

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN

KHOA KỸ THUẬT MÁY TÍNH

LỚP CE118.N21.2



BÁO CÁO LAB 3

THIẾT KẾ MẠCH TỔ HỢP PHỤC VỤ TÍNH TOÁN

MSSV: 21521810

HỌ TÊN: NGUYỄN QUỐC TRƯỜNG AN

I. 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.

II. 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.

THIẾT KẾ

-Ta có khối ALU gồm AU và LU.

-Ta thực hiện thiết kế khối AU gồm các phép tính: Cộng, cộng 1, trừ, trừ 1. Khối LU gồm các phép tính: And, or, nand, xor.

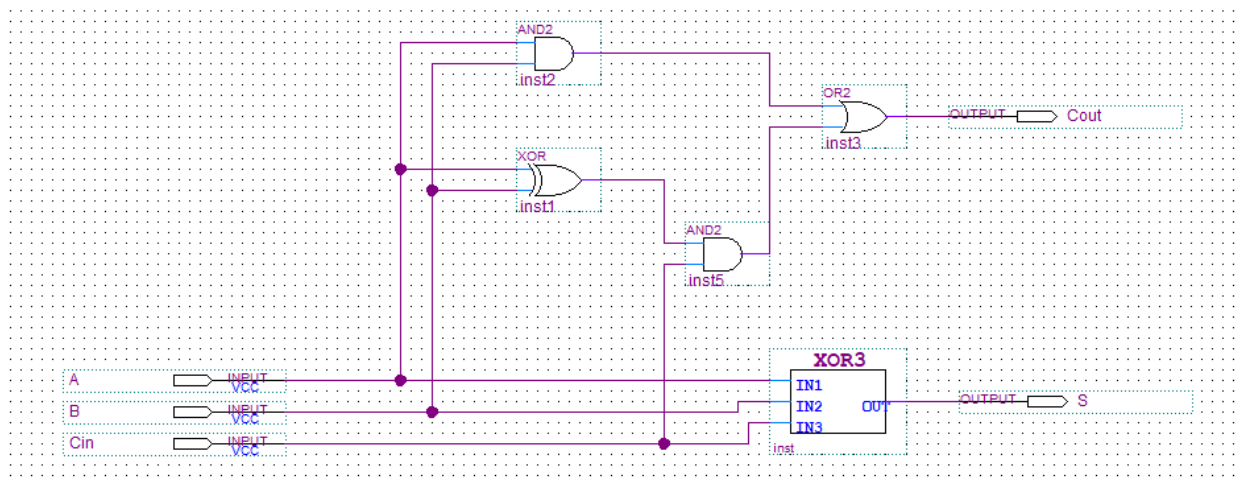
1) Thiết kế khối AU

a) Thiết kế bộ cộng

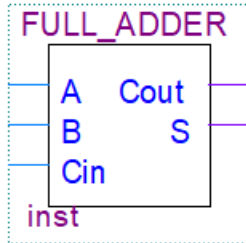
-Ta sử dụng 8 bộ full adder để thực hiện bộ cộng 8 bit, sau đó sử dụng 2 bộ cộng 8 bit để thực hiện bộ cộng 16 bit. Có được bộ cộng 16 bit thì ta giải quyết được 2 phép toán cộng, cộng 1. Thêm vào đó, ta kết hợp thêm cổng xor vào bộ cộng để có thể tích hợp thêm phép toán trừ. Tới đây đã giải quyết thêm được 2 phép toán trừ, trừ 1.

- Full adder:

+Schematic:

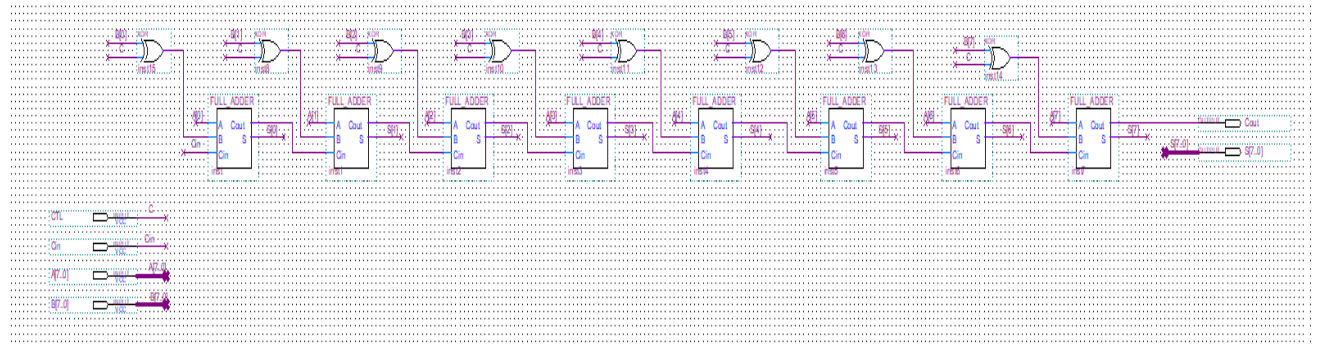


+Đóng gói:

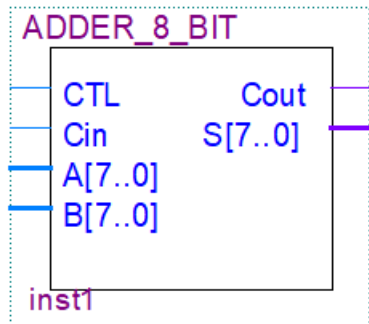


-Bộ cộng/trừ 8 bit:

+Schematic:

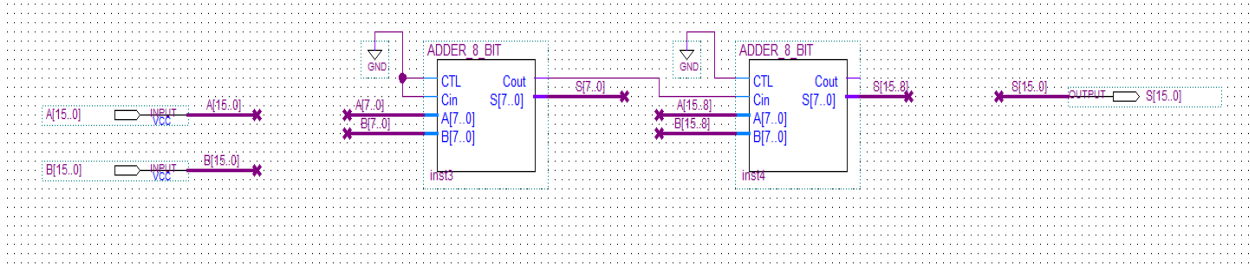


+Đóng gói:

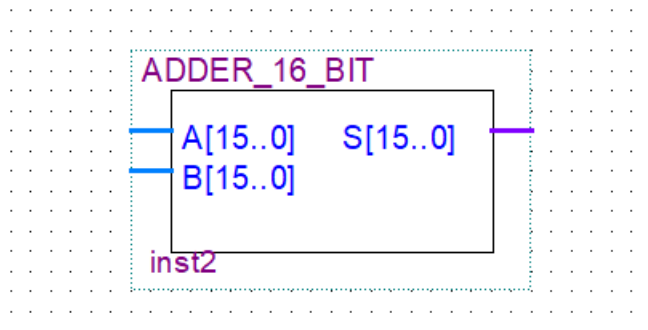


-Bộ cộng/trừ 16 bit:

+Schematic:

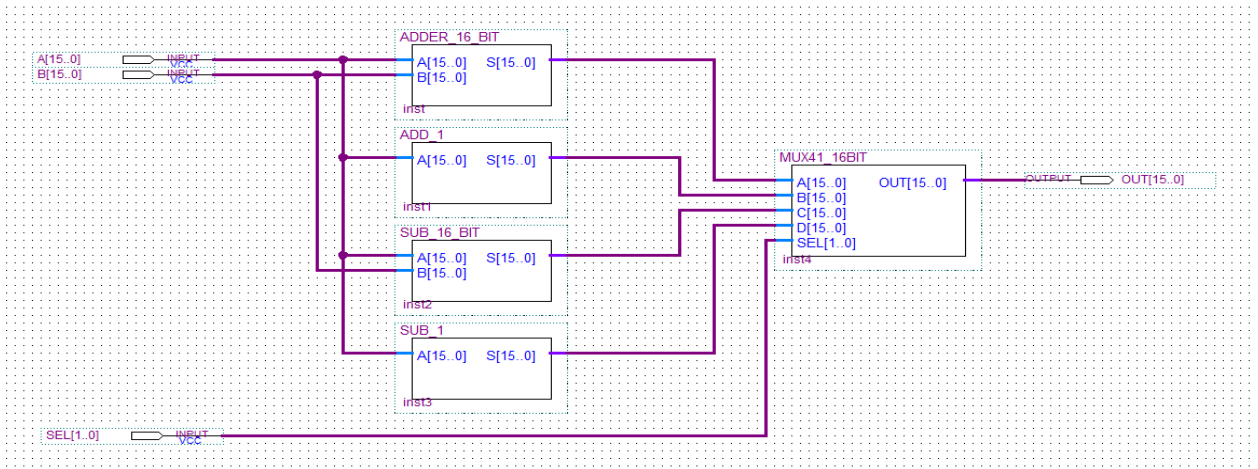


+Đóng gói:

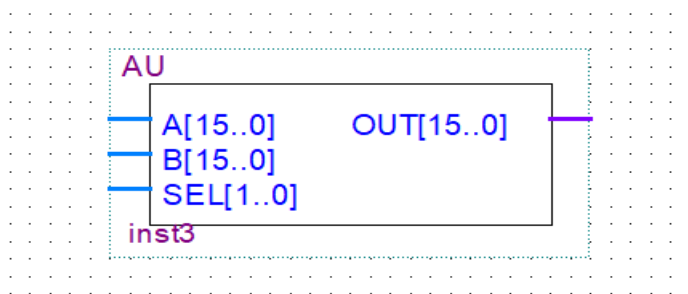


b) Tổng hợp khối AU:

-Schematic:



-Đóng gói:



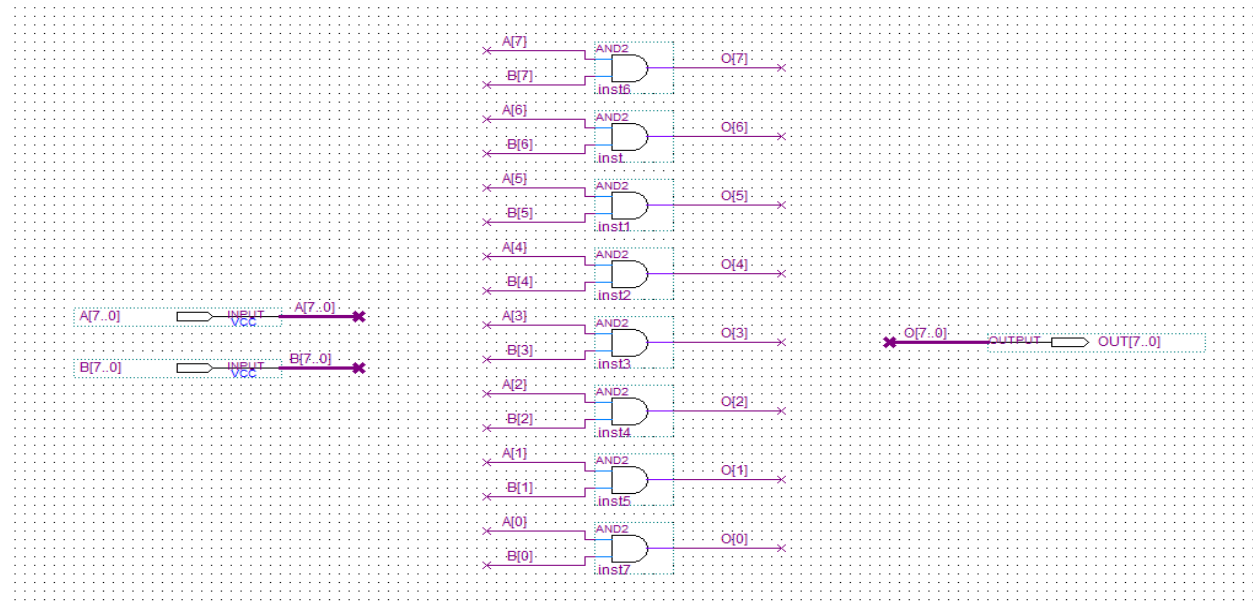
1) Thiết kế khối LU

a) Thiết kế khối and 16 bit

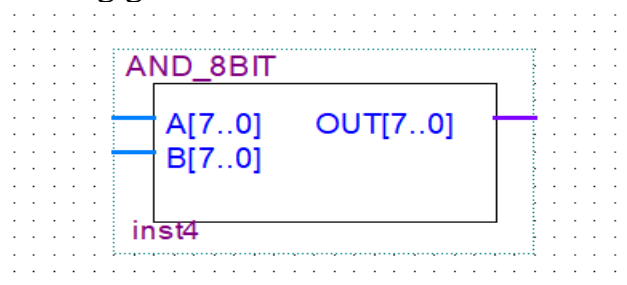
Ta thiết kế khối and 8 bit từ 8 cổng and 1 bit, từ đó thiết kế khối and 16 bit bằng 2 khối and 8 bit.

-Khối and 8 bit:

+Schematic:

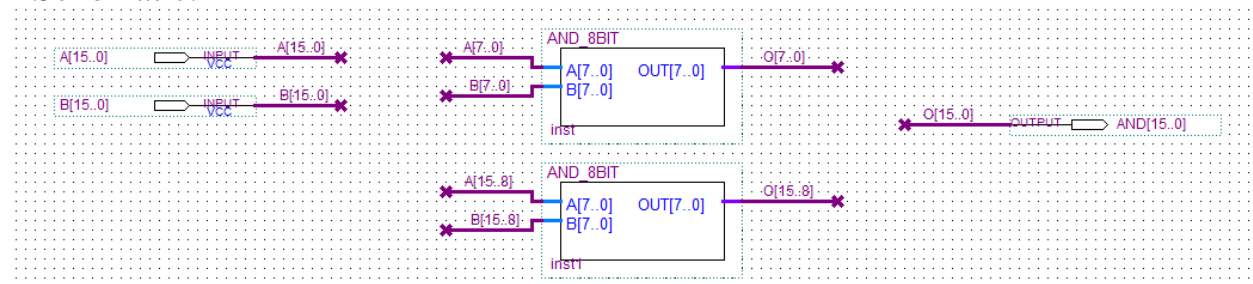


+Đóng gói:

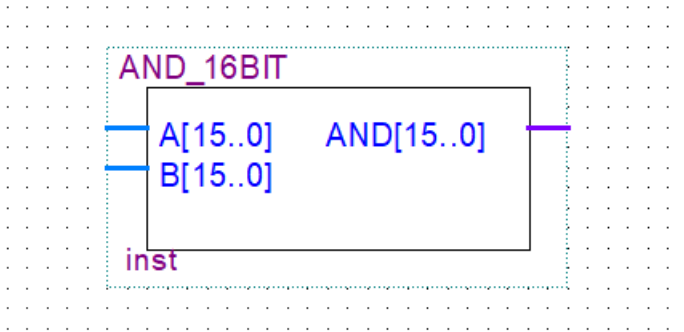


-Khối and 16 bit:

+Schematic:



+Đóng gói:

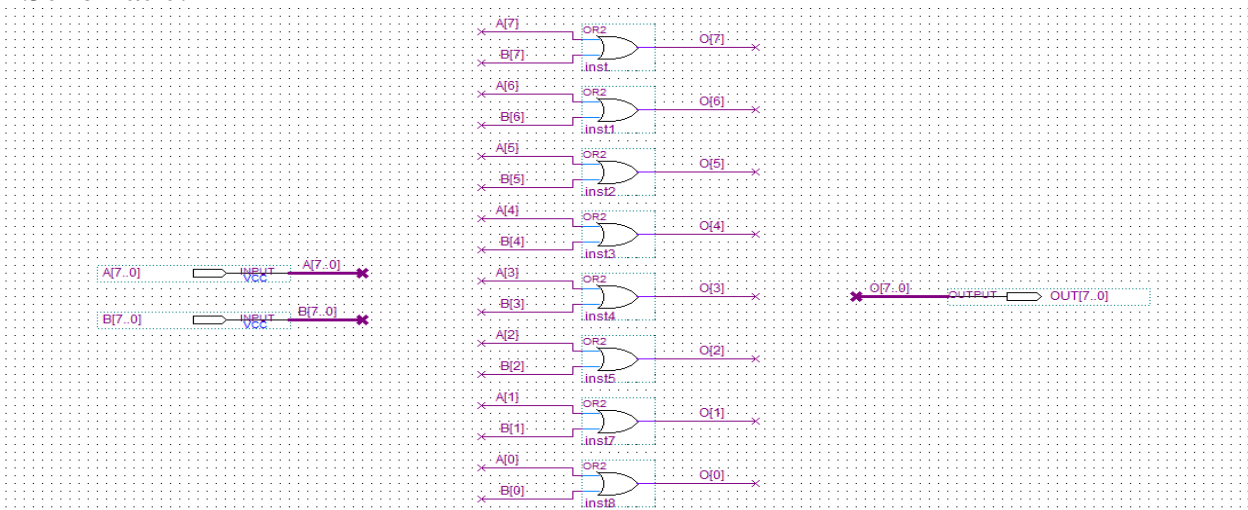


b) Thiết kế khối or 16 bit

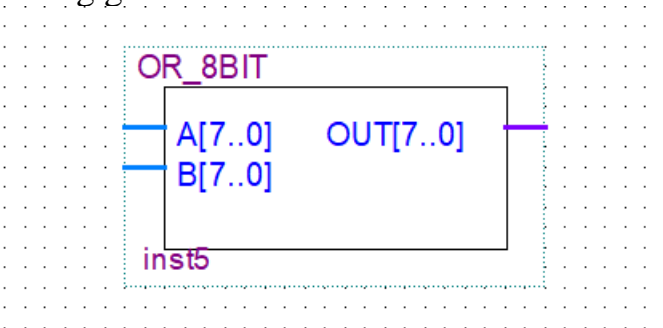
Ta thiết kế khối or 8 bit từ 8 cổng or 1 bit, từ đó thiết kế khối or 16 bit bằng 2 khối or 8 bit.

-Khối or 8 bit:

+Schematic:

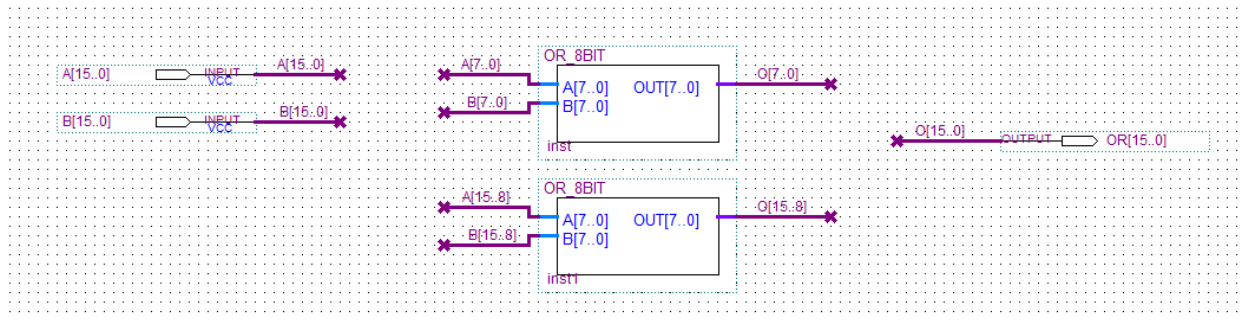


+Đóng gói:

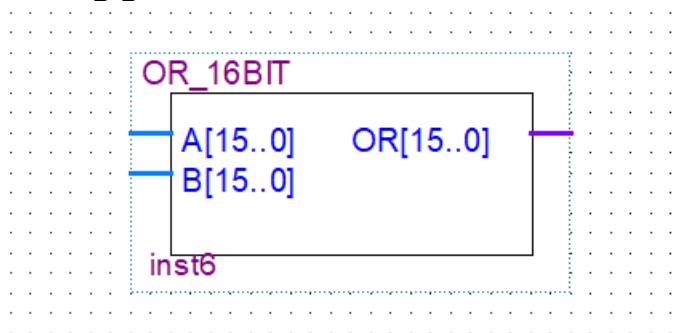


-Khối or 16 bit:

+Schematic:



+Đóng gói:

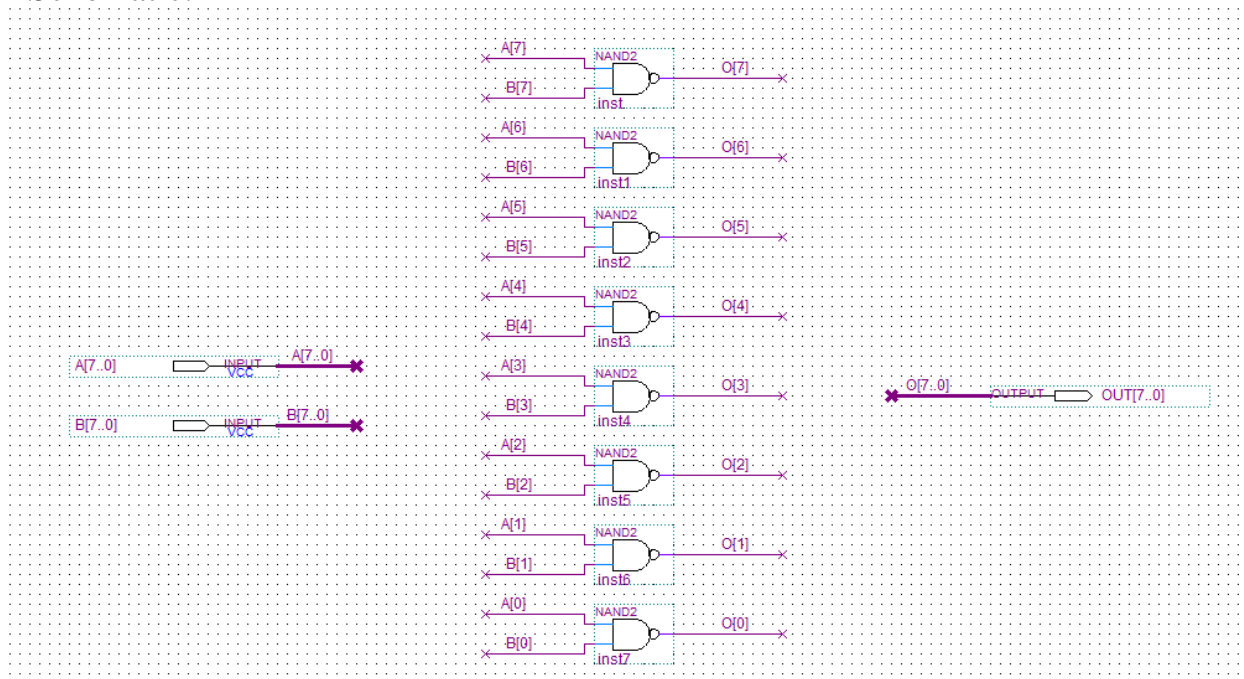


c) Thiết kế khối nand 16 bit

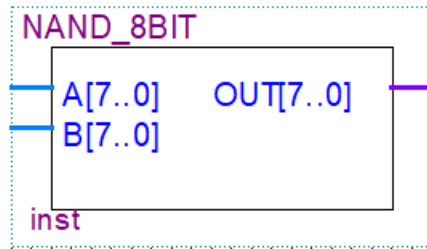
Ta thiết kế khối nand 8 bit từ 8 cổng nand 1 bit, từ đó thiết kế khối nand 16 bit bằng 2 khối nand 8 bit.

