



LAB 3: THIẾT KẾ MẠCH TỔ HỢP TÍNH TOÁN

Trần Minh Tiến -23521586
Trần Hoàng Thông-23521529

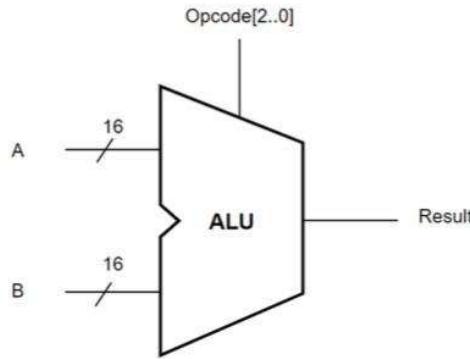
1.1 Mục tiêu

Trong bài lab này, SV thực hiện thiết kế bộ tính toán ALU. **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.

1.2 Nội dung thực hành

Sinh viên thực hiện 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, xor** theo đúng thứ tự tương ứng với tín hiệu điều khiển (Opcode) từ 0 đến 7.



Hình 1: Minh họa ALU 16-bit

Chú ý: Thiết kế thêm một ngõ ra Overflow báo hiệu tràn. Lệnh Add (cộng) và Subtract (trừ) được thực hiện trên 2 số có dấu 4-bit A và B. Kết quả sẽ được biểu diễn trong số có dấu 4bit (R).



Thực hành: Thiết kế Luận lý Số (CE118)

Cờ báo add_sub_overflow sẽ được bật lên 1 khi mạch phát hiện có overflow xảy ra. Ngoài ra sinh viên thực hiện một mạch nhân hai số 4 bit với kết quả ngõ ra lưu trong 8 bit. Gợi ý: sử dụng các bộ dịch và bộ cộng để thực hiện mạch nhân này.

1.3 Sinh viên chuẩn bị

1. Vẽ mạch thiết kế ALU 4-bit như mô tả trên và mô phỏng kiểm tra trên phần mềm Quartus.
2. Xem lại nguyên lý thực hiện phép nhân hai số và vẽ mạch thực hiện bộ nhân trên phần mềm Quartus.

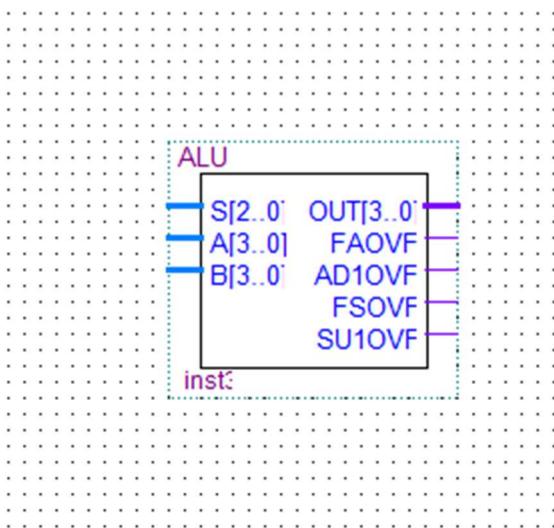
1.4 Hướng dẫn thực hành

1. Tạo một project mới, đặt tên: E/CE118_lab/lab3_MSSV
2. Thực thi thiết kế ALU và bộ nhân 4-bit với các chức năng trong phần 1.2.
3. Thiết kế và kiểm chứng hoạt động của ALU 4-bit trên phần mềm Quartus II và thực thi trên KIT DE2.
4. Thiết kế và kiểm chứng hoạt động của bộ nhân hai số 4-bit trên phần mềm Quartus II và thực thi trên KIT DE2.

