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

BÀI BÁO CÁO CE118.P11.1 THIẾT LUẬN LÝ SỐ TUẦN 3

THỰC HÀNH LAB03: THIẾT KẾ MẠCH TỔ HỢP PHỤC VỤ TÍNH TOÁN

Sinh viên: Trương Thiên Quý
MSSV: 23521321
Lớp: CE118.P11.1
Giảng viên hướng dẫn: Tạ Trí Đức

CE118-Lab03

Thiết kế mạch tổ hợp phục vụ tính toán

1. Lý thuyết

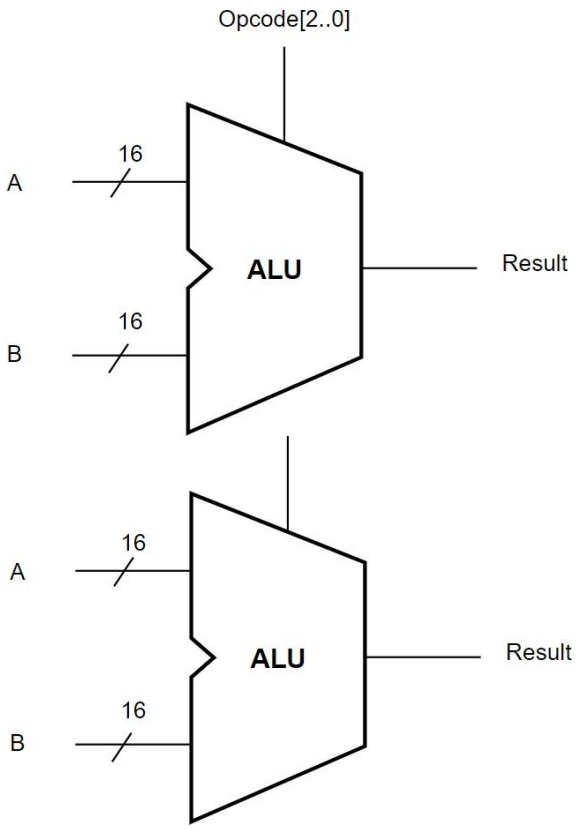
ALU - Arithmetic and Logic Unit là một mạch tổ hợp để thực hiện các tác vụ về toán học (cộng, trừ, nhân, chia,…) và logic (and, or, not, xor,…).

Một ALU đơn giản sẽ bao gồm 2 phần là khối AU (Arithmetic Unit) chịu trách nhiệm thực hiện các tác vụ về toán học và khối LU (Logic Unit) chịu trách nhiệm thực hiện các tác vụ về logic.

ALU thường sẽ có 2 toán hạng và phép toán được ALU thực hiện sẽ được điều khiển thông qua tín hiệu Opcode.

2. Thực hành

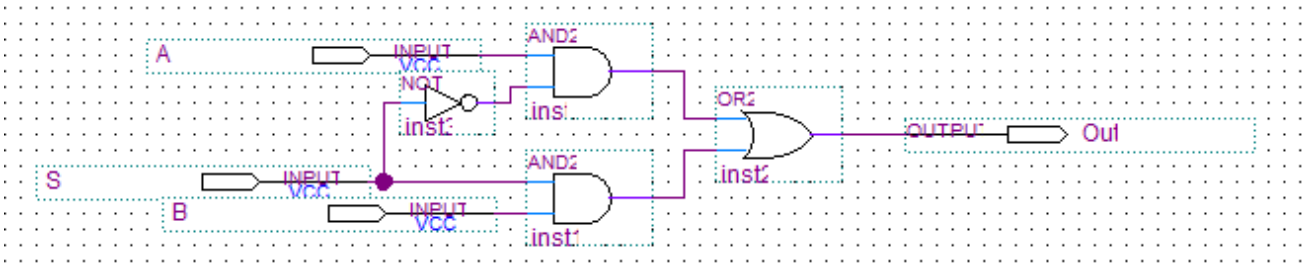
Sinh viên thực hiện thiết kế và mô phỏng một ALU có 2 toán hạng (16 bit) và các phép toán **cộng, cộng 1, trừ, trừ 1, and, or, nand, xor** theo đúng thứ tự tương ứng với tín hiệu điều khiển (Opcode) từ 0□7.



Hình 1 - Symbol của ALU 16 bit

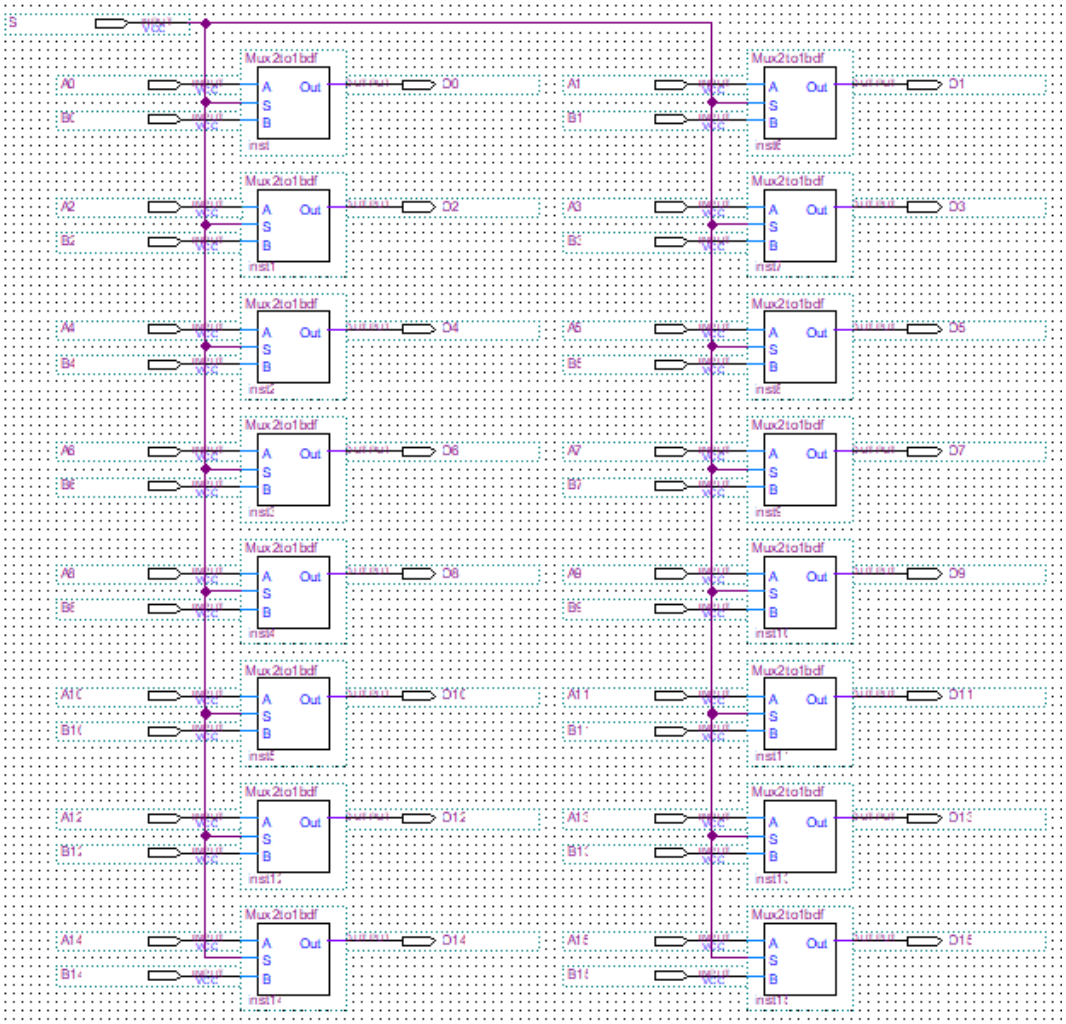
**Thiết kế MUX2 16 bit:

-Bộ MUX2 1 bit:



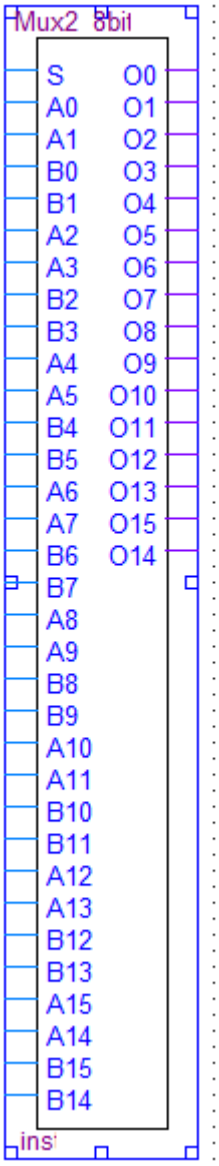
Hình 2.1 *– MUX2 1 bit*

-Bộ MUX2 16 bit:



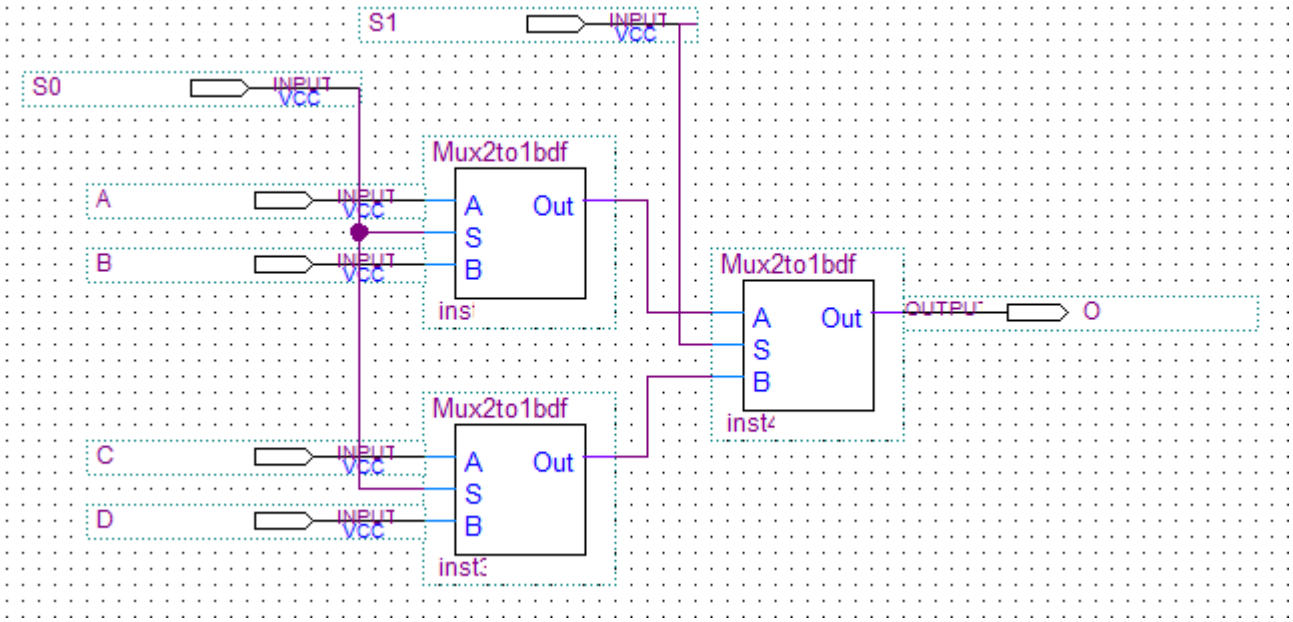
Hình 2.2 *– MUX2 16 bit*

Ký hiệu MUX2 16 bit:



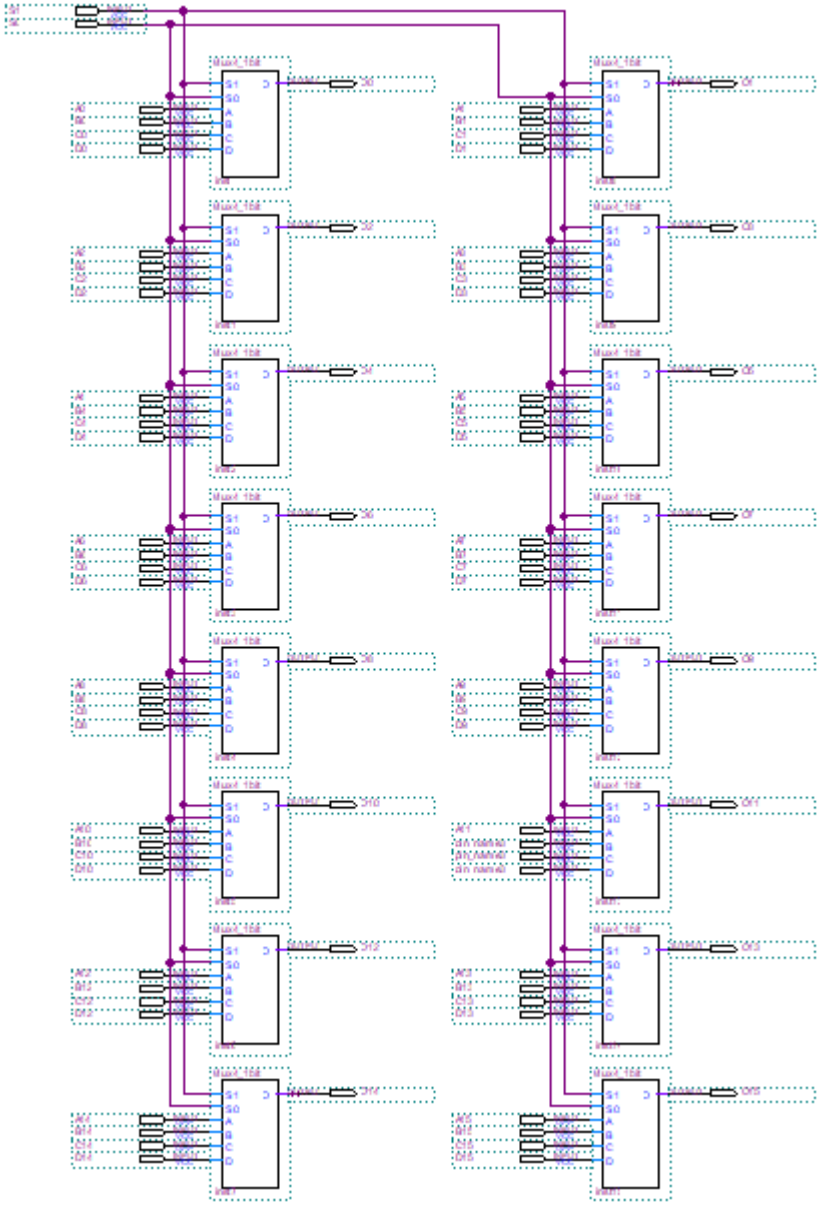
Hình 2.3 *– Ký hiệu MUX2 16 bit*

****Thiết kế MUX4 1 bit:**



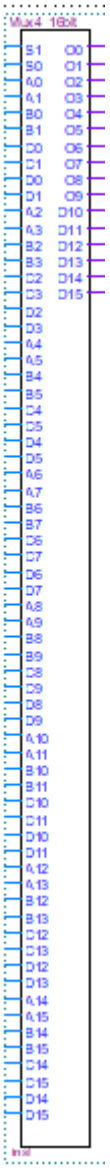
Hình 2.4 *– MUX4 1 bit*

****Thiết kế MUX4 16 bit:**



Hình 2.5 *–MUX4 16 bit*

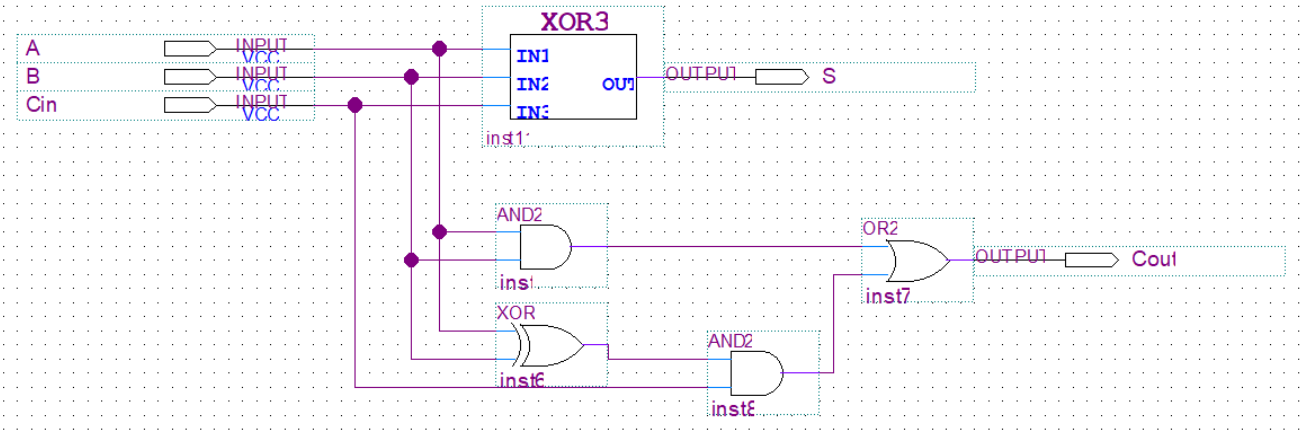
Ký hiệu bộ MUX4 16 bit:



Hình 2.6 *– Ký hiệu MUX4 16 bit*

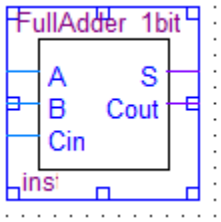
***Thiết kế ALU 16 bit:

Thiết kế bộ cộng 1 bit:



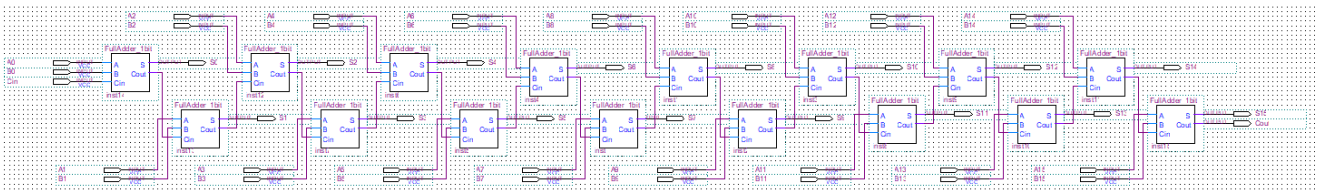
Hình 2.7 *– Bộ cộng 1 bit*

Ký hiệu bộ cộng 16 bit:



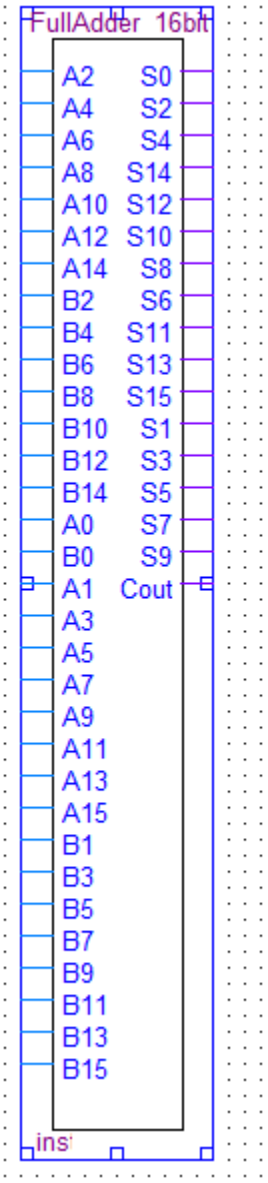
Hình 2.8 *– Ký hiệu Bộ cộng 1 bit*

**Thiết kế bộ cộng 16 bit:



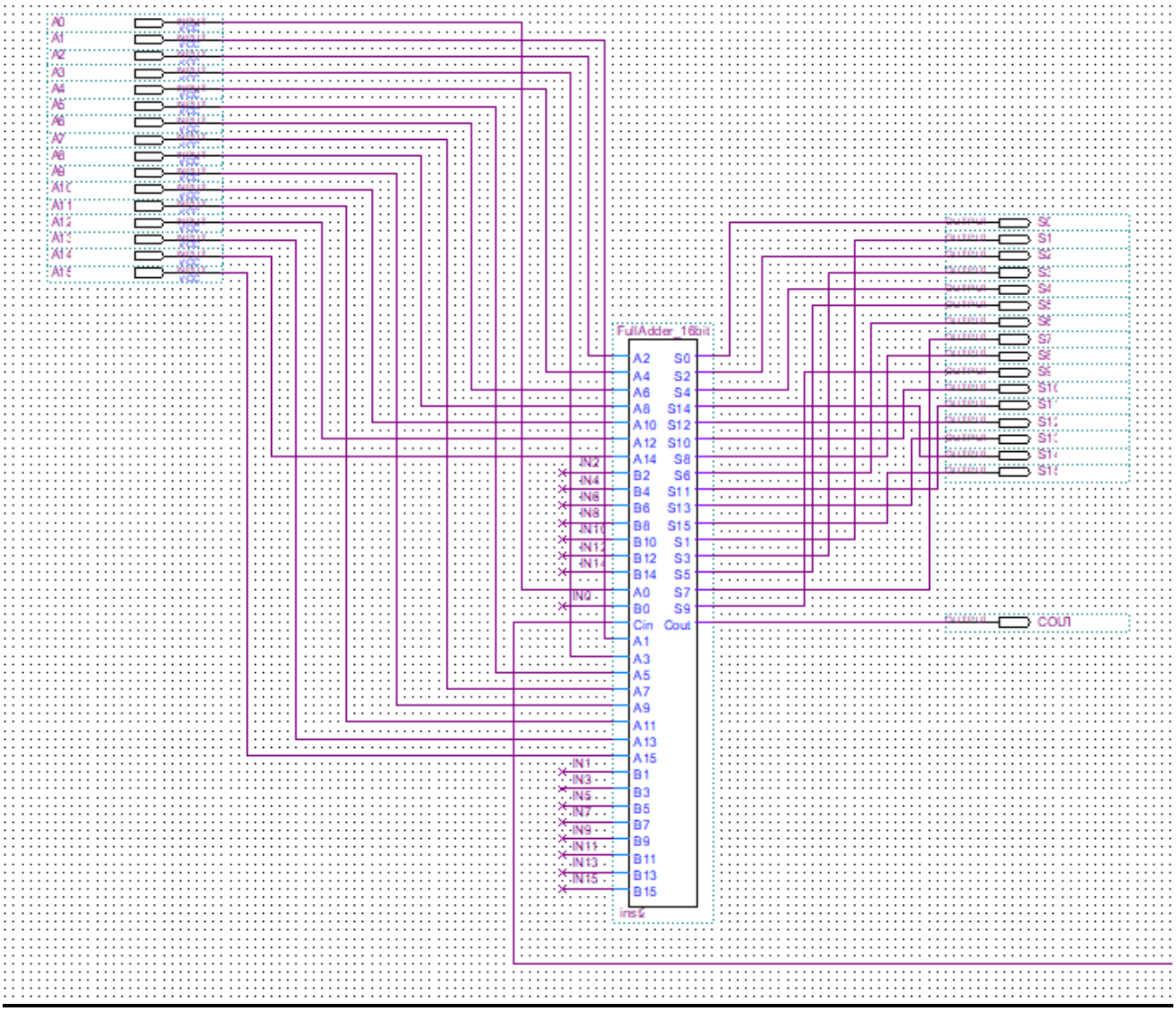
Hình 2.9 *–Bộ cộng 16 bit*

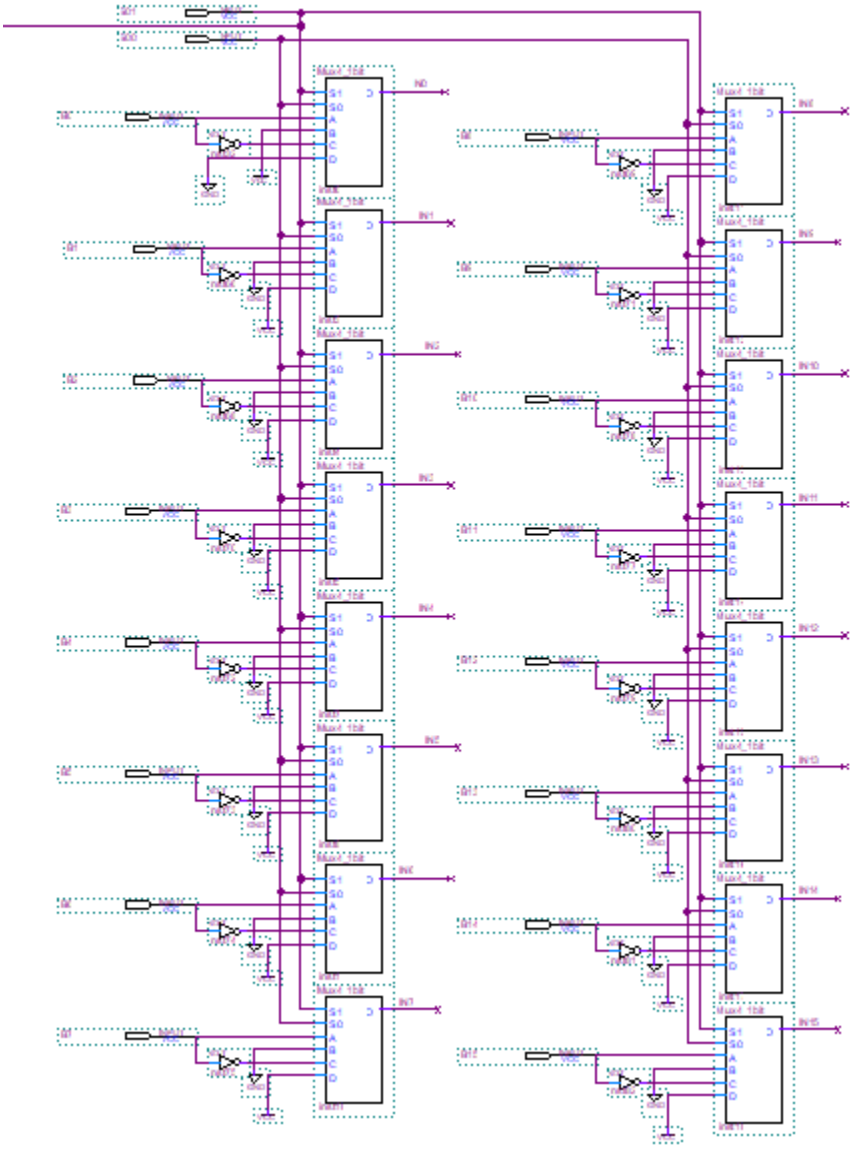
Ký hiệu bộ cộng 16 bit:



Hình 2.10 *–Ký hiệu Bộ cộng 16 bit*

******Thiết kế AU 16 bit:**



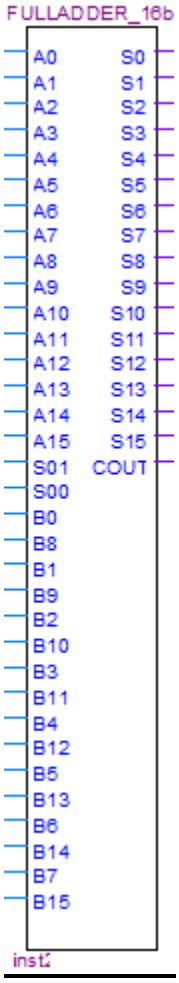


Tín hiệu S01-S00	Bit[0]	Bit[15:1]	Cin
00	B[0]	B[15:1]	0
01	1	0	0
10	Not B[0]	Not B[15:1]	1
11	0	1	1

