

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



SQUARE-ROOT APPROXIMATION PROCESSOR

SINH VIÊN: NGUYỄN GIA BẢO NGỌC – 21520366

GIẢNG VIÊN HƯỚNG DẪN:

TS-LÂM ĐỨC KHẢI

TP. HỒ CHÍ MINH – Tháng 6 năm 2023

MỤC LỤC:

Danh mục ảnh:II

Danh mục bảng: III

I. TỔNG QUAN THỰC HIỆN:1

I.1. Ý tưởng thực hiện:1

I.2. Các bước thực hiện:2

I.3. Tổng quan thiết kế:3

II. CHI TIẾT THỰC HIỆN:4

II.1. Thực hiện khối Datapath:4

II.2.1 THỰC HIỆN KHỐI CONTROLLER:10

III. KẾT QUẢ CHẠY MÔ PHỎNG15

Danh mục ảnh:

Hình 1 - Thuật toán thực hiện.....	1
Hình 2 - Lưu đồ giải thuật	1
Hình 3 - Các bước thực hiện.....	2
Hình 4 - Tổng quan thiết kế kết.....	3
Hình 5 - Khối Datapath	4
Hình 6 - Mạch "Sub_module"	5
Hình 7 - Mạch Add_or_Sub_module	5
Hình 8 - Mạch Mux2-1	6
Hình 9 - Mạch Mux2-1 8bit.....	6
Hình 10 - Khối AU1	7
Hình 11 - Khối AU2	8
Hình 12 - Bảng điều khiển khối AU2.....	8
Hình 13 - Mạch Shift_right1	9
Hình 14 - Mạch Shift_right3	9
Hình 15 - Khối Controller	10
Hình 16 - Lưu đồ chuyển trạng thái	10
Hình 17 - Khối OUTPUT_FUNCT	13
Hình 18 - Khối REG.....	14
Hình 19 - Khối NEXT_STA.....	15

Danh mục bảng:

Bảng 1 - Giảm đồ thời gian	2
Bảng 2 - Bảng hoạt động khối AU1 thiết kế kết hợp pipeline khối chức năng và pipeline đường dữ liệu	7
Bảng 3 - Bảng chân trị của các tín hiệu điều khiển ứng với từng trạng thái	11
Bảng 4 - Bảng chân trị cho mạch chuyển trạng thái khối Controller	14

VI XỬ LÝ TÍNH GIÁ TRỊ XẤP XỈ CĂN BẬC HAI CỦA HAI SỐ NGUYÊN

I. TỔNG QUAN THỰC HIỆN:

I.1. Ý tưởng thực hiện:

Vi xử lý dựa trên thuật toán sau thực hiện tính toán giá trị căn bậc hai của 2 số nguyên bất kì:

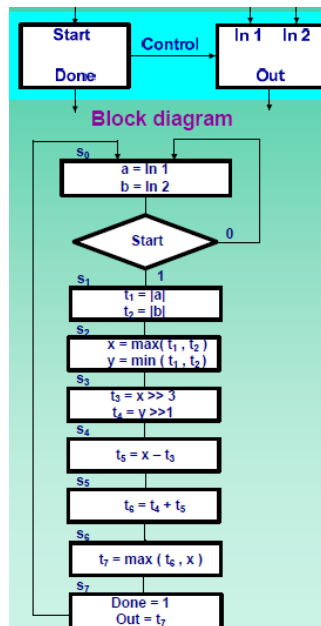
$$\sqrt{a^2+b^2} \approx \max((0,875x+0,5y),x)$$

where $x = \max(a,b)$

$y = \min(a,b)$

Hình 1 - Thuật toán thực hiện

Sau khi xác định được công thức thực hiện, bước tiếp theo ta xác định lưu đồ thuật toán để thực hiện giải thuật trên. Thuật toán trên có thể được thực hiện thông qua lưu đồ sau:



Hình 2 - Lưu đồ giải thuật

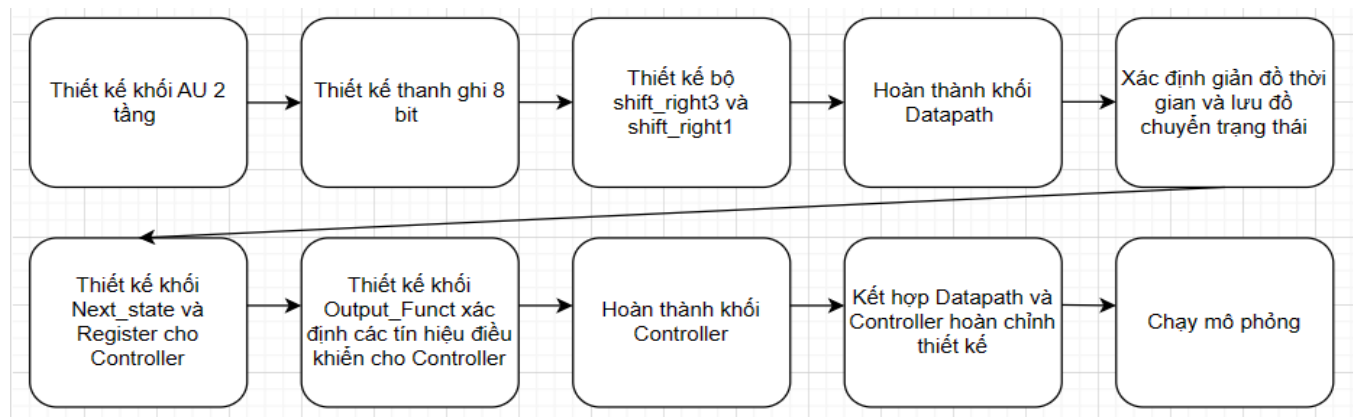
Kiến trúc của vi xử lý có thể được tổng quát gồm 2 thành phần chính là khối Controller và khối Datapath. Theo đó, khối Controller sẽ là được thiết kế như là một máy trạng thái, Datapath sẽ bao gồm các thành phần tính toán và thành phần lưu trữ và sẽ được điều khiển bởi các tín hiệu được sinh ra phụ thuộc trạng thái tại khối Controller. Datapath sẽ được thiết kế để phục vụ kỹ thuật Pipeline nhằm tăng hiệu suất hoạt động. Để cho ra kết quả cuối cùng cần sự phối hợp chính xác giữa Datapath

và Controller. Dưới đây, ta xác định được lưu đồ thời gian, các công việc ứng với từng trạng thái hoạt động.

Bảng 1 - Giản đồ thời gian

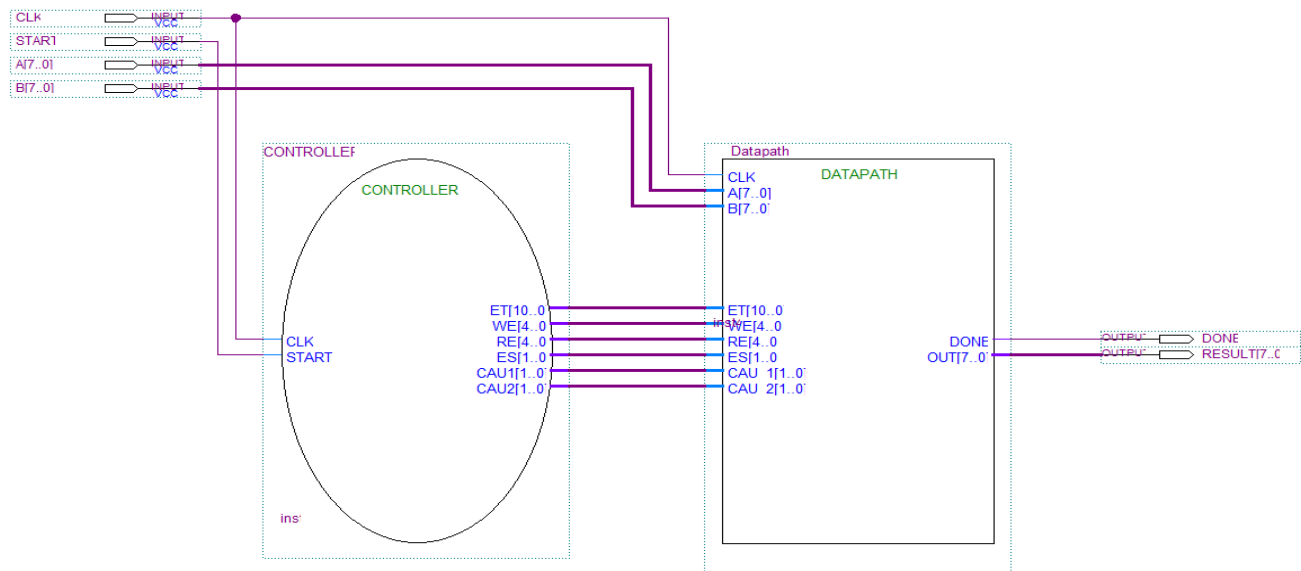
	S_0	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	S_9	S_{10}	S_{11}	S_{12}
Đọc R_1		a			t_1	t_1			a			t_1	t_1
Đọc R_2			b		t_2	t_2				b		t_2	t_2
AU_1 tầng 1		a	b		min	max			a	b		min	max
AU_1 tầng 2			a	b		min	max			a	b		min
Dịch bit						>>1	>>3						>>1
Ghi vào R_1	a		t_1					a		t_1			
Ghi vào R_2	b			t_2				b			t_2		
Đọc R_3								t_3		t_5		t_6	
Đọc R_4								x				x	
Đọc R_5										t_4			
AU_2 tầng 1								-		+			
AU_2 tầng 2											+		
Ghi vào R_3							t_3		t_5		t_6		
Ghi vào R_4							x						
Ghi vào R_5						t_4							t_4
Xuất ngõ ra													t_7

1.2. Các bước thực hiện:



Hình 3 - Các bước thực hiện

1.3. Tổng quan thiết kế:

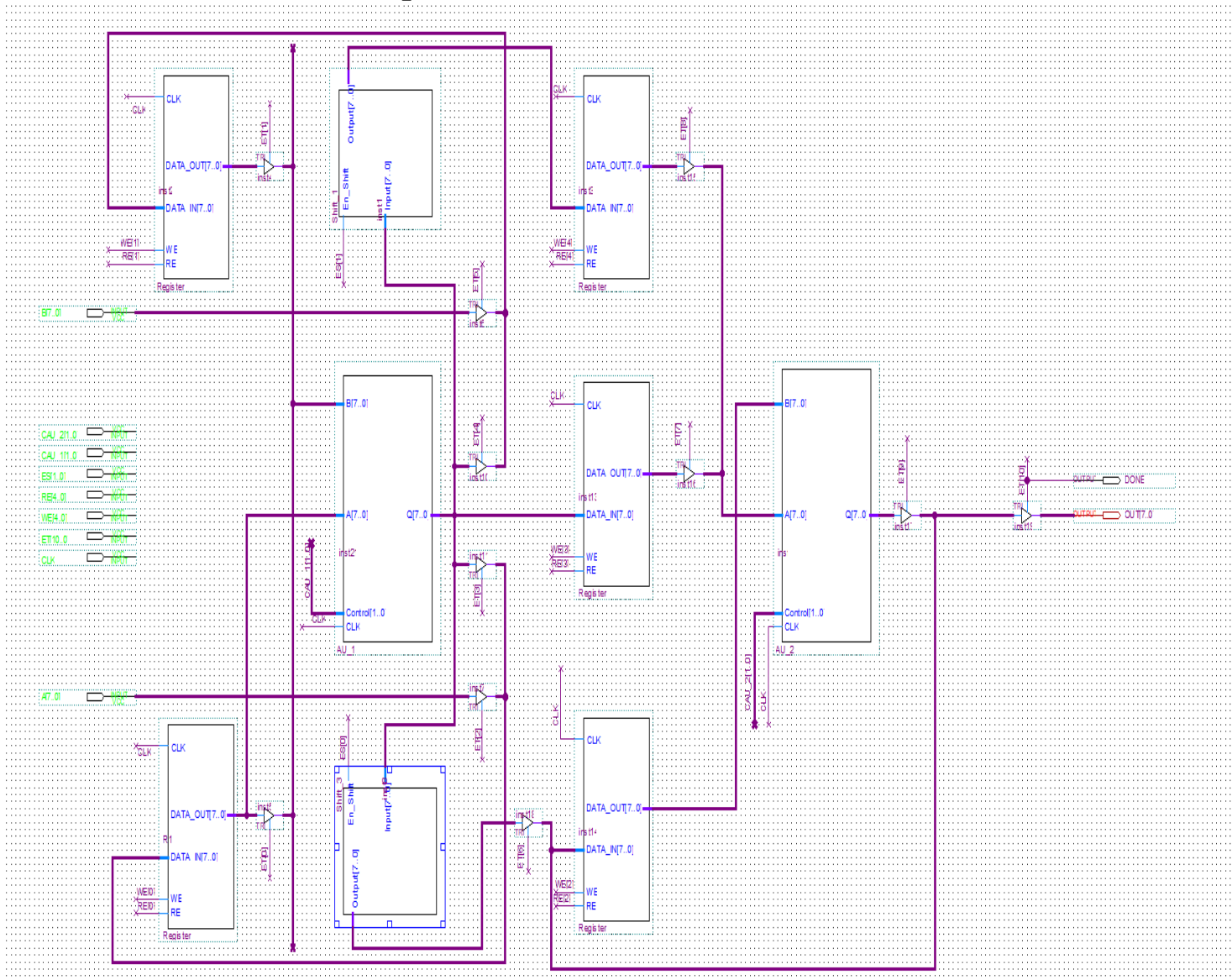


Hình 4 - Tổng quan thiết kế kết

- Ngõ vào A[7..0] mang giá trị toán hạng 1.
- Ngõ vào B[7..0] mang giá trị toán hạng 2.
- Tín hiệu Start, để khởi đầu hệ thống.
- Ngõ ra RESULT[7..0] là kết quả của phép tính.
- Tín hiệu DONE để báo hoàn thành việc tính toán.

II. CHI TIẾT THỰC HIỆN:

II.1. Thực hiện khối Datapath:

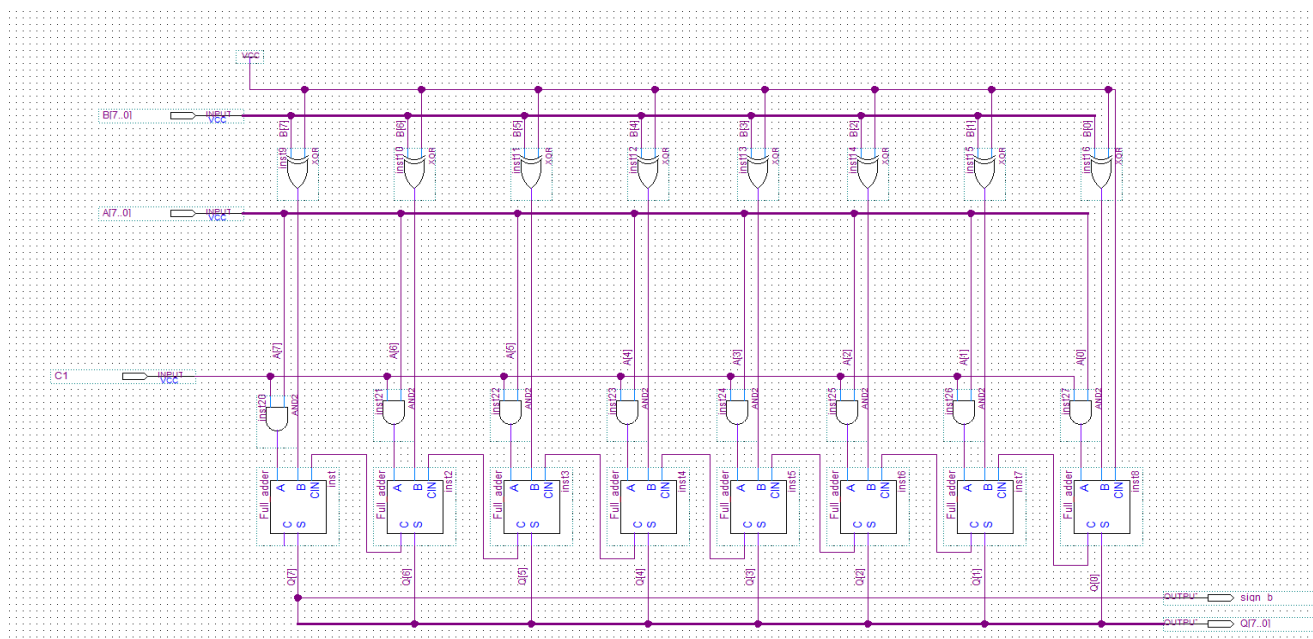


Hình 5 - Khối Datapath

Tín hiệu ET[10] đồng thời cũng là tín hiệu DONE.

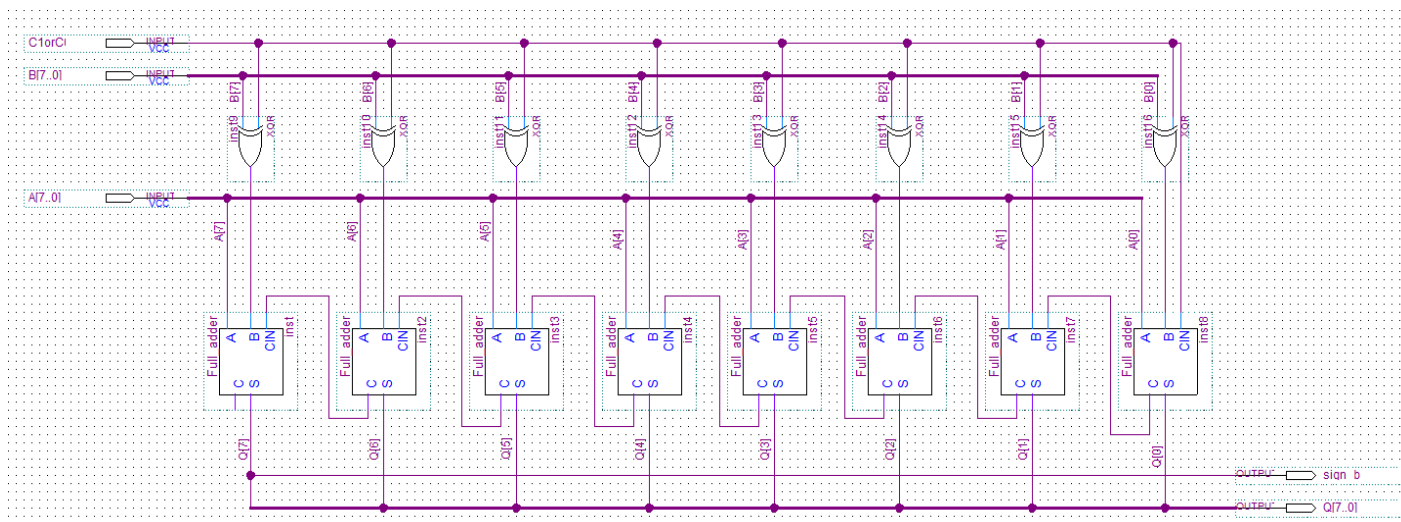
II.1.1 Các khối cơ bản tính toán cơ bản (“Sub_module”, “Add_or_Sub_module”, “Mux2-1_8bit”):

Khối “Sub_module” : có chức năng thực hiện phép trừ dựa vào tín hiệu điều khiển C1. Khối thực hiện phép trừ bằng cách chuyển số bị trừ thành dạng bù 2 sau đó dùng các “Full_adder” để thực hiện phép cộng. Đồng thời ghi nhận “bit dấu” của kết quả.



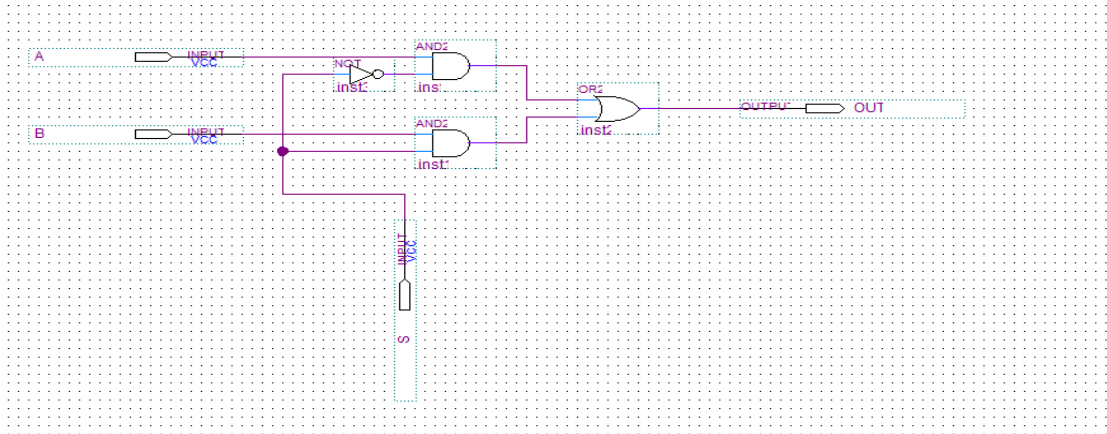
Hình 6 - Mạch "Sub_module"

Khối “Add_or_Sub_module” : có chức năng thực hiện phép cộng hoặc phép trừ dựa vào tín hiệu điều khiển “C₁ or C0”. Khối thực hiện phép cộng như bình thường bằng các khối “Full_adder”. Khối thực hiện phép trừ bằng cách chuyển số bị trừ thành dạng bù 2 sau đó dùng các “Full_adder” để thực hiện phép cộng. Đồng thời ghi nhận “bit dấu” của kết quả.

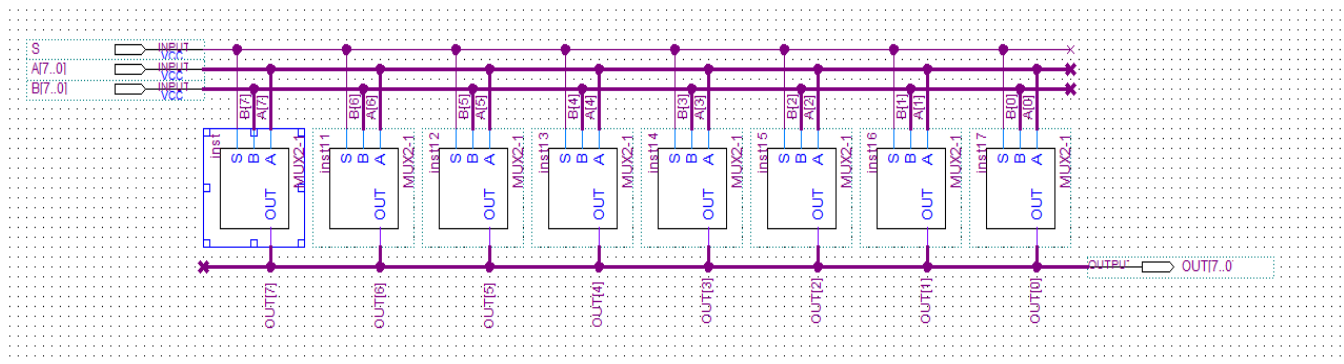


Hình 7 - Mạch Add_or_Sub_module

Khối “Mux2-1_8bit” : Tạo thành từ các phần tử là các mạch “Mux2-1” có tác dụng chọn dữ liệu đầu ra từ 2 dữ liệu đầu vào dựa trên tín hiệu điều khiển.



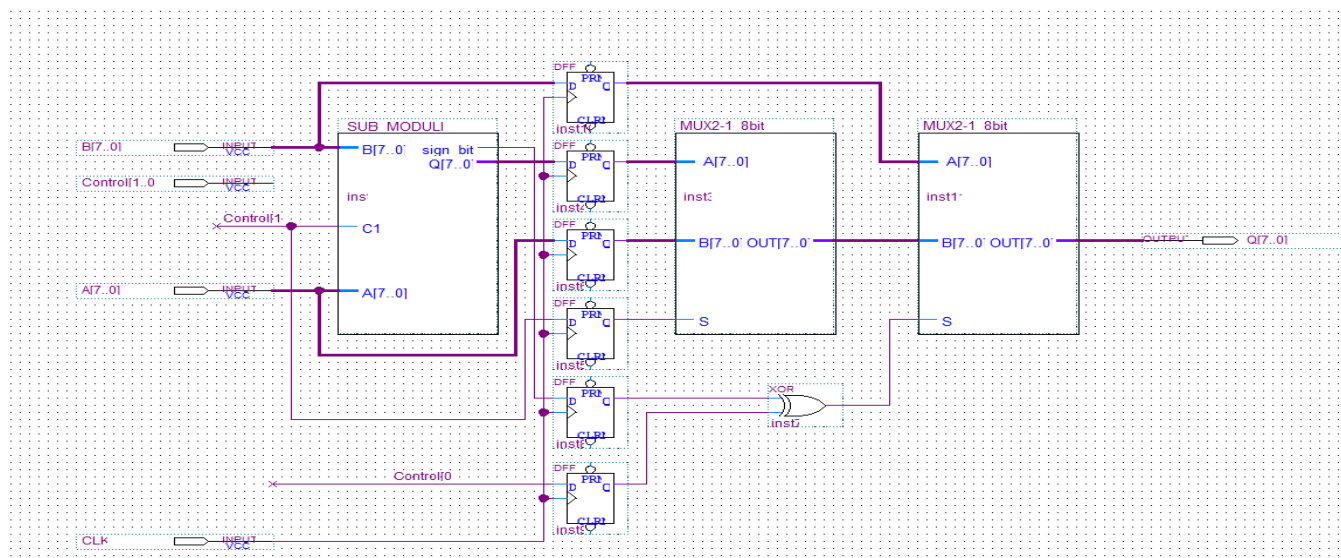
Hình 8 - Mạch Mux2-1



Hình 9 - Mạch Mux2-1 8bit

II.1.2 Khối AU1 (thực hiện các phép tính min, max, trị tuyệt đối):

Phép toán cơ bản của khối là **thực hiện phép trừ**, sau đó từ kết quả tính toán số học qua các bộ “Mux2-1_8bit” để chọn ra **giá trị min, max và trị tuyệt đối**. Thiết kế của khối được chia thành 2 tầng bởi các mạch flip-flop, các flip-flop các tác dụng lưu trữ các giá trị tính toán từ bộ tính toán số học và các tín hiệu điều khiển cho các bộ “Mux2-1 8bit”, do đó bộ “Sub_module” có thể phép tính mới mà không cần đợi các bộ “Mux2-1 8 bit” thực hiện xong.



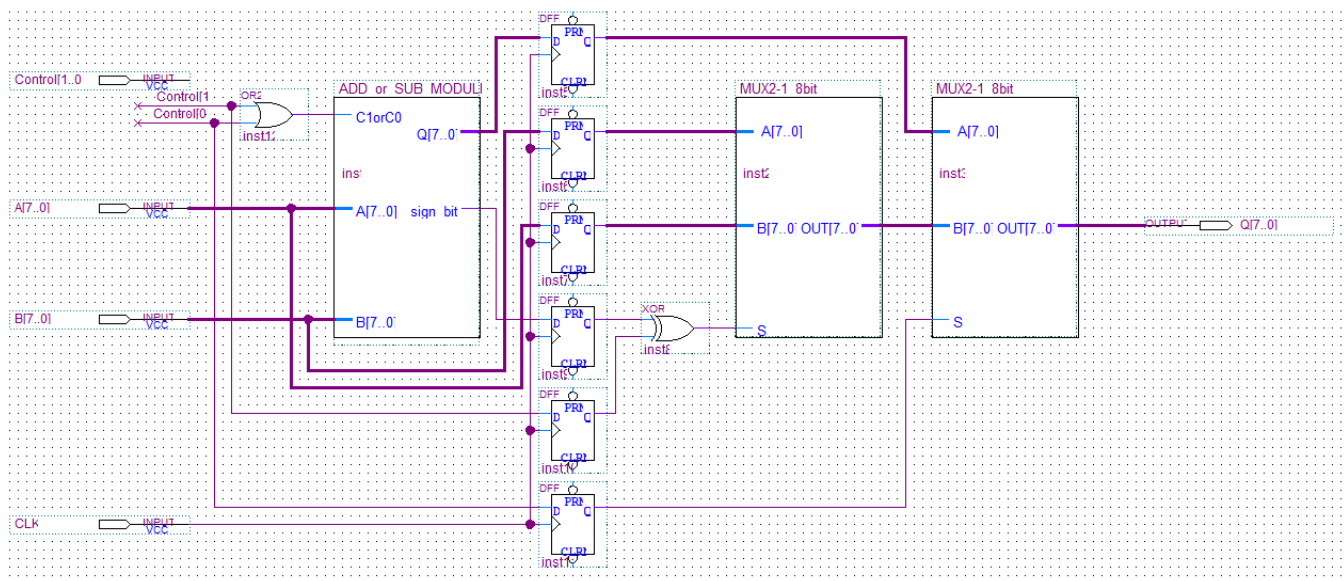
Hình 10 - Khối AU1

Bảng 2 - Bảng hoạt động khối AU1 thiết kế kết hợp pipeline khối chức năng và pipeline đường dữ liệu

Control[1]	Control[0]	Toán hạng
0	1	<i>Absolute</i>
1	0	<i>Minimum</i>
1	1	<i>Maximum</i>

II.1.3 Khối AU2 (thực hiện các phép tính cộng, trừ, min, max):

Phép toán cơ bản của khối là thực hiện phép **cộng và trừ**, sau đó từ kết quả tính toán số học qua các bộ “Mux2-1 8bit” để chọn ra **giá trị min, max**. Thiết kế của khối được chia thành 2 tầng bởi các mạch flip-flop, các flip-flop các tác dụng lưu trữ các giá trị tính toán từ bộ tính toán số học và các tín hiệu điều khiển cho các bộ “Mux2-1 8bit”, do đó bộ “Add_or_Sub_module” có thể phép tính mới mà không cần đợi các bộ “Mux2-1 8 bit” thực hiện xong.

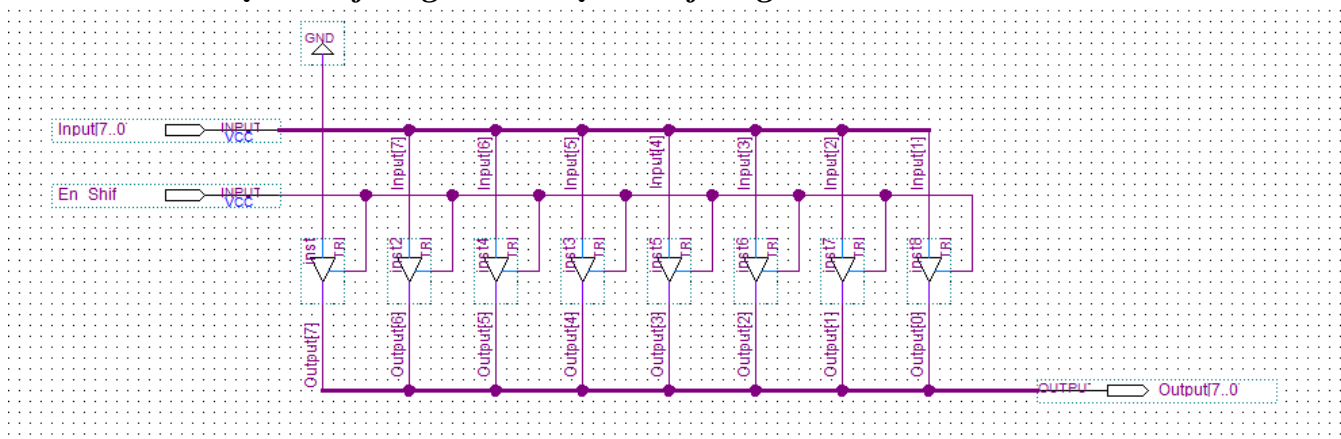


Hình 11 - Khối AU2

c_1	c_0	Operation
0	0	addition
0	1	minimum
1	0	subtraction
1	1	maximum

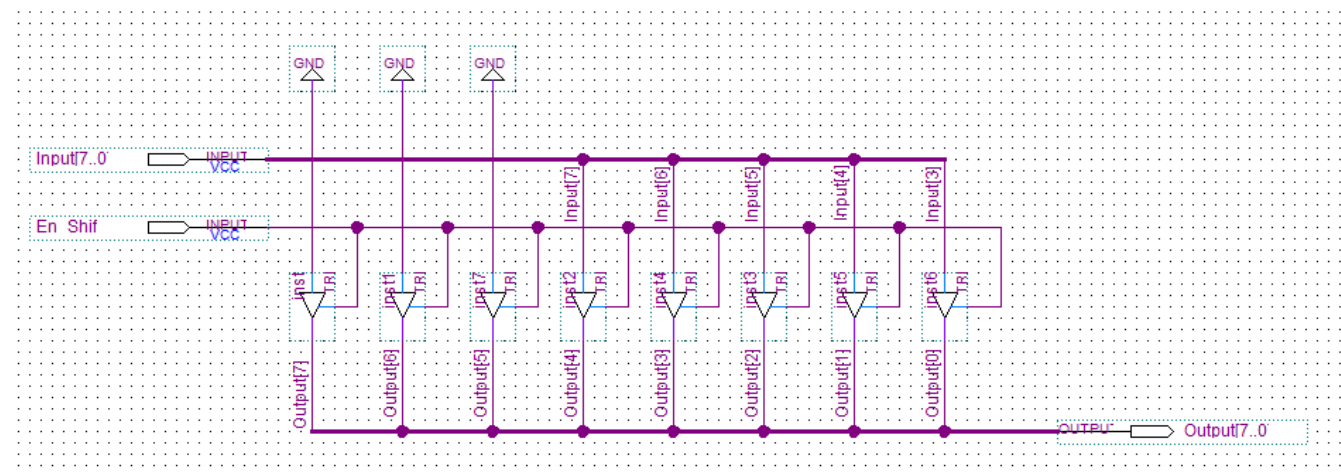
Hình 12 - Bảng điều khiển khối AU2

II.1.4 Mạch Shift_right3 và mạch Shift_right1:



Hình 13 - Mạch Shift_right1

Mạch Shift_right1: Tín hiệu En_Shif điều khiển việc có cho phép dịch phải 1 bit hay không.

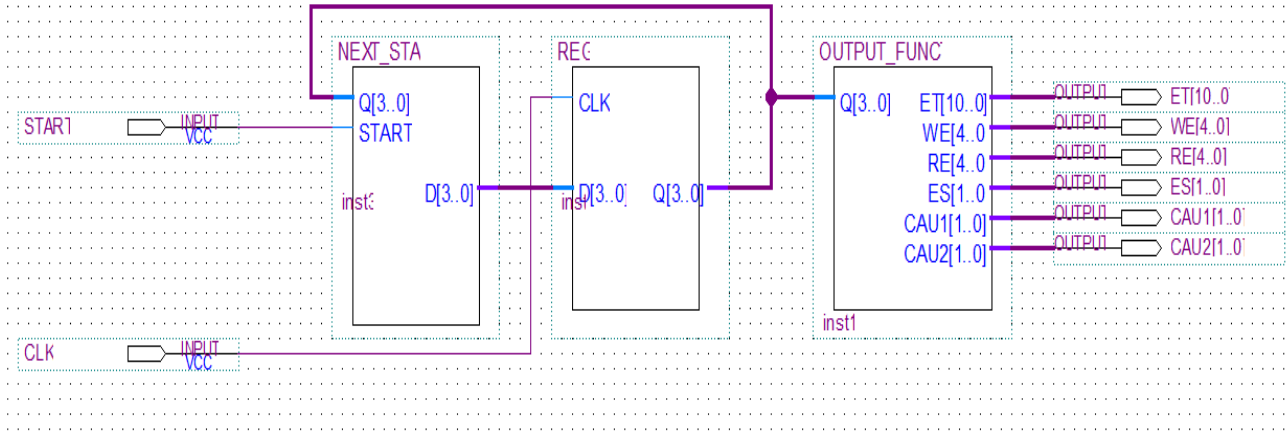


Hình 14 - Mạch Shift_right3

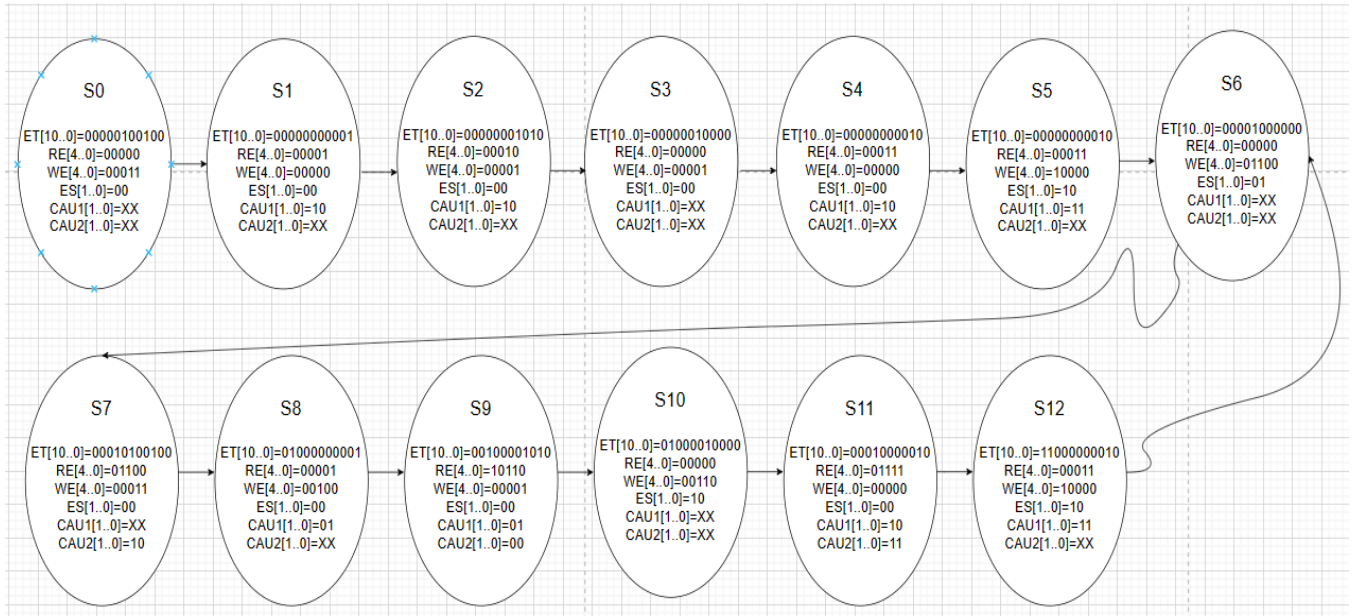
Mạch Shift_right3: Tương tự như mạch Shift_right1 tín hiệu En_Shif điều khiển việc có cho phép dịch phải 3 bit hay không.

II.2.1 THỰC HIỆN KHỐI CONTROLLER:

CONTROLLER thiết kế theo kiến trúc FSM gồm 3 khối NEXTSTATE, REG, OUTPUT_FUNC



Hình 15 - Khối Controller



Hình 16 - Lưu đồ chuyển trạng thái

- ET[10..0] là mảng các tín hiệu điều khiển các cổng tri-sta.
- RE[4..0] là mảng các tín hiệu điều khiển các việc đọc các thanh ghi theo thứ tự R5, R4, R3, R2, R1.
- WE[4..0] là mảng các tín hiệu điều khiển các việc ghi các thanh ghi theo thứ tự R5, R4, R3, R2, R1.

- ES[1..0] trong đó ES[1] điều bộ dịch phải 3 bit, ES[0] điều khiển bộ dịch phải 1 bit.
- CAU1[1..0] là các tín hiệu điều khiển khối AU tầng 1.
- CAU2[1..0] là các tín hiệu điều khiển khối AU tầng 2.

Từ sơ đồ chuyển trạng thái trên ta có thể rút ra được bảng chân trị cho từng ngõ điều khiển ứng với từng trạng thái như sau:

Bảng 3 - Bảng chân trị của các tín hiệu điều khiển ứng với từng trạng thái

	S_0	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	S_9	S_{10}	S_{11}	S_{12}
ET_0		1							1				
ET_1			1		1	1				1		1	1
ET_2	1							1					
ET_3			1							1			
ET_4				1							1		
ET_5	1							1					
ET_6							1						
ET_7								1				1	
ET_8										1		1	
ET_9									1		1		1
ET_{10}													1
WR_0	1		1					1		1			
WR_1	1			1				1			1		
WR_2							1		1		1		
WR_3							1						
WR_4						1							1
RE_0		1			1	1			1			1	1
RE_1			1		1	1				1		1	1
RE_2								1		1		1	
RE_3								1				1	
RS_4										1			
ES_0							1						
ES_1						1							1
CAU1	XX	01	01	XX	10	11	XX	XX	01	01	XX	10	11
CAU2	XX	XX	XX	XX	XX	XX	XX	10	XX	00	XX	11	XX

Ta có hàm ngõ ra cho các tín hiệu điều khiển cũng chính là ngõ ra khối **OUTPUT của Controller**, có dạng:

$$ET_0 = \overline{Q_3} \overline{Q_2} \overline{Q_1} Q_0 + Q_3 \overline{Q_2} \overline{Q_1} \overline{Q_0}$$

$$ET_1 = \overline{Q_3} \overline{Q_2} Q_1 \overline{Q_0} + \overline{Q_3} Q_2 \overline{Q_1} + Q_2 \overline{Q_1} \overline{Q_0} + Q_3 \overline{Q_2} Q_0$$

$$ET_2 = \overline{Q_3} \overline{Q_2} \overline{Q_1} \overline{Q_0} + \overline{Q_3} Q_2 Q_1 Q_0$$

$$ET_3 = \overline{Q_3} \overline{Q_2} Q_1 \overline{Q_0} + Q_3 \overline{Q_2} \overline{Q_1} Q_0$$

$$ET_4 = \overline{Q_3} \overline{Q_2} Q_1 Q_0 + Q_3 \overline{Q_2} \overline{Q_1} \overline{Q_0}$$

$$ET_5 = \overline{Q_3} \overline{Q_2} \overline{Q_1} Q_0 + \overline{Q_3} Q_2 Q_1 Q_0$$

$$ET_6 = \overline{Q_3} Q_2 Q_1 \overline{Q_0}$$

$$ET_7 = Q_3 \overline{Q_2} Q_1 Q_0 + \overline{Q_3} Q_2 Q_1 Q_0$$

$$ET_8 = Q_3 \overline{Q_2} Q_0$$

$$ET_9 = Q_3 \overline{Q_2} \overline{Q_0} + Q_3 \overline{Q_1} \overline{Q_0}$$

$$ET_{10} = Q_3 Q_2 \overline{Q_1} \overline{Q_0}$$

$$WE_0 = Q_3 \overline{Q_2} \overline{Q_1} Q_0 + \overline{Q_3} Q_2 Q_1 Q_0 + \overline{Q_3} \overline{Q_2} \overline{Q_0}$$

$$ALU_1[0] = \overline{Q_3} \overline{Q_2} + \overline{Q_3} Q_0 + Q_3 \overline{Q_1}$$

$$ALU_1[1] = Q_2 + Q_1 Q_0$$

$$ALU_2[0] = \overline{Q_2} Q_1$$

$$ALU_2[1] = Q_1$$

$$WE_1 = \overline{Q_3} \overline{Q_2} \overline{Q_1} \overline{Q_0} + Q_3 \overline{Q_2} Q_1 \overline{Q_0} + \overline{Q_3} Q_1 Q_0$$

$$WE_2 = \overline{Q_3} Q_2 Q_1 \overline{Q_0} + Q_3 \overline{Q_2} \overline{Q_0}$$

$$WE_3 = \overline{Q_3} Q_2 Q_1 \overline{Q_0}$$

$$WE_4 = \overline{Q_3} Q_2 \overline{Q_1} Q_0 + Q_3 \overline{Q_2} Q_1 \overline{Q_0}$$

$$RE_0 = Q_3 \overline{Q_2} \overline{Q_1} Q_0 + Q_3 \overline{Q_1} \overline{Q_0} + Q_3 \overline{Q_2} \overline{Q_0} + \overline{Q_3} Q_2 \overline{Q_1}$$

$$RE_1 = \overline{Q_3} \overline{Q_2} Q_1 \overline{Q_0} + Q_3 \overline{Q_2} Q_0 + Q_2 \overline{Q_1} \overline{Q_0}$$

$$RE_2 = \overline{Q_3} Q_2 Q_1 Q_0 + Q_3 \overline{Q_2} Q_0$$

$$RE_3 = \overline{Q_3} Q_2 Q_1 Q_0 + Q_3 \overline{Q_2} Q_1 Q_0$$

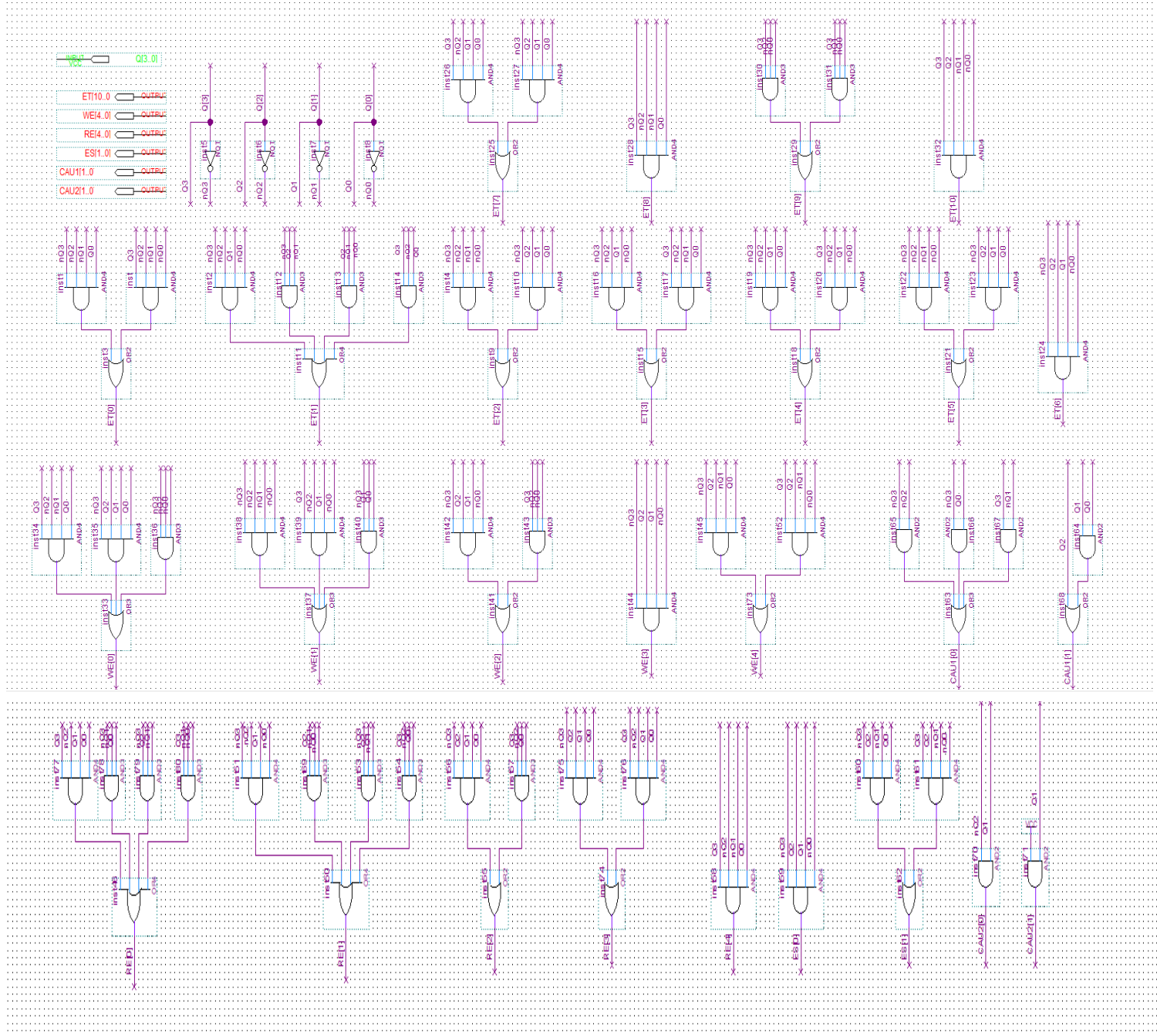
$$RE_4 = Q_3 \overline{Q_2} \overline{Q_1} Q_0$$

$$ES_0 = \overline{Q_3} Q_2 Q_1 \overline{Q_0}$$

$$ES_1 = \overline{Q_3} Q_2 \overline{Q_1} Q_0 + Q_3 Q_2 \overline{Q_1} \overline{Q_0}$$

I. Khối OUTPUT_FUNCT:

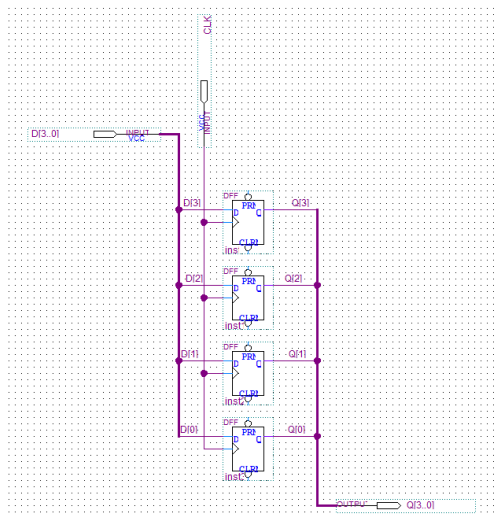
Từ hàm ngõ ra của các tín hiệu điều khiển trên, ta có thiết kế mạch OUTPUT_FUNCT, như sau:



Hình 17 - Khối OUTPUT_FUNCT

II.2.2 Khối REG:

Khối REG có chức năng lưu trữ trạng thái hiện tại, là ngõ vào cho khối OUTPUT_FUNCT tạo tín hiệu điều khiển, theo lưu đồ chuyển trạng thái ta có 13 trạng thái nên cần dùng 4 fliplop để lưu trữ.



Hình 18 - Khối REG

II.2.3 Khối NEXT_STA:

Khối NEXT_STA có chức năng chuyển trạng thái hiện tại trong khối REG thành trạng thái kế tiếp mong muốn. Từ lưu đồ chuyển trạng thái ta có thể rút ra được bảng chuyển trạng thái như sau:

Bảng 4 - Bảng chân trị cho mạch chuyển trạng thái khối Controller

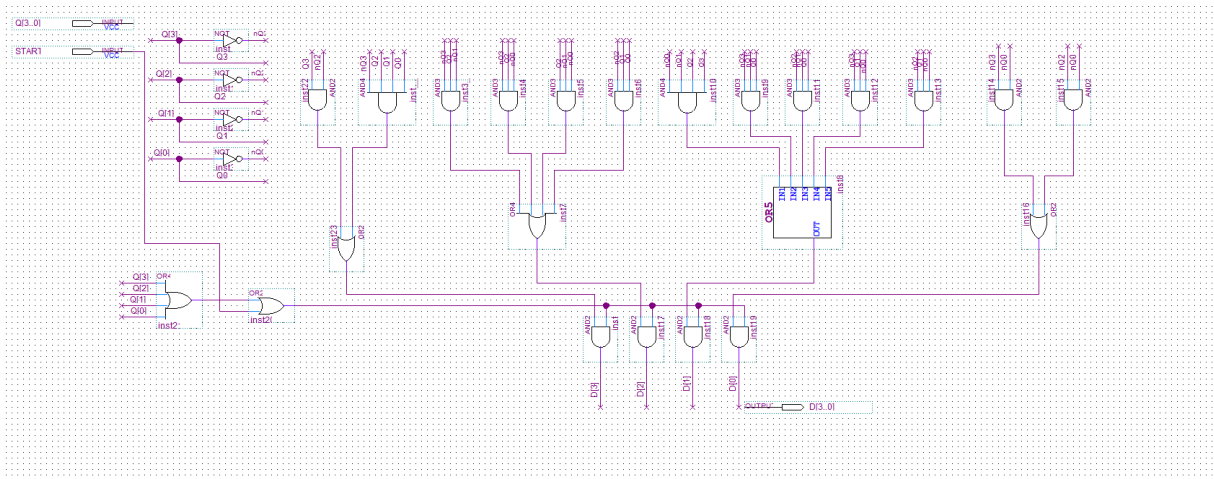
Trạng thái hiện tại				Trạng thái kế tiếp			
Q ₃	Q ₂	Q ₁	Q ₀	D ₃	D ₂	D ₁	D ₀
0	0	0	0	0	0	0	1
0	0	0	1	0	0	1	0
0	0	1	0	0	0	1	1
0	0	1	1	0	1	0	0
0	1	0	0	0	1	0	1
0	1	0	1	0	1	1	0
0	1	1	0	0	1	1	1
0	1	1	1	1	0	0	0
1	0	0	0	1	0	0	1
1	0	0	1	1	0	1	0
1	0	1	0	1	0	1	1
1	0	1	1	1	1	0	0
1	1	0	0	0	1	1	0

$$D_3 = Q_3 \overline{Q_2} + \overline{Q_3} Q_2 Q_1 Q_0$$

$$D_2 = \overline{Q_3} Q_2 \overline{Q_1} + \overline{Q_3} Q_2 \overline{Q_0} + Q_2 \overline{Q_1} \overline{Q_0} + Q_2 Q_1 Q_0$$

$$D_1 = Q_3 Q_2 \overline{Q_1} \overline{Q_0} + \overline{Q_3} \overline{Q_1} Q_0 + \overline{Q_2} \overline{Q_1} Q_0 + \overline{Q_2} Q_1 \overline{Q_0}$$

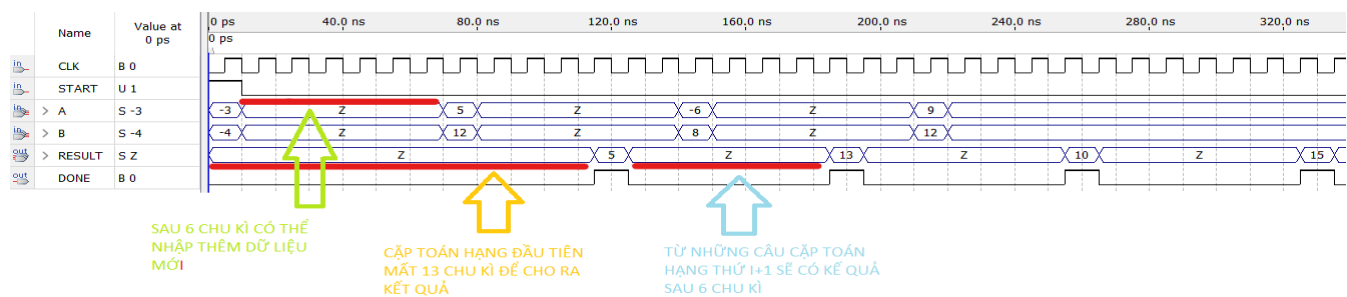
$$D_0 = \overline{Q_3} \overline{Q_0} + \overline{Q_2} \overline{Q_0}$$



Hình 19 - Khối NEXT_STA

Tín hiệu Start có chức năng khởi động giá trị lúc đầu chỉ có giá trị khi ở trạng thái S₀, hệ thống sẽ hoạt động khi ở trạng thái S₀ tín hiệu START bật lên 1.

III. KẾT QUẢ CHẠY MÔ PHỎNG



Ta có thể thấy : $(-3)^2 + (-4)^2 = 5^2$

$$5^2 + 12^2 = 13^2$$

$$(-6)^2 + 8^2 = 10^2$$

$$9^2 + 12^2 = 40^2$$