-Khối nand 8 bit:

+Schematic:

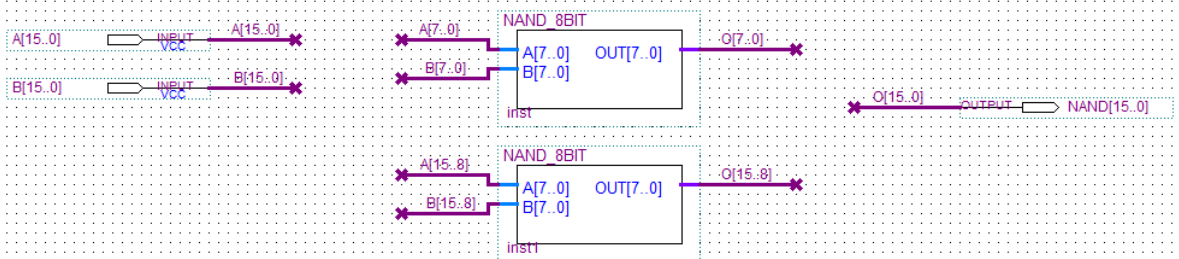


+Đóng gói:

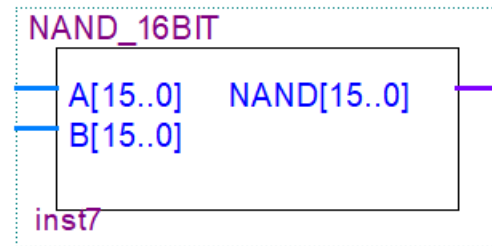


-Khối nand 16 bit:

+Schematic:



+Đóng gói:

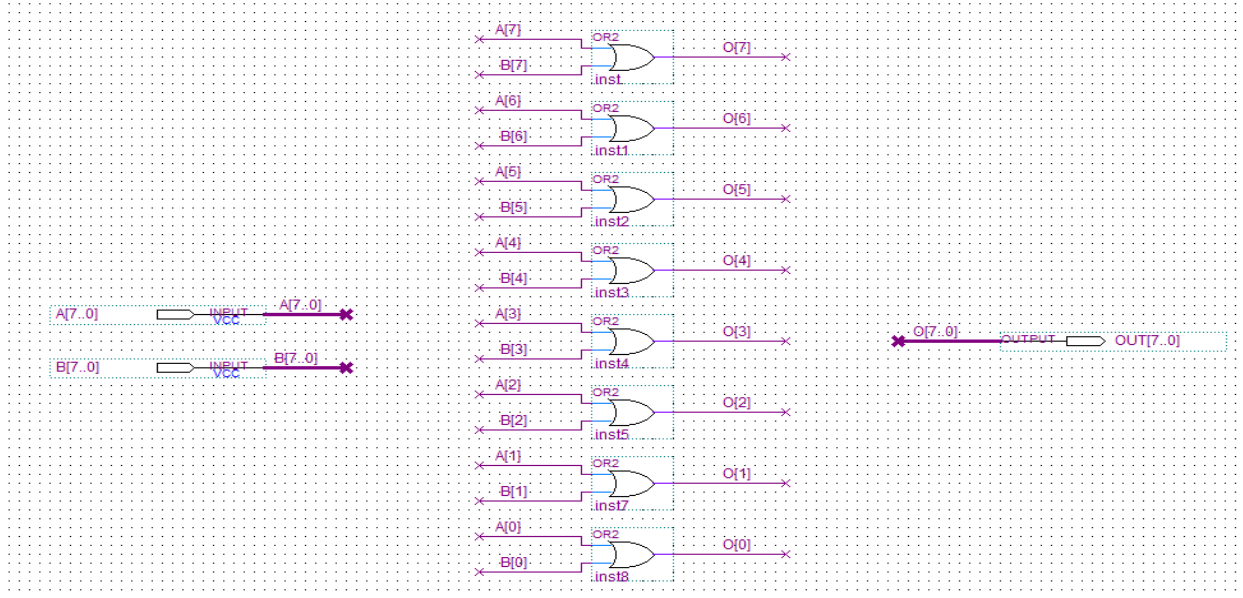


d) Thiết kế khối xor 16 bit

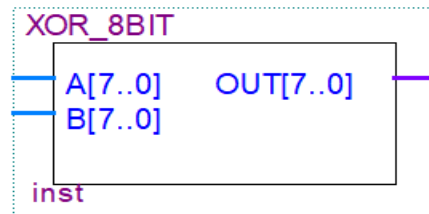
Ta thiết kế khối xor 8 bit từ 8 cổng xor 1 bit, từ đó thiết kế khối xor 16 bit bằng 2 khối xor 8 bit.

