Gabe G., Lawrence L., David Y.

ESE 463

5.4.25

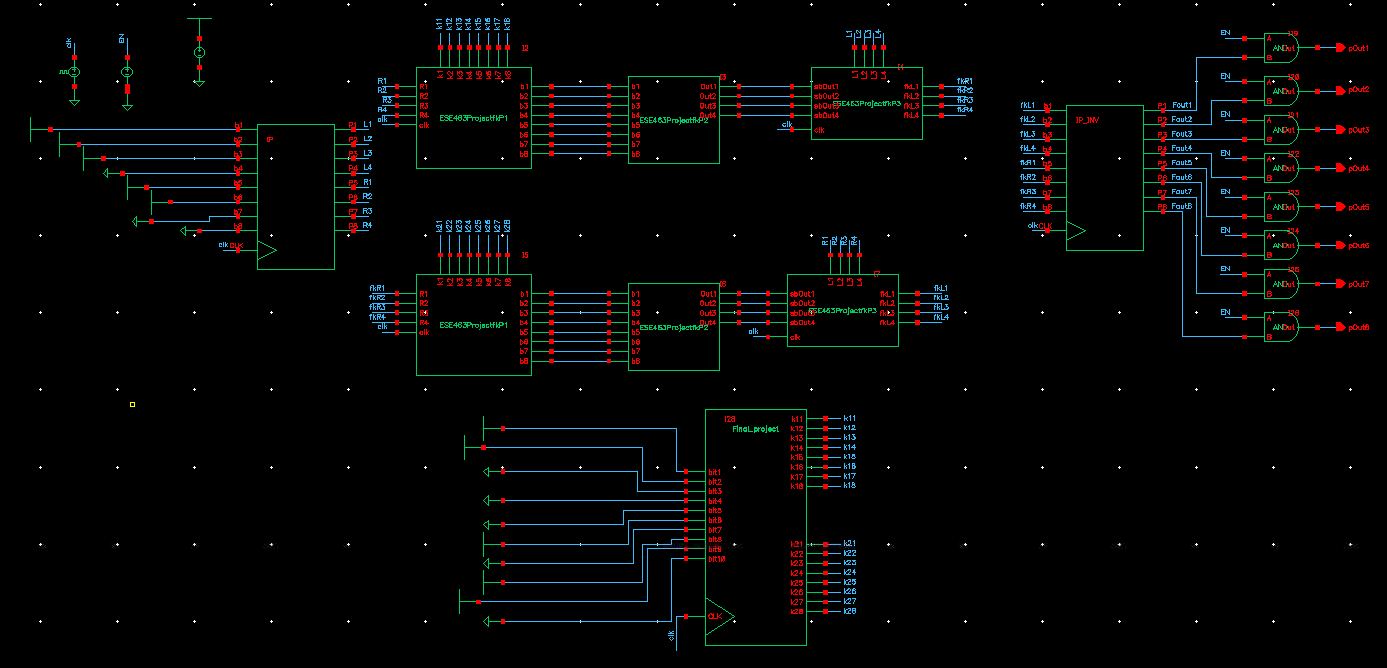
ESE 463 – Final Project Report

**Introduction:**

The *Data Encryption Standard* (DES) is a symmetric-key block-cipher algorithm that was developed by IBM in the early 1970s [1]. IBM submitted DES to the U.S. National Bureau of Standards (NBS, now NIST) in 1975; after the bureau consulted with the National Security Agency (NSA), a slightly modified version was adopted and published in 1977 as **FIPS PUB 46**. Soon after its adoption, DES was criticized for its 56-bit effective key length, which many researchers argued was too short to resist exhaustive, brute-force search as computer power improved. As of today, DES is no longer the encryption standard. In this project, we aim to understand the simplified DES and design a chip capable of performing the algorithm, where we provide an 8-bit binary input and a 10-bit binary input as the key and receive an 8-bit binary number as the output.

**Chip and Circuit Description:**

The chip will require a VDD of 3.3V, and the inputs will be using logic high and low. Thus, the logic voltages will be 3.3V and 0V respectively. The outputs will also be logic high and low. The Following figure shows the overall schematic of our DES chip. In this schematic, we used symbols, as well as some circuit elements (voltage sources) for the different sections of the algorithm. For example, we have a symbol for the fk functions, IP permutations, etc. In the report below we show the screenshots of each block symbol as well as the schematic of the blocks. The layouts for each of the components are included as well.

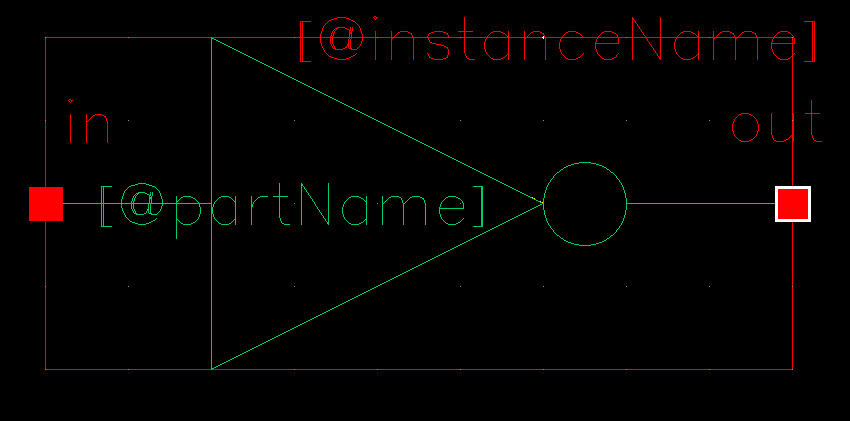


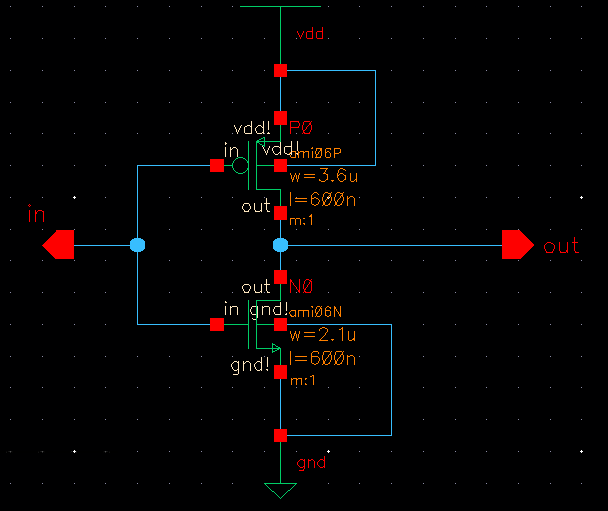
**Figure 1 -** Overall Diagram of DES Chip



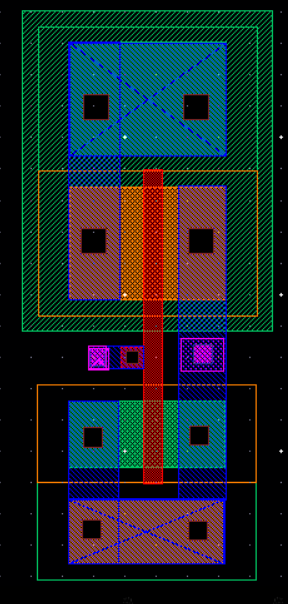
**Figure 2 -** Overall Chip Layout

The following figures show the INV symbol, the transistor schematic, and the layout of the inverter, respectively.

**Figure 3 –** INV Symbol

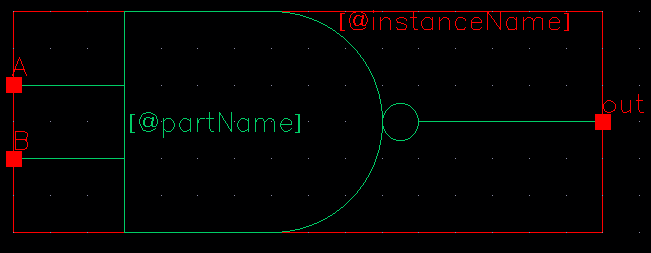


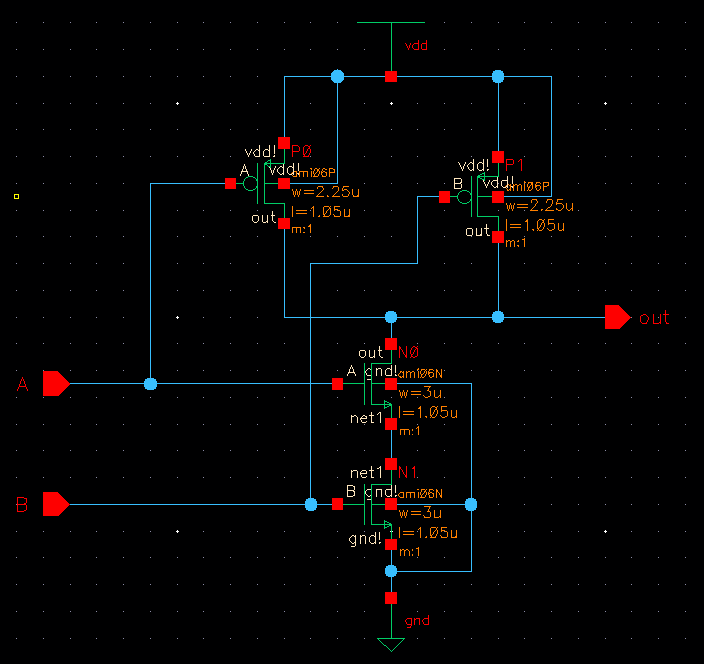
**Figure 4 –** INV Circuit Schematic

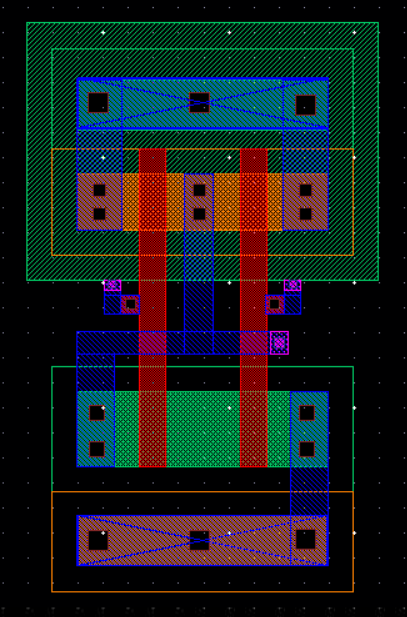


**Figure 5 -** INV Layout

The following figures show the NAND gate symbol, the transistor schematic, and the layout of the NAND gate, respectively.

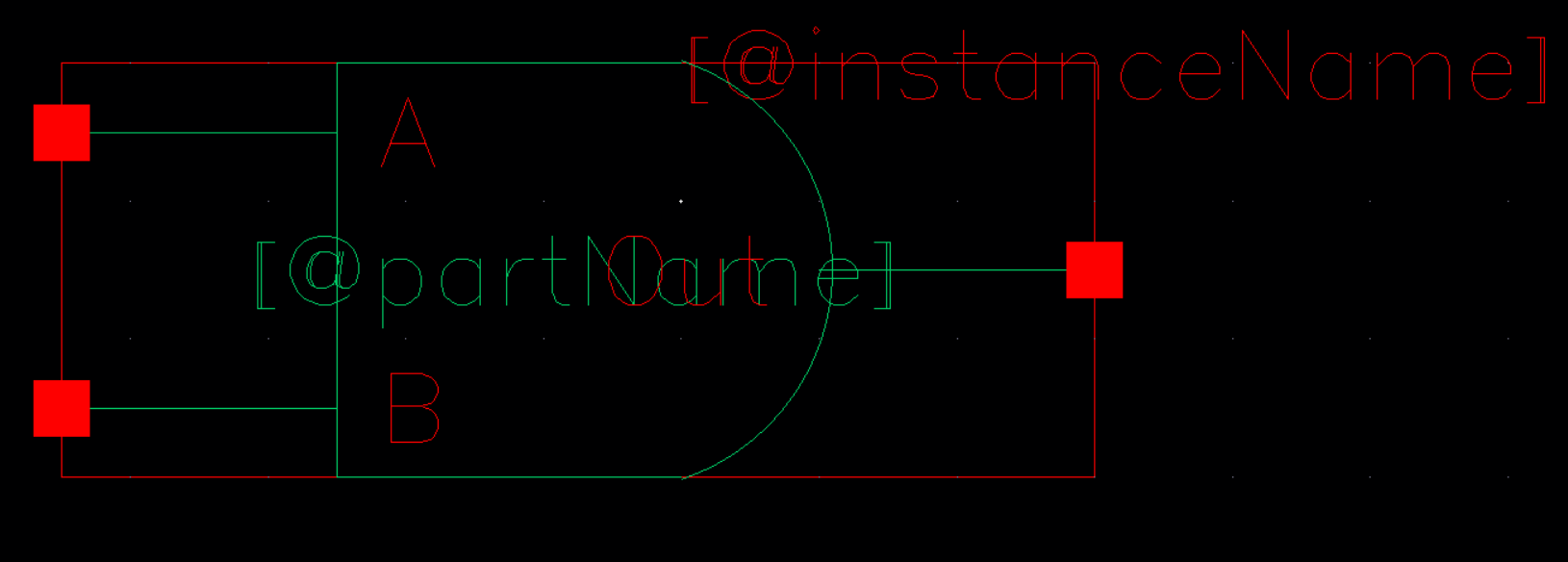
**Figure 6 -** NAND Symbol

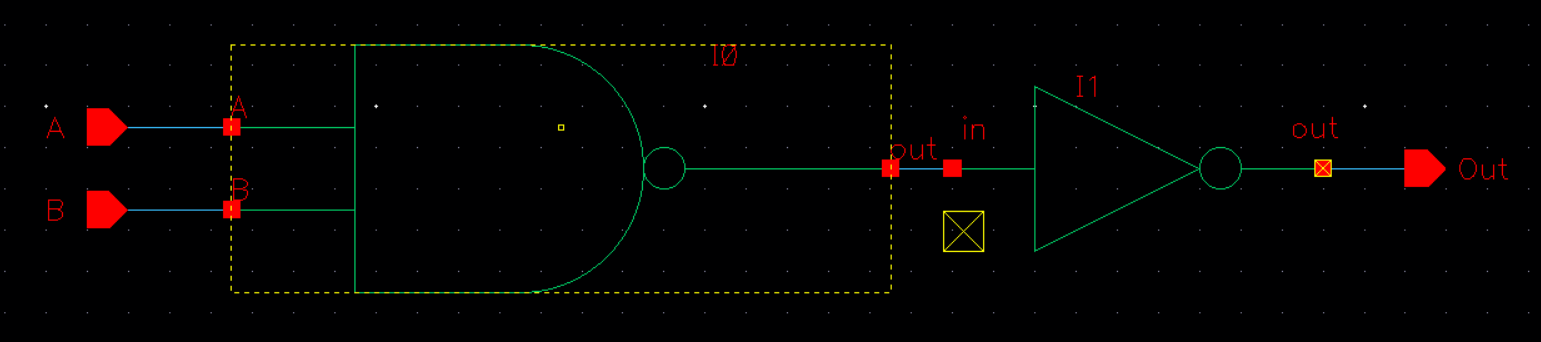
**Figure 7 -** NAND Circuit Schematic

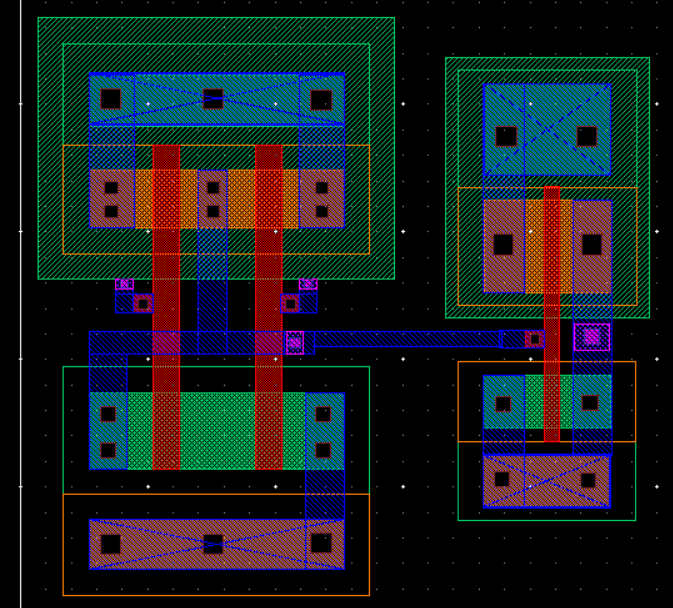


**Figure 8 -** NAND Layout

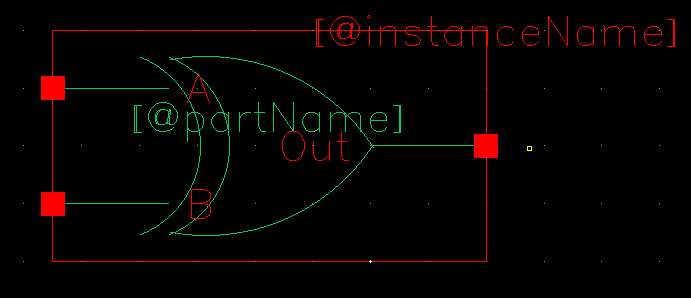
The following figures shows the symbol, schematic, and layout of our AND gate.

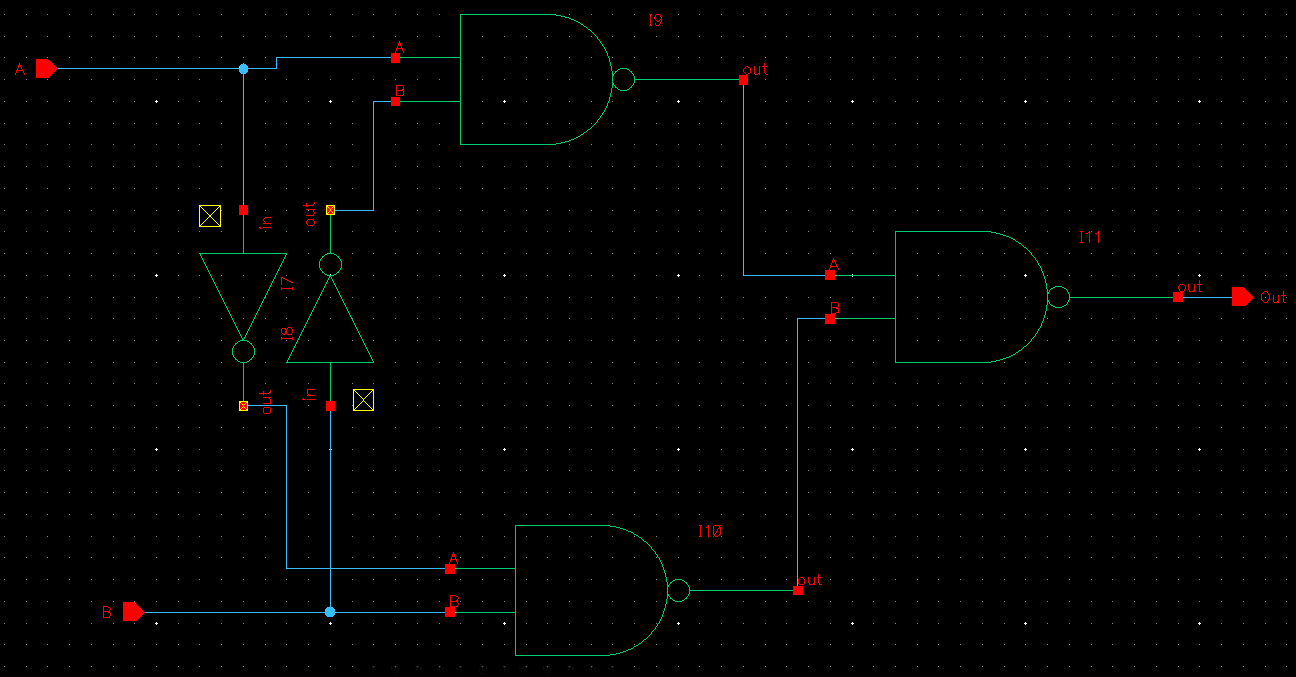
**Figure 9 -** AND Symbol

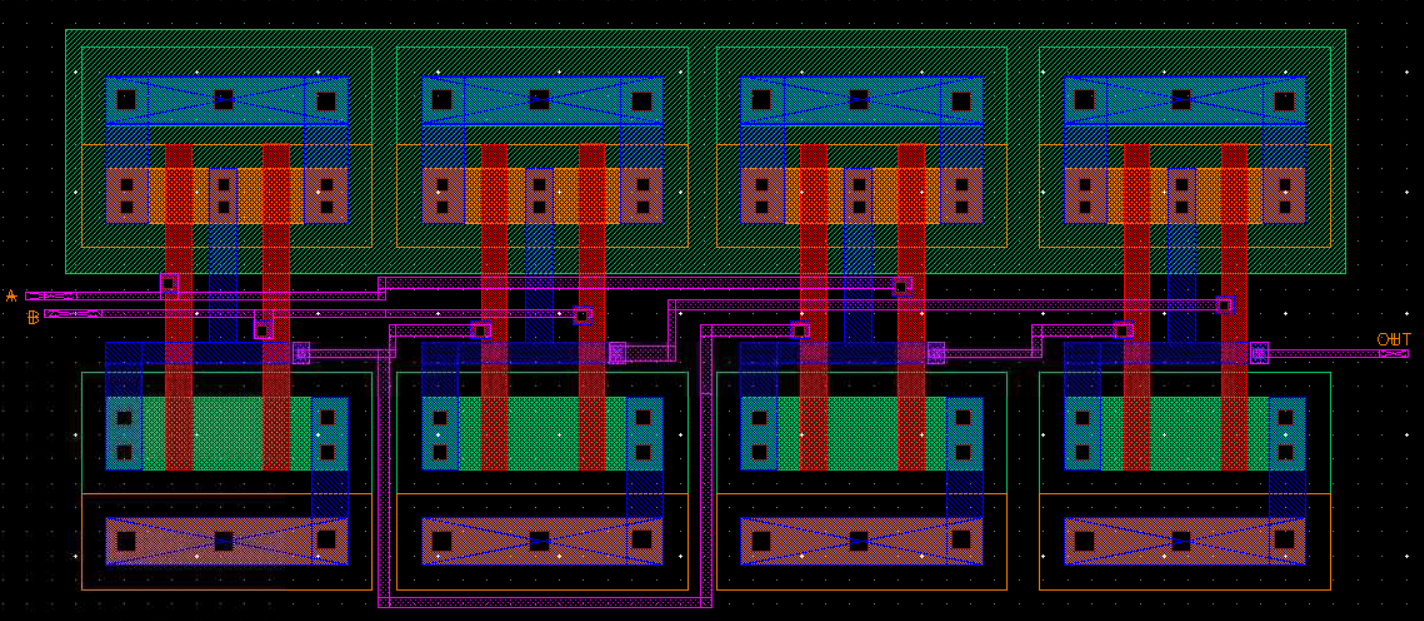
**Figure 10 -** AND Schematic

**Figure 11 -** AND Layout

The following figures show the symbol for our XOR gate, the transistor level circuit, and the layout, respectively.

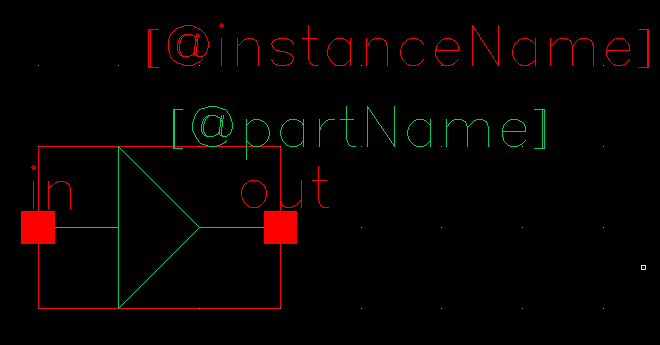
**Figure 12 –** XOR Symbol

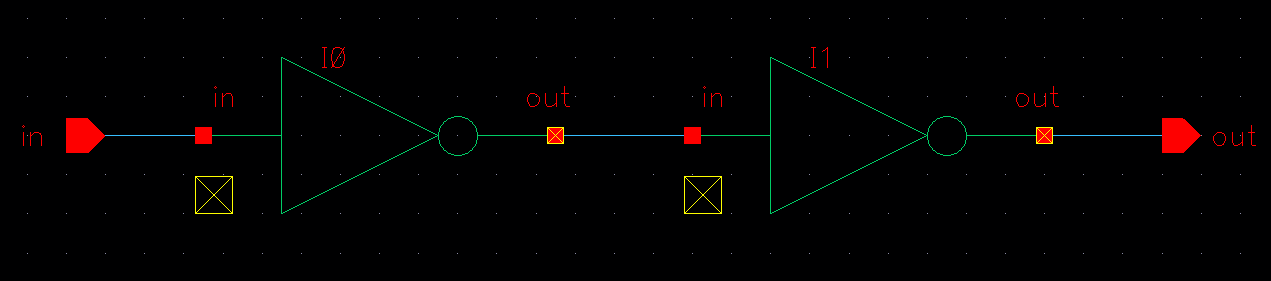
**Figure 13 – XOR Circuit Schematic**

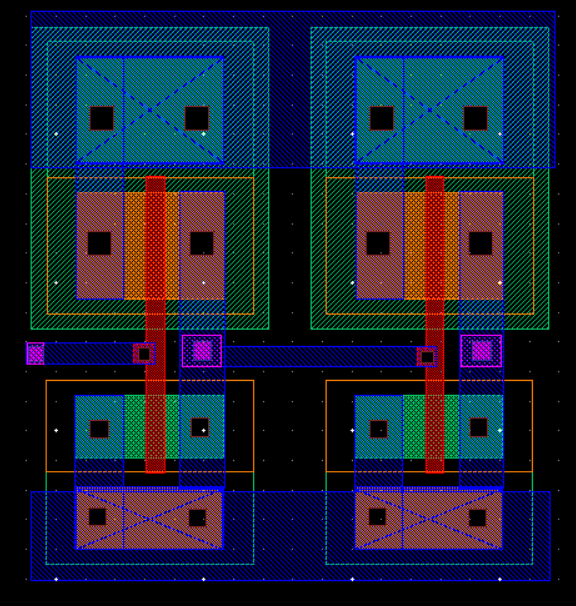


**Figure 14 -** XOR Layout

The Following figures shows the buffer symbol, the buffer schematic, and the layout which is used to stabilize signals by bring a signal, that may hover between 0 and 3.3V and bring it to the rails. .

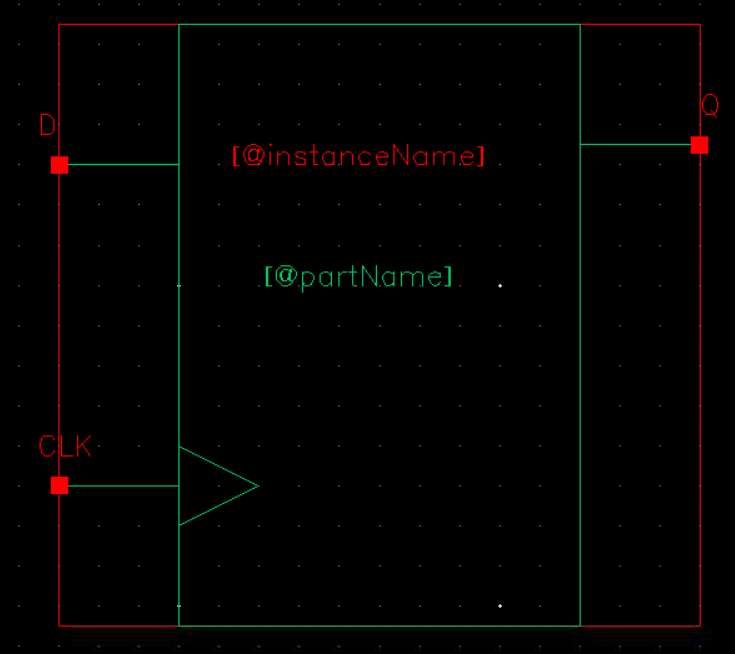
**Figure 15 –** Buffer Symbol

**Figure 16 –** Buffer Schematic

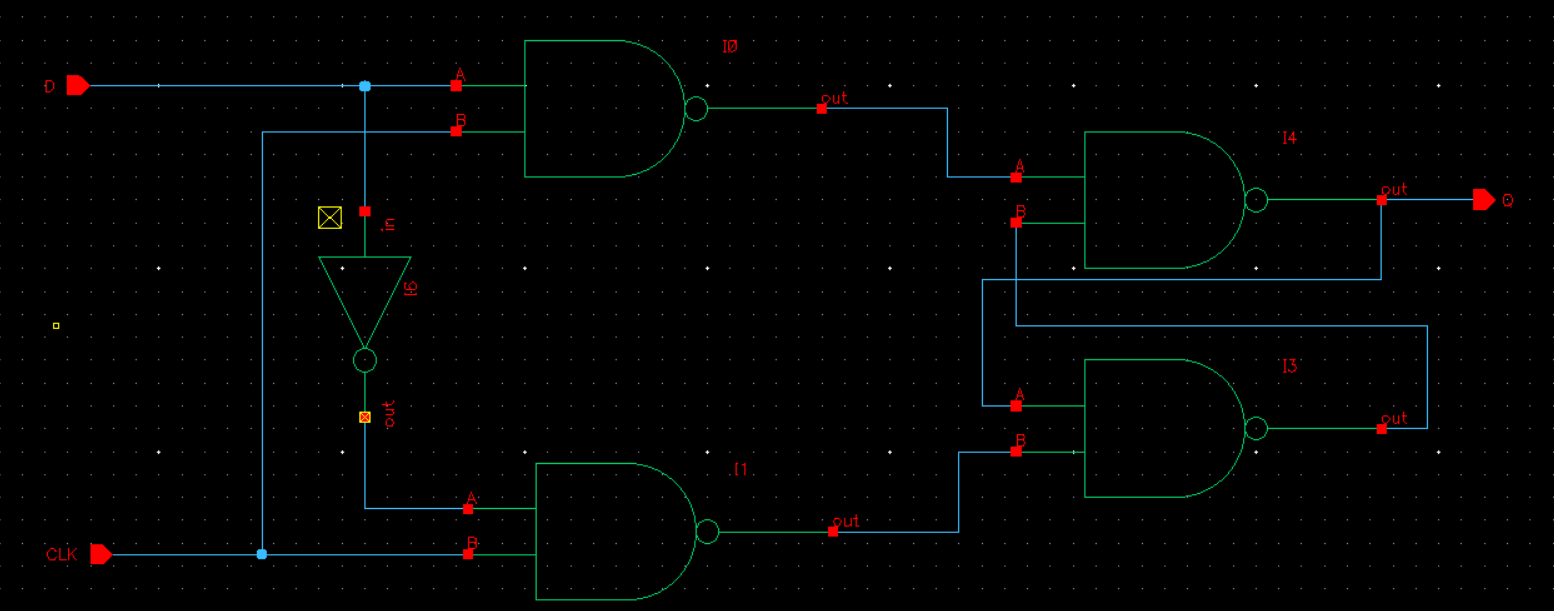


**Figure 17 -** Buffer Layout

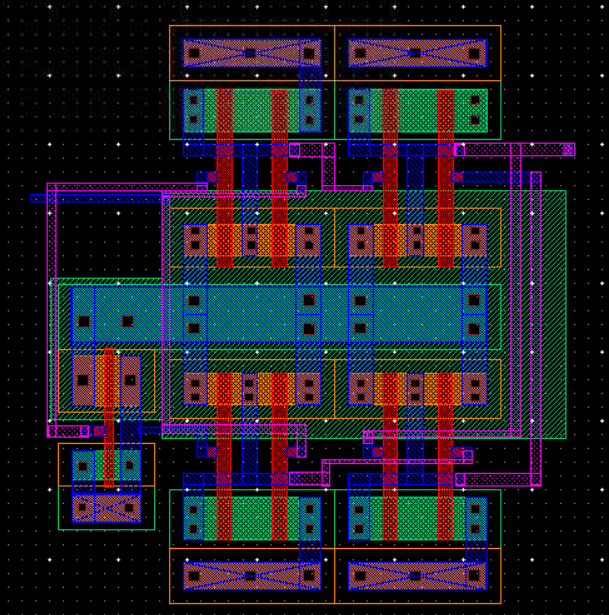
The following figure shows the symbol block, circuit schematic, and the layout of the D flip-flop (DFF).



**Figure 18 -** DFF Symbol

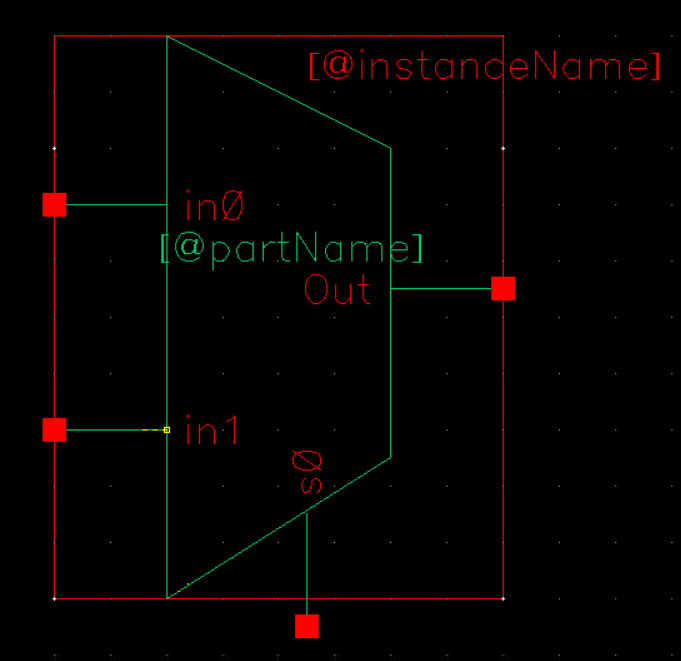


**Figure 19 -** DFF Schematic

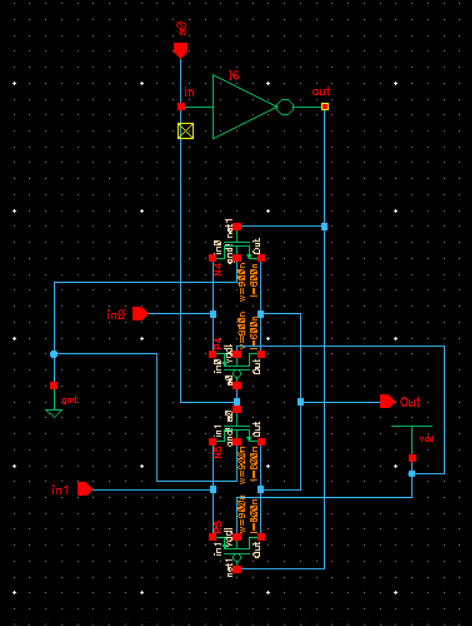


**Figure 20 -** DFF Layout

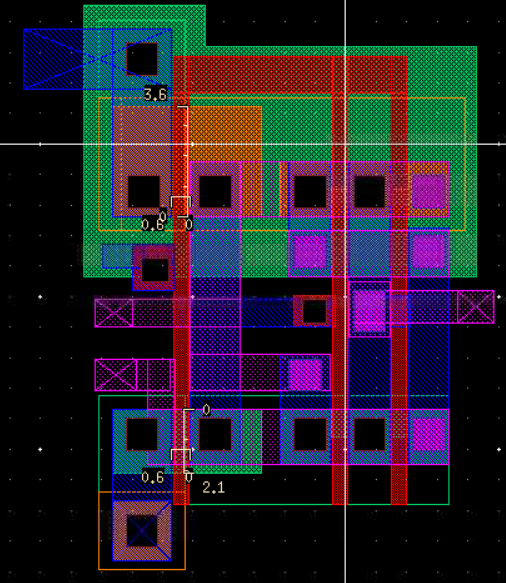
The following figures shows the symbol, schematic, and layout of our 2-to-1 MUX that is only used in our S-boxes.



**Figure 21 -** MUX Symbol

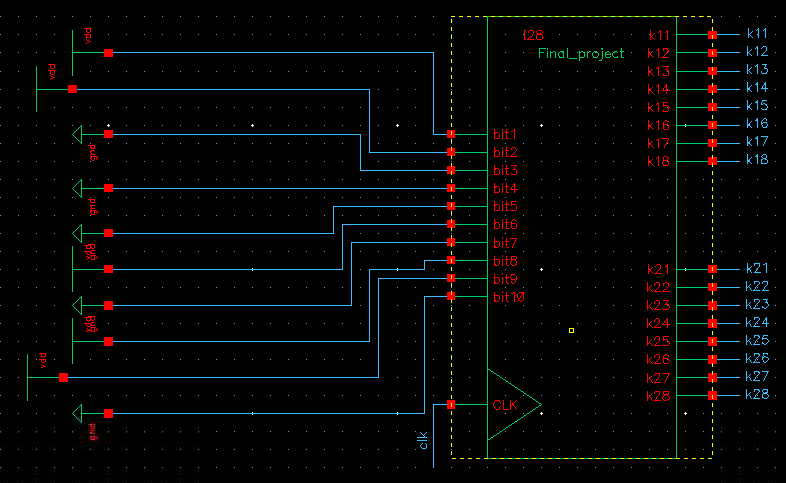


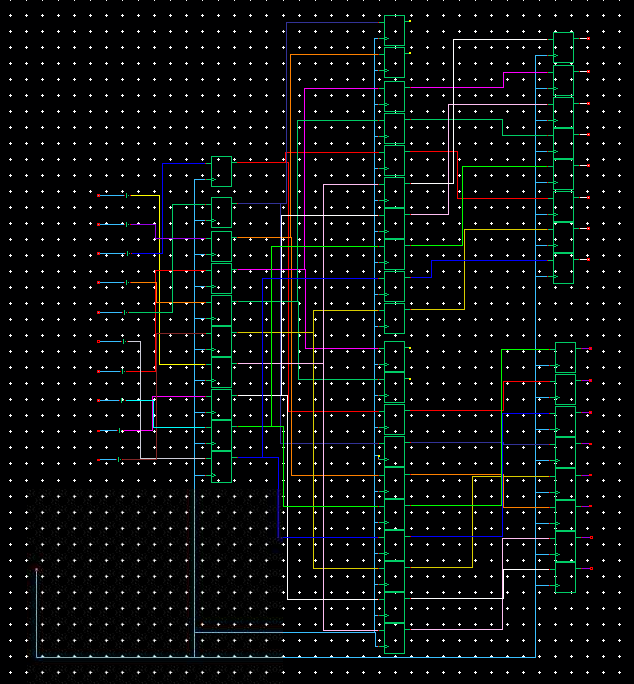
**Figure 22 -** MUX Schematic



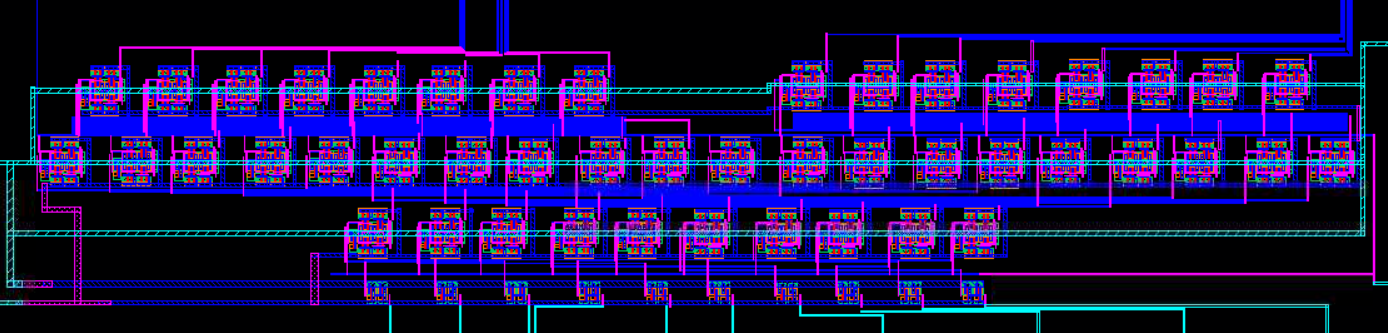
**Figure 23 -** MUX Layout

The following figures show the Key generation symbol block, the schematic, and the layout for the key generation circuit schematic. In the circuit schematic, we perform the P10 permutation. Then to get key 1, we circle shift the left 4 bits and then circle shift the right 4 bits then perform the P8 permutation to get key 1. To get key 2, we take the result from the P10 permutation and circle shift the left 4 bits three times and circle shift the right 4 bits three times. Then we perform the P8 permutation to get key2.

**Figure 24 –** Key Generation Symbol Block

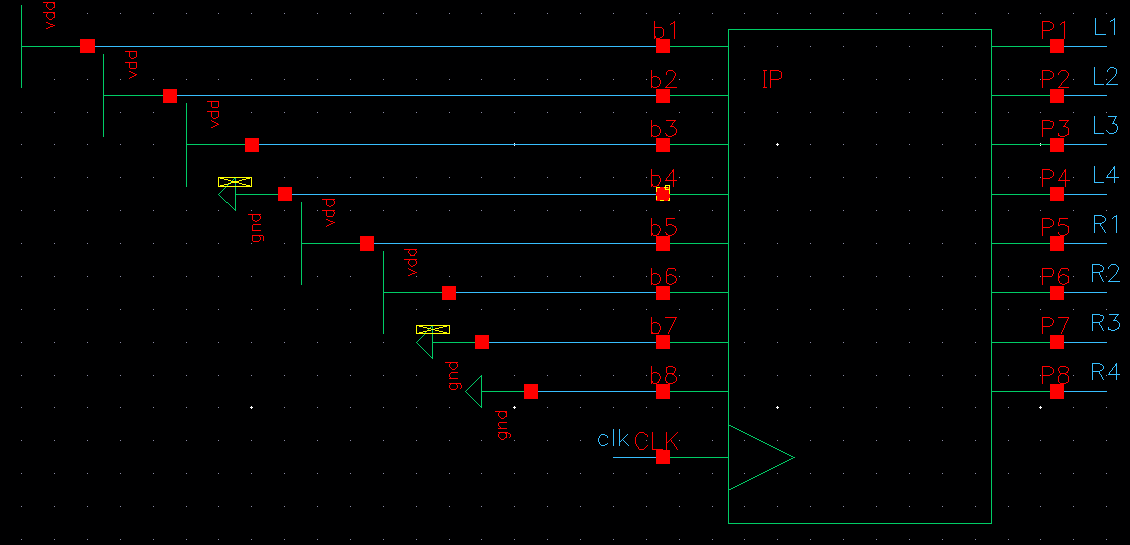


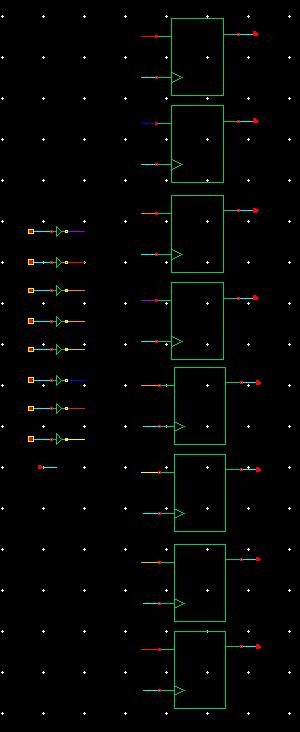
**Figure 25 –** Key Generation Circuit Schematic



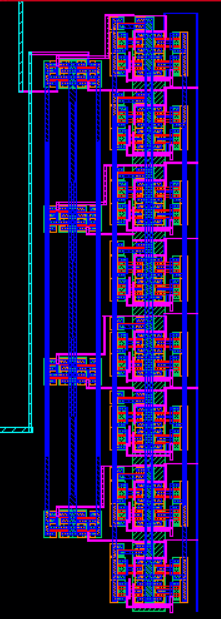
**Figure 26 - Key Generation Layout**

The following figure shows the IP block and the schematic, and the layout for the IP block. Here we perform the IP permutation, respectively.

**Figure 27 -** IP Block

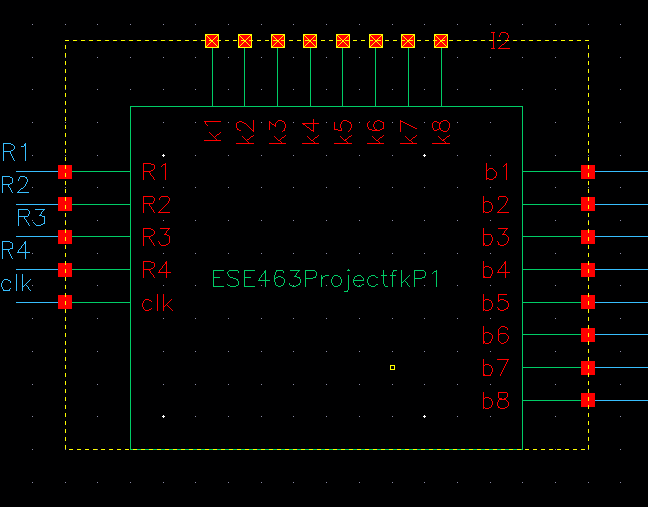


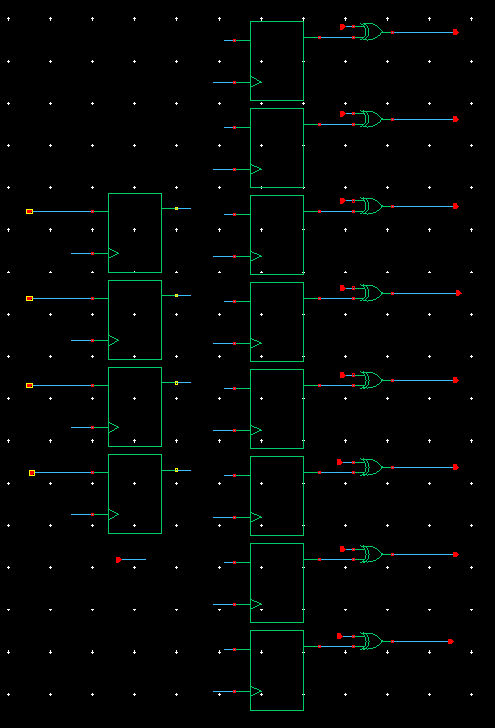
**Figure 28 -** The Schematic for the IP Block



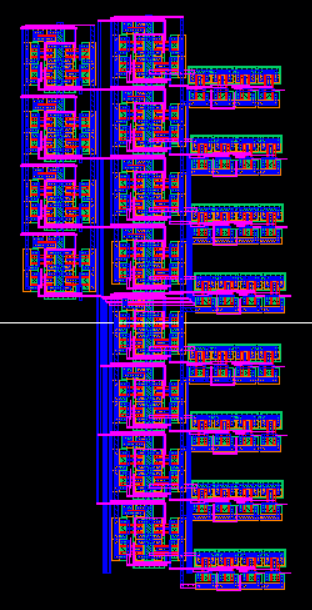
**Figure 29 -** IP Block Layout

The following figures shows the symbol, schematic, and layout for “ESE463ProjectfkP1” where we perform the expansion permutation “E/P” and then we XOR the result with key 1 if we are performing function fk1, or we XOR the result with key 2 if we are performing function fk2.

**Figure 30 -** ESE463ProjectfkP1

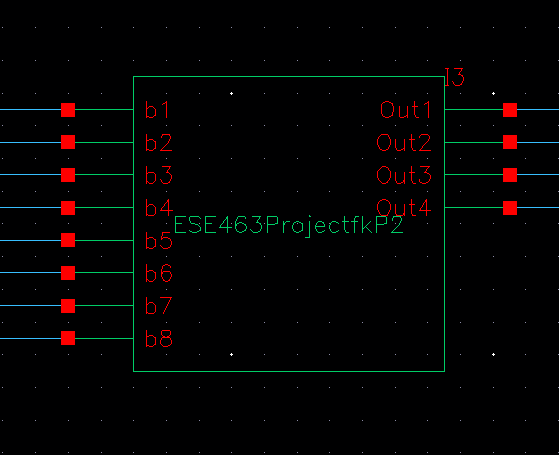


**Figure 31 -** Schematic of “ESE463ProjectfkP1” Block

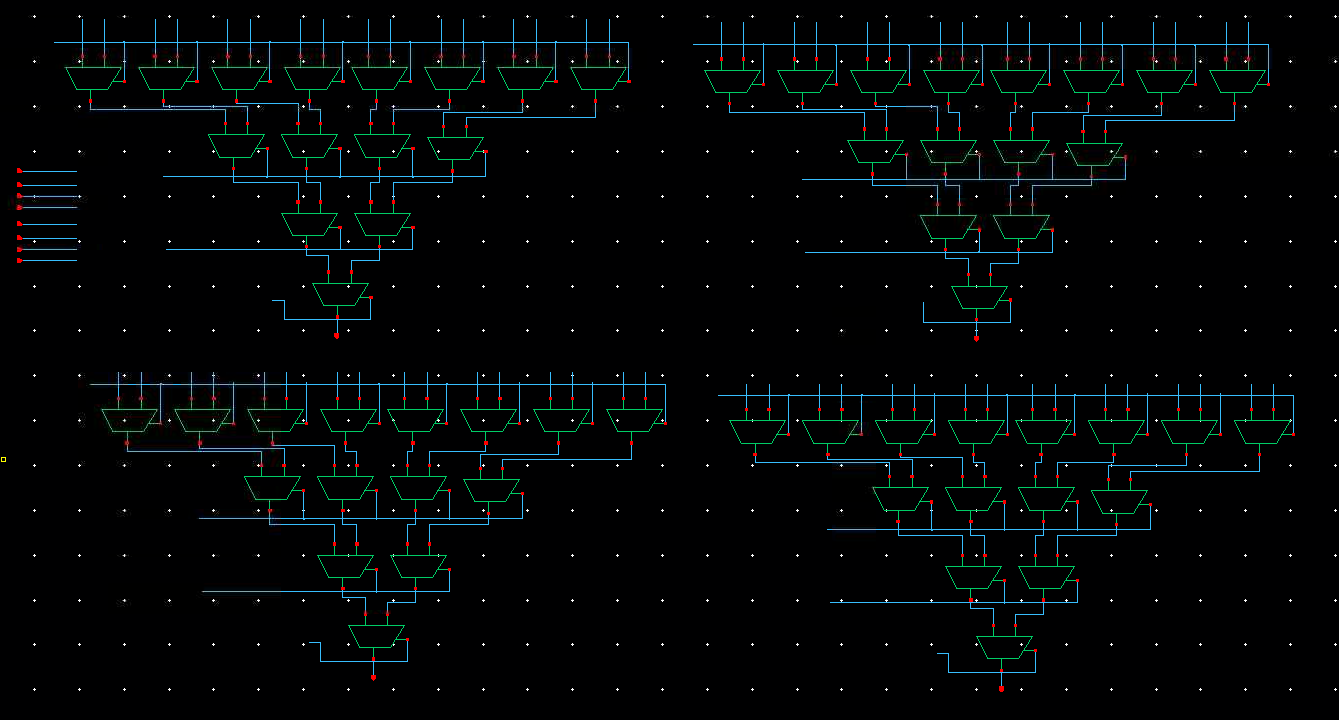


**Figure 32 -** ESE463ProjectfkP1 Layout

The following figures shows the symbol, schematic, and layout for “ESE463ProjectfkP2” which is the combinational logic for the SBOXes. The top two LUTs are the logic for SBOX1, where the left side outputs sbOut1 and the right outputs sbOut2. The bottom two LUTs are for SBOX2. The left LUT outputs sbOut3 and the right LUT outputs sbOut4. The result from “ESE463ProjectP2” is [sbOut1 sbOut2 sbOut3 sbOut4].



**Figure 33 -** ESE463ProjectfkP2



**Figure 34 –** Schematic for “ESE463ProjectfkP2” Block

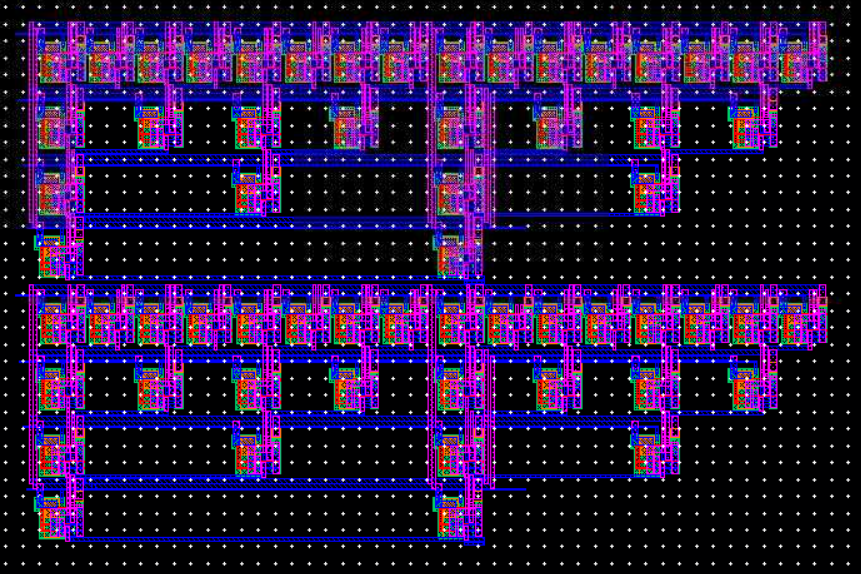
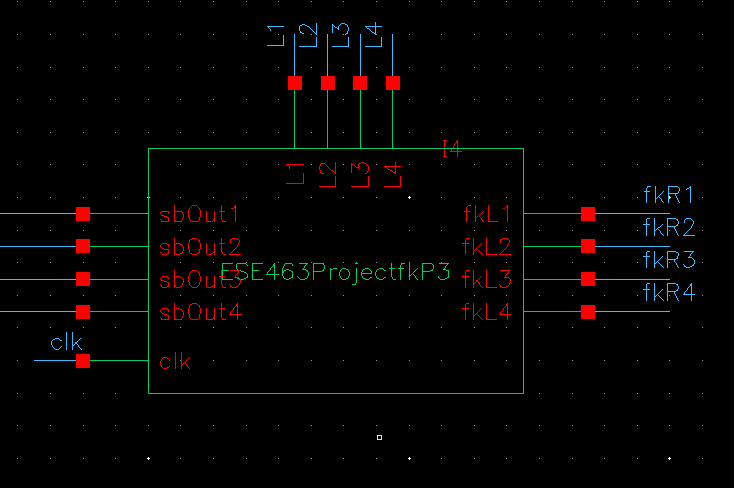
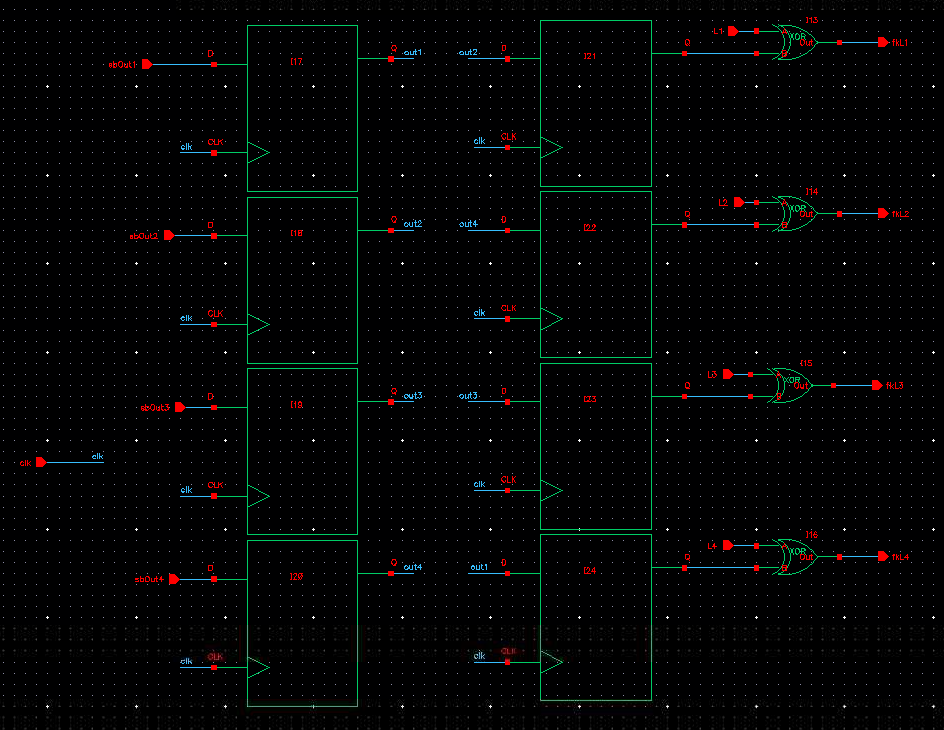


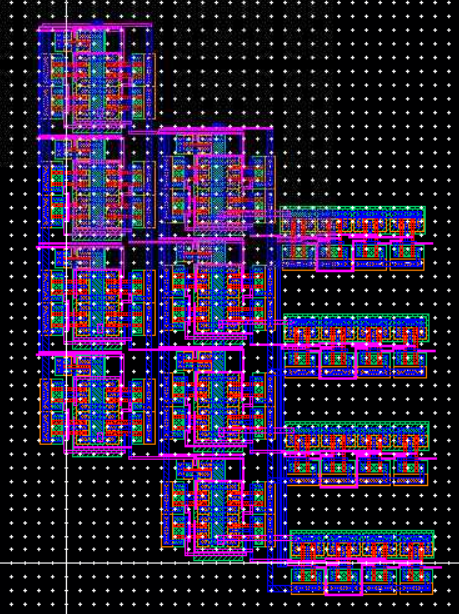
Figure 35 - ESE463ProjectfkP2 Layout

The following figures shows the symbol, schematic, and layout for “ESE463ProjectP3” where we perform permutation P4 and then XOR the result with L which is the Left 4 bits from the IP permutation. When we perform function fk2, we XOR the result of P4 with left 4 bits from function fk1.

**Figure 36 -** ESE463ProjectfkP3

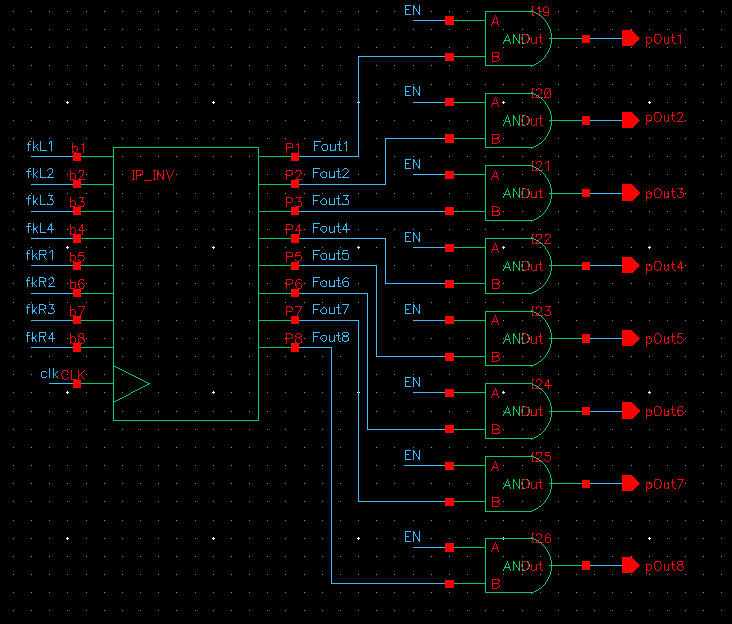


**Figure 37 -** Schematic for “ESE463ProjectP3” Block

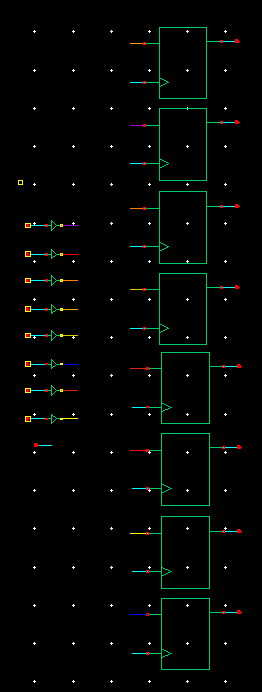


**Figure 38 -** ESE463ProjectfkP3 Layout

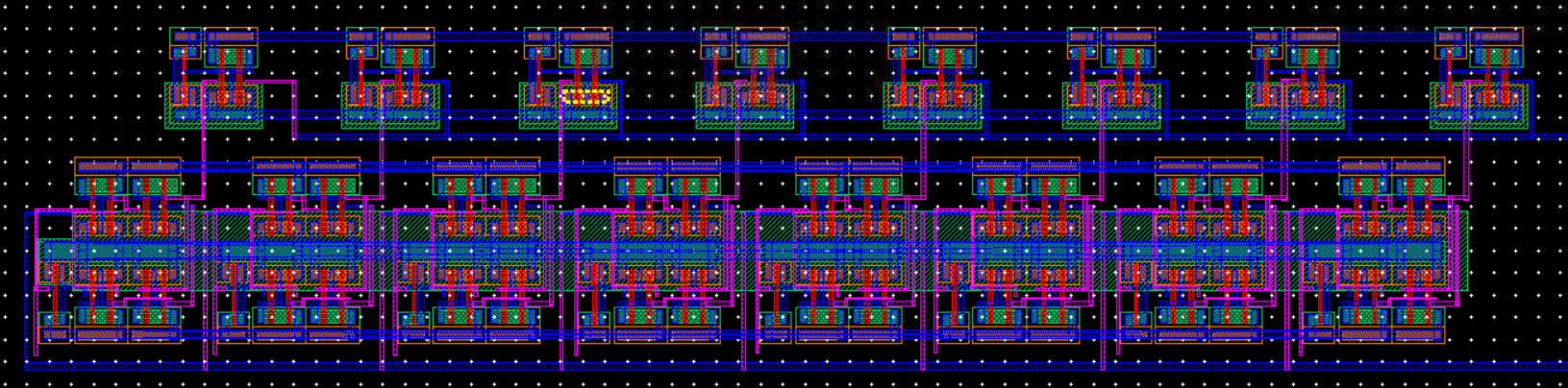
The following figures shows symbol, schematic, and layout of the IP inverse Block along with the enable functionality of our chip. POut1 – pOut8 are the outputs of our chip.



**Figure 39 -** IP Inverse and Enable Functionality of Chip



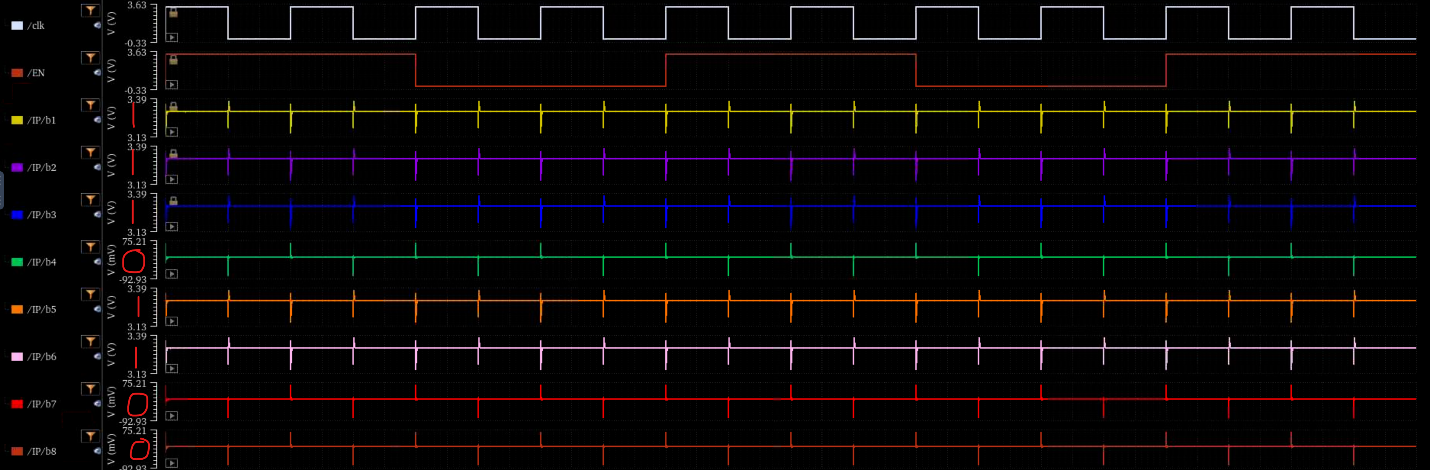
**Figure 40 -** IP Inverse Permutation Block Schematic

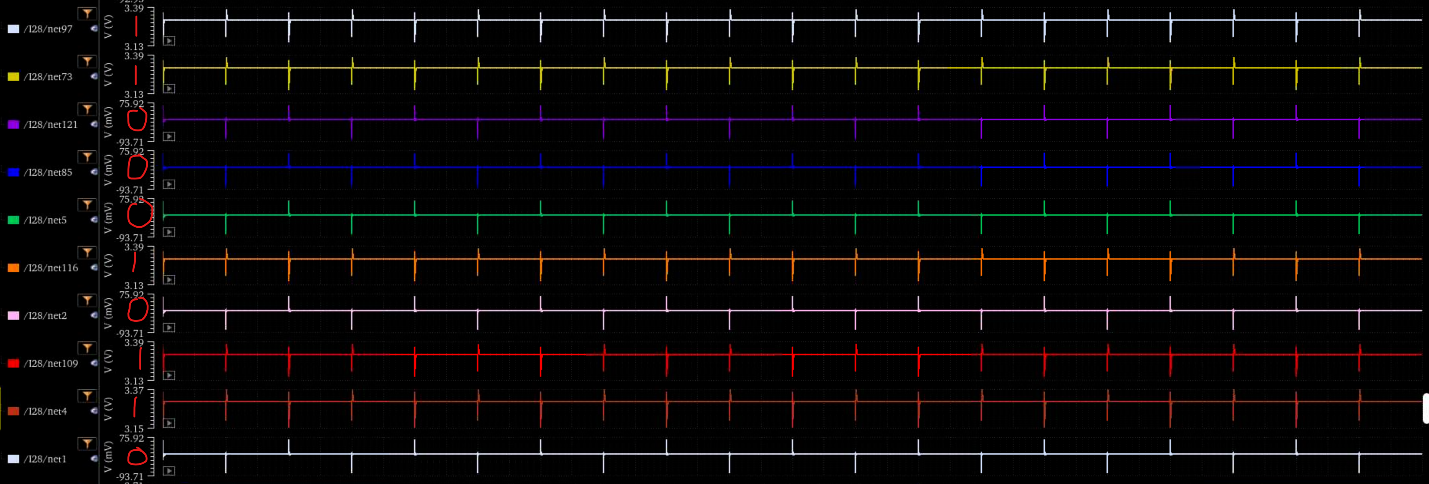


**Figure 41 -** IP Inverse Permutation Block Layout

**Circuit Timing:**

After creating our circuit, we did a transient simulation with a 10MHz clock and an enable (EN) signal with a frequency of 2.5MHz. The enable signal is on for 50% of each cycle.



**Figure 42 –** Circuit Timing Plots

Above we see the complete transient simulation with added comments on the picture for readability. In the first two plots we use the clock, EN, and our 8-bit input sequence (11101100). The follow plot shows us the input key (1100010110) which is consistent throughout the entire simulation. The last plot is the output (IP-1) which shows the output 11001011 which took around 25 ns to produce. It can be noted that there is some fluctuation in the beginning of the signal, and this is due to our overall system initializing. Once we have progressed to the end of our circuit, the output will be constant until our EN signal becomes low. Once the low EN is entered, the output signal is 00000000. Then when the EN becomes high again, the circuit produces our expected output.

**References**

[1] "Data Encryption Standard — Simplified DES." *Wikipedia: The Free Encyclopedia*, 11 Apr. 2025, <https://en.wikipedia.org/wiki/Data_Encryption_Standard#Simplified_DES>. Accessed 3 May 2025.