**Integrated Circuit Course Project Report**

*Yousef Ghanem [1], Asaad Halayqa [2]*

1,2 Student, Electrical & Computer Engineering Department, Birzeit University

Ramallah, Palestine

1172333 [1], 1172102 [2]

**1.**Introduction

A Digital Comparator is a combinational logic circuit that is used for comparison of two binary values. Basically, it generates the desired signal (either low or high) at the output when compares two digital values provided at its input.

Majorly a digital comparator is of two types:

1. Identity Comparator
2. Magnitude Comparator
   * 1-bit Magnitude Comparator
   * 2-bit Magnitude Comparator
   * n-bit Magnitude Comparator

**Identity Comparator**

A digital comparator that compares only the equality of two applied signals at its inputs is known as identity comparator. It has 2 input and only 1 output pin. The output pin shows a logic high signal when the two values are equal otherwise it shows a low signal.

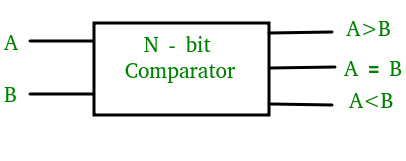
More specifically, we can say, for two inputs P and Q, if P = Q then output HIGH, and if P ≠ Q then output LOW.

**Magnitude Comparator**

Basically, a magnitude comparator makes the comparison by considering all the factors. It shows results for either greater, equal or lesser than value by comparing the magnitude of two inputs. Hence contains 3 output pins and accordingly, any one of the 3 output pins of a magnitude comparator becomes high.

Suppose P and Q are the two inputs of magnitude comparator. And the 3 outputs will be P > Q, P = Q and P < Q. And depending upon the comparison performed, any one of the given outputs will be high.

The figure 1 below represents the block diagram of a magnitude comparator having 2 inputs P and Q:



**Figure 1: N-bit Comparator**

# 2.Specifications

We built different types of comparators as a 1-bit,4-bit and 8-bit Comparators with different components, n-bit comparators depend on less bit comparators.

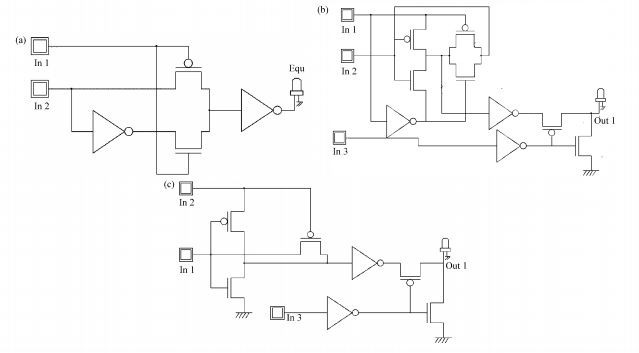
And here are the components we used to build the comparator:

* **PMOS** logic is a family of digital circuits based on p-channel, enhancement mode metal–oxide–semiconductor field-effect transistors. In the late 1960s and early 1970s, PMOS logic was the dominant semiconductor technology for large-scale integrated circuits before being superseded by NMOS and CMOS devices.
* **NMOS** (N-type metal-oxide-semiconductor) logic uses n-type (-) MOSFETs (metal-oxide-semiconductor field-effect transistors) to implement logic gates and other digital circuits. These NMOS transistors operate by creating an inversion layer in a p-type transistor body. This inversion layer, called the n-channel, can conduct electrons between n-type "source" and "drain" terminals. The n-channel is created by applying voltage to the third terminal, called the gate. Like other MOSFETs, NMOS transistors have four modes of operation: cut-off (or subthreshold), triode, saturation (sometimes called active), and velocity saturation.
* **AND** gate is a basic digital logic gate that implements logical conjunction - it behaves according to the truth table to the right. A HIGH output results only if all the inputs to the AND gate are HIGH. If none or not all inputs to the AND gate are HIGH, LOW output results.
* **OR** gate is a digital logic gate that implements logical disjunction – it behaves according to the truth table to the right. A HIGH output (1) results if one or both the inputs to the gate are HIGH (1). If neither input is high, a LOW output (0) results. In another sense, the function of OR effectively finds the maximum between two binary digits, just as the complementary AND function finds the minimum.
* **NOR** gate is a digital logic gate that implements logical NOR - it behaves according to the truth table to the right. A HIGH output results if both the inputs to the gate are LOW; if one or each input is HIGH, a LOW output results. NOR is the result of the negation of the OR operator.
* A logical inverter, sometimes called a **NOT** gate to differentiate it from other types of electronic inverter devices, has only one input. It reverses the logic state. If the input is 1, then the output is 0. If the input is 0, then the output is 1.

In the 4-bit Comparator, our Targeted area is below 1142µm2, as for power is below 21.7µW, for performance we targeted a delay below 104ps.

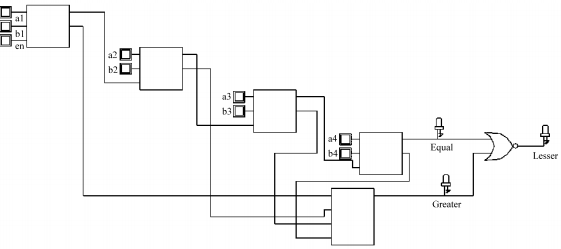
# 3. Background-others systems

In another System, three models were used for a 1-bit Comparator, all three models used the pass-transistor logic style which is popular design approach for the area aspect. All shown in figure 2a.

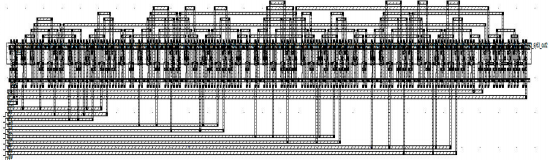


**Figure 2a: 3 models of 1-bit Comparator**

As in 4-bit, it was designed by cascading four 1-bit comparators and using 1 four input OR gate and NOR gate as in figure 2b and the output was obtained at the resulting LED, a layout diagram is shown in figure 2c.

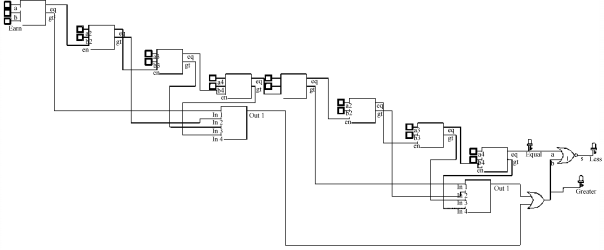


**Figure 2b: 4-bit Comparator Schematic**

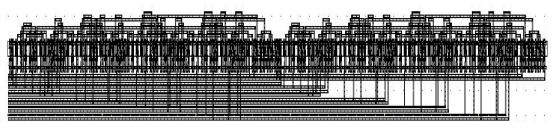


**Figure 2c: 4-bit Comparator layout**

Similarly, in 8-bit Comparator, it was designed by cascading 8 1-bit comparators it’s schematic is shown in figure 2d and layout is shown in figure 2e.



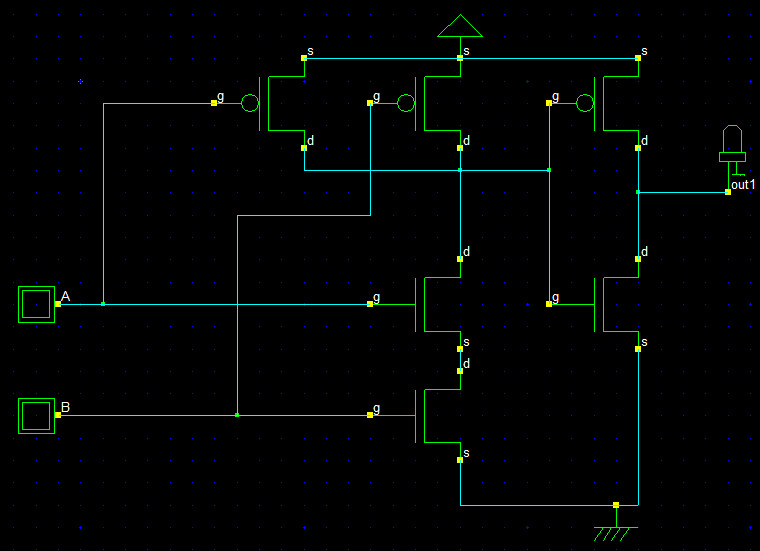
**Figure 2d: 8-bit Comparator Schematic**



**Figure 2e: 8-bit Comparator layout**

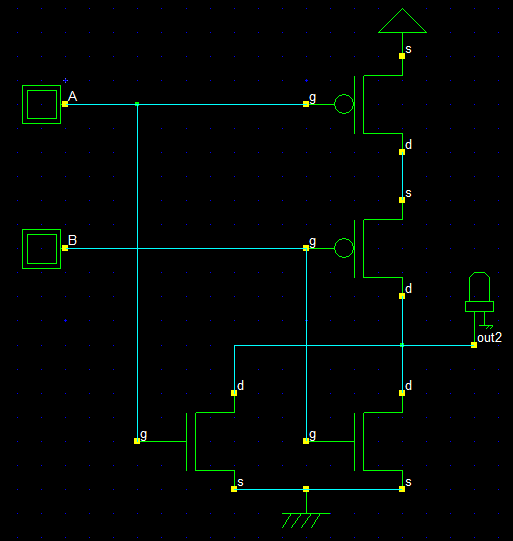
# 4. Proposed Design

In Our design strategy we used the 65n rule in all our schematics in DSCH as we tried different rules and it turned out to be the most power saving and smallest area with the best performance possible.

Initially, we designed AND gate using CMOS and PMOS gates, as shown in the figure 3a.

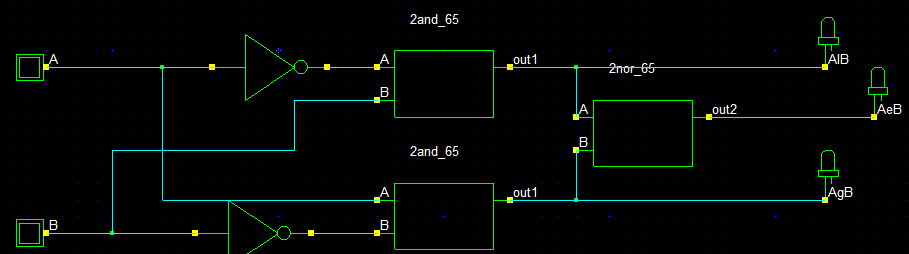
**Figure 3a: AND gate Schematic**

We designed NOR gate using CMOS and PMOS gates, as shown in the figure 3b.



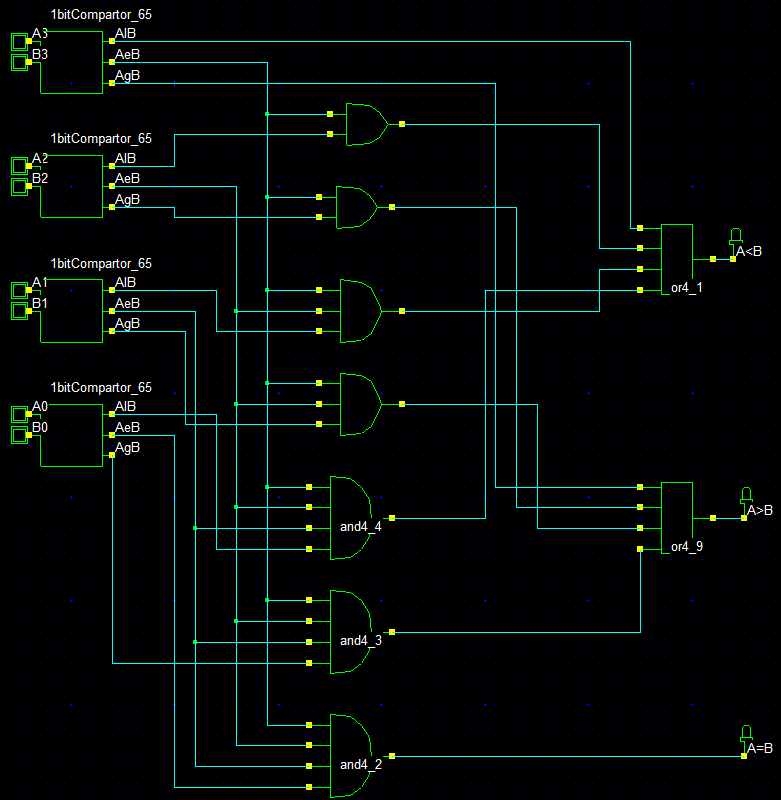
**Figure 3b: NOR gate Schematic**

After that, we designed a 1-bit comparator using the previous AND & NOR gates with two inverter gates, as shown in the figure 3c.



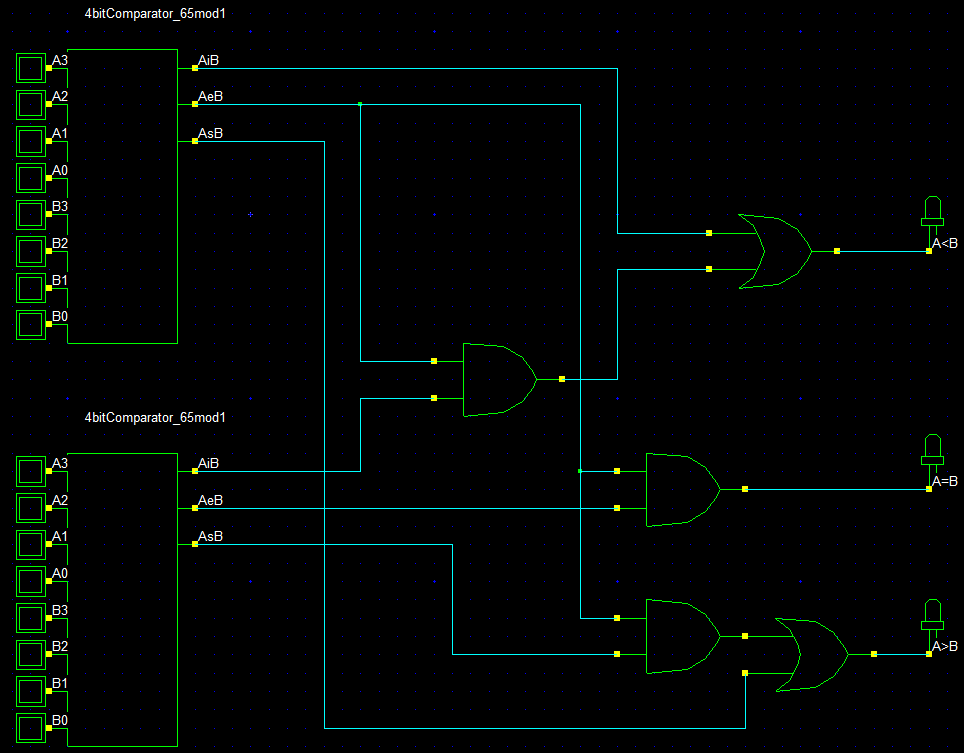
**Figure 3c: 1-bit Comparator Schematic**

Then we used the previous 1-bit comparator and some gates (AND & OR) to design a 4-bit comparator, as shown in the figure 3d.



**Figure 3d: 4-bit Comparator Schematic**

Finally, we designed an 8-bit comparator using the previous 4-bit comparator and some gates (AND & OR), as shown in the figure 3e.



**Figure 3e: 8-bit Comparator Schematic**

# 5. Testing strategy and results

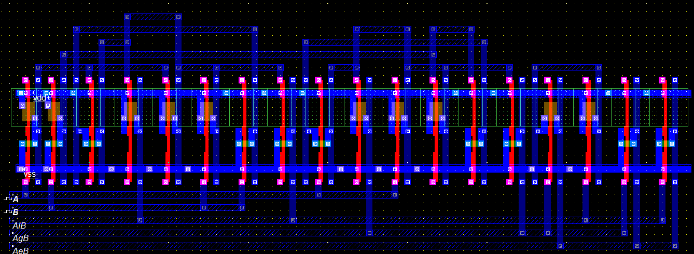
We tested our Comparator’s designs in Microwind as we drew a layout and simulated the results for each design and recorded the area, power and delay for each design.

## 5.1. 1-bit Comparator

First, we calculated the area for the 1-bit Comparator as:

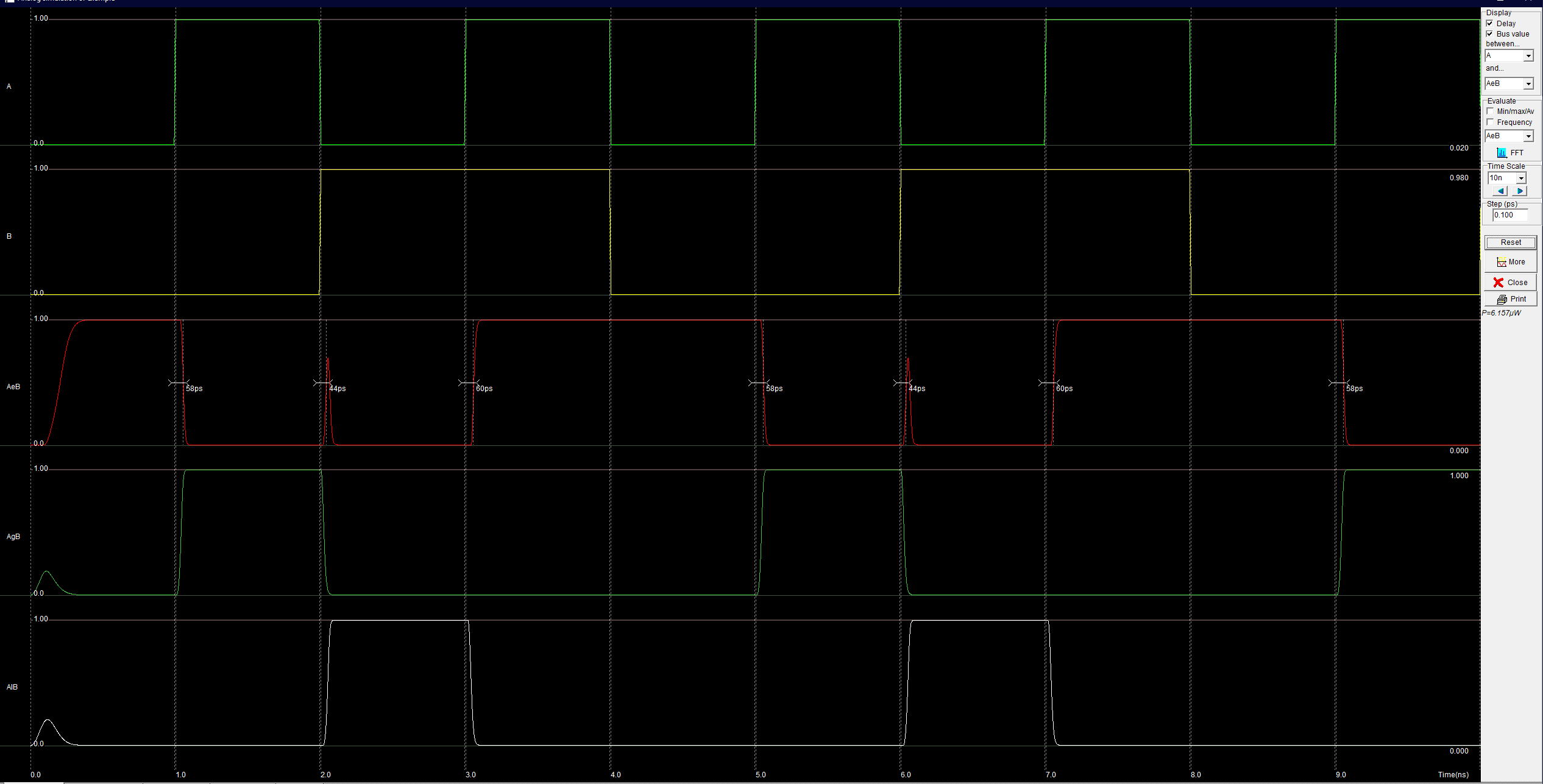
Layout width=16µm, height= 7µm. so, area = 112µm2

It’s shown in figure 4a.



**Figure 4a: 1-bit Comparator Layout**

The power is recorded as 6.15µW and max delay is 60ps as shown in figure 4b.



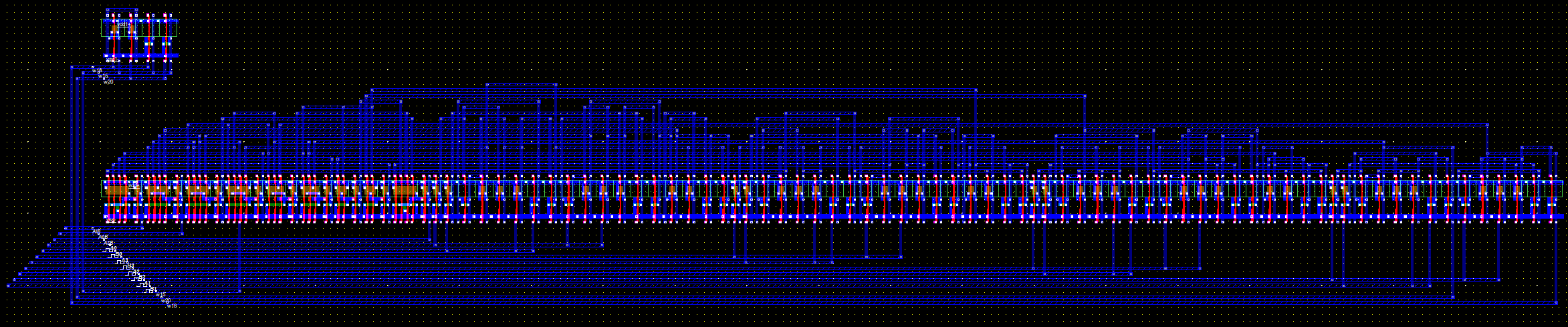
**Figure 4b: 1-bit Comparator simulation**

## 5.2. 4-bit Comparator

As before, we calculated the area for the 4-bit Comparator as:

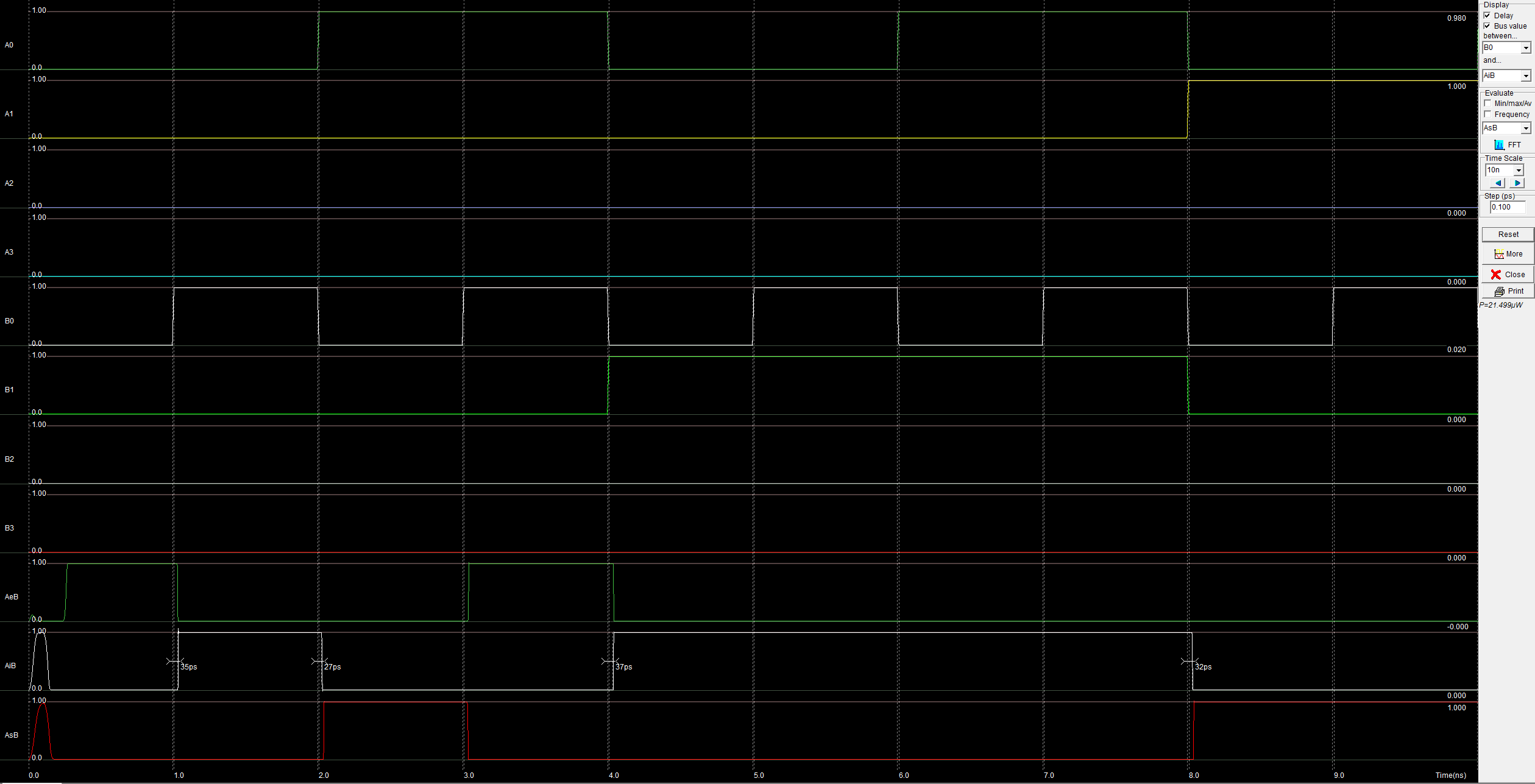
Layout width=76µm, height=14µm. so, area = 1064µm2

It’s shown in figure 4c.



**Figure 4c: 4-bit Comparator**

The power is recorded as 21.49µW and max delay is 40ps as shown in figure 4d.



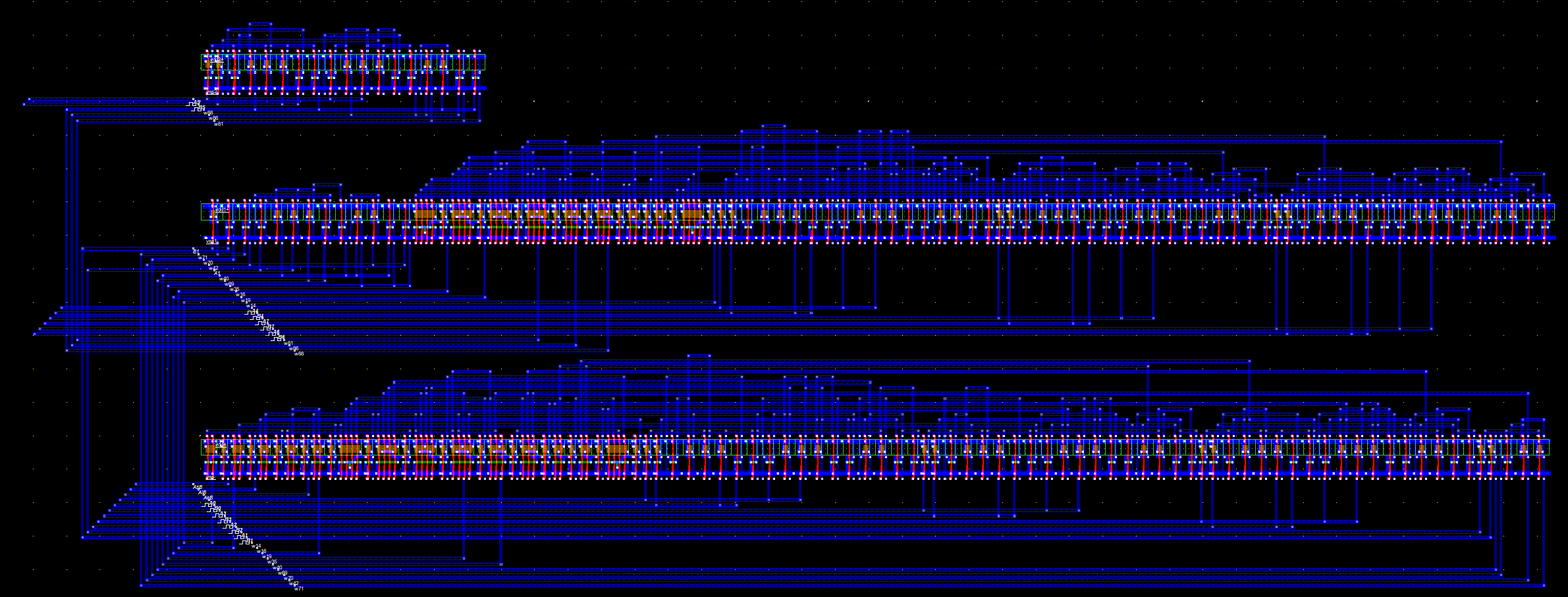
**Figure 4d: 4-bit Comparator Simulation**

## 5.3. 8-bit Comparator

we calculated the area for the 8-bit Comparator as:

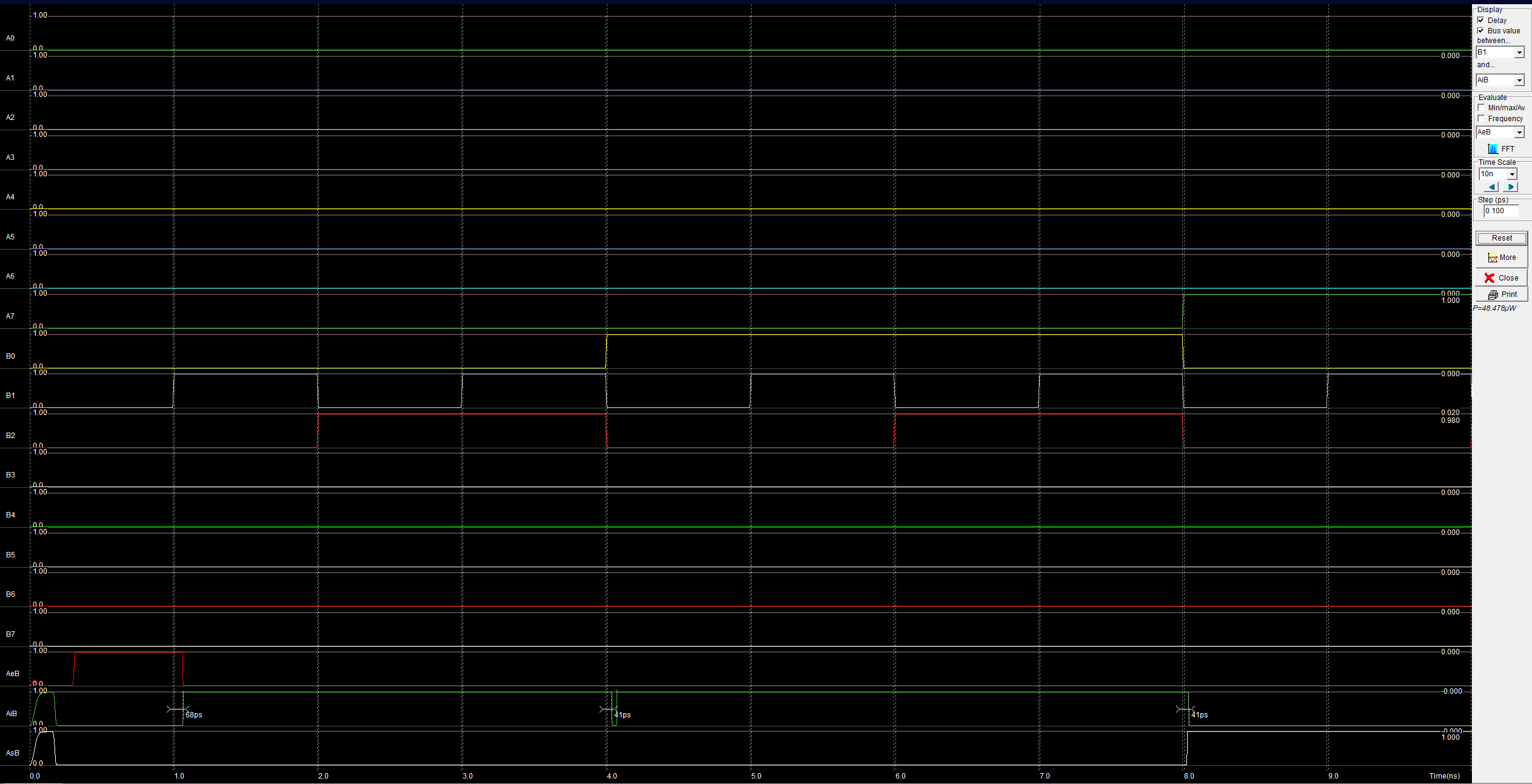
Layout width=80µm, height=29µm. so, area = 2320µm2

It’s shown in figure 4e.



**Figure 4e: 8-bit Comparator Layout**

The power is recorded as 48.47µW and max delay is 68ps as shown in figure 4f.



**Figure 4f: 8-bit Comparator Layout**

# 6. Optimization strategy and results

First, we tried different rules and recorded the area, power and performance of each at different designs and we found out that the most optimized and efficient rule is 65n in all the aspects.

Also, we designed multiple designs for both 1-bit and 4-bit Comparators as we tried in the 1-bit a design that contains of NMOS and PMOS and a design of self-designed AND-gate and NOR-gate and took the second one as it had less area and more power efficiency. In 4-bit Comparator we built it using the best 1-bit comparator design we built, and we tried two equivalent designs to see which is more efficient (AND – OR), (NAND – NAND) designs. The first design turned out to be the most efficient in power and area aspects.

In compare with the background design, we talked about before, we got less area, power and better performance than that model in 4-bit version in table (1) as follows:

|  |  |  |  |
| --- | --- | --- | --- |
|  | Area | Power | Speed (Delay) |
| Background system | 1142µm2 | 21.77µW | 104ps |
| Our design | 1064µm2 | 21.49µW | 40ps |

**Table (1)**

# 7. Conclusion

To summarize, after finishing this task we have acquired good knowledge about Designing, simulating digital circuits using Microwind, and DSCH, from implementing 1-bit,4-bit and 8-bit Comparators.

Also, we learned to optimize and find better solutions to an existing circuit and how to improve it in many aspects.

# 8. References

1. https://electronicscoach.com/digital-comparator.html
2. https://whatis.techtarget.com/definition/logic-gate-AND-OR-XOR-NOT-NAND-NOR-and-XNOR
3. https://en.wikipedia.org/wiki/NOR\_gate
4. https://en.wikipedia.org/wiki/NMOS\_logic
5. https://en.wikipedia.org/wiki/PMOS\_logic