-Khối xor 8 bit:

+Schematic:

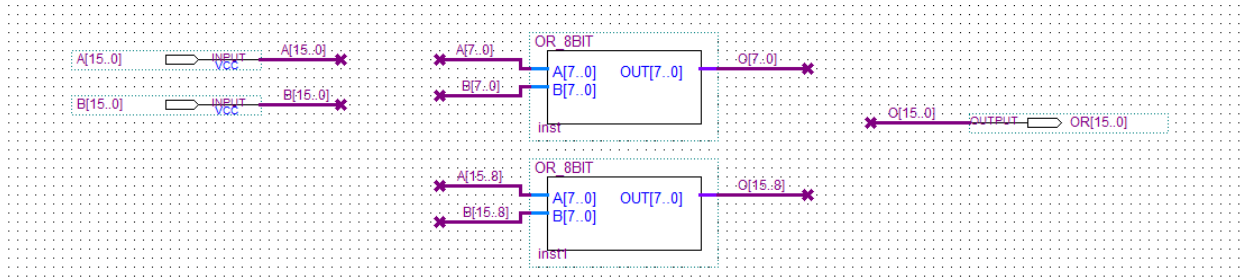


+Đóng gói:

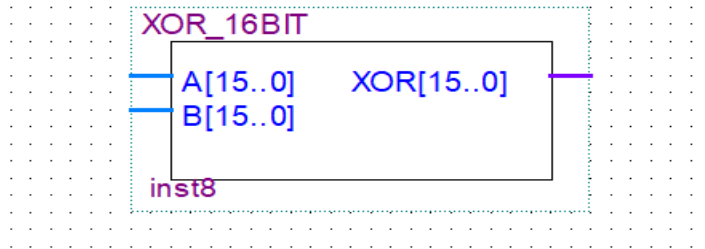


-Khối xor 16 bit:

+Schematic:

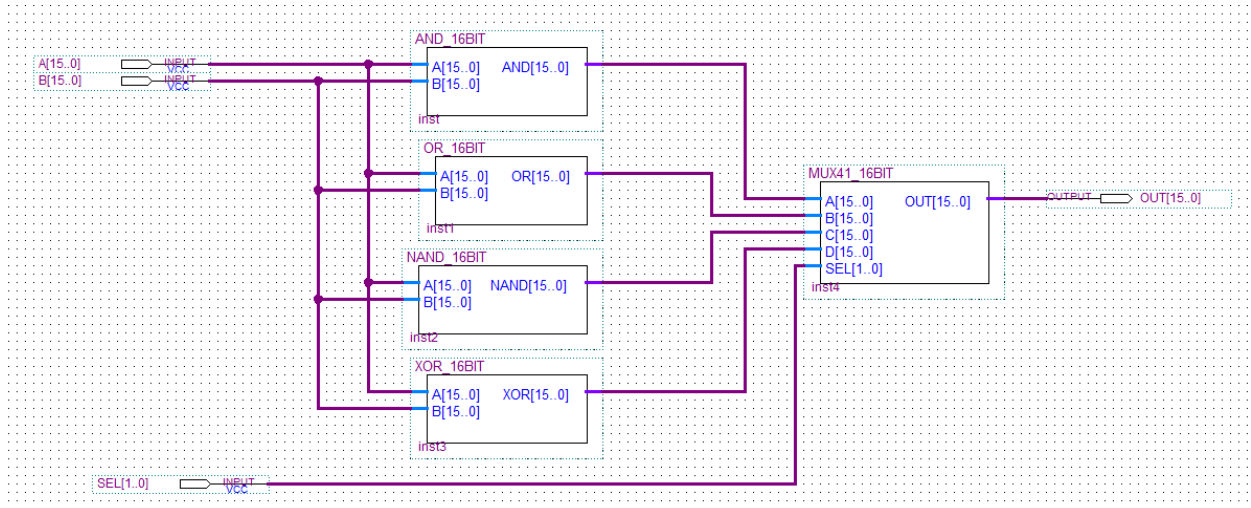


+Đóng gói:

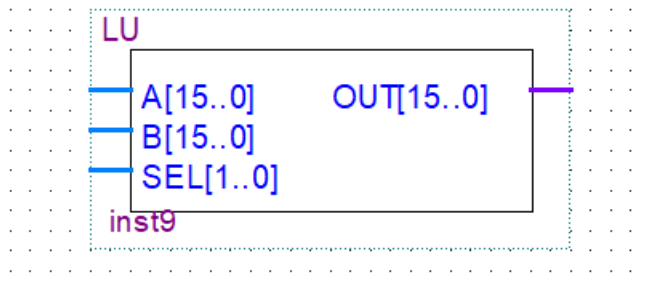


e) Tổng hợp khối LU

-Schematic:



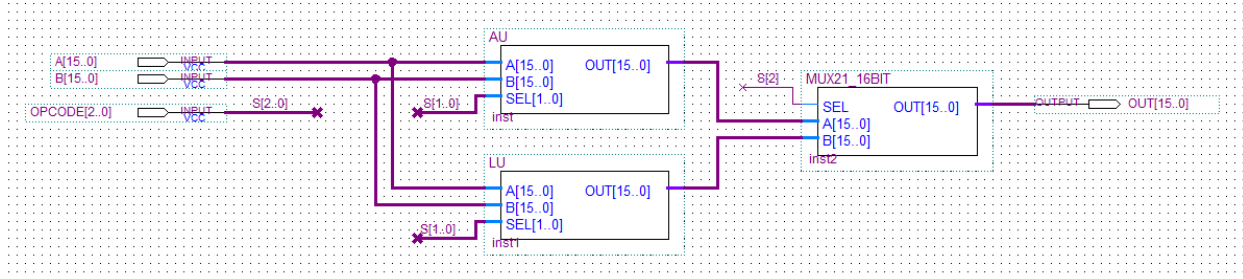
-Đóng gói:



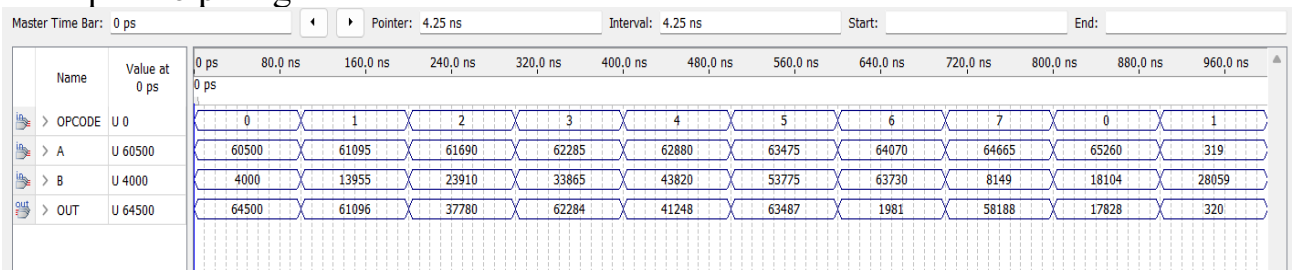
2) Thiết kết ALU

Ta kết nối khối AU và LU, dùng Mux để lựa chọn kết quả đầu ra phù hợp với yêu cầu phép toán.

-Schematic:



-Kết quả mô phỏng:



III. BÀI TẬP LÀM THÊM

Thực hiện thiết kế mạch nhân 2 số bất kì (4 bit) bằng phương pháp dịch – cộng.

THIẾT KẾ

1) Ý tưởng thực hiện

Ý tưởng nhân 2 số 4 bit bất kì sử dụng thuật toán dịch – cộng như sau:

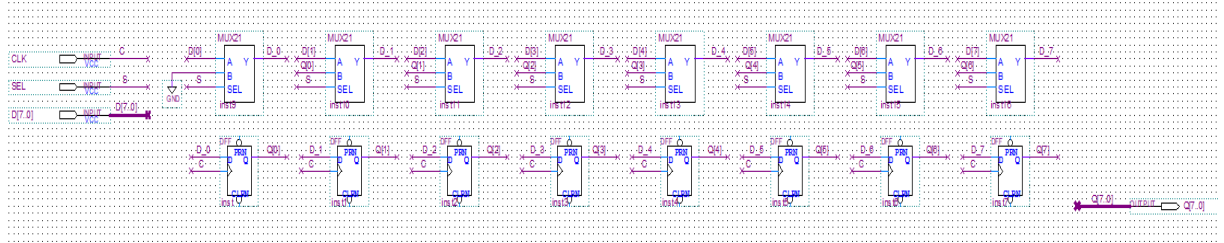
- Giả sử thực hiện phép tính $A[3..0] * B[3..0]$
 - 1) Kiểm tra bit $B[0]$ là 1 thì thực hiện $KQ = KQ + A$
 - 2) Dịch trái A 1 bit ($A \ll 1$), dịch phải B 1 bit ($B \gg 1$)
 - 3) Nếu $B \neq 0$ quay lại bước 1
 - 4) Được $KQ = A * B$ và thoát

2) Thiết kế bộ nhân

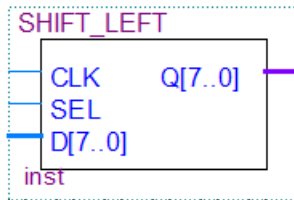
Ta sử dụng 1 thanh ghi – dịch trái 8 bit, 1 thanh ghi – dịch phải 8 bit, 1 thanh ghi 8 bit, 1 bộ cộng 8 bit và tri-state để cho phép hiển thị kết quả khi thực hiện xong phép nhân.

a) Thiết kế thanh ghi – dịch trái 8 bit

-Schematic:

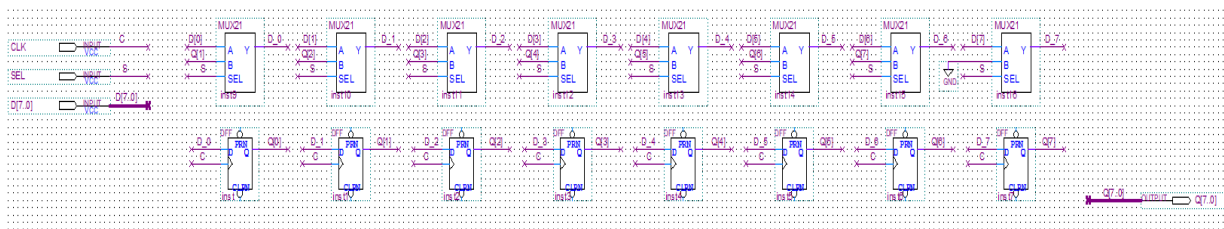


-Đóng gói:

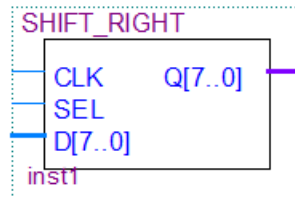


b) Thiết kế thanh ghi – dịch phải 8 bit

-Schematic:



-Đóng gói:



c) Thiết kế thanh ghi 8 bit

-Schematic:

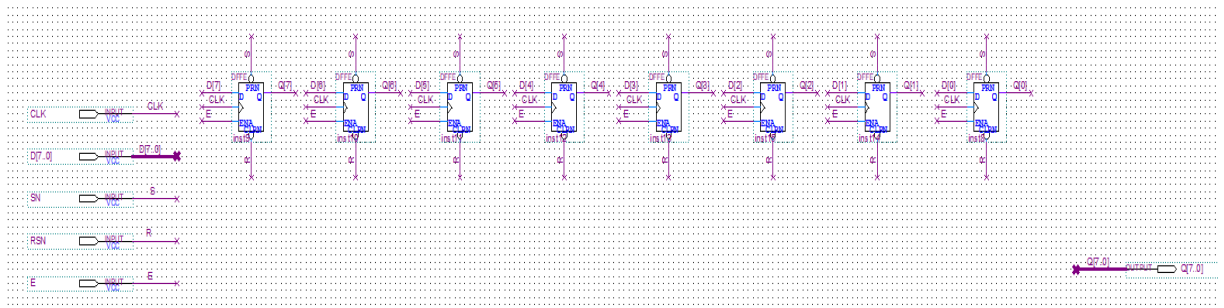
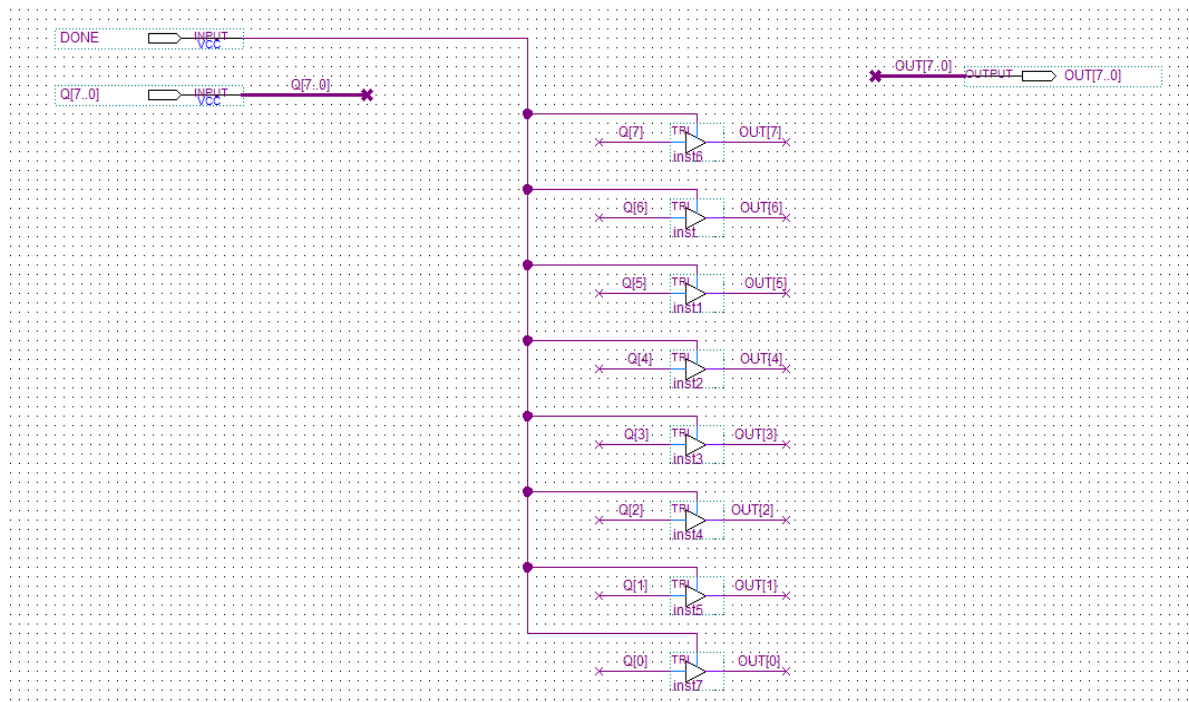


Diagram of a REGISTER component. The component is a black rectangle with a dashed border, labeled "REGISTER" at the top. It has five inputs on the left: CLK (clock), D[7..0] (data bus), SN (set), RSN (reset), and E (enable). It has one output on the right: Q[7..0] (data bus). The component is labeled "inst2" at the bottom left.

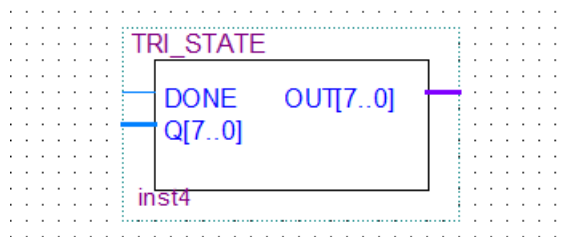
Ta tái sử dụng lại bộ cộng 8 bit đã thiết kế ở bài thực hành phía trên

Diagram of an 8-bit adder component named **ADDER_8_BIT**. The component has four inputs on the left: **CTL** (blue), **Cin** (blue), **A[7..0]** (blue), and **B[7..0]** (blue). It has two outputs on the right: **Cout** (blue) and **S[7..0]** (purple). The component is labeled **inst3** at the bottom left.

-Schematic:

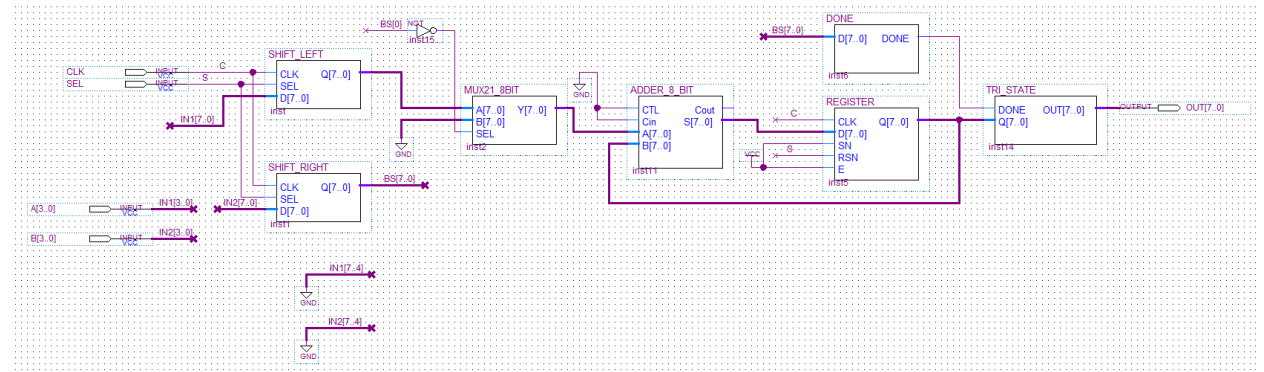


-Đóng gói:



f) Tổng hợp thành bộ nhân 2 số 4 bit bất kì

-Schematic:



-Kết quả mô phỏng:

