

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN
KHOA KỸ THUẬT MÁY TÍNH



BÁO CÁO THỰC HÀNH LAB3
THIẾT KẾ MẠCH TỔ HỢP TÍNH TOÁN
MÔN CE118 – THIẾT KẾ LUẬN LÝ SỐ
Giảng viên hd: Ths. HỒ NGỌC DIỄM

Sinh viên thực hiện :

Nguyễn Đình Anh. MSSV: 23520057

Nguyễn Hoàng Quốc Cường. MSSV: 23520200

Lớp CE118.P21

NỘI DUNG THỰC HÀNH

1. Thiết kế và mô phỏng một ALU có 2 toán hạng (4 bit) và các phép toán cộng, cộng 1, trừ, trừ 1, and, or, nand, nor.

- Vì ALU thực hiện 8 phép toán bao gồm hai loại phép toán đó là phép luận lí và phép số học , nên ta thiết kế chia thành hai khối lớn chính với opcode được xác định như sau:

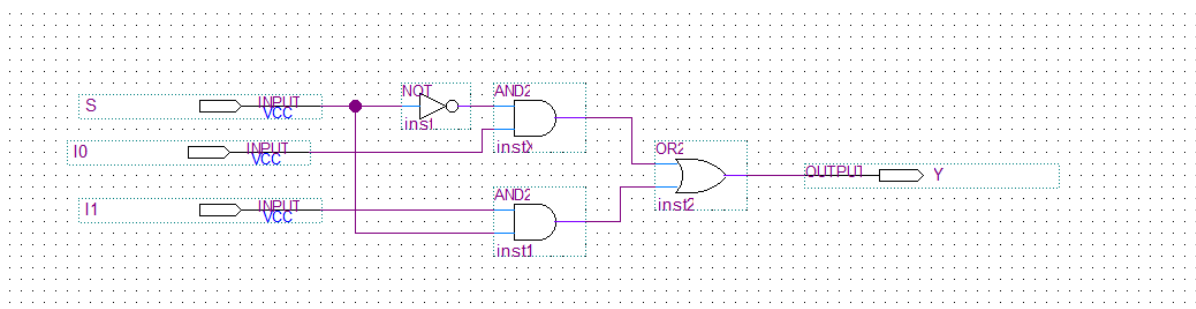
Opcode[2]	Opcode[1:0]	Phép toán
0	00	A+B
0	01	A+1
0	10	A-B
0	11	A-1
1	00	A&B
1	01	A B
1	10	A NAND B
1	11	A XOR B

Hình 1.

2. Các khối cần thiết kế và đóng gói.

a. Thiết kế bộ AU

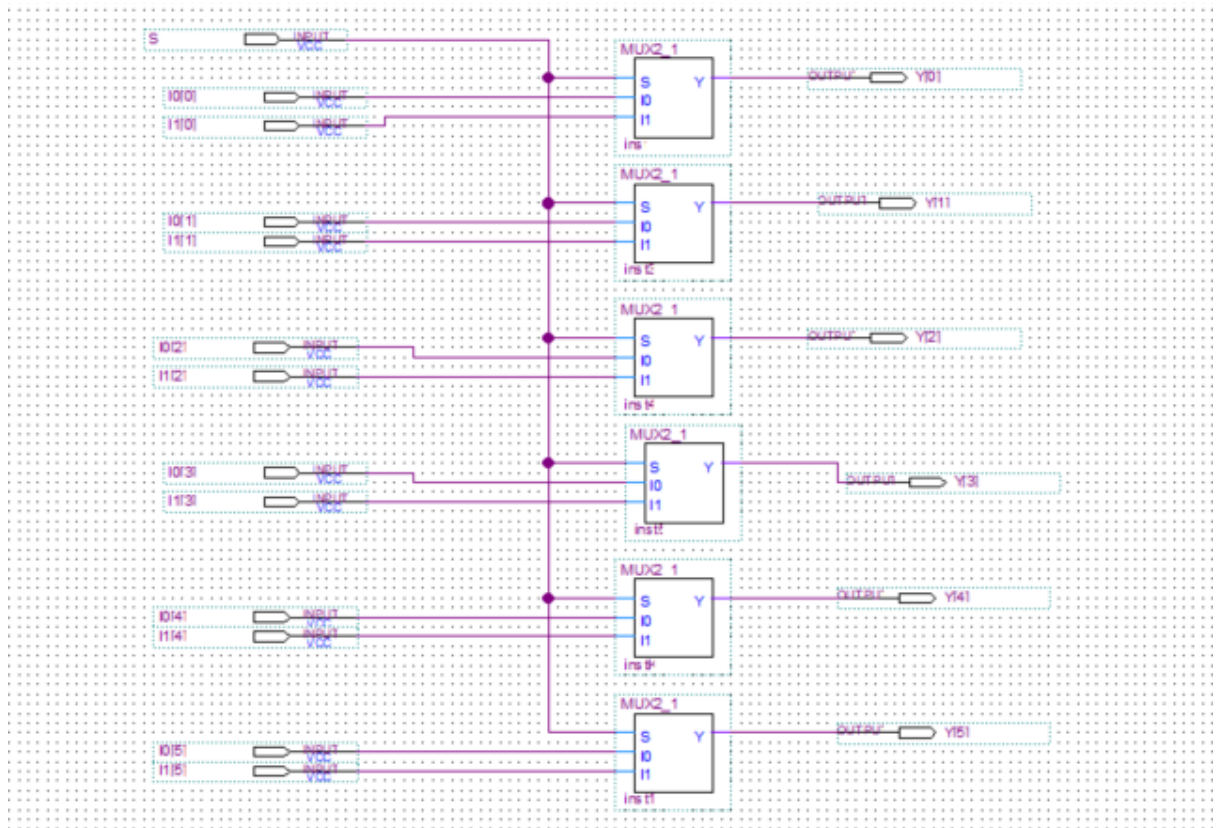
- Thiết kế bộ MUX4_1 4 bit : ta có thể sử dụng ba bộ MUX 2_1 4 bit.
- Trước tiên để thiết kế được bộ MUX2_1 4 bit thì cần thiết kế MUX2_1 1 bit rồi ghép nối song song 4 bộ lại với nhau.



Hình 2. MUX2_1 một bit

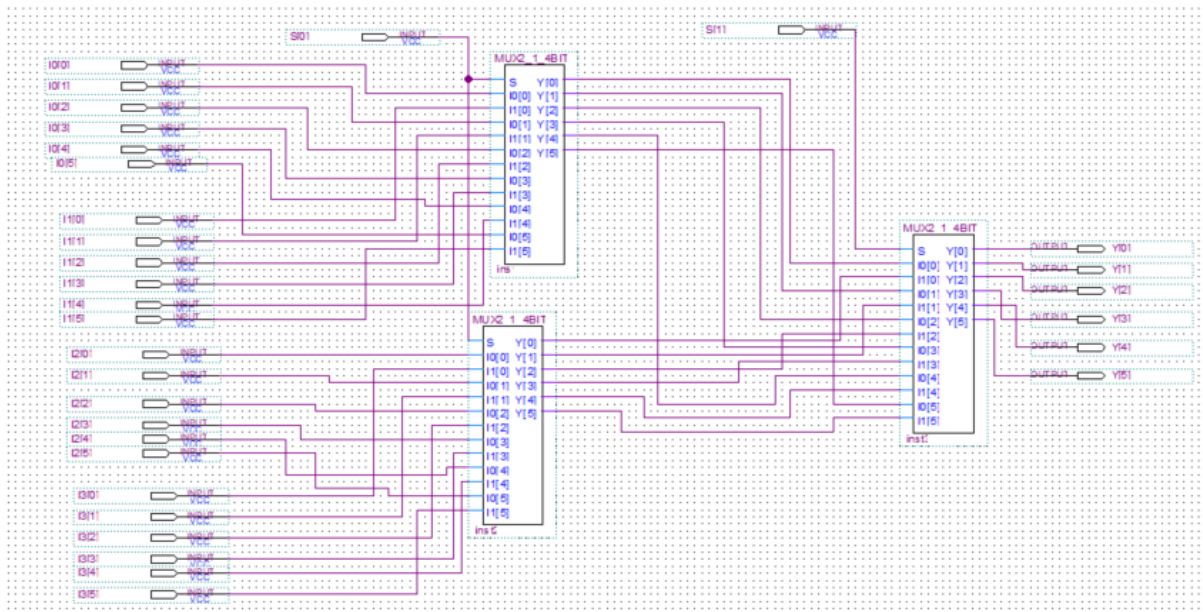
- Vì đối với phép cộng hoặc trừ chúng ta cần phải biểu diễn nó trong 5 bit với 1 bit dấu và 4 bit giá trị, ngoài ra thiết kế thêm một ngõ **over** để xác

định tràn, vậy tổng cộng ngõ ra cần 6 bit, do đó cần thiết kế bộ mux 6 bit.



Hình 3. MUX 2_1 sáu bit

- Vì mỗi khối có 4 phép tính toán nên ta cần thiết kế MUX4_1 sáu bit. Ta có thể thiết kế thông qua việc sử dụng MUX2_1 sáu bit (do trong quá trình đóng gói thì đặt tên khối là mux2_1 4 bit nhưng thực chất có 6 bit)



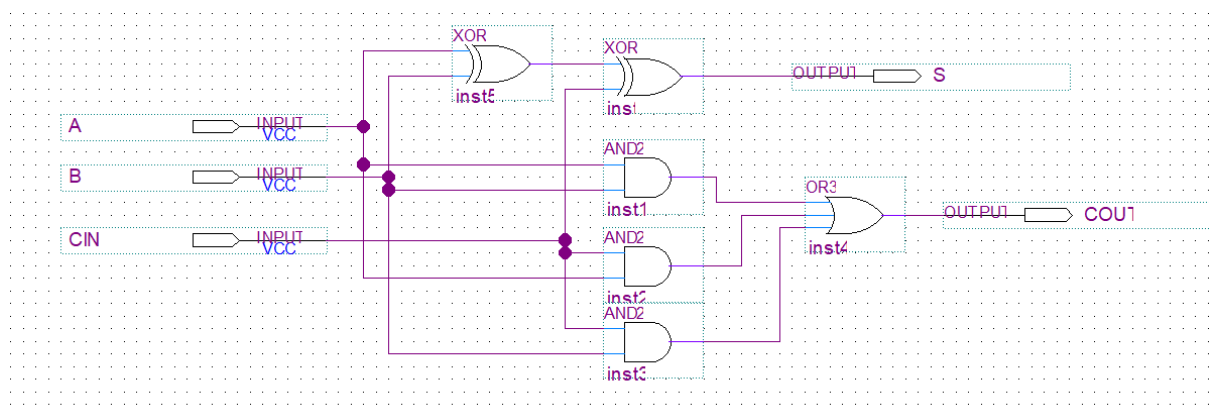
Hình 4. MUX4_1 sáu bit.

- Để thực hiện phép cộng, trước tiên ta thiết kế bộ cộng FULL ADDER 1 bit trước rồi sau đó ghép 4 bộ cộng lại với nhau để thành bộ cộng 4 bit. Ta có bảng chân trị:

CIN	A	B	S	COUT+1
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	0

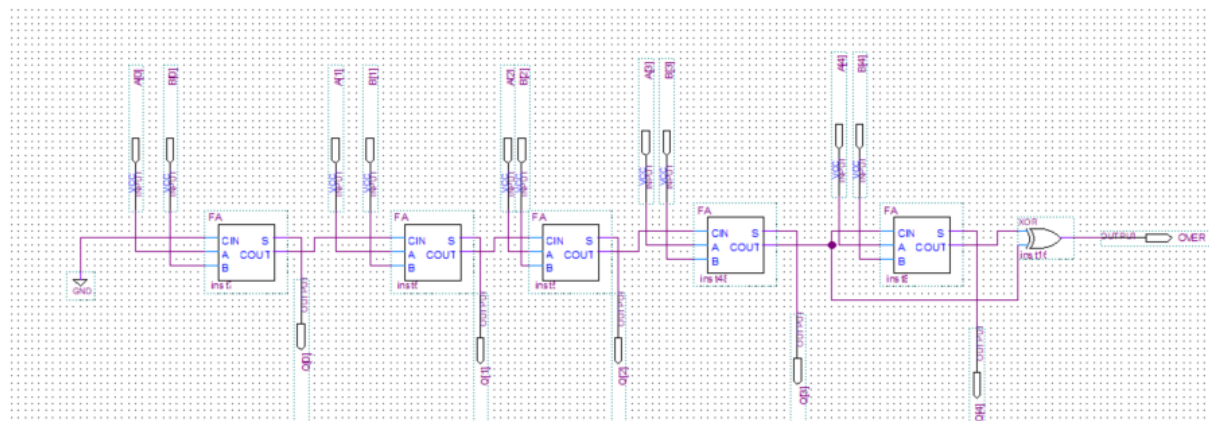
→ $S = A \text{ XOR } B \text{ XOR } \text{CIN}$

→ $\text{COUT} = AB + ACIN + BCIN$



Hình 6. Full adder 1 bit

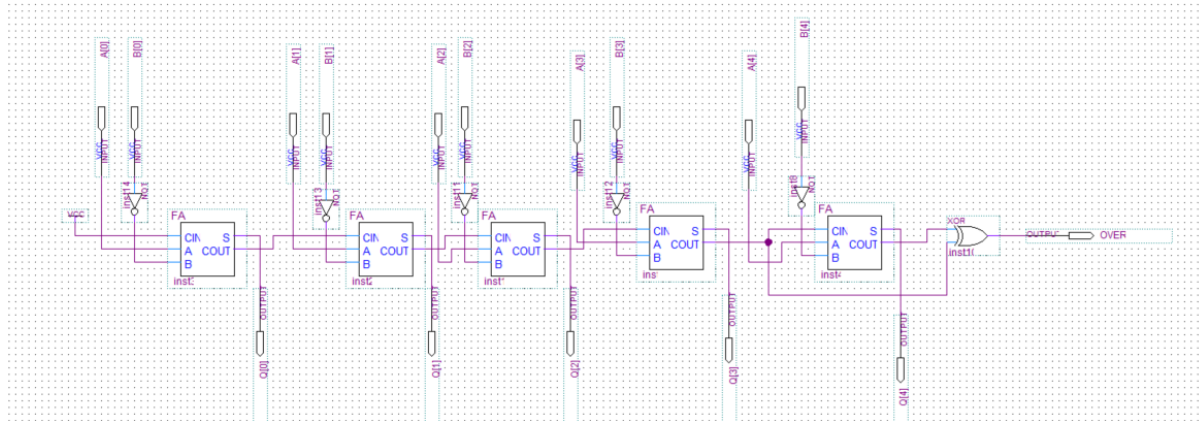
- Ngõ ra có 6 bit trong đó 5 bit thấp là biểu diễn kết quả của phép cộng, còn 1 bit cao còn lại là cờ báo tràn **over**



Hình 7. Adder 5 bit (Có chứa 1 bit dấu) .

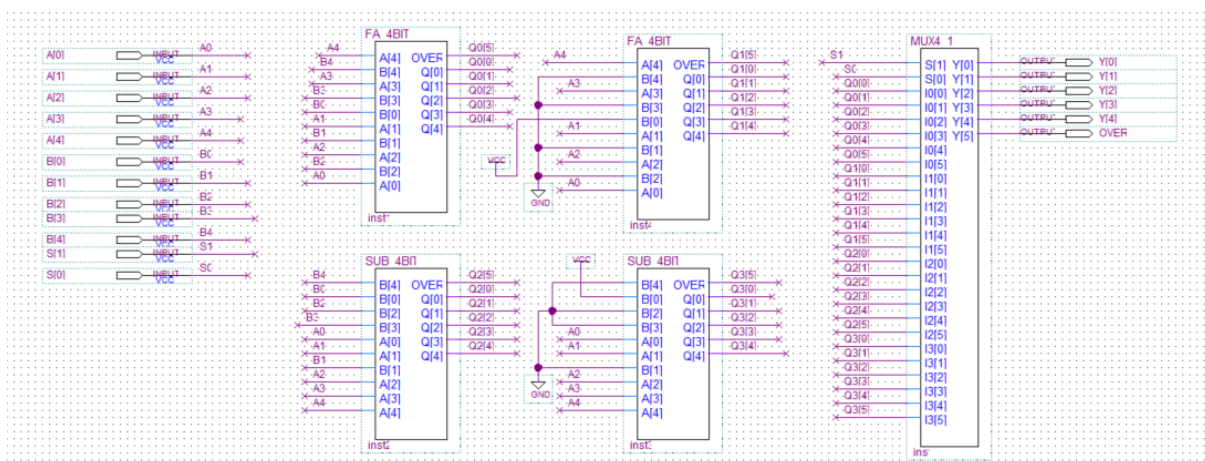
- Đối với phép trừ, ta thực hiện phép cộng với số bù 2, ví dụ $A - B = A + (-B) = A + B' + 1$.

Cũng tương tự như bộ cộng, bộ trừ với 6 ngõ ra, trong đó 5 bit thấp thể hiện kết quả của phép trừ, còn 1 bit cao thể hiện cờ báo tràn **OVER**



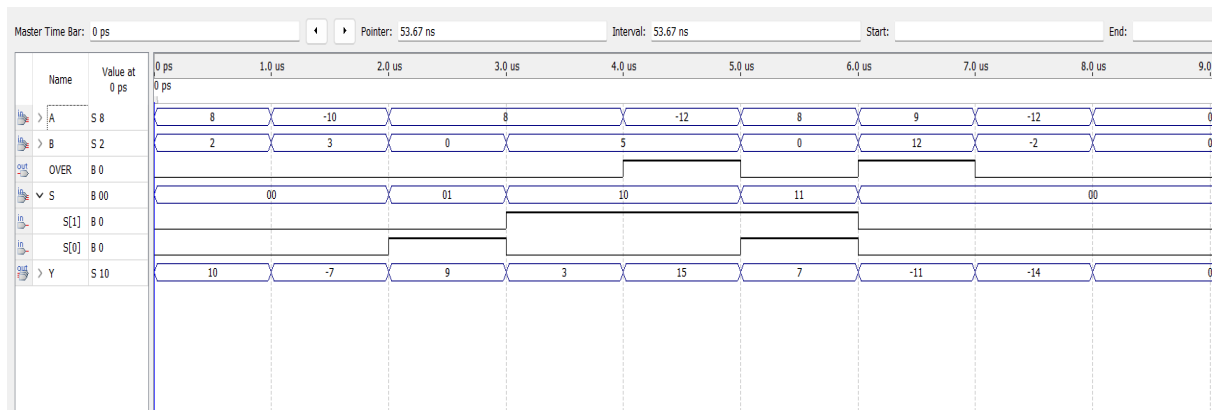
Hình 8. Bộ trừ

- Đối với bộ cộng 1 và trừ, thì ta chỉ cần sử dụng lại bộ cộng và trừ nhưng ở đầu toán hạng thứ hai ta đặt giá trị là 1.
- Sau khi đã đóng gói các mạch cần thiết, ta tiến hành thiết kế bộ AU với các thành phần là khối MUX4_1 sáu bit để chọn lựa phép toán, bộ cộng, bộ trừ



Hình 8. Bộ AU

- Opcode của các phép toán là $[1:0] s = 00 \rightarrow A+B$, $S = 01 \rightarrow A+1$, $S = 10 \rightarrow A - B$, $S = 11 \rightarrow A-1$;

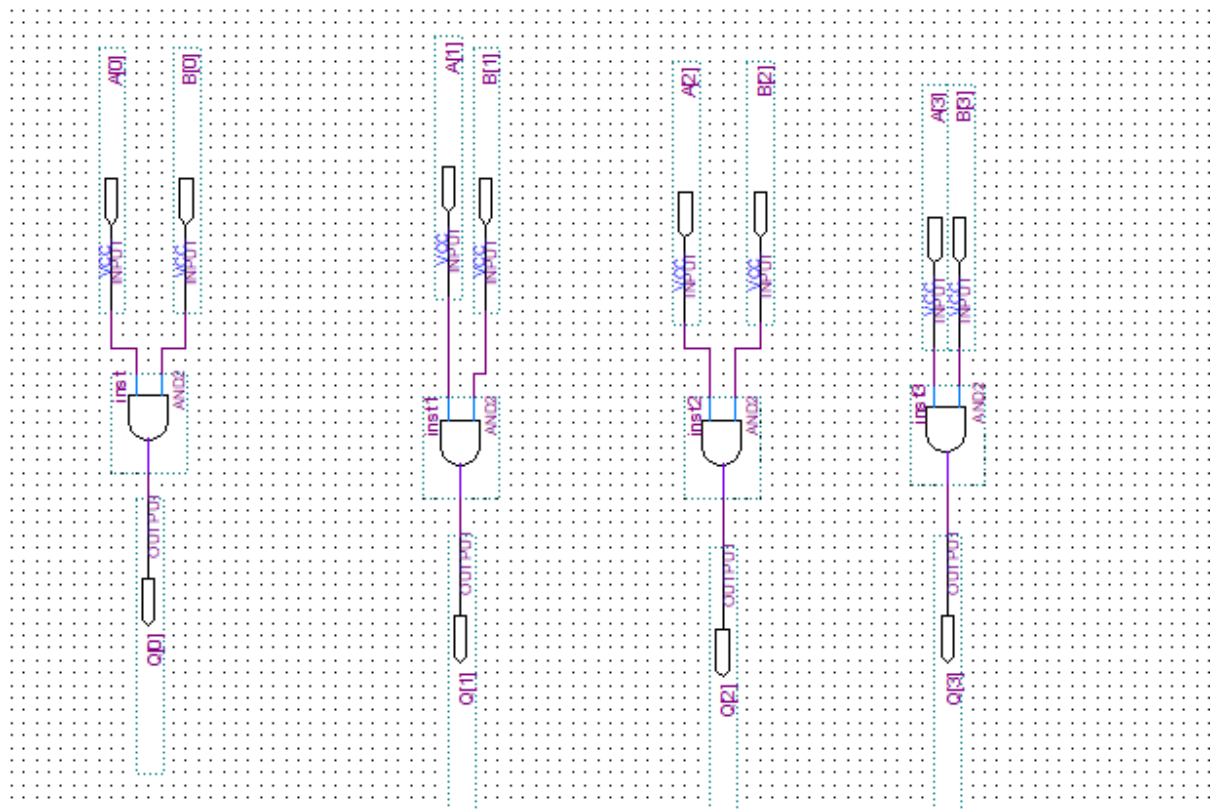


Hình 9. Waveform kiểm tra thiết kế.

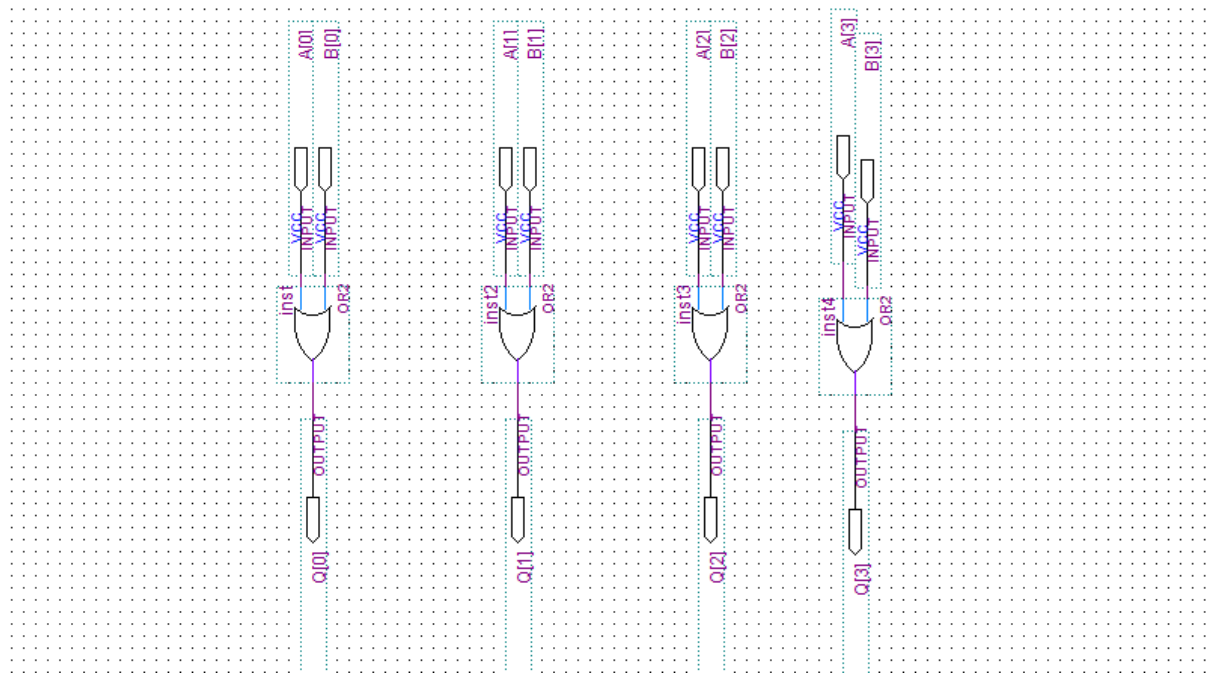
➔ Bộ AU đã chạy đúng như yêu cầu của các phép toán

b. Thiết kế bộ LU

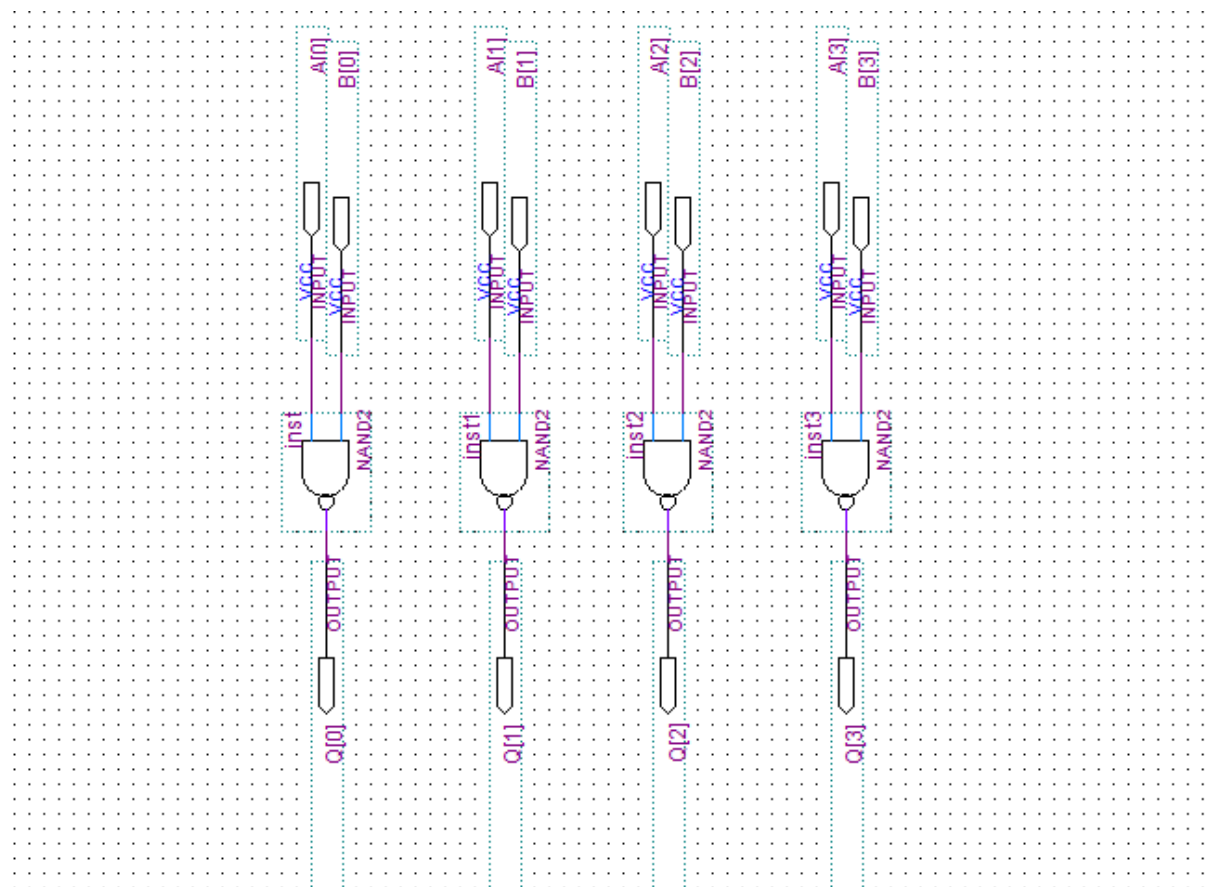
- Các phép toán logic cần thiết kế gồm and, or, nand và xor. Với các khối này thì ta chỉ cần ghép song song các cổng lại với nhau để thành khối 4 bit.



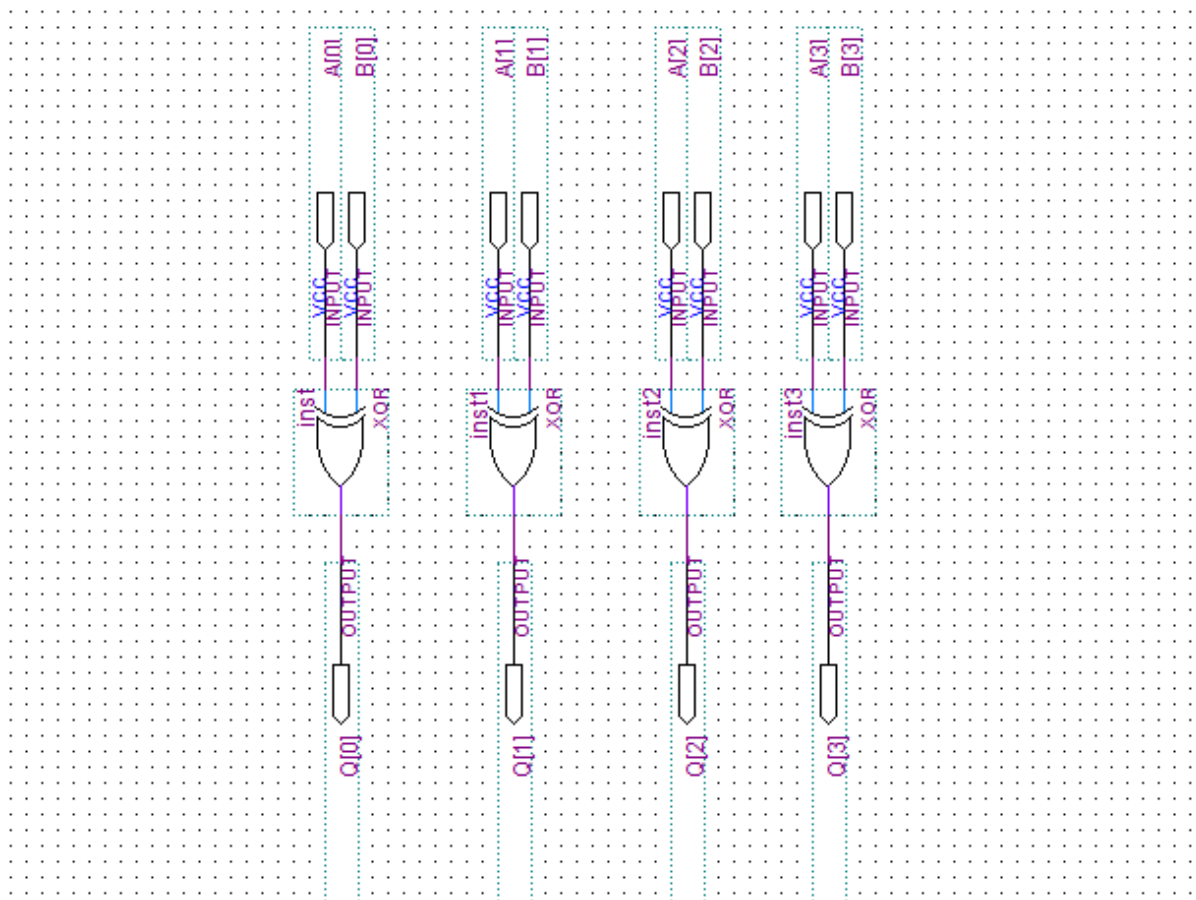
Hình 10. And 4 bit.



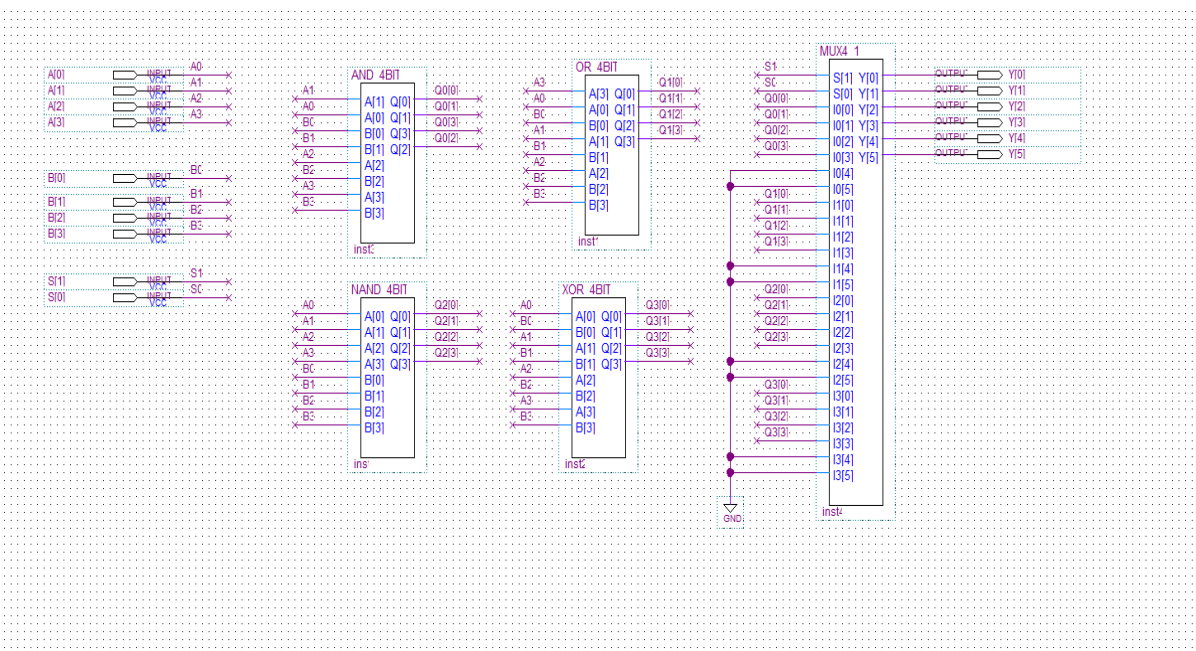
Hình 11. Or 4 bit



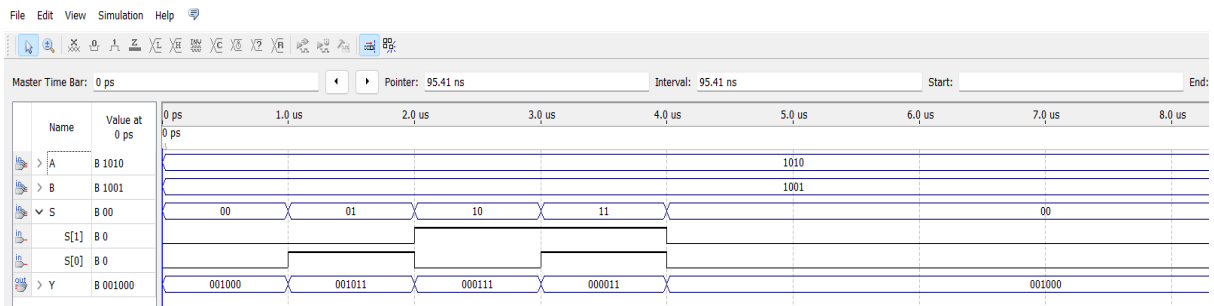
Hình 12. NAND 4 bit



Hình 13. Xor 4 bit

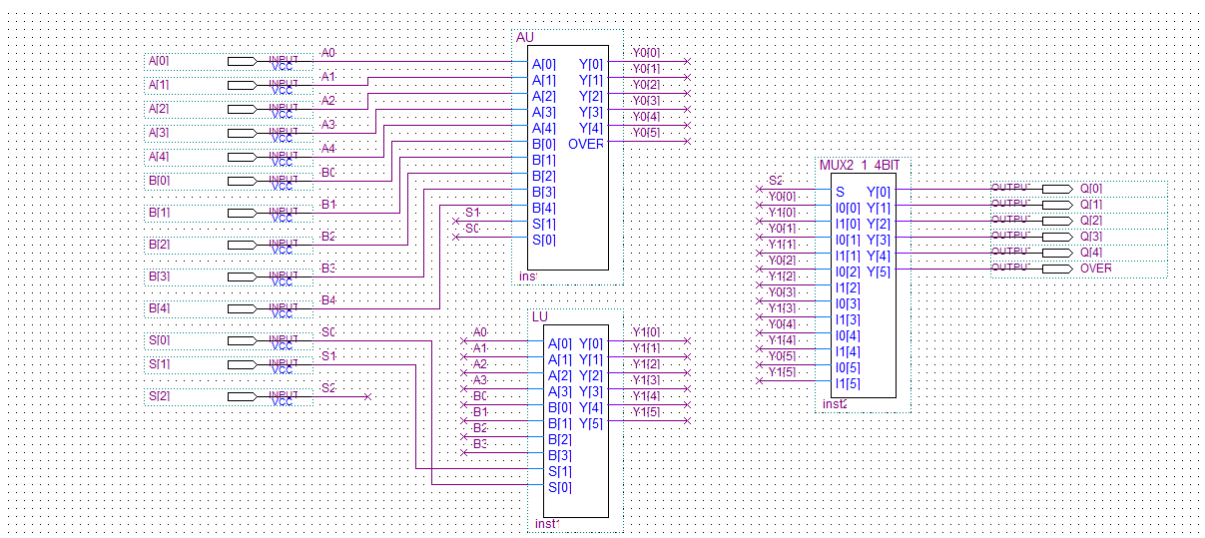


Hình 14. Bộ LU sau khi thiết kế .

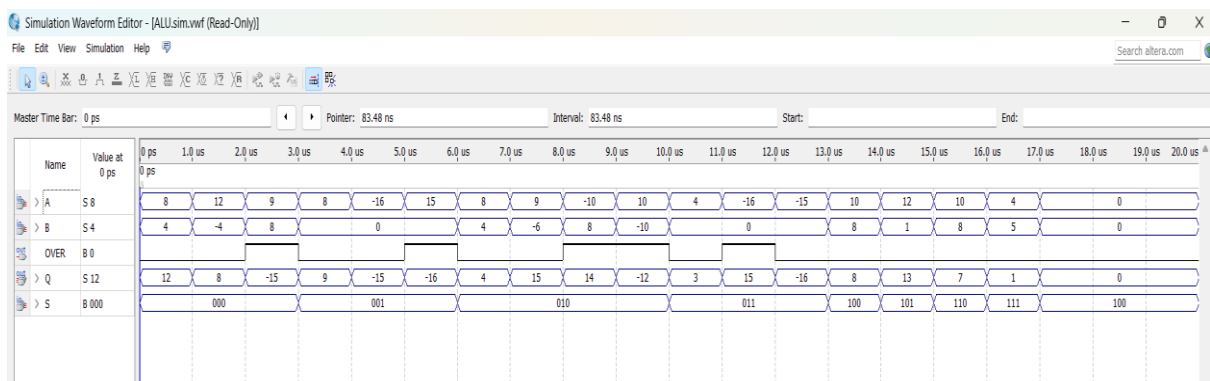


Hình 15. Waveform của bộ LU.

- Sau khi đã thiết kế xong bộ LU và AU ta tiến hành ghép lại các bộ lại với nhau để tạo thành ALU.



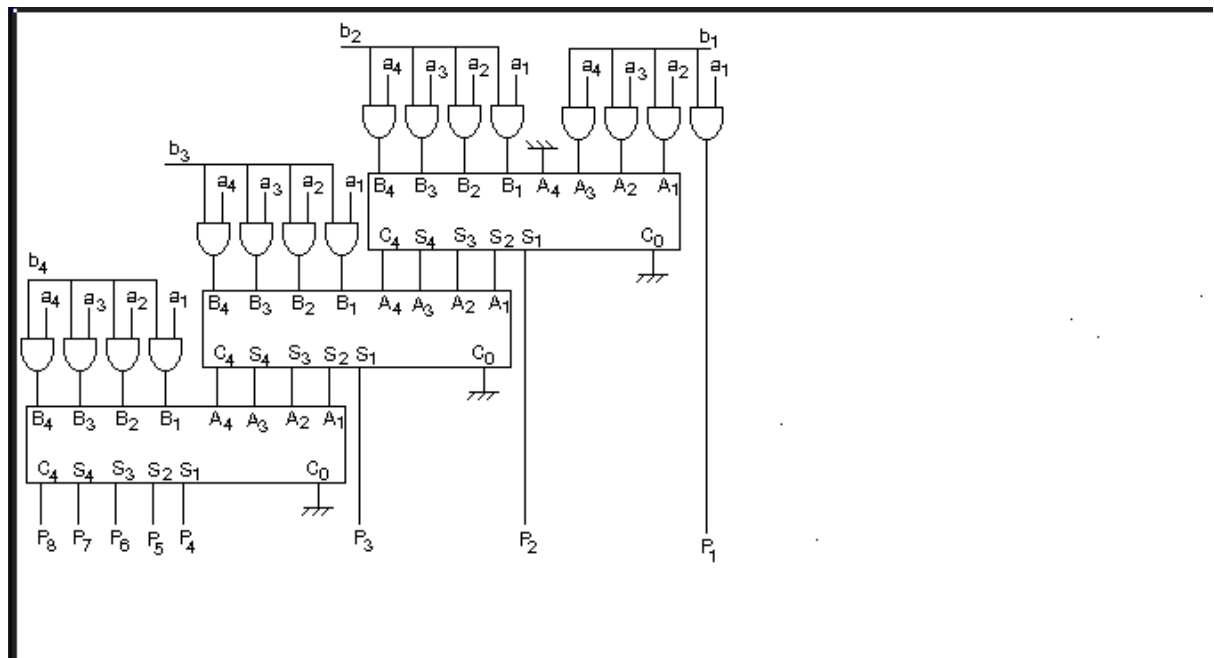
Hình 16. Thiết kế khi hoàn thành



Hình 17. Waveform của bộ ALU

c. Thiết kế mạch nhân

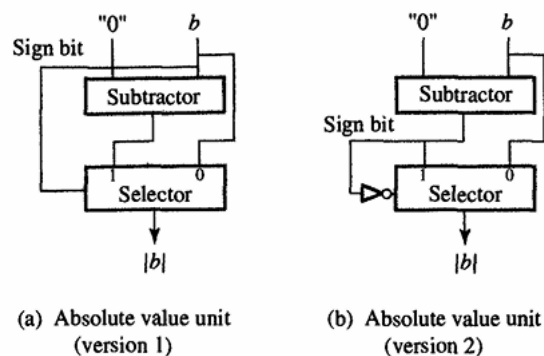
- Ý tưởng thiết kế (Hình 18)



Hình 18.

- Để thực hiện với phép nhân có dấu thì chúng ta đưa về trị tuyệt đối của chúng rồi thực hiện phép nhân, sau đó chuyển kết quả về dạng bù 2.
- Vậy chúng ta cần phải thiết kế thêm một mạch tính giá trị tuyệt đối và một mạch tính bù 2.

Section 8.4 Synthesis from



Hình 19. Mạch tính giá trị tuyệt đối.

- Để xác định dấu, ta thực hiện phép xor giữa 2 bit cuối cùng.
 - Bảng chân trị mạch tính bù 2 = bù 1 + 1.
- Vậy nếu dấu là dấu âm thì ta thực hiện bù 1 từng bit và cộng thêm 1, nếu là dấu dương thì ta giữ nguyên và cộng thêm 0.

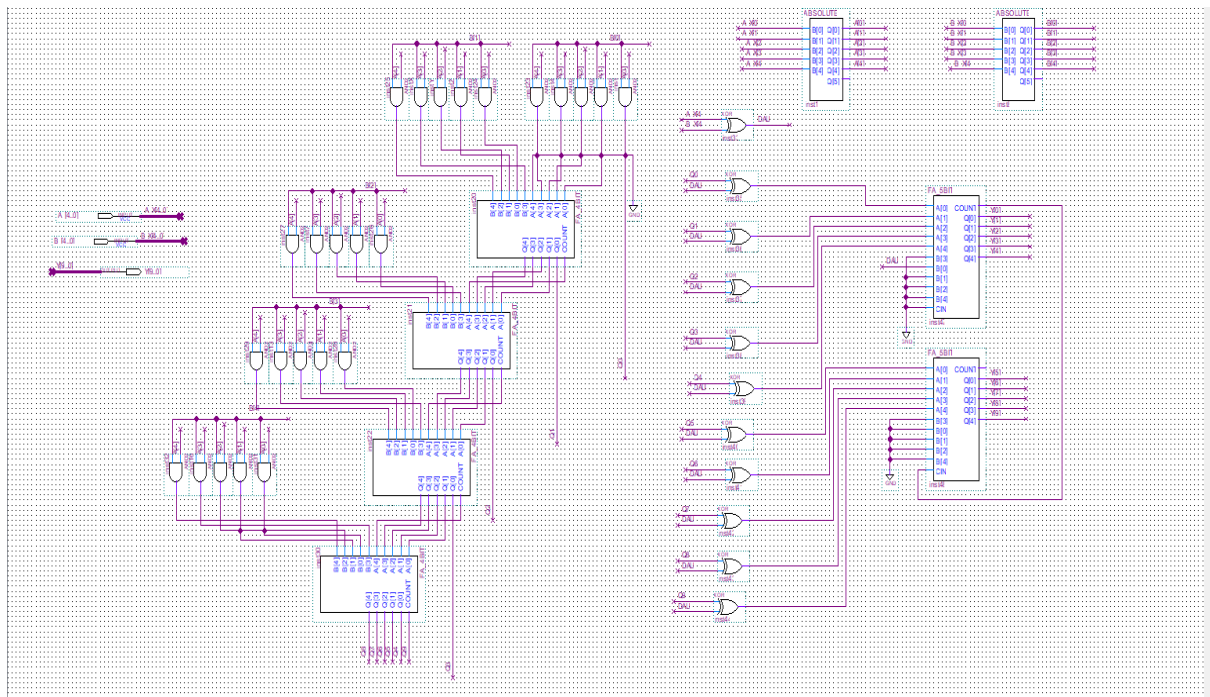
Dấu	Bít	Ouput (mạch bù 1, nếu dấu	Giá trị cộng thêm (1 nếu là

		dương thì giữ nguyên, dấu âm thì bù bit)	dấu âm và 0 nếu dấu dương)
0	0	0	0
0	1	1	0
1	0	1	1
1	1	0	1

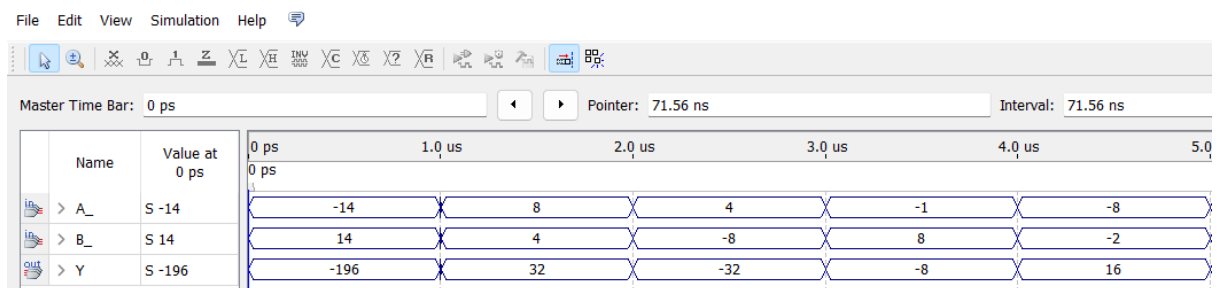
Hình 20. Bảng chân trị

Output = dấu xor bit

Toán hạng thứ 2 của mạch cộng = dấu. (Nếu dấu âm cộng 1, nếu dấu dương thì cộng 0)



Hình 21. Toàn bộ thiết kế mạch nhân dạng bù 2.



Hình 22. Waveform mô phỏng thiết kế.