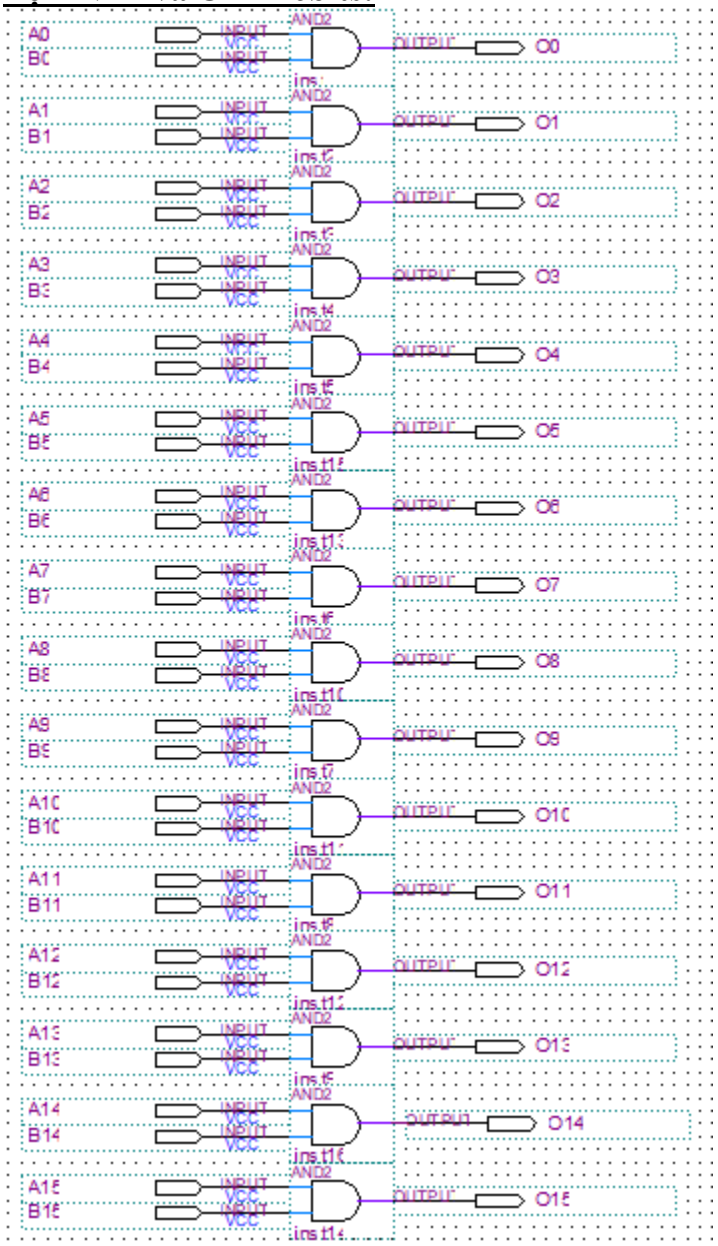
- Khi tín hiệu điều khiển là 00, thực hiện phép cộng hai số A+B. Lúc này, toán hạng thứ hai sẽ là giá trị của B với 16 bit.
- Với tín hiệu 01, thực hiện phép tính A+1. Ở đây, toán hạng thứ hai được đặt thành số có 15 bit cao là 0 và bit thấp nhất là 1.

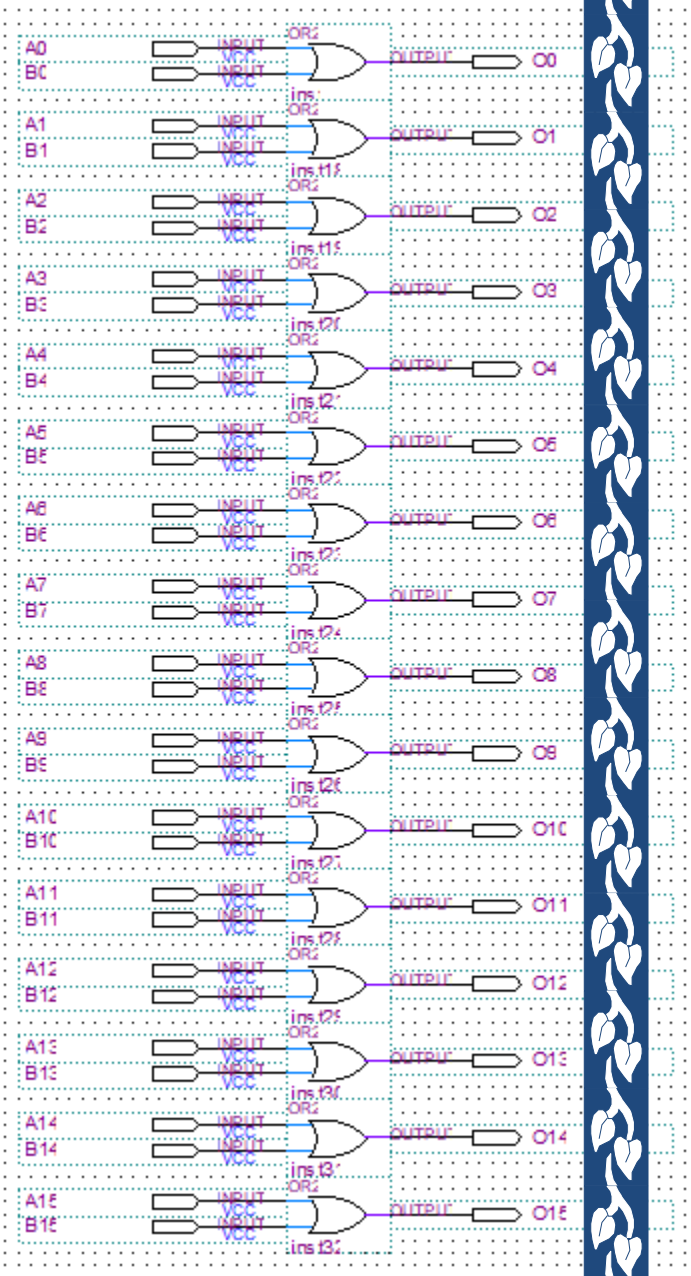
- Khi tín hiệu là 10, thực hiện phép trừ A− B. Toán hạng thứ hai sẽ là giá trị đảo bit (NOT) của B và Cin=1, tương ứng với phép bù hai của B (NOT B + 1 ở Cin).
- Khi tín hiệu là 11, thực hiện phép tính A−1. Toán hạng thứ hai lúc này sẽ là số có 15 bit cao là 1 và bit thấp nhất là 0, cùng với Cin=1, tương đương với A+(−1) và cho kết quả là A−1.

Ký hiệu AL 16 bit:

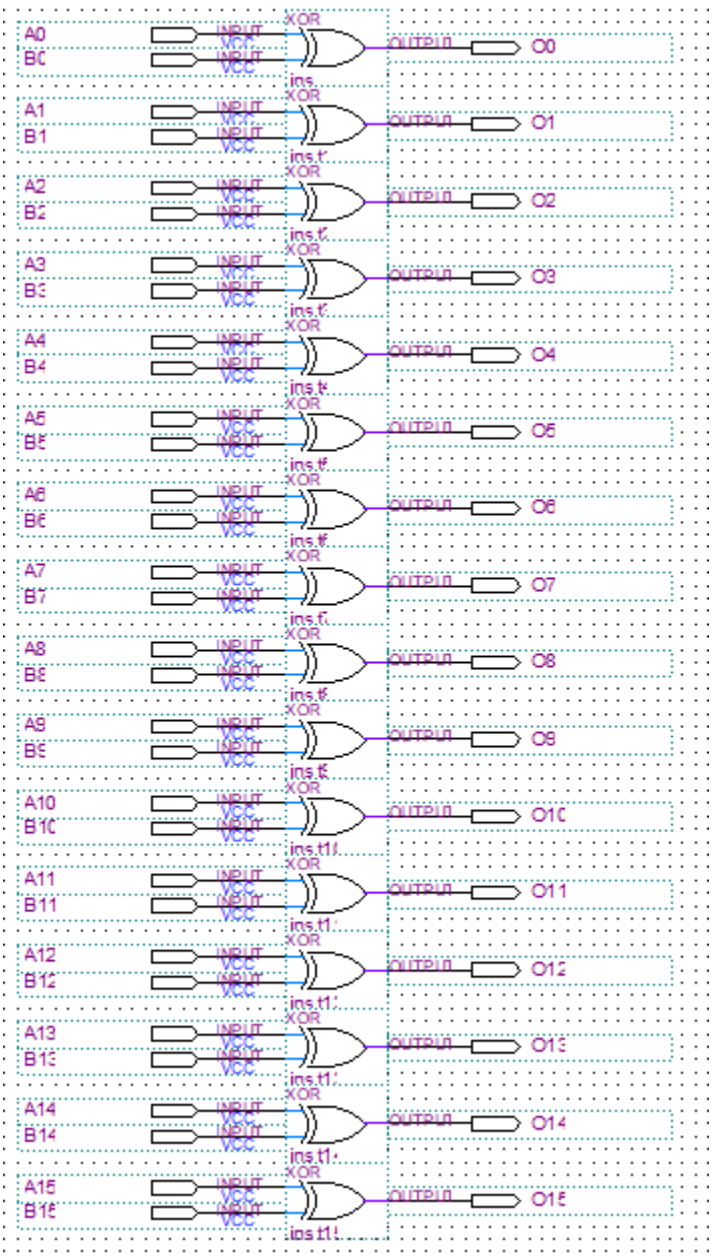


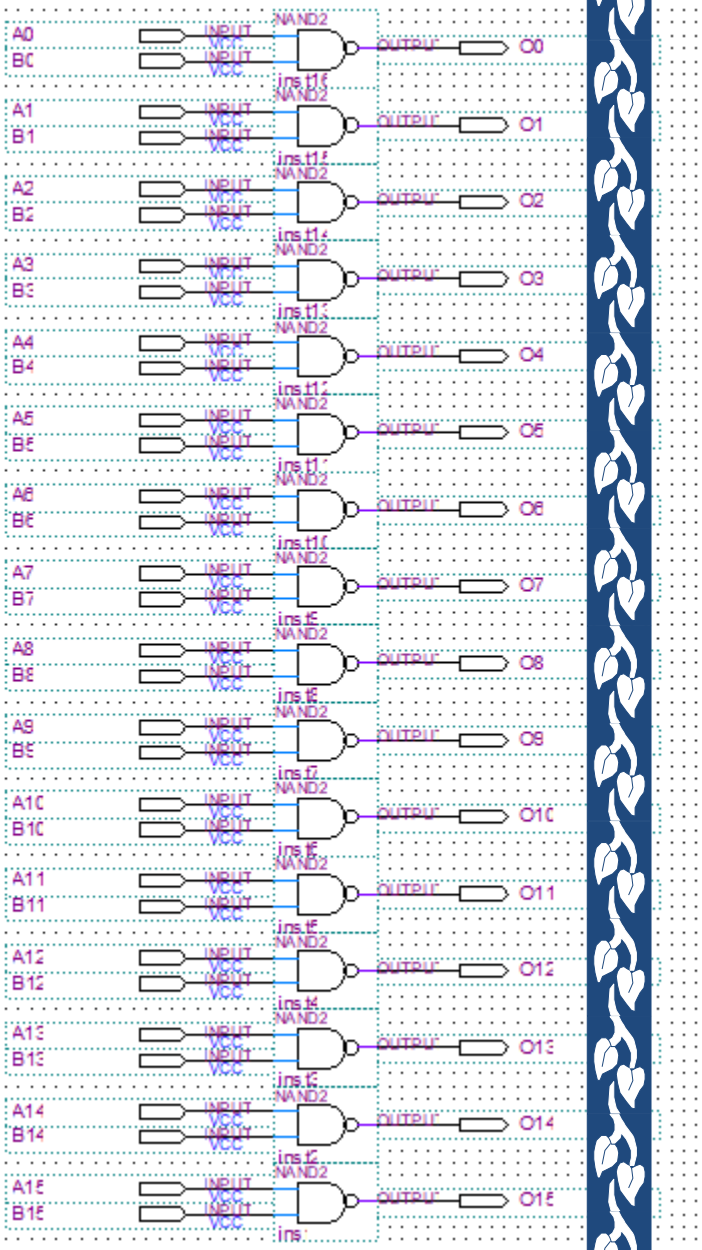
Bộ AND2 và OR2 16bits:



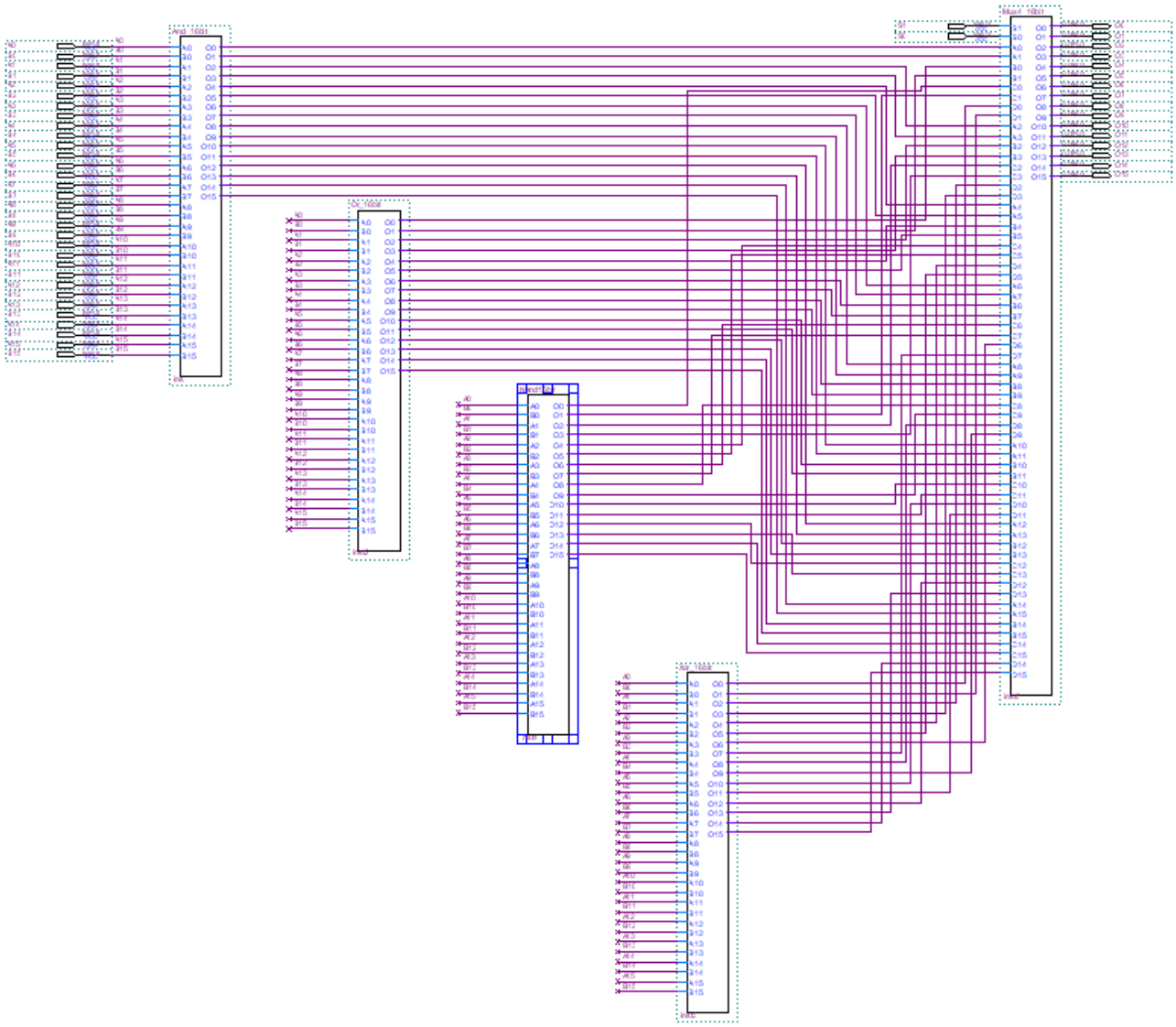


Bộ NAND2 và XOR2 16bits:





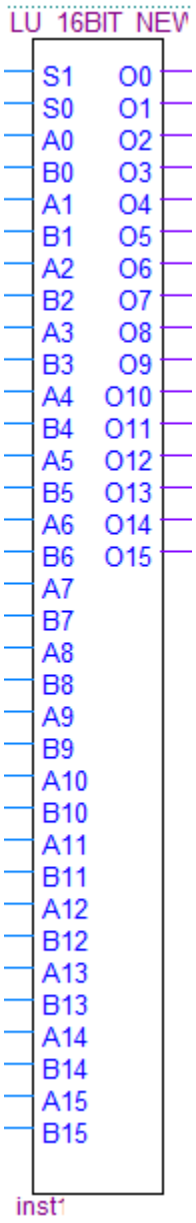
******Thiết kế LU 16 bit:**



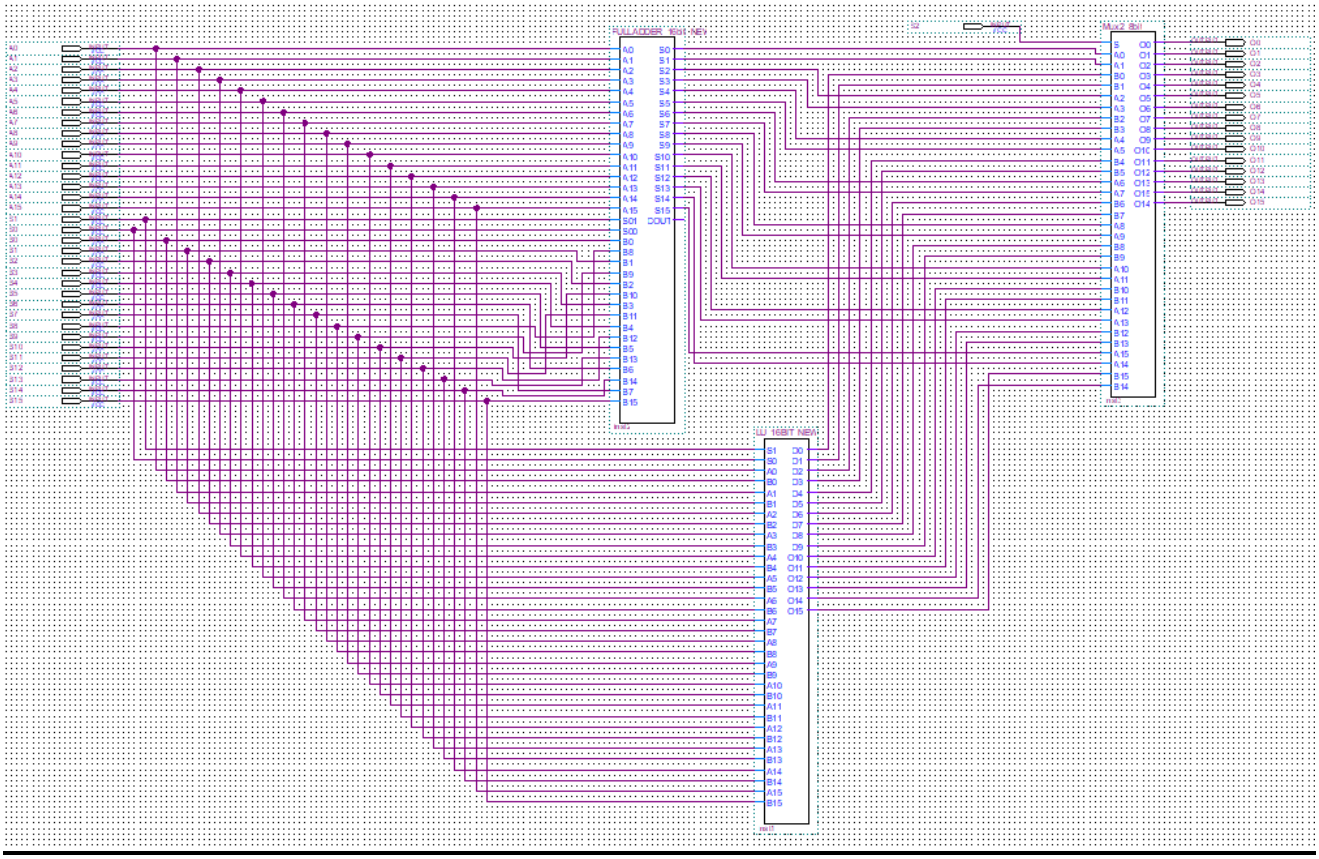
Dữ liệu đầu vào của hai toán hạng A và B được đưa vào các cổng logic **AND2, OR2, NAND2, và XOR2** đã được tích hợp. Sau đó, các tín hiệu đầu ra của các cổng này được truyền đến bộ **Mux4** để chọn lựa đầu ra cuối cùng.

