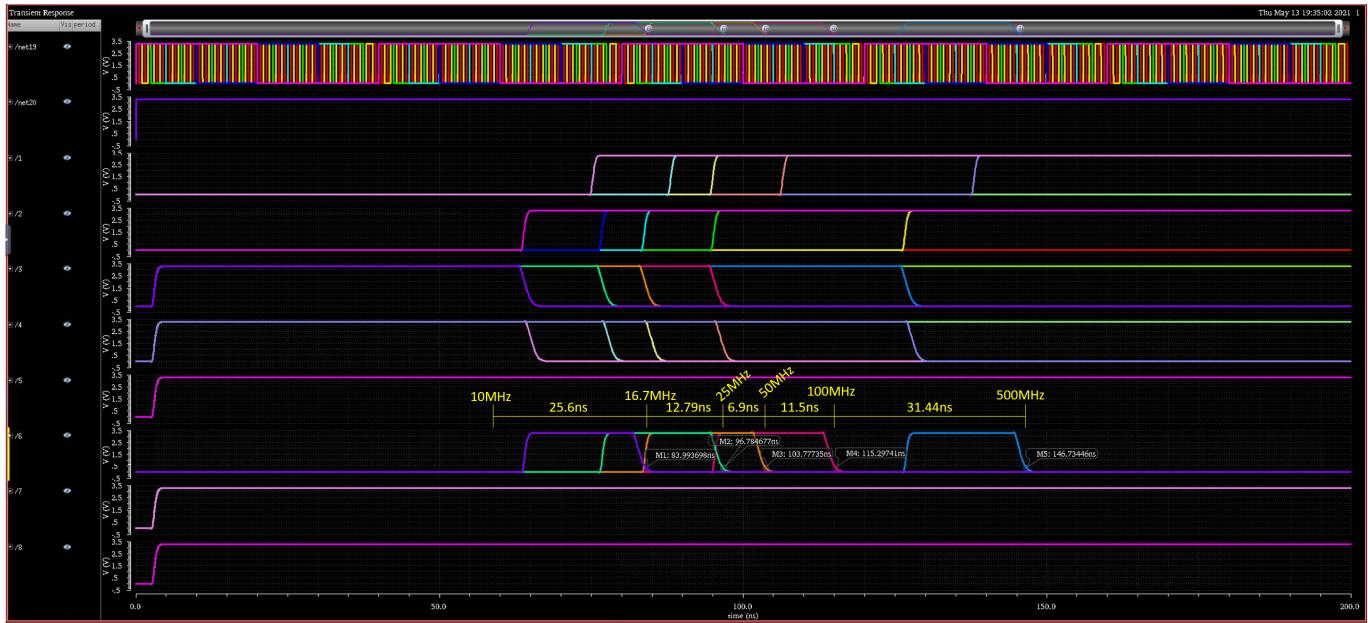


Figure 49: Faster Clock Settling Time

### 5.3 VDD = 5V

During our testing we also explored using a higher VDD in the form of 5V. We found that by using a higher VDD, the response time of the chip increased. This is seen in that the time it took the flip flops and thus the outputs to settle was much quicker and the outputs became stable earlier than when the chip was tested with 3.3V as the VDD. We saw an increase in performance from the 58.4ns time at 3.3V to 40.7ns at 5V.



Figure 50: Faster Clock Settling Time