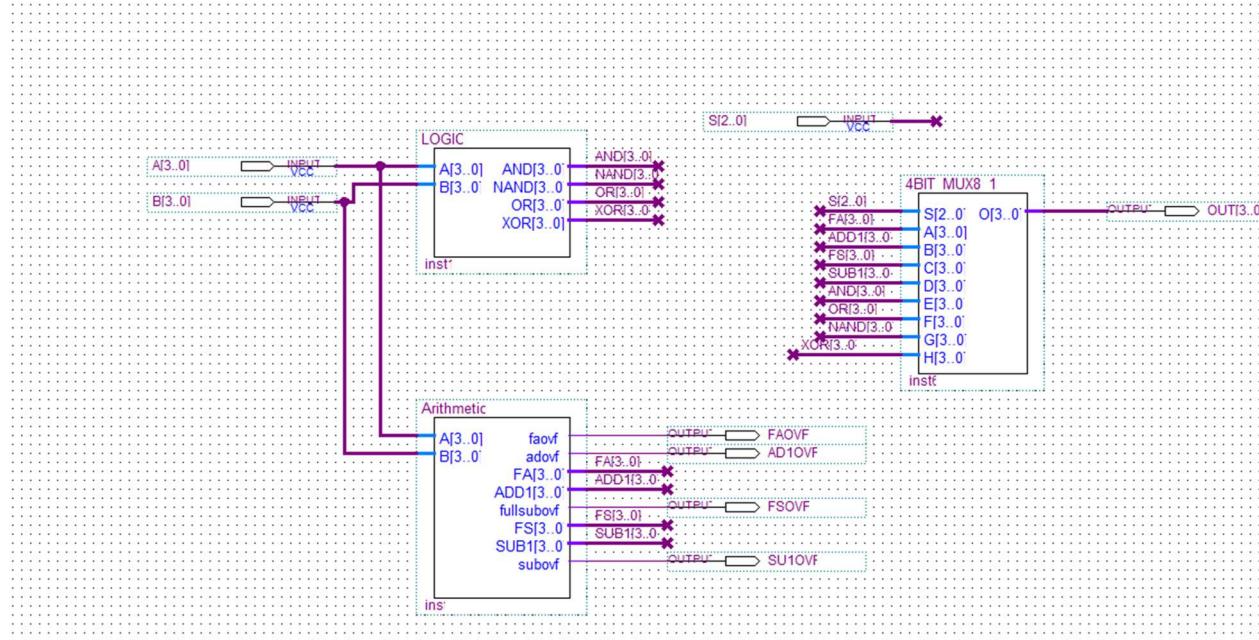
Khối ALU 4bits:



Thực hành: Thiết kế Luận lý Số (CE118)



A[3..0] và **B[3..0]** là 2 toán hạng đầu vào, **S[2..0]** là opcode để lựa chọn phép toán thực thi, **OUT[3..0]** là kết quả của sau khi thực thi, **FAOVF** là cò báo tràn của **full adder**, **AD1OVF** là cò báo tràn của **cộng 1**, **FSOVF** là cò báo tràn của **trừ**, **SU1OVF** là cò báo tràn của **trừ 1**.





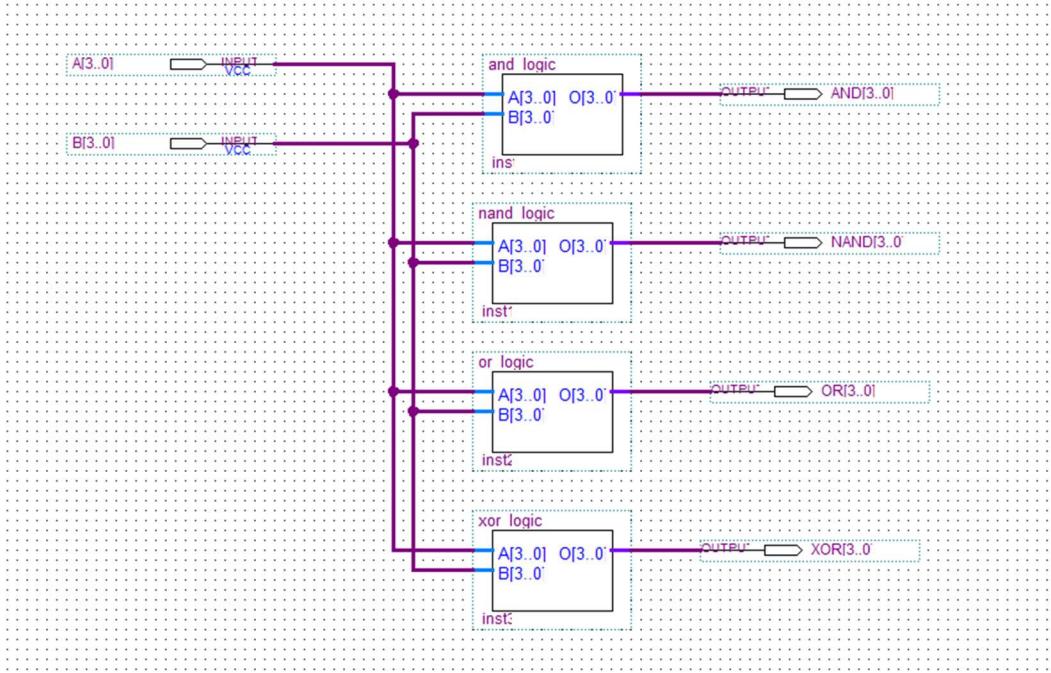


0

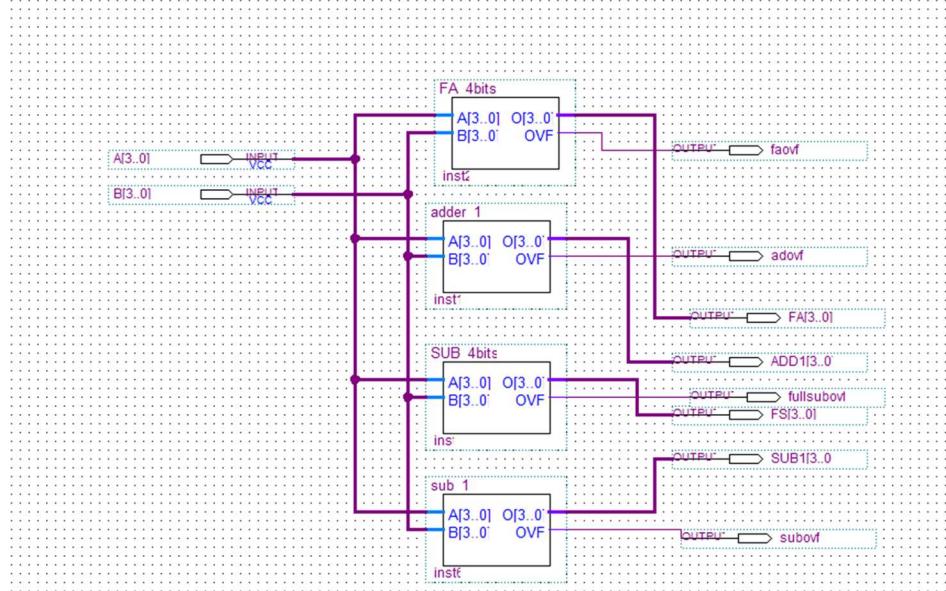
Khối Logic:



Thực hành: Thiết kế Luận lý Số (CE118)



Khối Arithmetic:



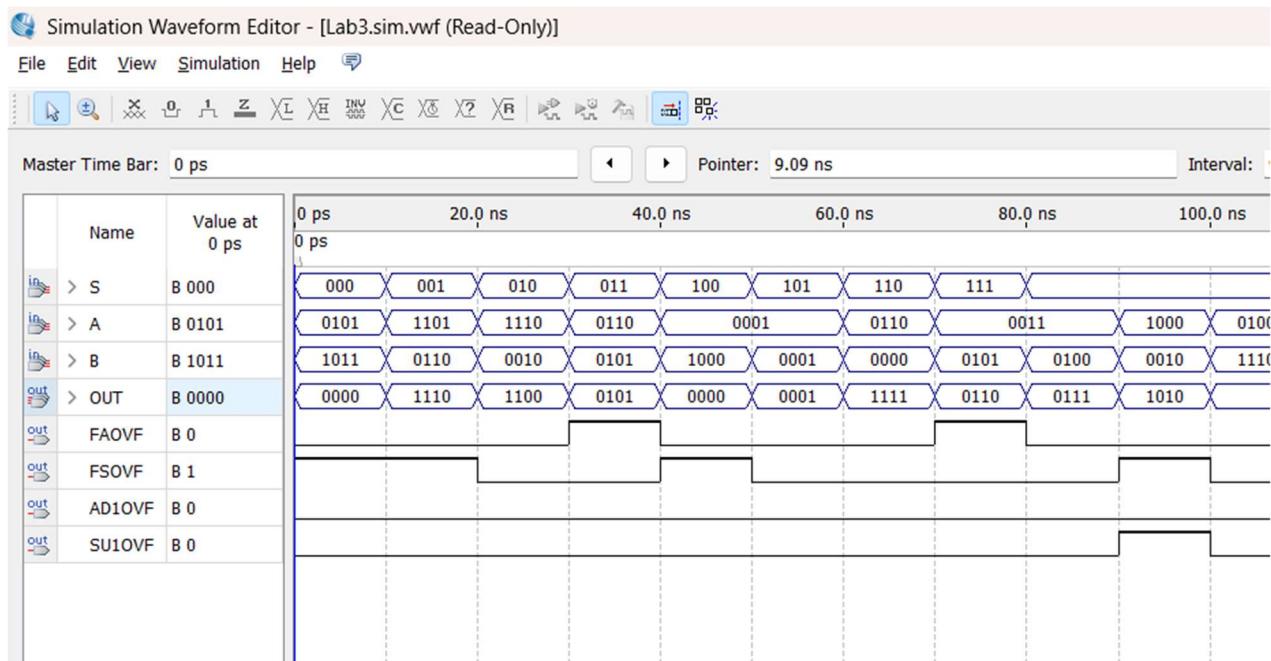
BẢNG HOẠT ĐỘNG CỦA ALU:



Thực hành: Thiết kế Luận lý Số (CE118)

A	B	C	D
S2	S1	S0	OUT
0	0	0	A+B
0	0	1	A+1
0	1	0	A-B
0	1	1	A-1
1	0	0	A AND B
1	0	1	A OR B
1	1	0	A NAND B
1	1	1	A XOR B

Waveform kiểm tra hoạt động của ALU

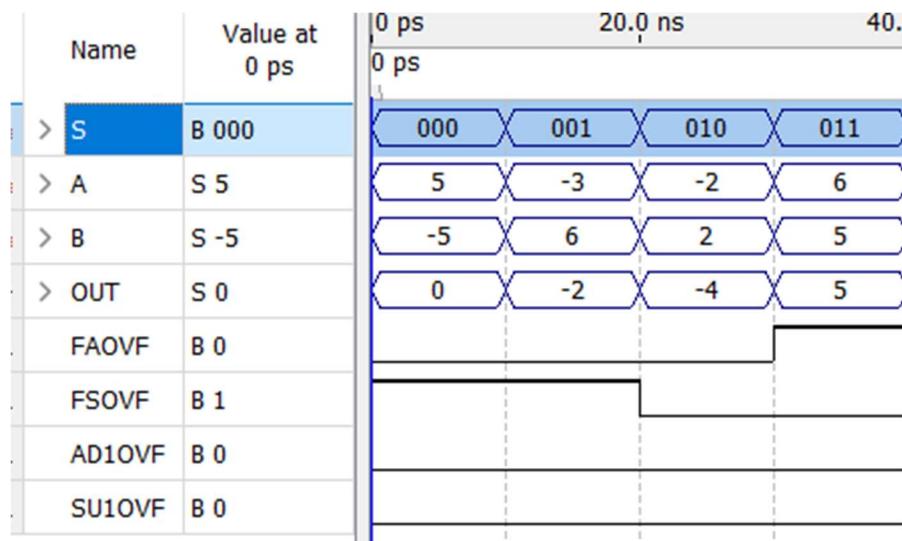


có thể đưa các chân báo tràn của các phép tính toán và logic vào cổng mux để chọn chỉ in 1 cờ, nhưng ở đây không đưa vào vì để nhận thấy các phép toán và logic đều được thực hiện đồng thời nhưng ALU chỉ chọn 1 phép ra tương tự như 1 cổng MUX to.

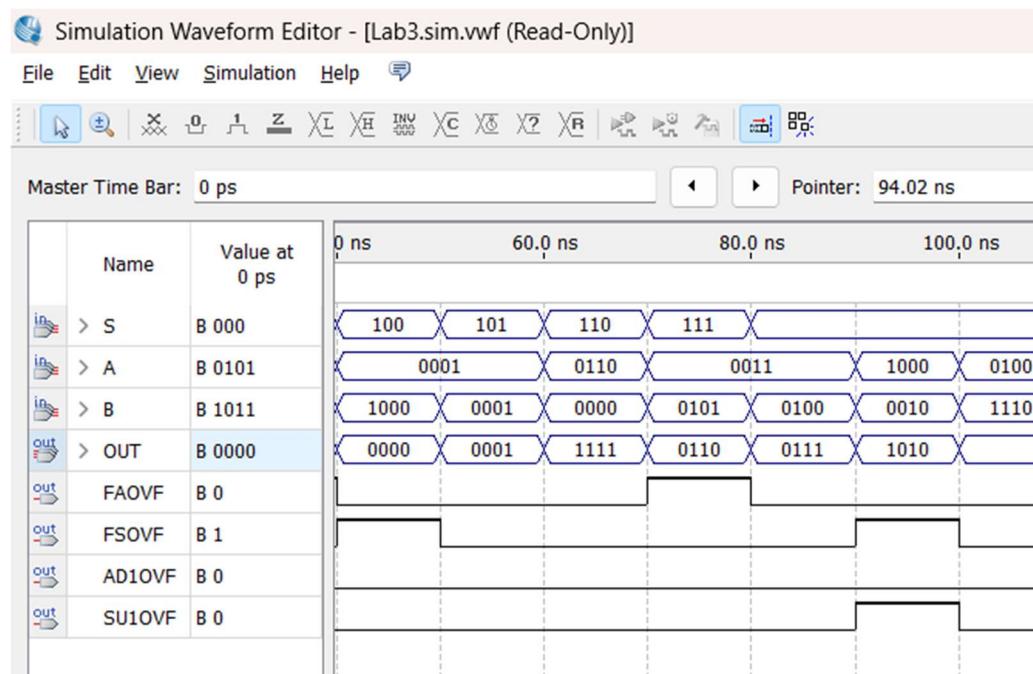


Thực hành: Thiết kế Luận lý Số (CE118)

Waveform hoạt động của khối Arithmetic:

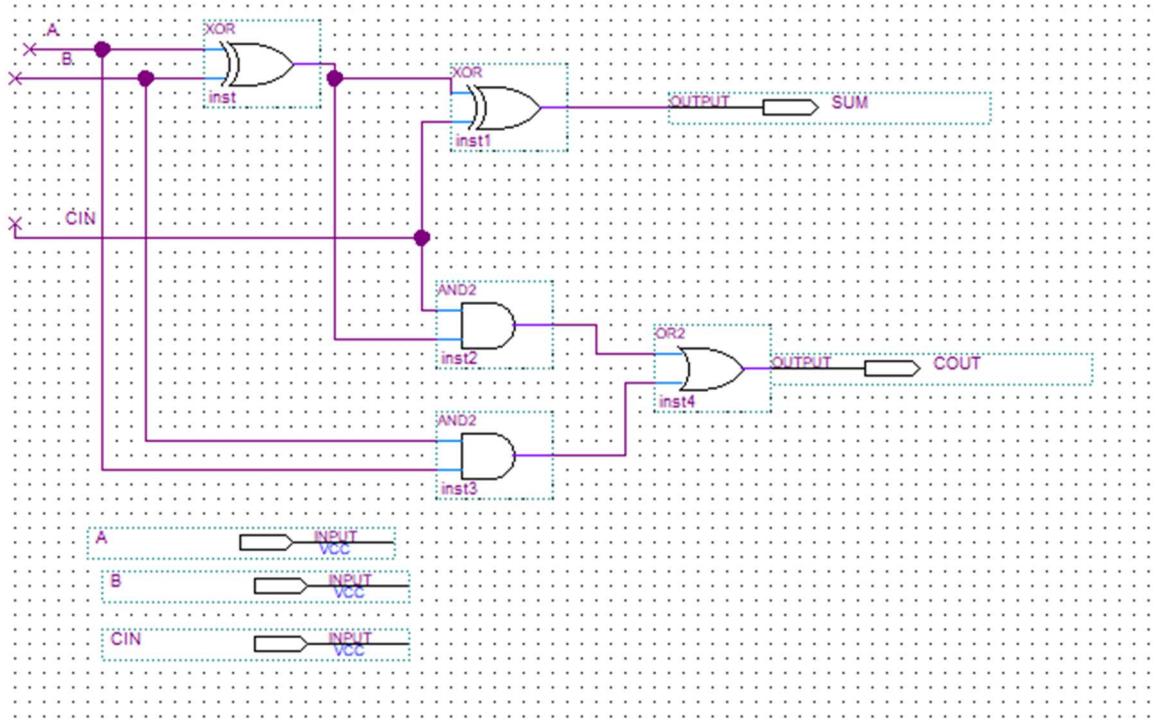


Waveform hoạt động của khối Logic:

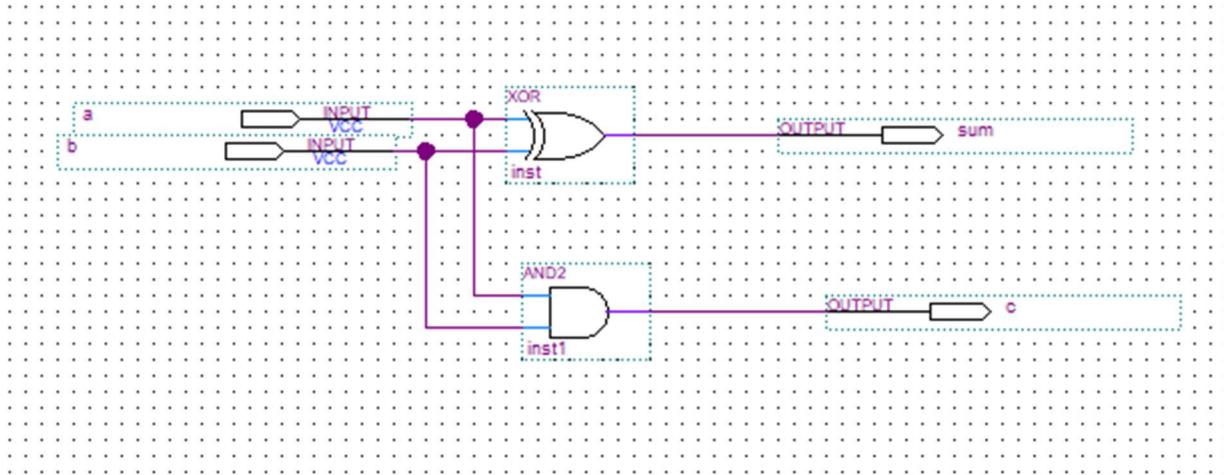


Thiết kế mạch nhân 4 bit

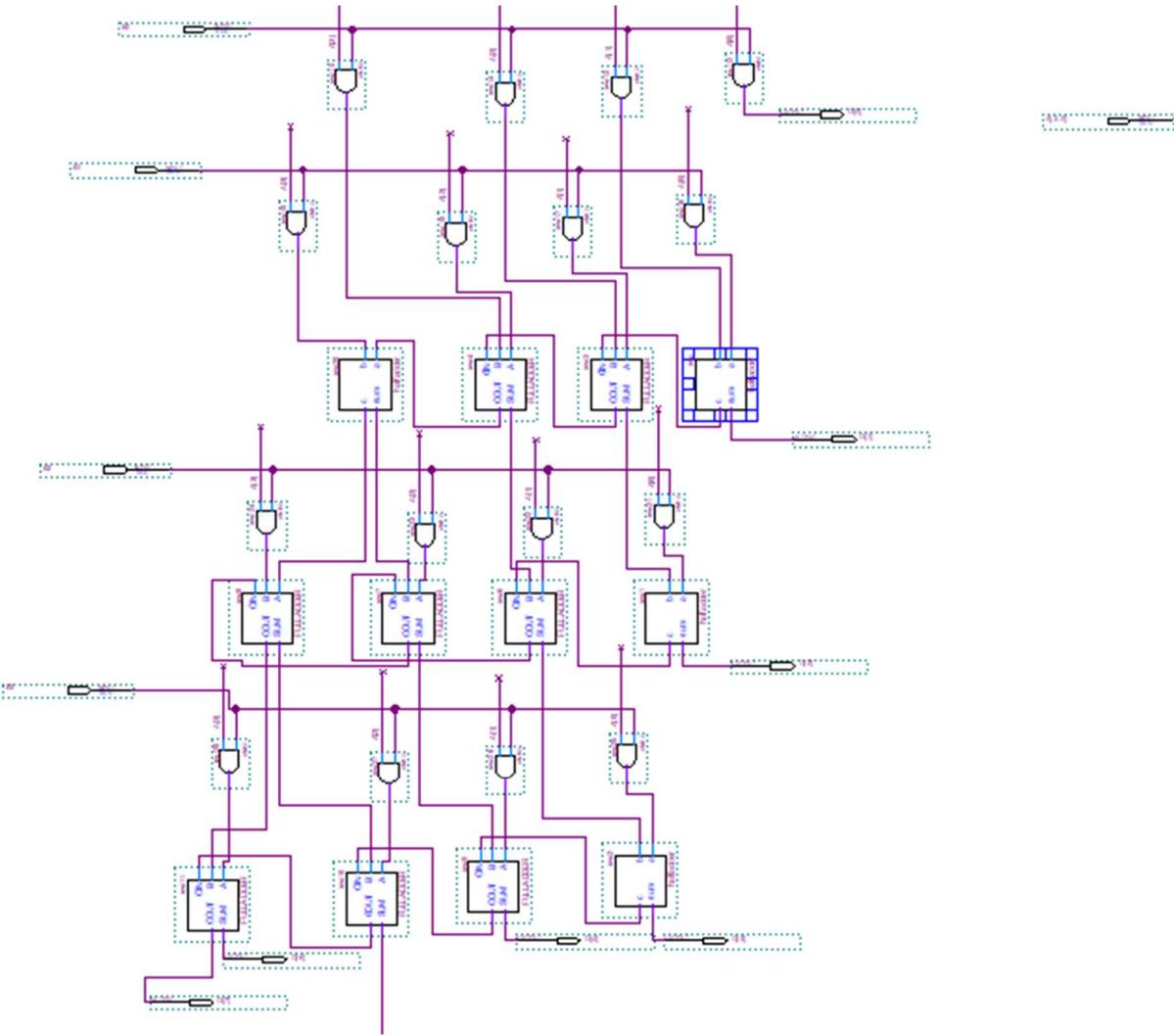
-Thiết kế Full Adder và Half Adder



(Full Adder logic)



(Half Adder logic)



Bộ nhân 4 bit theo đầu vào A[3..0] và B[3..0] và đầu ra 8 bit O[7..0]



Thuật toán dựa trên

		A3	A2	A1	A0				
	x	B3	B2	B1	B0	Inputs			
		C	B0 x A3	B0 x A2	B0 x A1	B0 x A0			
+		B1 x A3	B1 x A2	B1 x A1	B1 x A0				
	C	sum	sum	sum	sum	Internal Signals			
+		B2 x A3	B2 x A2	B2 x A1	B2 x A0				
	C	sum	sum	sum	sum				
+		B3 x A3	B3 x A2	B3 x A1	B3 x A0				
	C	sum	sum	sum	sum				
	P7	P6	P5	P4	P3	P2	P1	P0	Outputs

Ý tưởng thuật toán Multiplier là bit LSB của B sẽ AND với từng bit của A tương tự phép nhân 1 bit và sau đó giữ lại bit LSB của kết quả phép nhân và lặp lại tiếp tục đến khi đến MSB của B tương tự như 1 mạch Shift.