Tín hiệu S1 – S0	Phép Logic
00	AND
01	OR
10	NAND
11	XOR

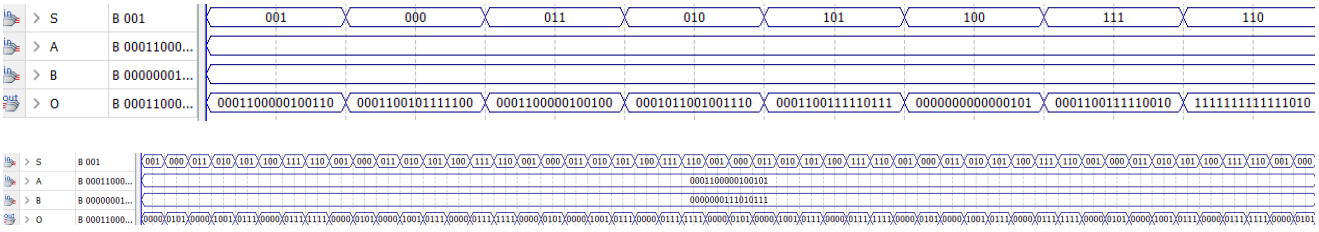
Ký hiệu bộ LU 16bit



******Thiết kế ALU 16bit: Kết hợp AU + LU + Mux2 = ALU



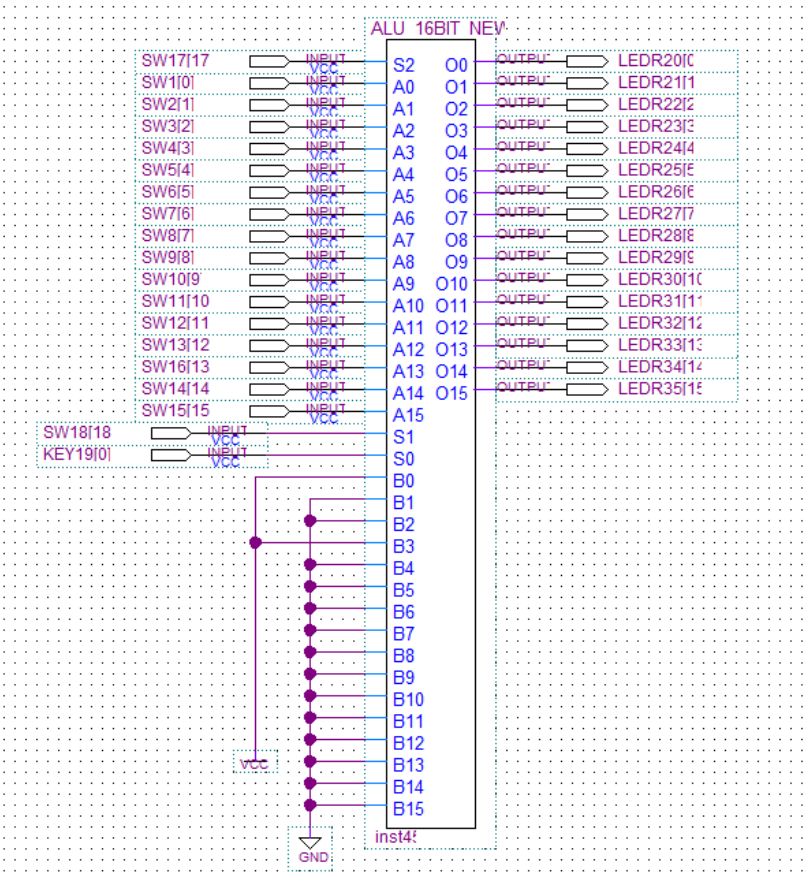
Mô phỏng hoạt động ALU:



- ✓ S2S1S0 = 000: **A + B:**
- ✓ S2S1S0 = 001: **A + 1:**
- ✓ S2S1S0 = 010: **A - B:**
- ✓ S2S1S0 = 011: **A - 1:**
- ✓ S2S1S0 = 100: **A and B:**
- ✓ S2S1S0 = 101: **A nand B:**
- ✓ S2S1S0 = 110: **A or B:**
- ✓ S2S1S0 = 111: **A xor B:**

Video nạp Kit: Lab3_ALU.mp4 - Google Drive

Nạp Kit:



3.Bài tập

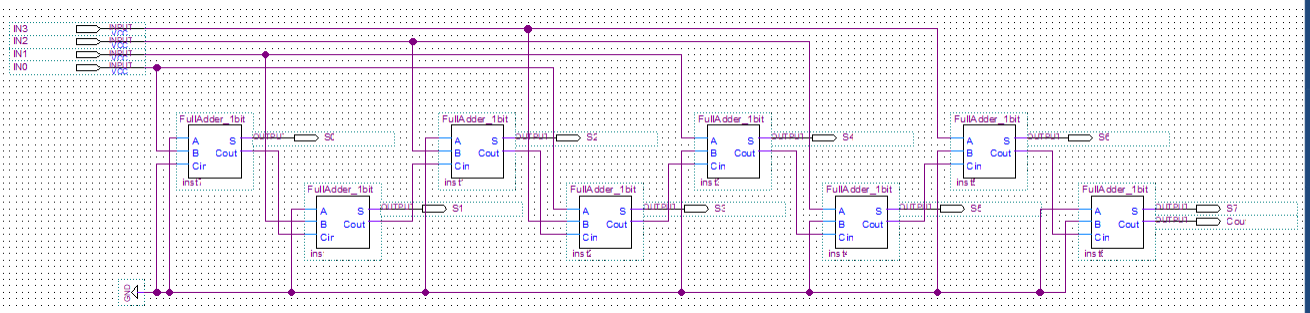
Sinh viên thực hiện thiết kế và mô phỏng một bộ nhân 4 bit đơn giản với hệ số nhân là số cuối cùng trong MSSV. Riêng đối với những bạn có số cuối MSSV là 0 (hoặc 1) thì thực hiện bộ nhân 8 (hoặc 9) tương ứng.

Input 4 bit, Ouput có thể lên tới 8 bit (bởi vì với 2 số 4 bit nhân nhau thì kết quả tối đa cần phải dùng 8 bit để hiển thị)

Ví dụ: Input là số 15 (1111) nhân với 9 (1001) thì kết quả là 135 (cần 8 bit để biểu diễn)

Gợi ý: phép nhân tức là phép dịch bit. Cụ thể, đối với phép nhân 8 thì ta có thể dịch trái 3 bit. Với phép nhân 9 thì ta có thể thực hiện dịch trái 3 bit, sau đó cộng với chính nó.

******Tiến hành thiết kế:**



Một số 4 bit, ký hiệu là **IN[3:0],** được đưa vào bộ cộng sau khi dịch **3 bit** sang trái. Lúc này, các bit trong IN được định vị lại để có trọng số cao hơn: **bit IN3** được đưa vào bộ cộng tại vị trí **bit thứ 7**, bit **IN2** tại vị trí bit thứ 6, bit **IN1** tại vị trí bit **thứ 5,** và **bit IN0** tại vị trí bit **thứ 4**. Điều này hoàn thành việc dịch trái 3 bit cho số đầu vào.

Ngoài ra, các bit **IN[3:0]** cũng được đưa trực tiếp vào toán hạng thứ hai của bộ cộng từ bit 3 đến bit 0, cho phép thực hiện phép cộng giữa số bị dịch trái và số gốc ban đầu. Nhờ đó, mạch này vừa thực hiện dịch trái 3 bit cho số ban đầu, vừa cộng với số ban đầu, tạo ra kết quả nhân với 9.

Bước tiếp theo là tiến hành mô phỏng: