

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

-----❧★❧-----



BÁO CÁO GIỮA KỲ
BÀI TẬP MÔN THIẾT KẾ LUẬN LÝ SỐ

Lớp: CE118.N21

Giảng viên hướng dẫn: Lâm Đức Khải

Nhóm sinh viên thực hiện:

- | | |
|--------------------------------|----------|
| 1. Đào Phước Tài (50%) | 21521391 |
| 2. Nguyễn Quốc Trường An (50%) | 21521810 |

TP.Hồ Chí Minh , tháng 04 năm 2023

MỤC LỤC

DANH MỤC HÌNH ẢNH.....	6
DANH MỤC BẢNG.....	8
BÀI TẬP 1. THIẾT KẾ BỘ ĐẾM MODULO-3	9
I. YÊU CẦU	9
II. CẤU TẠO	9
III. THIẾT KẾ.....	9
1. Vẽ sơ đồ trạng thái và bảng trạng thái	9
2. Rút gọn trạng thái.....	10
3. Mã hóa trạng thái.....	11
4. Lập bảng chuyển trạng thái sử dụng flipflop J-K	12
5. Rút gọn luận lý cho biểu thức ngõ vào flipflop và ngõ ra của mạch.....	12
6. Vẽ mạch luận lý	16
7. Mô phỏng	16
BÀI TẬP 2. THIẾT KẾ BỘ ĐẾM BCD 5 BIT UP/DOWN	17
I. YÊU CẦU	17
II. CẤU TẠO	17
III. THIẾT KẾ.....	17
1. Thiết kế bộ cộng trừ HAS.....	17
2. Thiết kế bộ đếm	18
3. Thiết kế bộ nạp	19
4. Thiết kế bộ chọn MUX21 (5 bit)	20
5. Vẽ mạch tổng quát	21
6. Mô phỏng	21
BÀI TẬP 3. THIẾT KẾ RAM 32 BYTE (32x8)	22
I. YÊU CẦU	22
II. CẤU TẠO	22
III. THIẾT KẾ RAM 32 BYTES.....	22
1. Thiết kế word 8 bit.....	22

2. Thiết kế bộ giải mã.....	24
3. Thiết kế RAM 8x8	27
4. Thiết kế RAM 32x8 từ 4 thanh RAM 8x8	29
BÀI TẬP 4. THIẾT KẾ PUSH-DOWN STACK VỚI RAM 32x8.....	31
I. YÊU CẦU	31
II. CẤU TẠO	31
III. THIẾT KẾ.....	32
1. Cách Stack hoạt động với RAM 32x8	32
2. Thiết kế Stack	33
BÀI TẬP 5. THIẾT KẾ BỘ CỘNG TÍCH LŨY (1 ĐẾN 8).....	35
I. YÊU CẦU	35
II. CẤU TẠO	35
III. THIẾT KẾ KHỐI DỮ LIỆU.....	35
1. Thiết kế bộ Counter 5 bit	35
2. Thiết kế bộ cộng Adder 5 bit	35
3. Thiết kế thanh ghi 5 bit sử dụng flipflop D	37
4. Tổng hợp khối dữ liệu.....	38
5. Đóng gói khối dữ liệu	38
6. Mô phỏng chức năng	38
IV. THIẾT KẾ KHỐI ĐIỀU KHIỂN.....	39
1. Sơ đồ trạng thái.....	39
2. Mã hóa trạng thái.....	40
3. Chọn flipflop và lập bảng chuyển trạng thái.....	40
4. Thiết kế khối trạng thái kế tiếp	41
5. Thiết kế khối lưu trạng thái	41
6. Thiết kế khối tạo tín hiệu điều khiển.....	42
7. Vẽ mạch toàn bộ hệ thống	43
8. Mô phỏng chức năng	43

BÀI TẬP 6. THIẾT KẾ KHỐI ĐẾM SỐ LƯỢNG BIT 1 TRONG DẪY 8 BIT	44
I. YÊU CẦU	44
II. CẤU TẠO	44
III. THIẾT KẾ KHỐI DỮ LIỆU	44
1. Ý tưởng	44
2. Thiết kế bộ MUX 4:1	44
3. Thiết kế thanh ghi dịch 8 bit	45
4. Thiết kế bộ đếm	46
5. Tổng hợp khối dữ liệu	47
6. Đóng gói khối dữ liệu	47
7. Mô phỏng chức năng	47
IV. THIẾT KẾ KHỐI ĐIỀU KHIỂN	48
1. Sơ đồ trạng thái	48
2. Mã hóa trạng thái	48
3. Chọn flipflop và lập bảng chuyển trạng thái	49
4. Vẽ mạch khối tạo trạng thái kế tiếp	50
5. Khối lưu trạng thái	50
6. Khối tạo ra tín hiệu điều khiển	51
7. Vẽ mạch toàn bộ hệ thống	52
8. Mô phỏng chức năng	52

DANH MỤC HÌNH ẢNH

BÀI TẬP 1

Hình 1.1 Sơ đồ trạng thái Modulo-3	9
Hình 1.2 Mạch giải mã 2 bit	12
Hình 1.3 Mạch ngõ vào các flipflop JK.....	14
Hình 1.4 Mạch ngõ ra Y	15
Hình 1.5 Mạch luận lý Modulo-3	16
Hình 1.6 Mô phỏng chức năng FSM Modulo-3	16

BÀI TẬP 2

Hình 2. 1 Sơ đồ mạch HAS.....	18
Hình 2.2 Đóng gói bộ HAS.....	18
Hình 2.3 Sơ đồ mạch đếm 5 bit	19
Hình 2.4 Đóng gói mạch Counter 5 bit.....	19
Hình 2.5 Sơ đồ mạch bộ nạp 5 bit.....	20
Hình 2.6 Bộ chọn MUX21 (5bit) sử dụng 5 bộ chọn MUX21 (1bit).....	20
Hình 2.7 Bộ chọn MUX 21 (5it).....	21
Hình 2.8 Mạch đếm 5 bit lên/xuống	21
Hình 2.9 Kết quả chạy mô phỏng	21

BÀI TẬP 3

Hình 3.1 Cấu tạo word 8 bit(trích Giáo trình Thiết kế luận lý số 2)	22
Hình 3.2 Mạch luận lý word 8 bit sử dụng D latch	23
Hình 3.3 Đóng gói khối word 8 bit.....	23
Hình 3.4 Mạch mô phỏng bộ giải mã 3:8	24
Hình 3.5 Bộ giải mã 3:8.....	25
Hình 3.6 Mạch mô phỏng bộ giải mã 2:4	26
Hình 3.7 Bộ giải mã 2:4.....	26
Hình 3.8 Mô phỏng RAM 8x8.....	27
Hình 3.9 Đóng gói khối RAM 8x8	27
Hình 3.10 Kết quả mô phỏng RAM 8x8.....	28
Hình 3.11 Mạch mô phỏng RAM 32x8	29
Hình 3.12 Đóng gói khối RAM 32x8	29
Hình 3.13 Kết quả mô phỏng RAM 32x8 hình 1	30
Hình 3.14 Kết quả mô phỏng RAM 32x8 hình 2	30

BÀI TẬP 4

Hình 4.1 Cấu tạo ngăn xếp có khả năng lưu trữ 32 word 8 bit được thực hiện bằng RAM 32x8 được thiết kế trước đó.....	31
Hình 4.2 Mạch luận lý tổng quát mô hình Stack	33
Hình 4.3 Khối Stack.....	33
Hình 4.4 Kết quả mô phỏng Stack với RAM 32x8 hình 1	34

Hình 4.5 Kết quả mô phỏng Stack với RAM 32 Byte hình 2.....	34
---	----

BÀI TẬP 5

Hình 5.1 Khối luận lý Full Adder 1 bit.....	36
Hình 5.2 Khối Full Adder 1 bit.....	36
Hình 5.3 Khối luận lý Full Adder 5 bit.....	36
Hình 5.4 Bộ cộng Full Adder 5 bit.....	37
Hình 5.5 Mạch miêu tả thanh ghi 5 bit sử dụng flipflop D.....	37
Hình 5.6 Thanh ghi 5 bit sử dụng flipflop D.....	37
Hình 5.7 Khối dữ liệu cho bộ tính sum(1 đến 8).....	38
Hình 5.8 Khối dữ liệu tính tổng từ 1 đến 8.....	38
Hình 5.9 Kết quả mô phỏng bộ tính tổng Sum (1 đến 8).....	38
Hình 5.10 Sơ đồ trạng thái.....	39
Hình 5.11 Mạch miêu tả khối tạo trạng thái kế tiếp.....	41
Hình 5.12 Khối trạng thái kế tiếp.....	41
Hình 5.13 Mạch miêu tả khối lưu trạng thái.....	41
Hình 5.14 Khối lưu trạng thái.....	42
Hình 5.15 Mạch miêu tả khối tín hiệu điều khiển.....	42
Hình 5.16 Khối tín hiệu điều khiển.....	42
Hình 5.17 Mạch luận lý toàn bộ hệ thống.....	43
Hình 5.18 Mô phỏng chức năng khối tính tổng 1 đến 8.....	43

BÀI TẬP 6

Hình 6.1 Mạch miêu tả MUX 4:1.....	45
Hình 6.2 Khối MUX 4:1.....	45
Hình 6.3 Mạch miêu tả thanh ghi dịch 8 bit.....	46
Hình 6.4 Khối thanh ghi dịch 8 bit.....	46
Hình 6.5 Khối đếm 5 bit.....	46
Hình 6.6 Mạch miêu tả tổng quát khối đếm số bit 1.....	47
Hình 6.7 Đóng gói khối dữ liệu.....	47
Hình 6.8 Kết quả mô phỏng khối dữ liệu.....	47
Hình 6.9 Sơ đồ trạng thái.....	48
Hình 6.10 Mạch miêu tả khối tạo trạng thái kế tiếp.....	50
Hình 6.11 Khối trạng thái kế tiếp.....	50
Hình 6.12 Mạch miêu tả khối lưu trạng thái.....	50
Hình 6.13 Khối lưu trạng thái.....	50
Hình 6.14 Mạch luận lý khối tạo tín hiệu điều khiển.....	51
Hình 6.15 Đóng gói khối tạo tín hiệu điều khiển.....	51
Hình 6.16 Mạch luận lý toàn bộ hệ thống.....	52
Hình 6.17 Kết quả chạy mô phỏng.....	52

DANH MỤC BẢNG

BÀI TẬP 1:

Bảng 1.1 Bảng trạng thái	10
Bảng 1.2 Bảng ngõ ra.....	10
Bảng 1.3 Bảng trạng thái kế tiếp.....	10
Bảng 1.4 Bảng trạng thái cho mã hóa trạng thái.....	11
Bảng 1.5 Bảng mã hóa trạng thái.....	11
Bảng 1.6 Bảng giải mã ngõ ra theo số đếm 0, 1, 2	11
Bảng 1.7 Bảng chuyển trạng thái sử dụng flipflop J-K	12
Bảng 1.8 Bảng ngõ vào J1	12
Bảng 1.9 Bảng ngõ vào K1	13
Bảng 1.10 Bảng ngõ vào J0	13
Bảng 1.11 Bảng ngõ vào K0	13
Bảng 1.12 Bảng ngõ ra Y	15

BÀI TẬP 2

Bảng 2.1 Bảng sự thật bộ cộng trừ HAS	17
Bảng 2.2 Bảng hoạt động của bộ đếm lên/ xuống 5 bit.....	18
Bảng 2.3 Bảng sự thật bộ nạp 5 bit.....	19

BÀI TẬP 3

Bảng 3.1 Bảng sự thật bộ giải mã 3:8	24
Bảng 3.2 Bảng sự thật bộ giải mã 2:4.....	25

BÀI TẬP 4

Bảng 4.1 Bảng hoạt động của Stack	32
Bảng 4.2 Bảng điều khiển của Stack với RAM	32

BÀI TẬP 5

Bảng 5.1 Bảng sự thật bộ Adder 5 bit.....	35
Bảng 5.2 Bảng mã hóa trạng thái.....	40
Bảng 5.3 Bảng trạng thái	40
Bảng 5.4 Bảng chuyển trạng thái.....	40
Bảng 5.5 Bảng sự thật khối tạo tín hiệu điều khiển.....	42

BÀI TẬP 6

Bảng 6.1 Bảng hoạt động MUX 4:1	44
Bảng 6.2 Bảng hoạt động của thanh ghi dịch	45
Bảng 6.3 Bảng mã hóa trạng thái.....	48
Bảng 6.4 Bảng trạng thái	49
Bảng 6.5 Bảng chuyển trạng thái.....	49
Bảng 6.6 Bảng sự thật khối tạo tín hiệu điều khiển.....	51

BÀI TẬP 1. THIẾT KẾ BỘ ĐẾM MODULO-3

I. YÊU CẦU

- Sử dụng flipflop J-K, Mealy FSM, mã hóa theo Encoding-B
- Bộ đếm có hai ngõ vào: enable counter (C), count direction (D).
- Khi $C = 1$, bộ đếm sẽ đếm theo tín hiệu của D, và dừng việc đếm khi $C = 0$.
- Bộ đếm sẽ đếm lên khi $D = 0$ và đếm xuống khi $D = 1$.
- Bộ đếm có một giá trị ngõ ra Y, ngõ ra $Y = 1$ khi:
 - o Tới 2 khi đếm xuống.
 - o Tới 0 khi đếm lên.

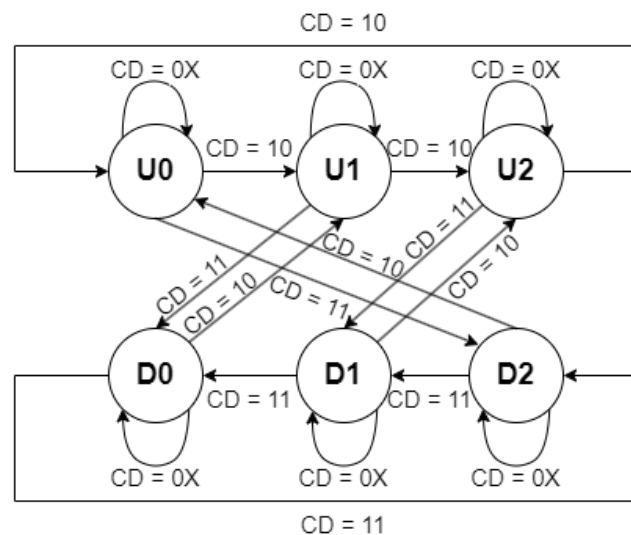
II. CẤU TẠO

Để thiết kế bộ đếm theo yêu cầu, chúng ta cần phải sử dụng mô hình FSM nhằm mô tả hành vi (mối quan hệ giữa ngõ vào và ngõ ra) của mạch tuần tự. Trong đó, chúng ta cần phải xác định mô hình này gồm các thành phần:

- Các trạng thái
- Hàm trạng thái kế tiếp
- Các ngõ vào
- Các ngõ ra
- Hàm ngõ ra

III. THIẾT KẾ

1. Vẽ sơ đồ trạng thái và bảng trạng thái



Hình 1.1 Sơ đồ trạng thái Modulo-3

Từ sơ đồ trạng thái, ta có được bảng miêu tả trạng thái kế tiếp và ngõ ra như sau:

Bảng 1.1 Bảng trạng thái

TTHT	TTKT/Ngõ ra		
	CD = 0X	CD = 10	CD = 11
u0	u0/0	u1/0	d2/1
u1	u1/0	u2/0	d0/0
u2	u2/0	u0/1	d1/0
d0	d0/0	u1/0	d2/1
d1	d1/0	u2/0	d0/0
d2	d2/0	u0/1	d1/0

2. Rút gọn trạng thái

Bảng 1.2 Bảng ngõ ra

Ngõ ra						
CD/TTHT	u0	u1	u2	d0	d1	d2
0X	0	0	0	0	0	0
10	0	0	1	0	0	1
11	1	0	0	1	0	0

Các trạng thái có chung ngõ ra: {u0, d0}, {u1,d1}, {u2,d2}.

Bảng 1.3 Bảng trạng thái kế tiếp

TTKT						
CD/TTHT	u0	u1	u2	d0	d1	d2
0X	u0	u1	u2	d0	d1	d2
10	u1	u2	u0	u1	u2	u0
11	d2	d0	d1	d2	d0	d1

Các trạng thái có chung trạng thái kế tiếp: {u0, d0}, {u1,d1}, {u2,d2}.

Các trạng thái trong các cặp trên là tương đương, sau khi rút gọn ta được các trạng thái u0, u1, u2.

3. Mã hóa trạng thái

Đặt tên các trạng thái: S0 (u0), S1 (u1), S2(u2).

Bảng 1.4 Bảng trạng thái cho mã hóa trạng thái

TTHT	TTKT/Ngõ ra		
	CD = 0X	CD = 10	CD = 11
S0	S0/0	S1/0	S2/1
S1	S1/0	S2/0	S0/0
S2	S2/0	S0/1	S1/0

Độ ưu tiên 1: Không có

Độ ưu tiên 2: {S1, S2}, {S0, S2}, {S0, S1}

Độ ưu tiên 3: {S1, S2}, {S0, S1}, {S0, S1, S2}

Bảng 1.5 Bảng mã hóa trạng thái

Trạng thái	Mã hóa
S0	00
S1	01
S2	11

Do trạng thái được mã hóa sẽ đếm theo các giá trị 0, 1, 3 không phù hợp yêu cầu đề bài, nên chúng ta có bảng giải mã ngược lại theo các giá trị 0, 1, 2:

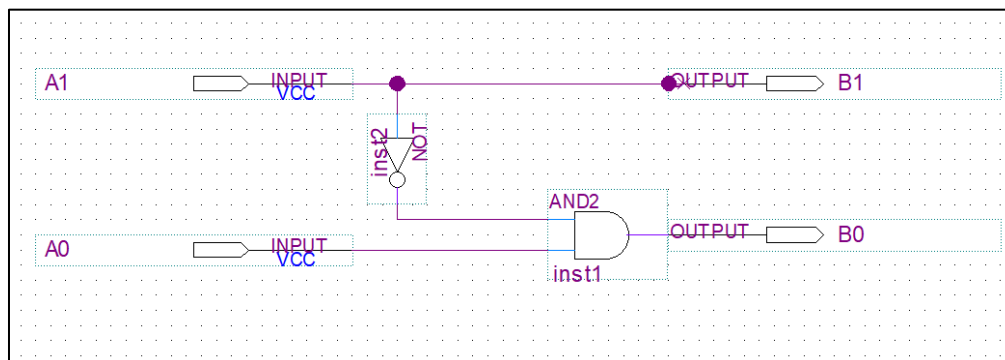
Bảng 1.6 Bảng giải mã ngõ ra theo số đếm 0, 1, 2

Ngõ vào		Ngõ ra	
A1	A0	B1	B0
0	0	0	0
0	1	0	1
1	0	X	X
1	1	1	0

⇒ B1 = A1

⇒ B0 = A1' A0

Từ các phương trình trên, ta có sơ đồ mạch giải mã như sau:



Hình 1.2 Mạch giải mã 2 bit

4. Lập bảng chuyển trạng thái sử dụng flipflop J-K

Bảng 1.7 Bảng chuyển trạng thái sử dụng flipflop J-K

TTHT	TTKT			Ngõ vào flipflop					
Q1Q0	Q1 ⁺ Q0 ⁺			J1K1			J0K0		
	CD	CD	CD	CD	CD	CD	CD	CD	CD
	= 0X	= 10	= 11	= 0X	= 10	= 11	= 0X	= 10	= 11
00	00	01	11	0X	0X	1X	0X	1X	1X
01	01	11	00	0X	1X	0X	X0	X0	X1
11	11	00	01	X0	X1	X1	X0	X1	X0
10	XX	XX	XX	XX	XX	XX	XX	XX	XX

5. Rút gọn luận lý cho biểu thức ngõ vào flipflop và ngõ ra của mạch

▪ Biểu thức ngõ vào các flipflop JK

Bảng 1.8 Bảng ngõ vào J1

J1				
Q1Q0/CD	00	01	10	11
00	0	0	0	1
01	0	0	1	0
11	X	X	X	X
10	X	X	X	X

$$\Rightarrow J1 = Q0'CD + Q0CD'$$

Bảng 1.9 Bảng ngõ vào K1

K1				
Q1Q0/CD	00	01	10	11
00	X	X	X	X
01	X	X	X	X
11	0	0	1	1
10	X	X	X	X

$\Rightarrow K1 = C$

Bảng 1.10 Bảng ngõ vào J0

J0				
Q1Q0/CD	00	01	10	11
00	0	0	1	1
01	X	X	X	X
11	X	X	X	X
10	X	X	X	X

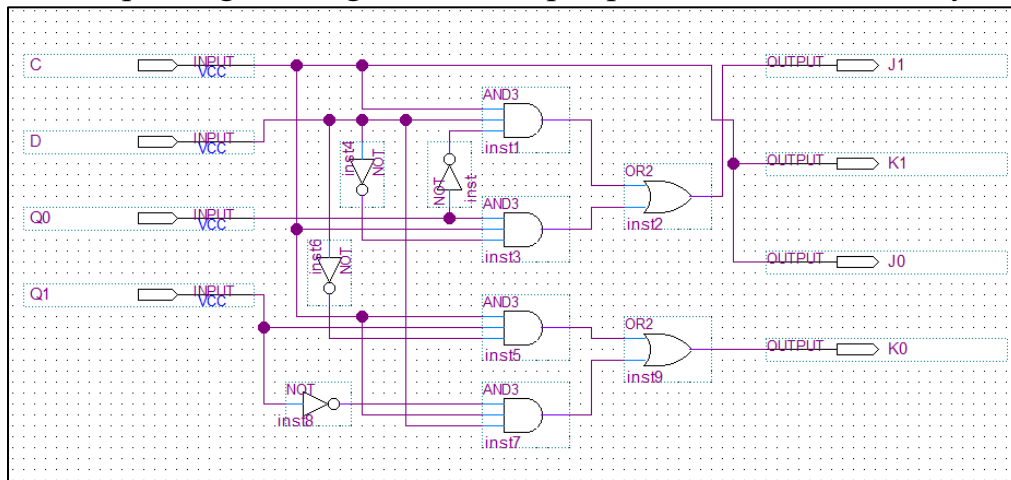
$\Rightarrow J0 = C$

Bảng 1.11 Bảng ngõ vào K0

K0				
Q1Q0/CD	00	01	10	11
00	X	X	X	X
01	0	0	0	1
11	0	0	1	0
10	X	X	X	X

$\Rightarrow K0 = Q1CD' + Q1'CD$

Từ các phương trình ngõ vào các flipflop JK, ta có mạch luận lý:



Hình 1.3 Mạch ngõ vào các flipflop JK

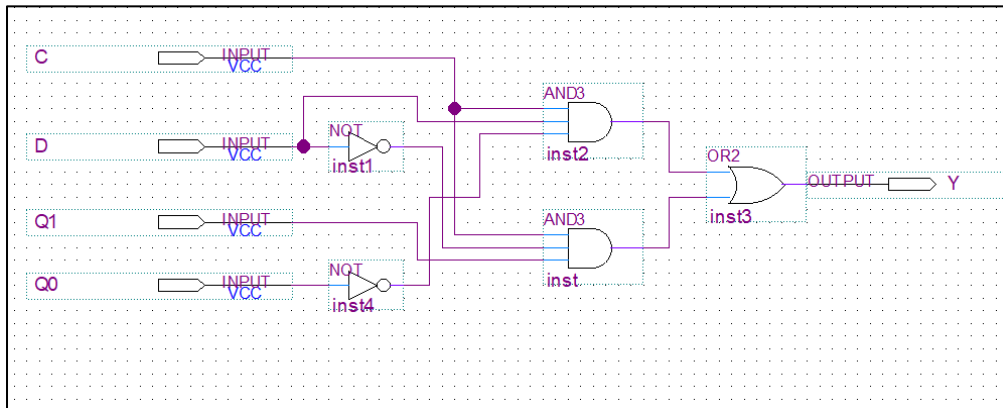
▪ **Biểu thức ngõ ra Y**

Bảng 1.12 Bảng ngõ ra Y

Ngõ ra Y			
Q1Q0/CD	0X	10	11
00	0	0	<u>1</u>
01	0	0	0
11	0	<u>1</u>	0
10	X	<u>X</u>	<u>X</u>

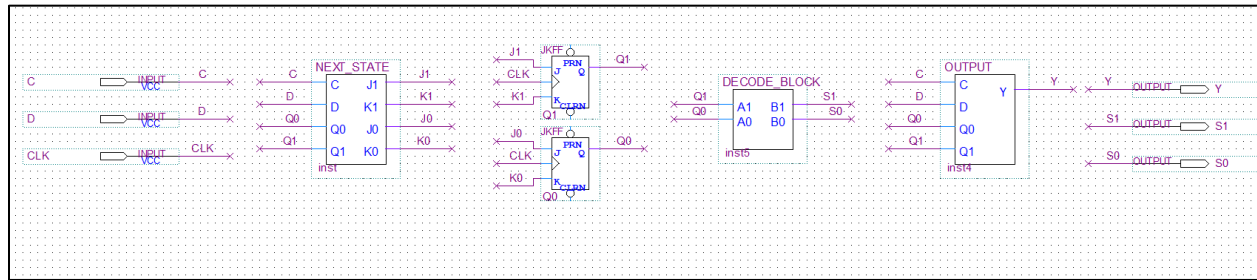
$\Rightarrow Y = Q1CD' + Q0'CD$

Từ biểu thức ngõ ra, ta có sơ đồ mạch ngõ ra Y như sau:



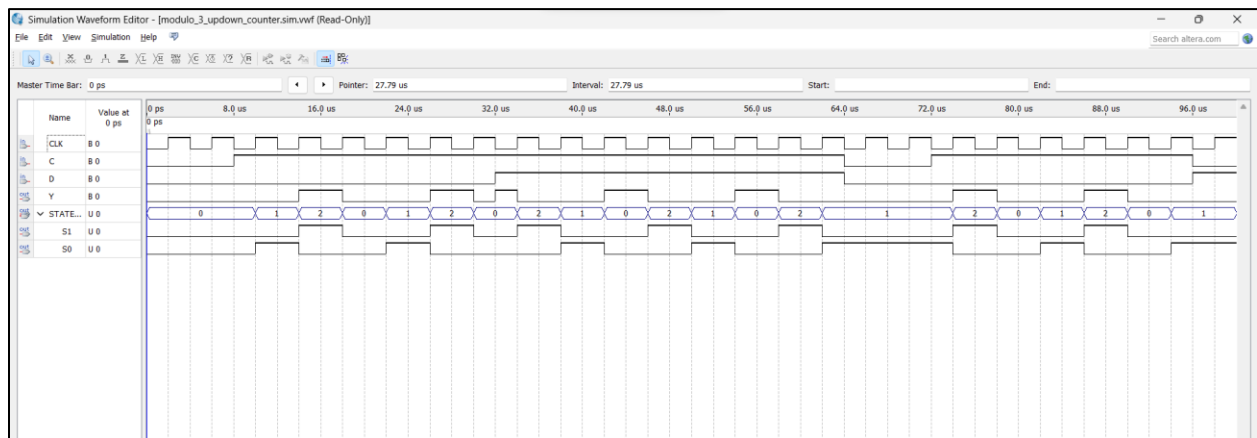
Hình 1.4 Mạch ngõ ra Y

6. Vẽ mạch luận lý



Hình 1.5 Mạch luận lý Modulo-3

7. Mô phỏng



Hình 1.6 Mô phỏng chức năng FSM Modulo-3

BÀI TẬP 2. THIẾT KẾ BỘ ĐẾM BCD 5 BIT UP/DOWN

I. YÊU CẦU

Bộ đếm BCD 5 bit up/down có khả năng đếm lên hoặc xuống giá trị từ 0 đến 9 khi ngõ vào Enable được kích hoạt ($E = 1$). Bộ đếm này đếm theo chiều tăng dần hoặc giảm dần tùy thuộc vào ngõ vào D ($D = 0$ thì đếm lên, $D = 1$ thì đếm xuống).

Bộ đếm 5 bit này được thực hiện đếm từ 0 đến 9. Do đó bộ đếm sẽ có giá trị 0 khi bộ đếm đến 9 và $D = 0$ (đếm lên), ngược lại bộ đếm sẽ về 9 khi bộ đếm có giá trị 0 và $D = 1$ (đếm xuống).

II. CẤU TẠO

Để hiện thực hóa bộ đếm trên, ta cần có các thành phần sau:

- Bộ cộng trừ HAS.
- Bộ đếm lên/xuống.
- Bộ nạp giá trị.
- Bộ chọn.

III. THIẾT KẾ

1. Thiết kế bộ cộng trừ HAS

a. Bảng sự thật (truth table)

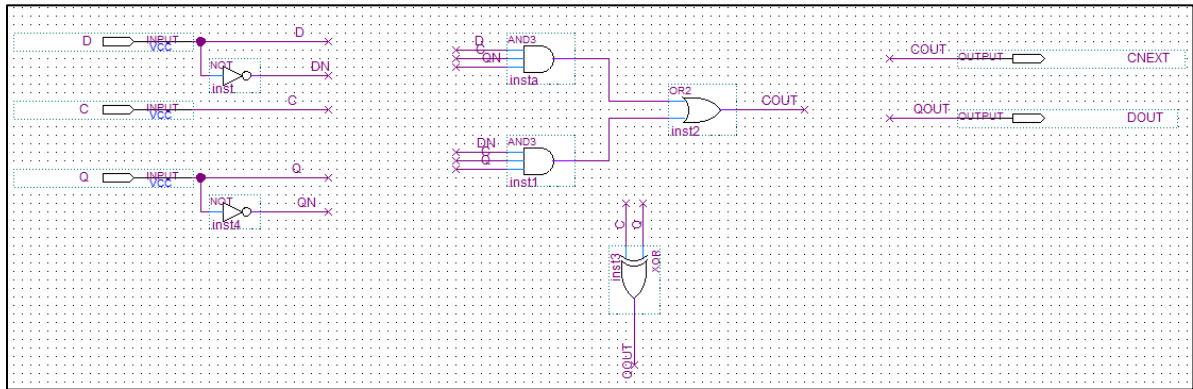
Bảng 2.1 Bảng sự thật bộ cộng trừ HAS

E	D	Q_i	C_i	C_{i+1}	D_i
1	0	0	0	0	0
1	0	0	1	0	1
1	0	1	0	0	1
1	0	1	1	1	0
1	1	0	0	0	0
1	1	0	1	1	1
1	1	1	0	0	1
1	1	1	1	0	0

$$\Rightarrow C_{(next)} = CD'Q + CDQ'$$

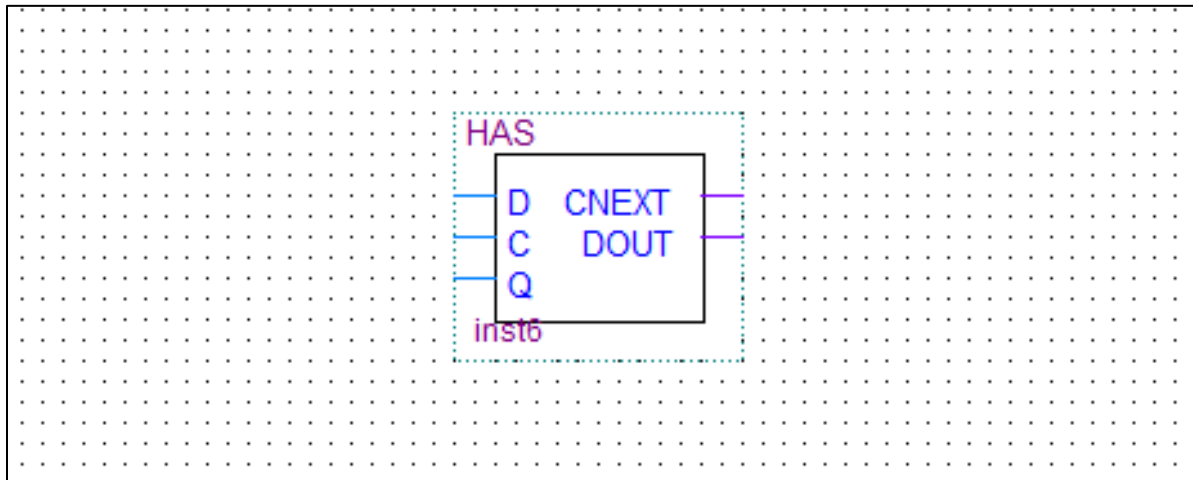
$$\Rightarrow D_{(Qnext)} = C \text{ XOR } Q$$

b. Sơ đồ mạch



Hình 2.1 Sơ đồ mạch HAS

c. Đóng gói



Hình 2.2 Đóng gói bộ HAS

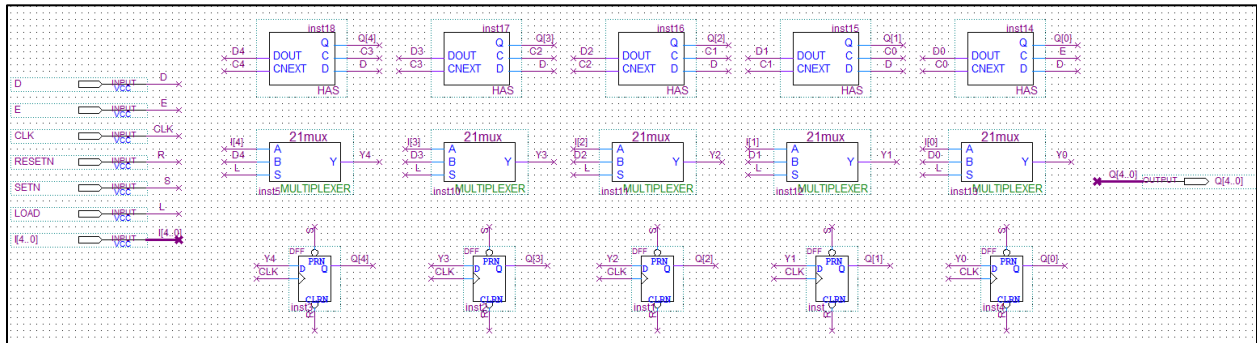
2. Thiết kế bộ đếm

a. Bảng hoạt động

Bảng 2.2 Bảng hoạt động của bộ đếm lên/ xuống 5 bit

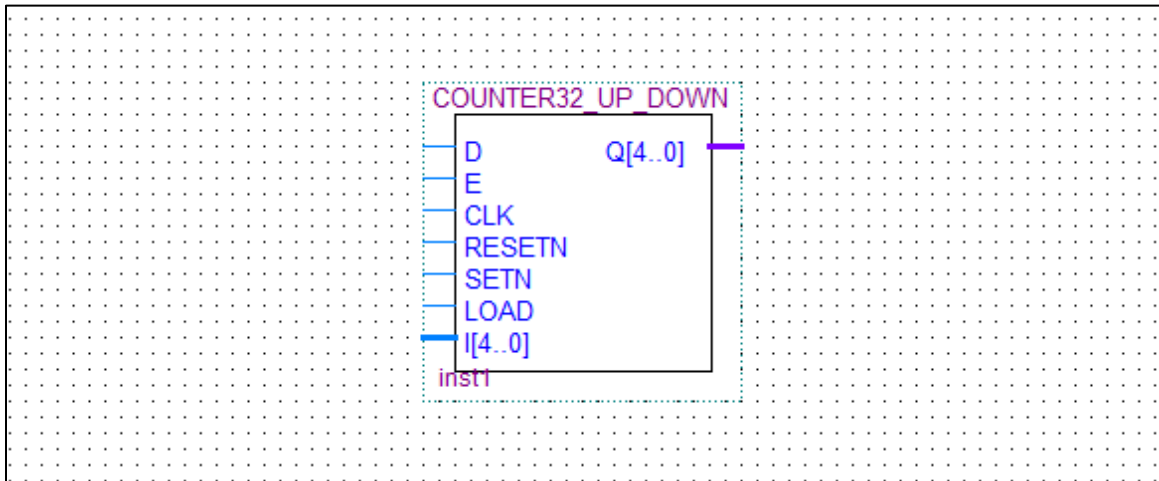
Load	E	D	Cách hoạt động
0	0	1	Không đổi
0	1	0	Đếm lên
0	1	1	Đếm xuống
1	X	X	Tải dữ liệu đầu vào

b. Sơ đồ mạch đếm 5 bit



Hình 2.3 Sơ đồ mạch đếm 5 bit

c. Đóng gói mạch



Hình 2.4 Đóng gói mạch Counter 5 bit

3. Thiết kế bộ nạp

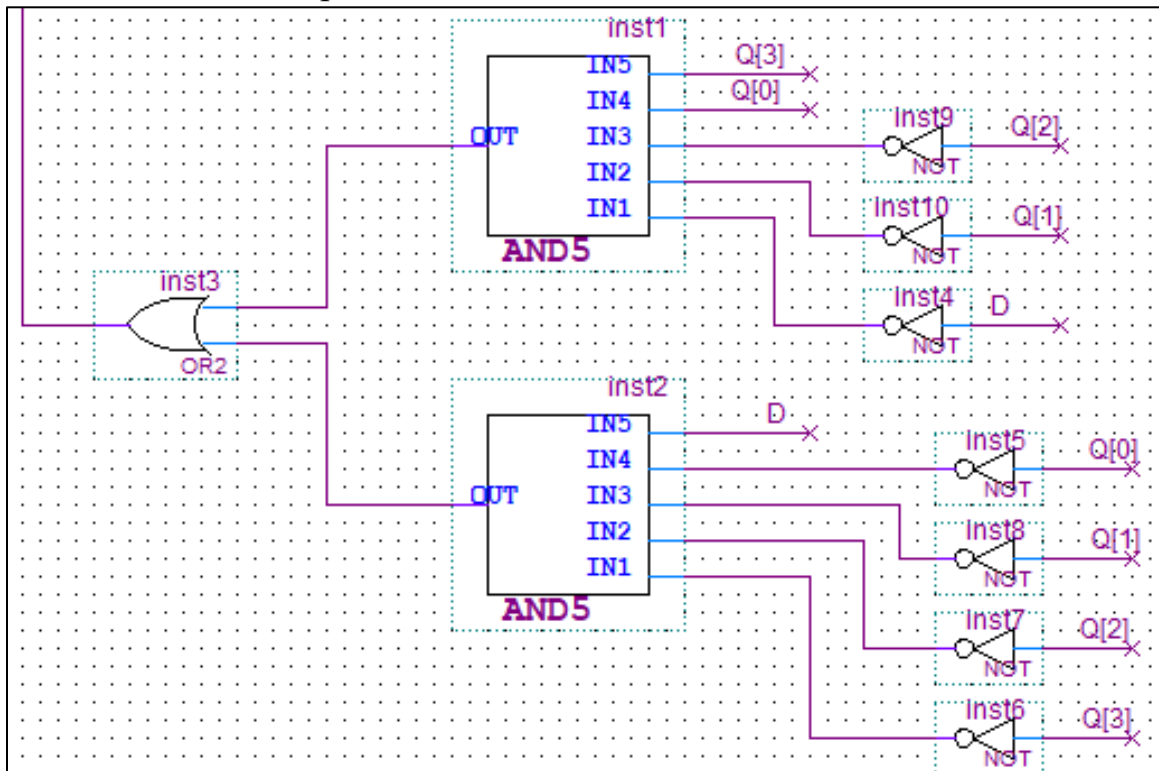
a. Bảng sự thật

Bảng 2.3 Bảng sự thật bộ nạp 5 bit

D	Q ₃	Q ₂	Q ₁	Q ₀	Load
0	1	0	0	1	1
1	0	0	0	0	1

$$\Rightarrow \text{Load} = D'Q_3Q_2'Q_1'Q_0 + DQ_3'Q_2'Q_1'Q_0'$$

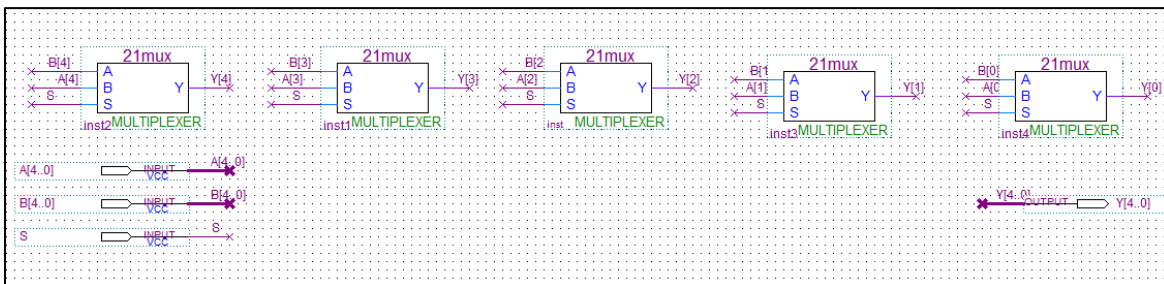
b. Sơ đồ mạch bộ nạp



Hình 2.5 Sơ đồ mạch bộ nạp 5 bit

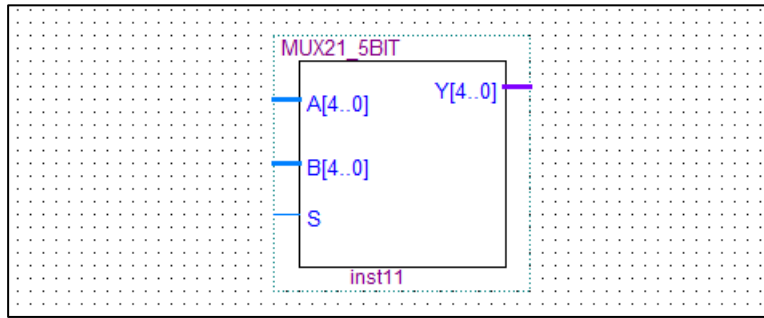
4. Thiết kế bộ chọn MUX21 (5 bit)

a. Sơ đồ mạch



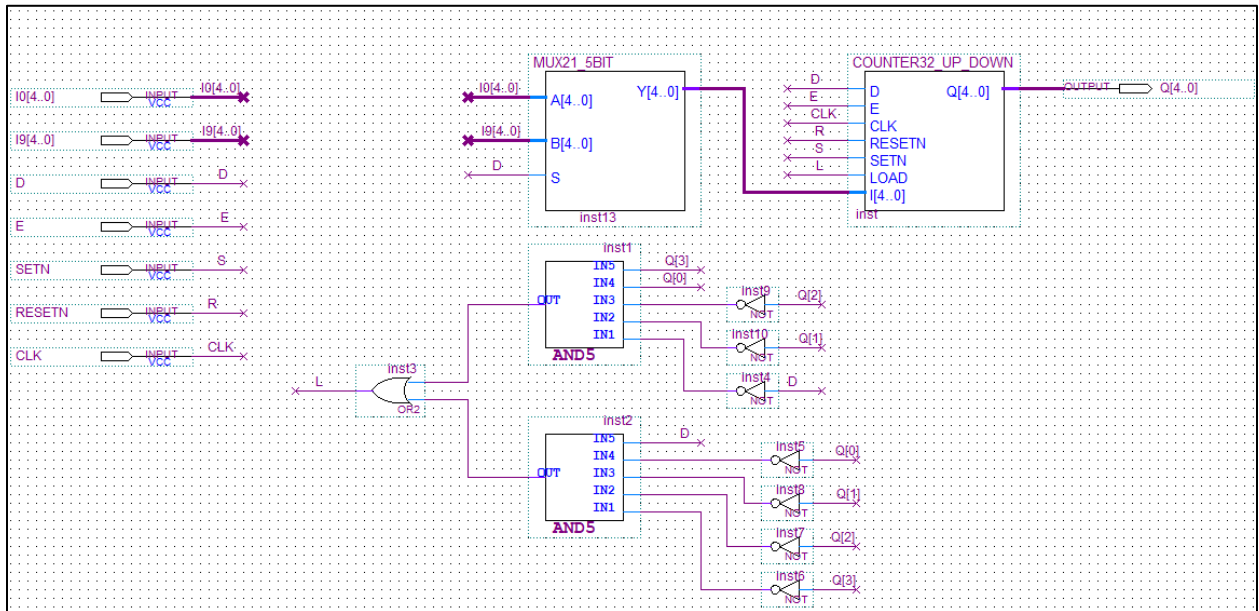
Hình 2.6 Bộ chọn MUX21 (5bit) sử dụng 5 bộ chọn MUX21 (1bit)

b. Đóng gói bộ chọn



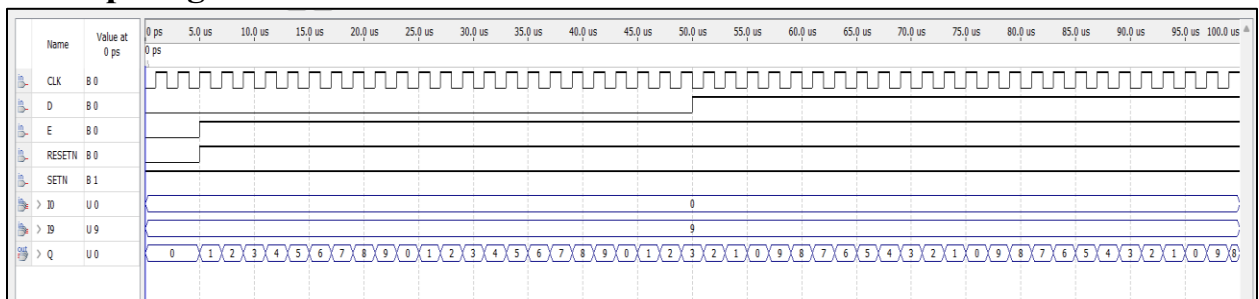
Hình 2.7 Bộ chọn MUX 21 (5bit)

5. Vẽ mạch tổng quát



Hình 2.8 Mạch đếm 5 bit lên/xuống

6. Mô phỏng



Hình 2.9 Kết quả chạy mô phỏng

BÀI TẬP 3. THIẾT KẾ RAM 32 BYTE (32x8)

I. YÊU CẦU

Thiết kế thành RAM 32 BYTE có khả năng đọc ghi dữ liệu.

II. CẤU TẠO

Để thiết kế RAM (có khả năng lưu trữ 32 word, mỗi word 8 bit), chúng ta cần những thành phần sau:

- Word 8 bit có khả năng lưu trữ dữ liệu, với mỗi bit được xử lý bởi D Latch tích cực theo mức.
- Bộ giải mã địa chỉ ở ngõ vào ADDR thành các tín hiệu điều khiển cho phép chọn Word nào được đọc hoặc ghi.
- Các tín hiệu điều khiển như CS, EN, W ...

III. THIẾT KẾ RAM 32 BYTES

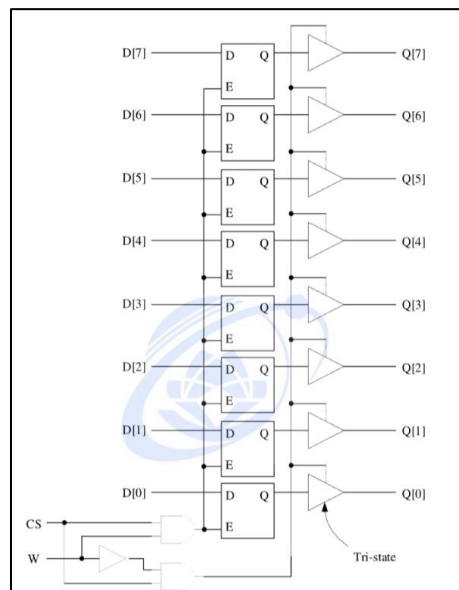
Dưới đây là cách thiết kế RAM 32 Byte (32x8) có khả năng lưu trữ 32 word, mỗi word 8 bit với cách ghép nối 4 thanh RAM 8x8 với nhau.

1. Thiết kế word 8 bit

a. Khái quát về word

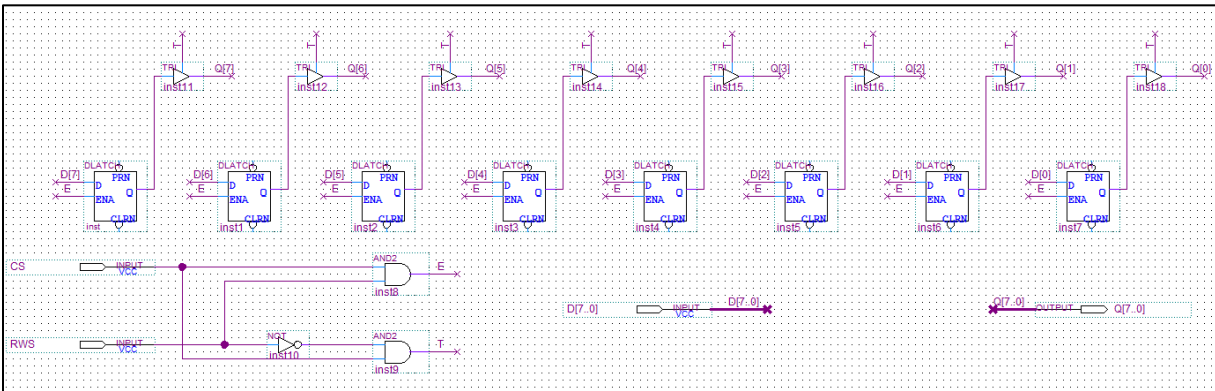
- Có chức năng lưu trữ dữ liệu.
- Một word được cấu tạo bởi các thành phần sau:
 - Các D latch được sử dụng để lưu 1 bit dữ liệu.
 - Các cổng 3 trạng thái (tri-state) vừa có chức năng xuất dữ liệu ra ngõ ra (khi CS = 1 và W = 0) vừa có chức năng ngăn cách dữ liệu với môi trường bên ngoài (khi CS = 0 hoặc W = 1).

b. Cấu tạo một word 8 bit



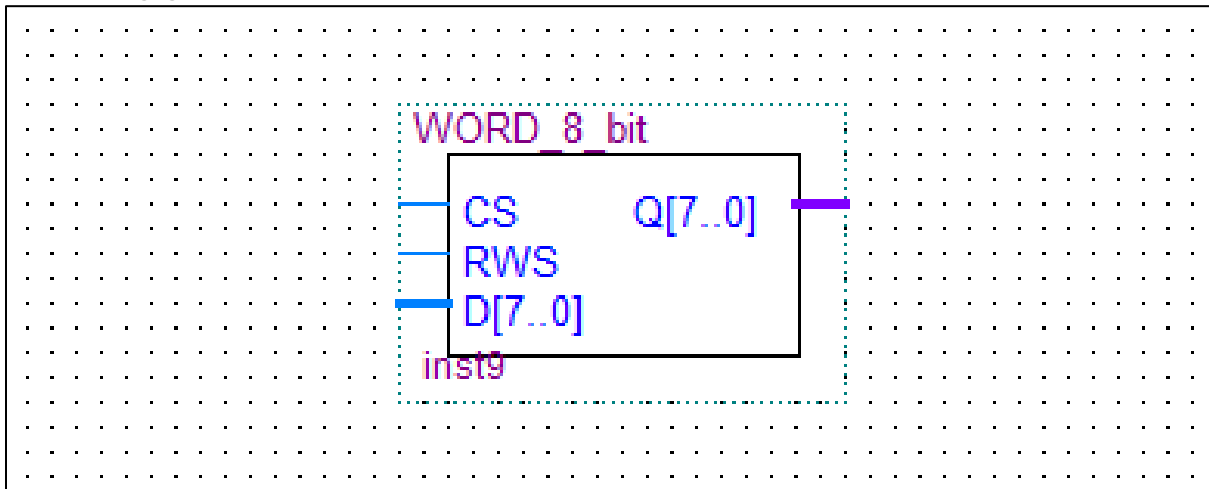
Hình 3.1 Cấu tạo word 8 bit(trích Giáo trình Thiết kế luận lý số 2)

c. Vẽ mạch luận lý word 8 bit sử dụng D latch



Hình 3.2 Mạch luận lý word 8 bit sử dụng D latch

d. Đóng gói mạch



Hình 3.3 Đóng gói khối word 8 bit

Chú thích:

- CS(Chip Select): Cho phép đọc hoặc ghi dữ liệu.
- RWS(Read Write Select): Read khi bằng 0 và Write khi bằng 1.
- D[7..0]: Ngõ vào dữ liệu 8 bit.
- Q[7..0]: Ngõ ra dữ liệu 8 bit.

2. Thiết kế bộ giải mã

a. Khái quát bộ giải mã

Bộ giải mã là một mạch tổ hợp có chức năng chuyển thông tin nhị phân từ các ngõ vào tới từng ngõ ra. Về tổng quát, một bộ giải mã với k ngõ vào thông tin thì có tối đa 2^k ngõ ra. Nếu thông tin được biểu diễn bởi k ngõ vào có các tổ hợp không được sử dụng thì số lượng ngõ ra có thể nhỏ hơn 2^k .

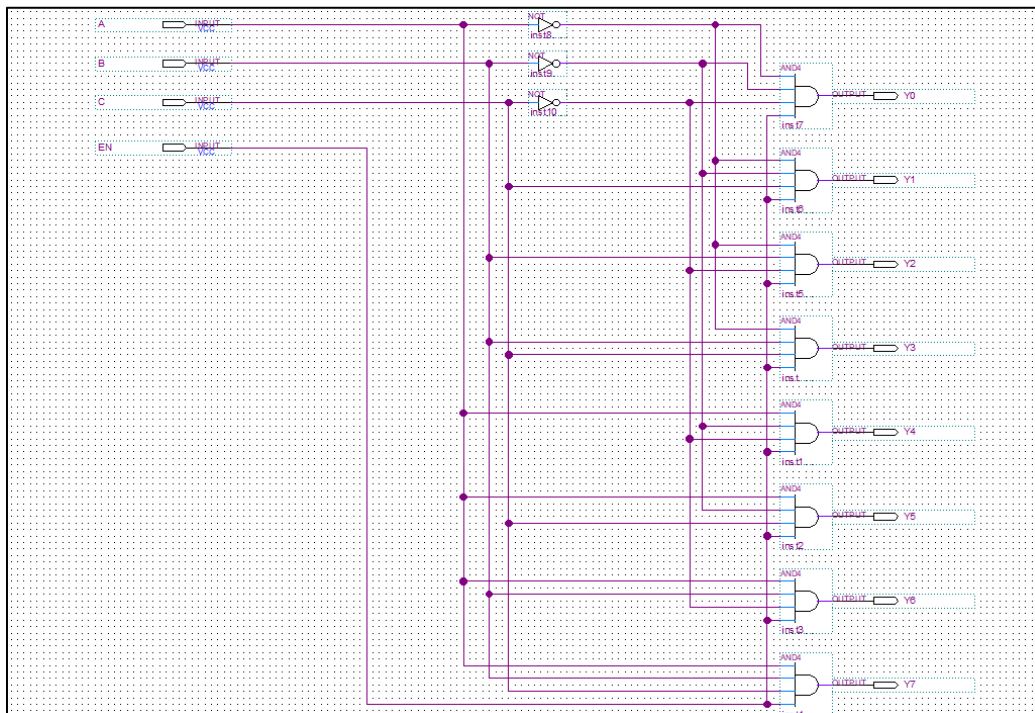
b. Bộ giải mã 3:8

▪ Bảng sự thật

Bảng 3.1 Bảng sự thật bộ giải mã 3:8

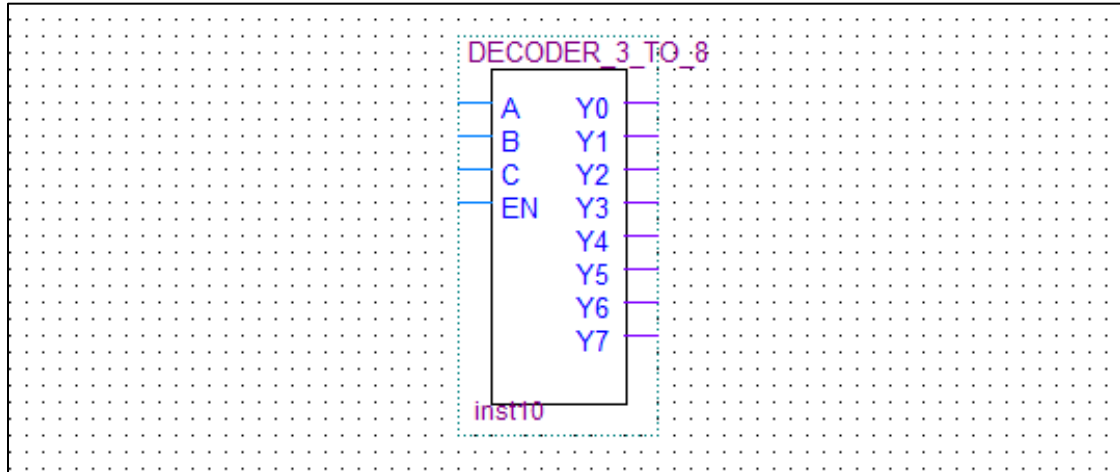
A	B	C	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0	1	0	0
1	1	0	0	0	0	0	0	0	1	0
1	1	1	0	0	0	0	0	0	0	1

▪ Vẽ mạch



Hình 3.4 Mạch mô phỏng bộ giải mã 3:8

- Đóng gói mạch



Hình 3.5 Bộ giải mã 3:8

Chú thích:

- A, B, C: 3 bit ngõ vào.
- Y0, Y1, ..., Y8: 8 bit ngõ ra.
- EN: tín hiệu cho phép phân giải địa chỉ khi EN = 1.

c. Bộ giải mã 2:4

- Bảng sự thật:

Bảng 3.2 Bảng sự thật bộ giải mã 2:4

EN	A	B	Y0	Y1	Y2	Y3
1	0	0	1	0	0	0
1	0	1	0	1	0	0
1	1	0	0	0	1	0
1	1	1	0	0	0	1

Từ bảng trên, ta có các phương trình ngõ ra Y như sau:

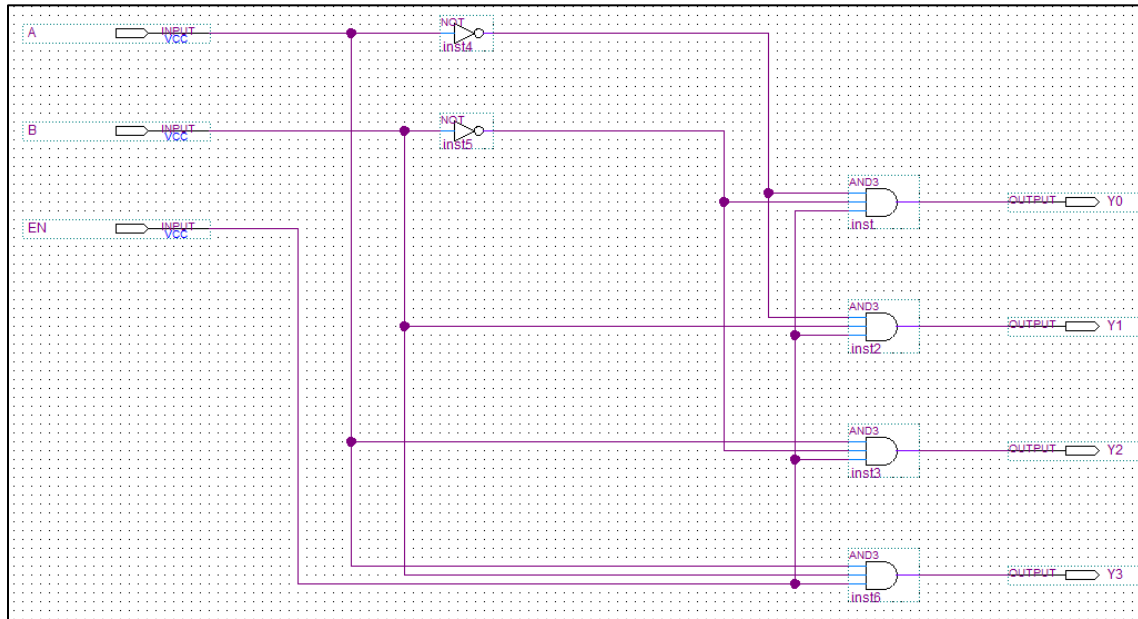
$$Y0 = EN.A'.B'$$

$$Y1 = EN.A'.B$$

$$Y2 = EN.A.B'$$

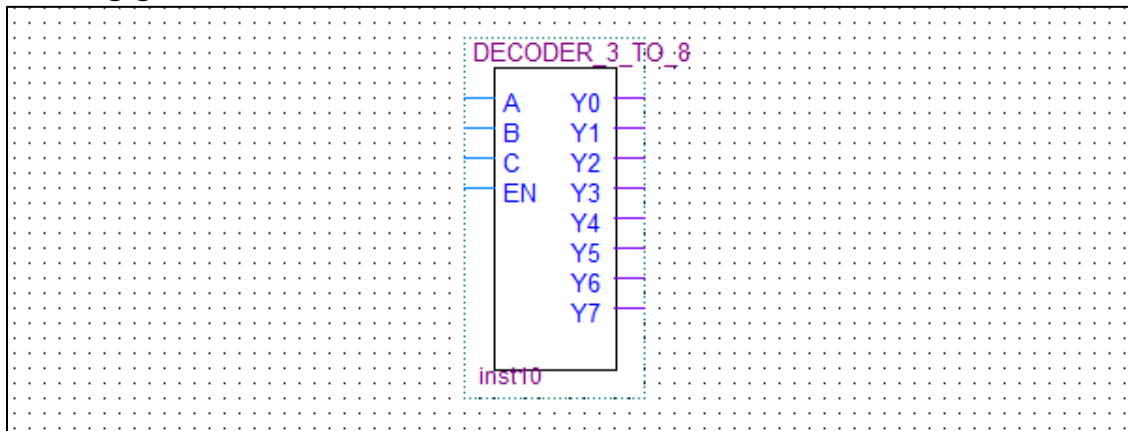
$$Y3 = EN.A.B$$

■ Vẽ mạch



Hình 3.6 Mạch mô phỏng bộ giải mã 2:4

■ Đóng gói mạch



Hình 3.7 Đóng gói bộ giải mã 2:4

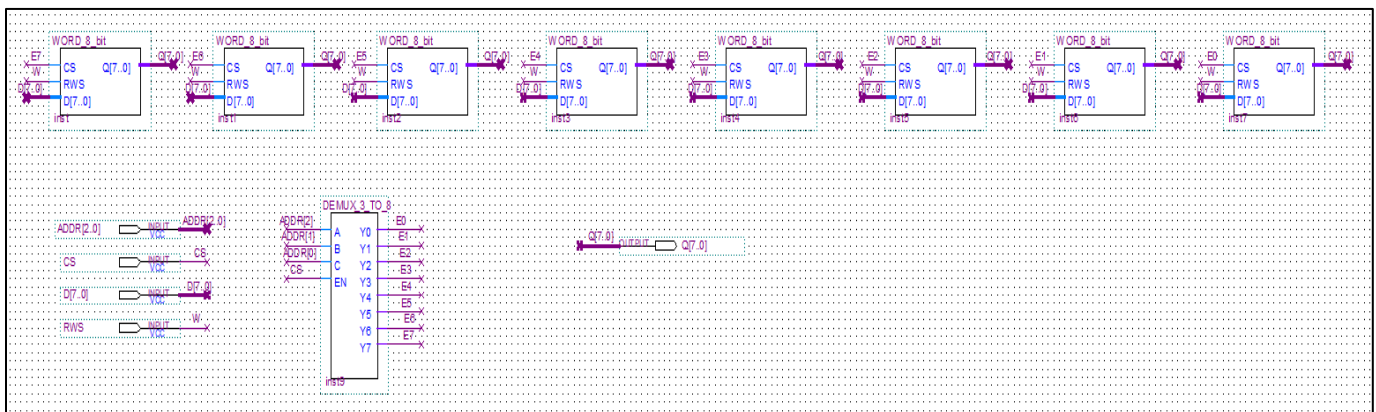
3. Thiết kế RAM 8x8

a. Cấu tạo và nguyên lý hoạt động

Một RAM 8x8 gồm các thành phần sau:

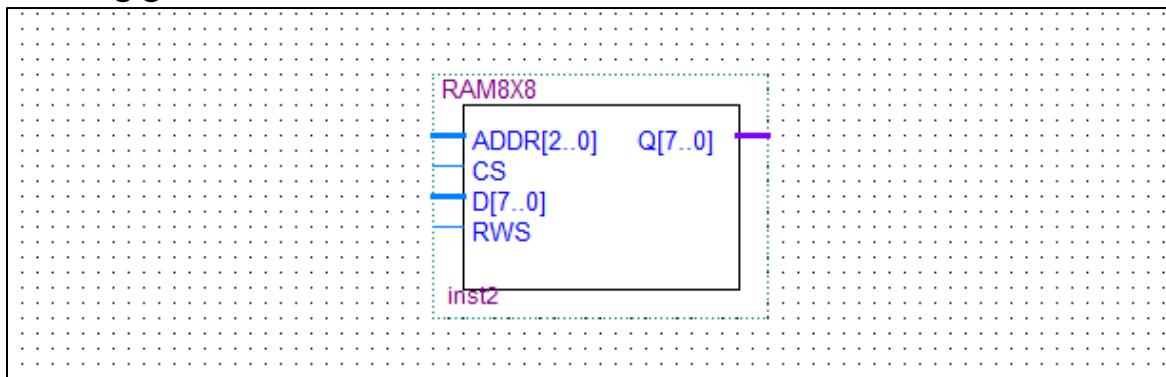
- Bộ giải mã 3:8 có chức năng giải mã địa chỉ ở ngõ vào ADDR thành các tín hiệu điều khiển cho phép chọn Word nào được đọc hoặc ghi.
- Word có chức năng lưu trữ dữ liệu.
- Để đọc dữ liệu từ một Word bất kì, ta thiết lập địa chỉ của Word cần đọc tới ADDR, thiết lập EN = 1 (ngõ vào CS – Chip Select cho phép giải mã địa chỉ) và W = 0 (đọc), ngay lập tức dữ liệu sẽ xuất hiện tại ngõ ra Q.
- Để ghi dữ liệu tới một Word bất kì, ta thiết lập địa chỉ của Word cần ghi tới ADDR, thiết lập EN = 1 (ngõ vào CS – Chip Select cho phép giải mã địa chỉ), đặt dữ liệu vào D và thiết lập W = 1 (ghi) thì dữ liệu sẽ được cập nhật vào Word mong muốn.

b. Vẽ mạch luận lý RAM 8x8



Hình 3.8 Mạch luận lý mô phỏng RAM 8x8

c. Đóng gói mạch

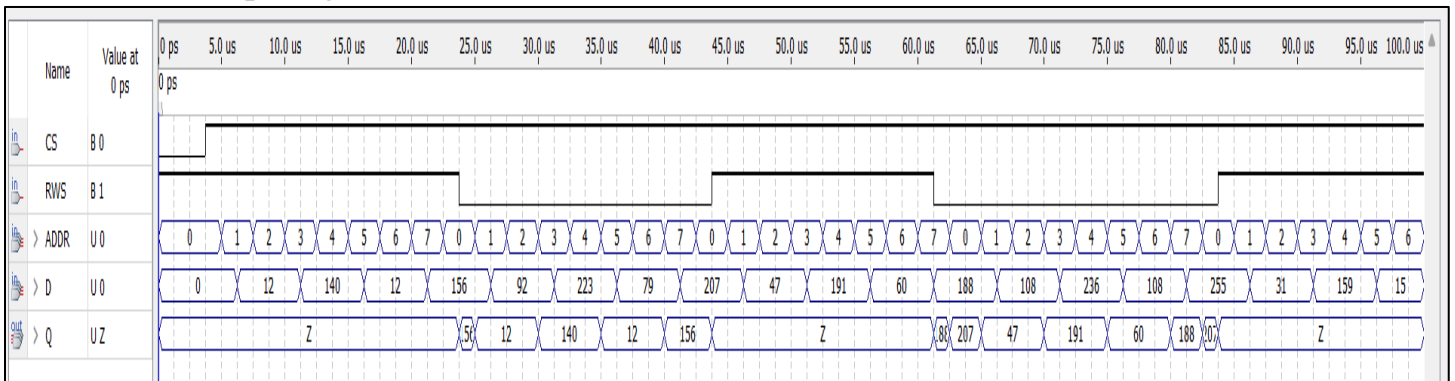


Hình 3.9 Đóng gói khối RAM 8x8

Chú thích:

- ADDR[2..0]: 3 bit đầu vào dùng để chọn 1 trong 8 word để đọc hoặc ghi.
- CS: Cho phép giải mã địa chỉ (cho phép đọc hoặc ghi).
- D[7..0]: Ngõ vào 8 bit dữ liệu.
- RWS (Read Write Select): Read bằng 0 và Write khi bằng 1.
- Q[7..0]: Ngõ ra 8 bit dữ liệu.

d. Mô phỏng



Hình 3.10 Kết quả mô phỏng RAM 8x8

4. Thiết kế RAM 32x8 từ 4 thanh RAM 8x8

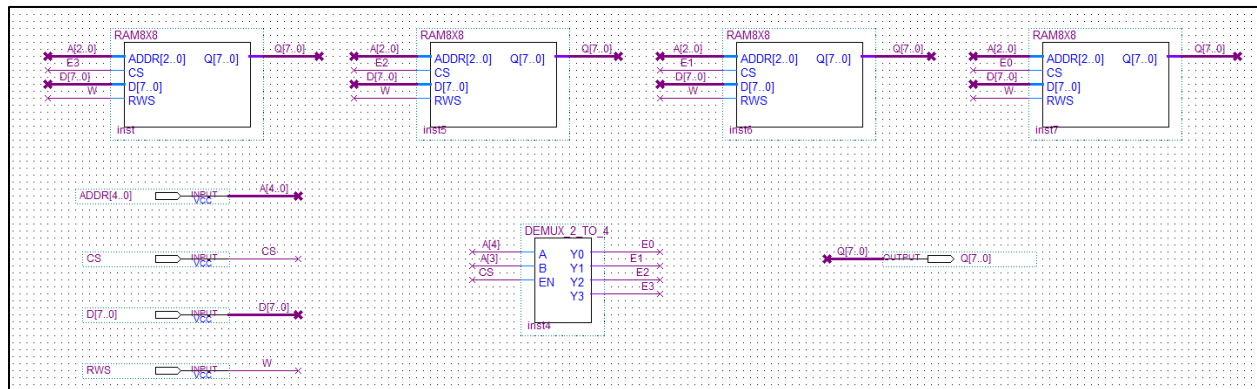
a. Cấu tạo

Để mở rộng khả năng lưu trữ thì chúng ta có thể ghép nối các RAM với nhau để tạo thành RAM có số word lớn hơn.

Một RAM 32x8 gồm các thành phần sau:

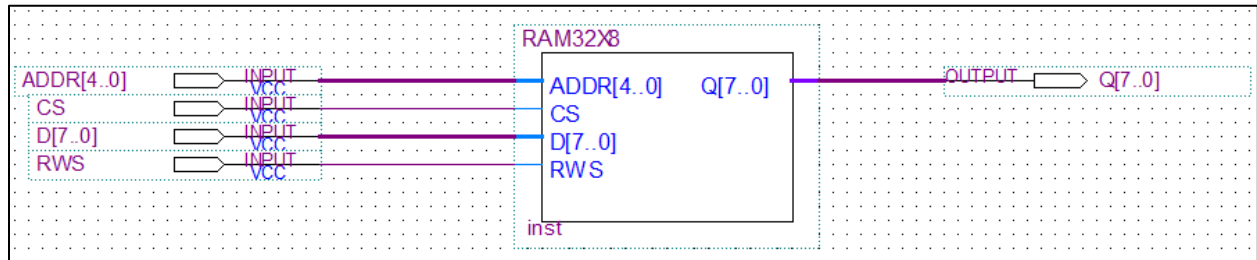
- Bốn RAM 8x8 được ghép nối với nhau.
- Bộ giải mã địa chỉ 2:4 để chọn RAM nào được đọc hoặc ghi.

b. Vẽ mạch



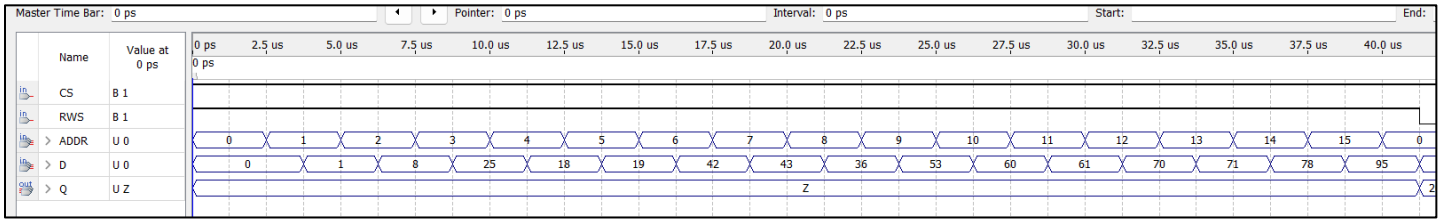
Hình 3.11 Mạch mô phỏng RAM 32x8

c. Đóng gói

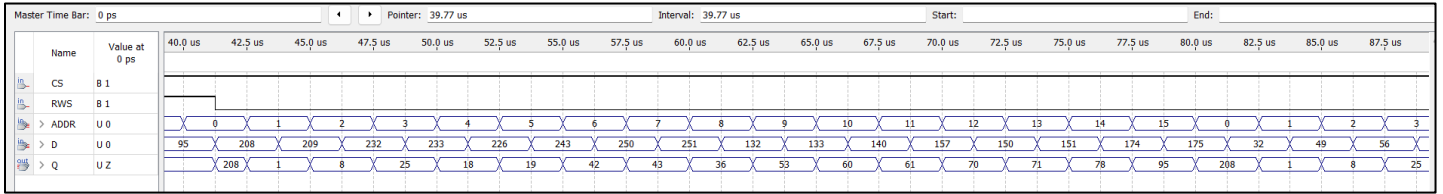


Hình 3.12 Đóng gói khối RAM 32x8

d. Mô phỏng



Hình 3.13 Kết quả mô phỏng RAM 32x8 hình 1



Hình 3.14 Kết quả mô phỏng RAM 32x8 hình 2

BÀI TẬP 4. THIẾT KẾ PUSH-DOWN STACK VỚI RAM 32x8

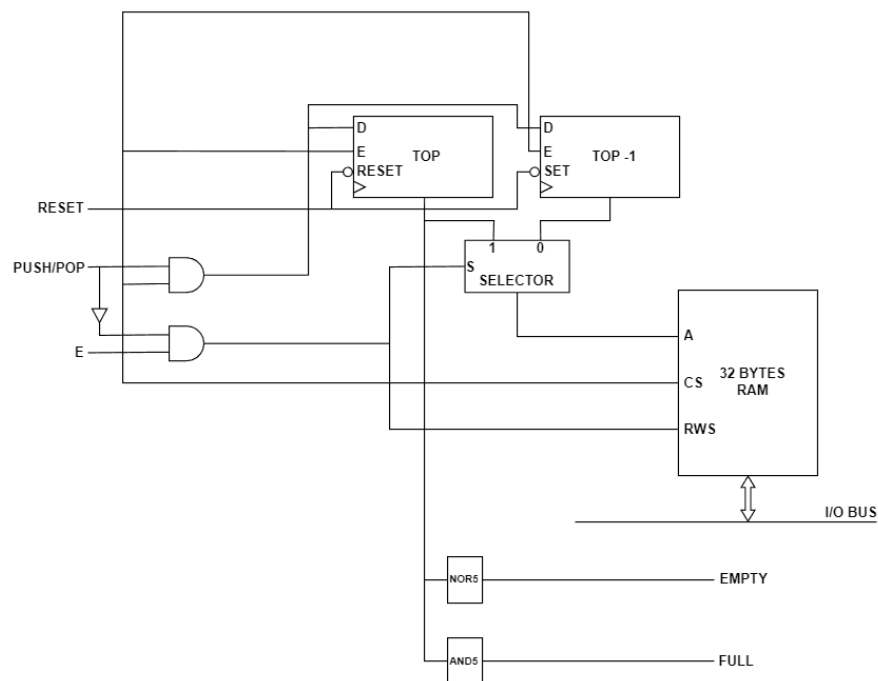
I. YÊU CẦU

Thiết kế bộ Stack có khả năng push/pop dữ liệu vào RAM 32x8 đã thiết kế trước đó theo cơ chế LIFO.

II. CẤU TẠO

Cấu tạo ngăn xếp có khả năng push/pop 32 word 8 bit (RAM 32x8) gồm:

- Khối TOP, TOP – 1 thực chất là các bộ đếm lên/ xuống và được sử dụng để lưu địa chỉ của đỉnh ngăn xếp và đỉnh của ngăn xếp – 1 tương ứng.
- Bộ chọn 2:1 được dùng để lựa chọn địa chỉ ghi/đọc cho RAM. Nếu đẩy dữ liệu vào ngăn xếp thì ngõ ra của bộ chọn là địa chỉ sẽ được ghi dữ liệu. Nếu kéo dữ liệu ra khỏi ngăn xếp thì ngõ ra của bộ chọn là địa chỉ sẽ được đọc dữ liệu.
- Ram được dùng để lưu trữ dữ liệu của ngăn xếp.



Hình 4.1 Cấu tạo ngăn xếp có khả năng lưu trữ 32 word 8 bit được thực hiện bằng RAM 32x8 được thiết kế trước đó.

III. THIẾT KẾ

1. Cách Stack hoạt động với RAM 32x8

Ban đầu ta bật tín hiệu Reset = 0, khi đó tín hiệu Reset ở Top sẽ tích cực dẫn đến bộ đếm Top sẽ có kết quả là 0 (00000) đồng thời tín hiệu Set ở Top – 1 sẽ tích cực dẫn đến bộ đếm Top – 1 sẽ có kết quả là 31 (11111).

Tiếp theo, nếu tín hiệu PUSH/POP = 0 (PUSH), bộ chọn sẽ chọn Top để làm địa chỉ ghi dữ liệu tới RAM. Nếu tín hiệu PUSH/POP = 1 (POP), bộ chọn sẽ chọn Top – 1 để làm địa chỉ đọc dữ liệu từ RAM.

Hai khối Top và Top – 1 sẽ tăng giảm tùy theo đọc hay ghi, nhưng nếu Top có giá trị 31 (11111). Tín hiệu Full sẽ tích cực ở ngõ ra. Tương tự nếu Top có giá trị 0 (00000). Tín hiệu Empty sẽ tích cực ở ngõ ra. Và địa chỉ 31 (11111) sẽ không được đọc hoặc ghi trong thiết kế này.

Ta có bảng hoạt động của Stack được mô tả như sau:

Bảng 4.1 Bảng hoạt động của Stack

Push/Pop	Enable	Hoạt động
X	0	Không thay đổi
0	1	Push
1	1	Pop

Thông qua bảng hoạt động của Stack, ta tổng hợp được bảng điều khiển Stack với RAM như sau:

Bảng 4.2 Bảng điều khiển của Stack với RAM

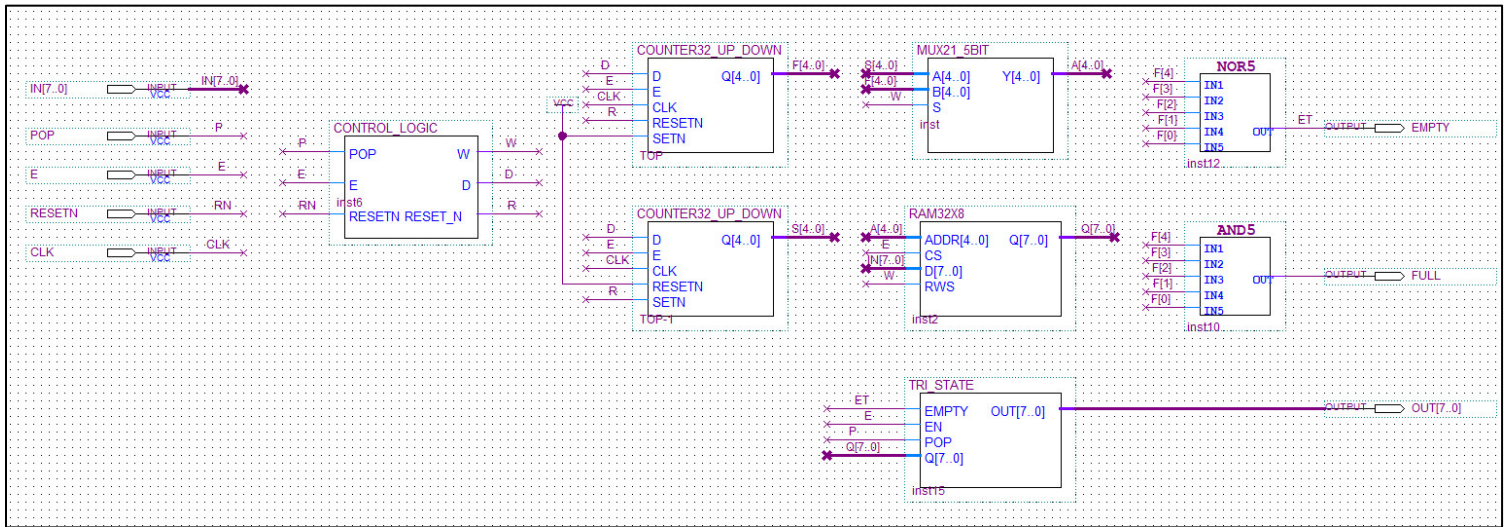
Push/Pop	Enable	Selector Control	Memory Controls		Counter Controls	
		S	CS	RWS	D	E
X	0	X	0	0	X	0
0	1	1	1	1	0	1
1	1	0	1	0	1	1

Từ bảng trên, ta thu được các biểu thức ngõ ra như sau:

- $S = \text{POP}' \cdot E$
- $CS = E$
- $RWS = \text{POP}' \cdot E$
- $D = \text{POP} \cdot E$
- $E = E$

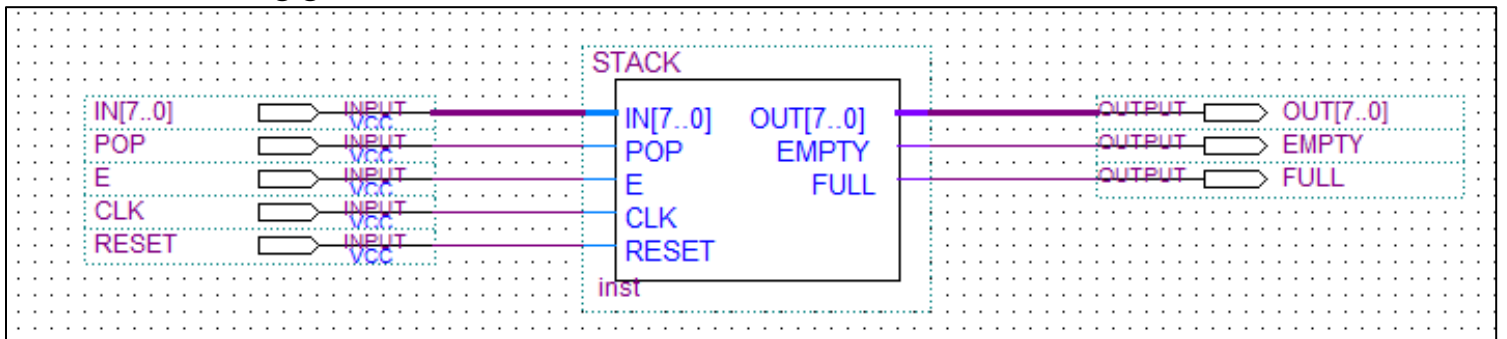
2. Thiết kế Stack

a. Mạch mô phỏng



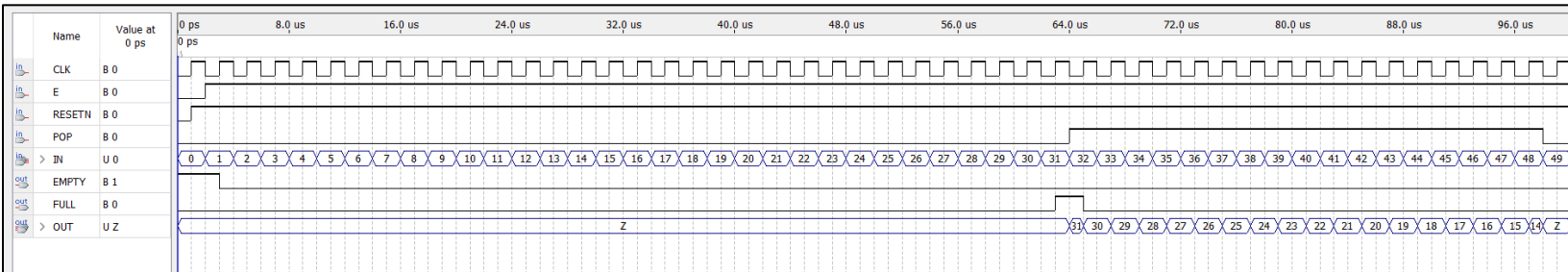
Hình 4.2 Mạch luận lý tổng quát mô hình Stack

b. Đóng gói mạch

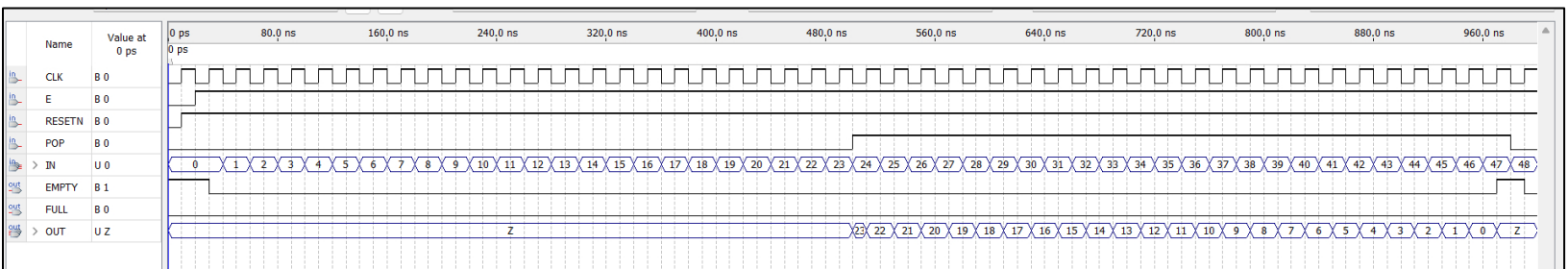


Hình 4.3 Khối Stack

c. Kết quả mô phỏng



Hình 4.4 Kết quả mô phỏng Stack với RAM 32x8 hình 1



Hình 4.5 Kết quả mô phỏng Stack với RAM 32 Byte hình 2

BÀI TẬP 5. THIẾT KẾ BỘ CỘNG TÍCH LŨY (1 ĐẾN 8)

I. YÊU CẦU

Thiết kế bộ cộng tích lũy có khả năng tính tổng các giá trị từ 1 đến 8 theo công thức $sum = sum + x_i$ với giá trị khởi tạo $sum = 0$. Tính toán và đưa kết quả ra màn hình mô phỏng.

II. CẤU TẠO

Để thiết kế bộ cộng theo yêu cầu, chúng ta cần những thành phần sau:

- **Khối dữ liệu:**
 - Bộ Counter để thực hiện việc tăng giá trị x_i .
 - Bộ cộng Adder để thực hiện việc cộng tích lũy.
 - Thanh ghi 5 bit để thực hiện việc lưu giá trị sau mỗi lần cộng.
- **Khối điều khiển:**
 - Khối trạng thái kế tiếp.
 - Khối lưu trạng thái.
 - Khối điều khiển ngõ ra.

III. THIẾT KẾ KHỐI DỮ LIỆU

1. Thiết kế bộ Counter 5 bit

Chúng ta sẽ dùng bộ Counter 5 bit đã thiết kế trong bài tập về BCD Counter 5 bit up/down.

2. Thiết kế bộ cộng Adder 5 bit

a. Bảng sự thật

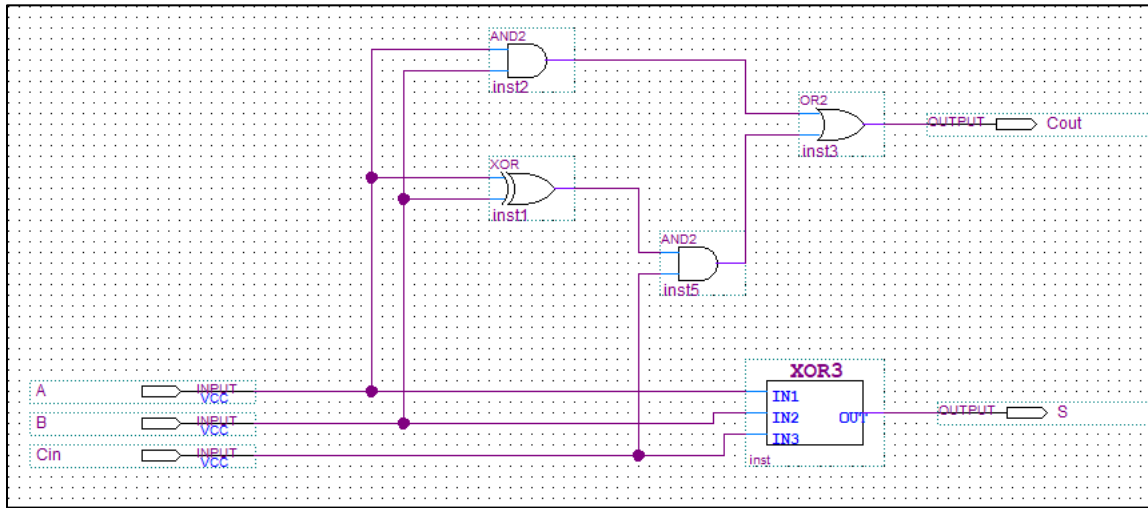
Bảng 5.1 Bảng sự thật bộ Adder

A	B	Cin	S	Cout
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	1

Ta suy ra được các phương trình ngõ ra như sau:

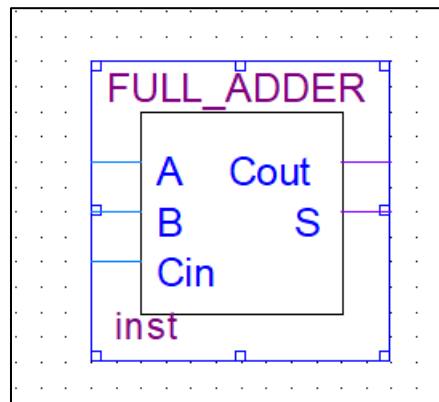
- $S = A \oplus B \oplus Cin$
- $Cout = Cin \vee (A + B) + AB$

b. Vẽ mạch luận lý khối Full Adder 1 bit



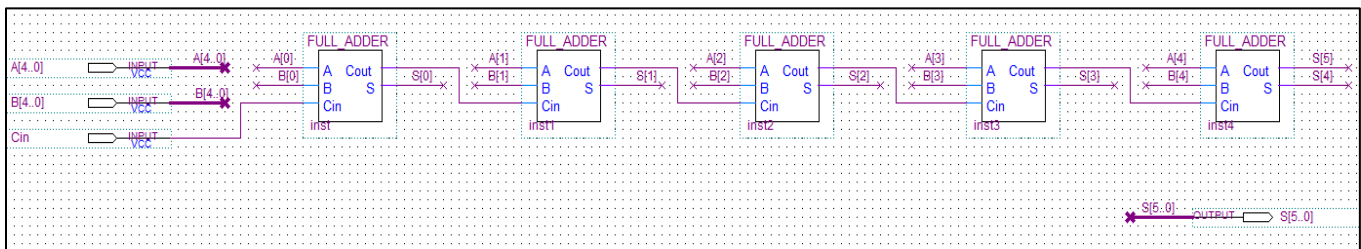
Hình 5.1 Khối luận lý Full Adder 1 bit

c. Đóng gói mạch Full Adder 1 bit



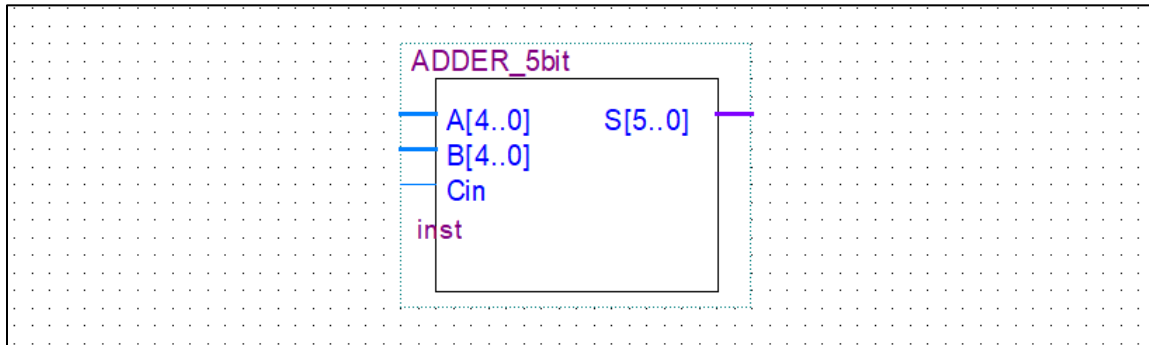
Hình 5.2 Khối Full Adder 1 bit

d. Vẽ mạch luận lý khối Full Adder 5 bit từ khối 1 bit



Hình 5.3 Khối luận lý Full Adder 5 bit

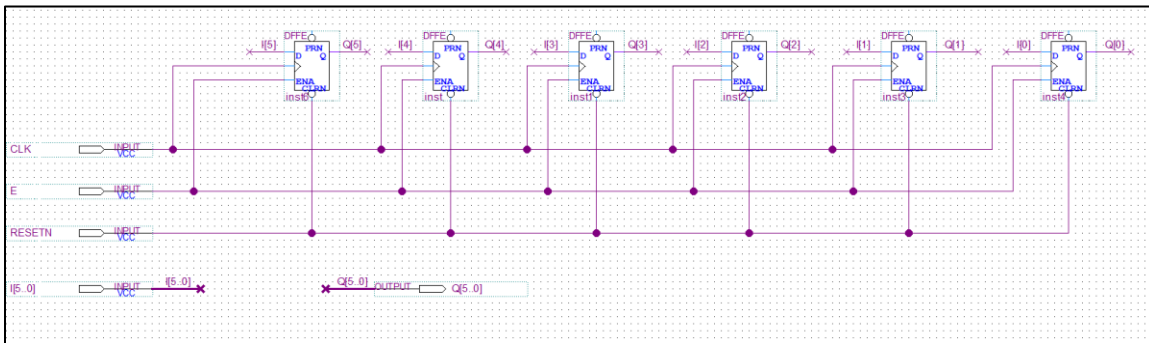
e. Đóng gói mạch Full Adder 5 bit



Hình 5.4 Bộ cộng Full Adder 5 bit

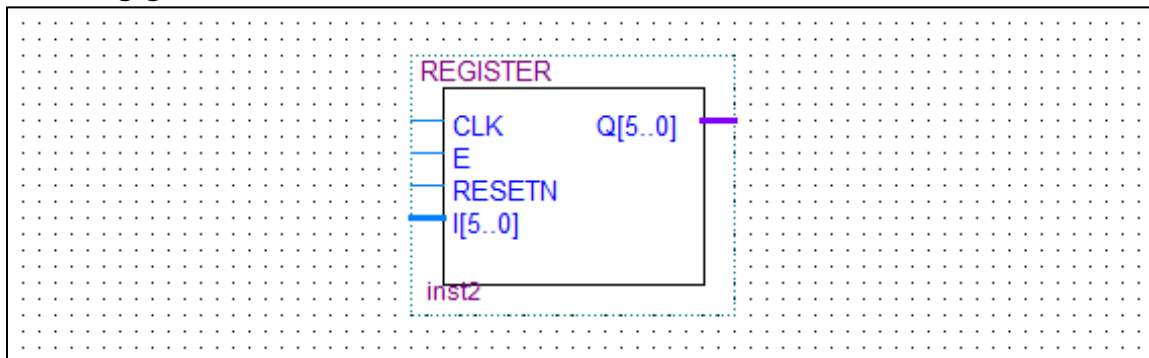
3. Thiết kế thanh ghi 5 bit sử dụng flipflop D

a. Vẽ mạch



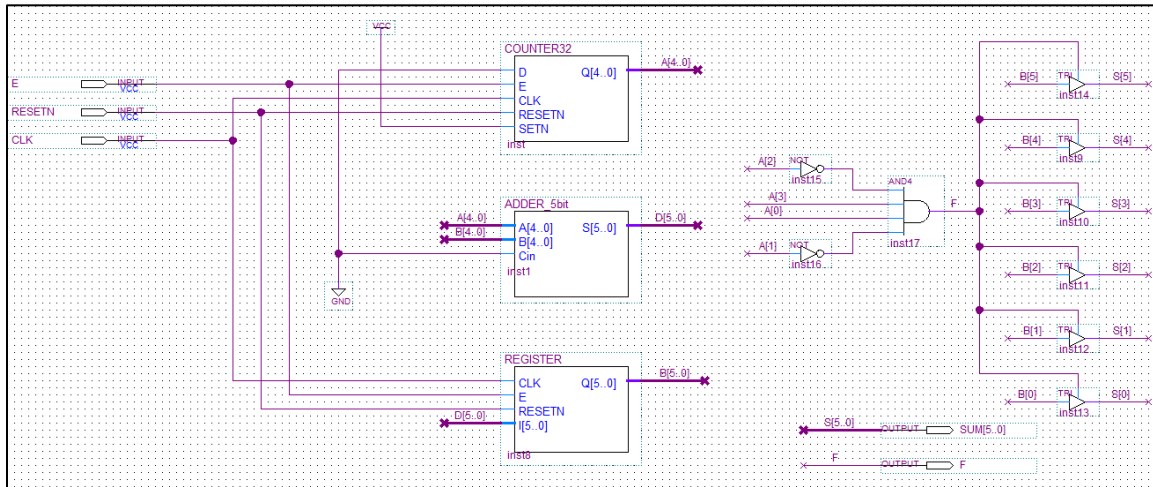
Hình 5.5 Mạch miêu tả thanh ghi 5 bit sử dụng flipflop D

b. Đóng gói mạch



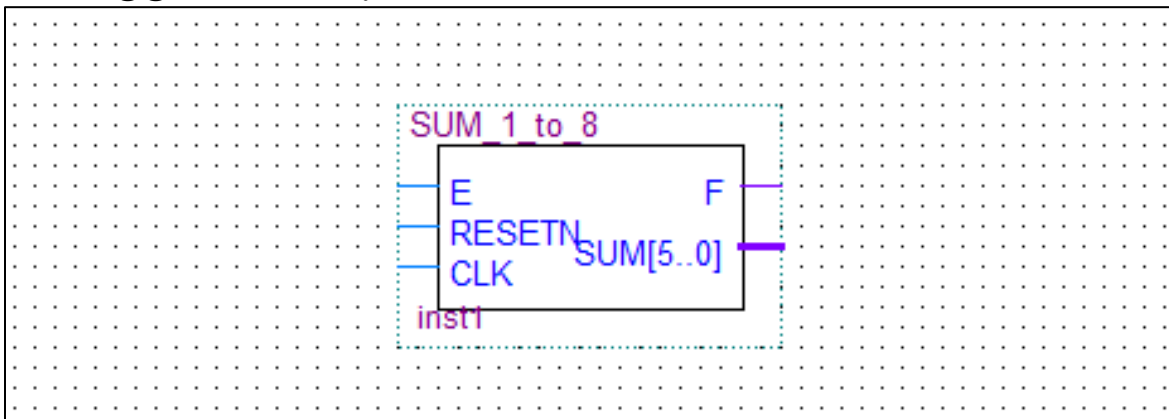
Hình 5.6 Thanh ghi 5 bit sử dụng flipflop D

4. Tổng hợp khối dữ liệu



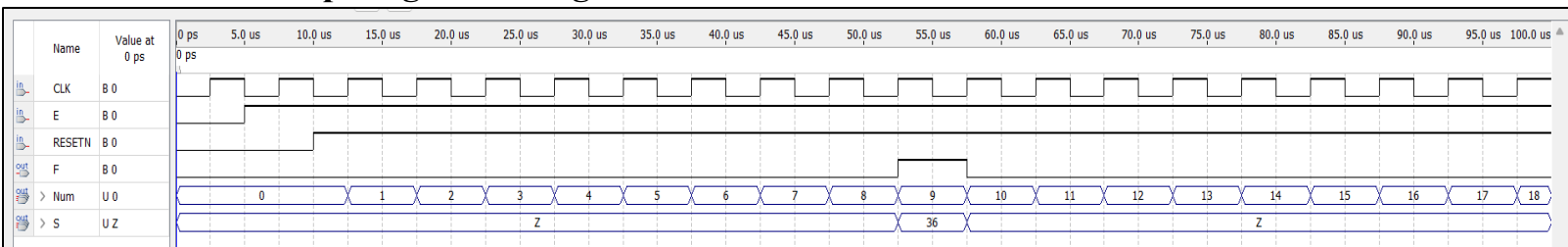
Hình 5.7 Khối dữ liệu cho bộ tính sum(1 đến 8).

5. Đóng gói khối dữ liệu



Hình 5.8 Khối dữ liệu tính tổng từ 1 đến 8

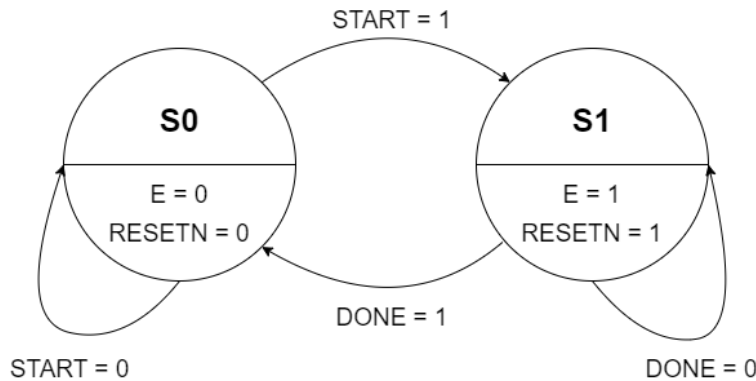
6. Mô phỏng chức năng



Hình 5.9 Kết quả mô phỏng bộ tính tổng Sum (1 đến 8)

IV. THIẾT KẾ KHỐI ĐIỀU KHIỂN

1. Sơ đồ trạng thái



Hình 5.10 Sơ đồ trạng thái

- Khối điều khiển có 2 trạng thái:
 - Trạng thái S0: Là trạng thái bắt đầu, ngõ ra trạng thái này gồm 2 tín hiệu điều khiển:
 - E (Enable) = 0: không cho phép việc đếm diễn ra.
 - RESETN = 0: reset giá trị trong thanh ghi về 0.
 - Trạng thái S1: Là trạng thái diễn ra việc tính toán ra kết quả, ngõ ra trạng thái này gồm 2 tín hiệu điều khiển:
 - E (Enable) = 1: cho phép đếm để tính toán.
 - RESETN = 1: tắt reset để thanh ghi được phép lưu các giá trị tính toán.
- Khối điều khiển có 2 điều kiện để chuyển trạng thái:
 - START: Tín hiệu cho phép bắt đầu việc tính toán.
 - DONE: Tín hiệu cho biết đã hoàn thành việc tính toán.

2. Mã hóa trạng thái

Bảng 5.2 Bảng mã hóa trạng thái

Trạng thái	Mã hóa
S0	0
S1	1

3. Chọn flipflop và lập bảng chuyển trạng thái

Từ sơ đồ trạng thái, ta lập được bảng trạng thái như sau:

Bảng 5.3 Bảng trạng thái

Trạng thái	SD			
	00	01	10	11
S0	S0	S0	S1	S1
S1	S1	S0	S1	S0

Để hiện thực khối Controller này, ta cần sử dụng 1 flipflop D để hiện thực hóa, khi đó ta có bảng chuyển trạng thái như sau:

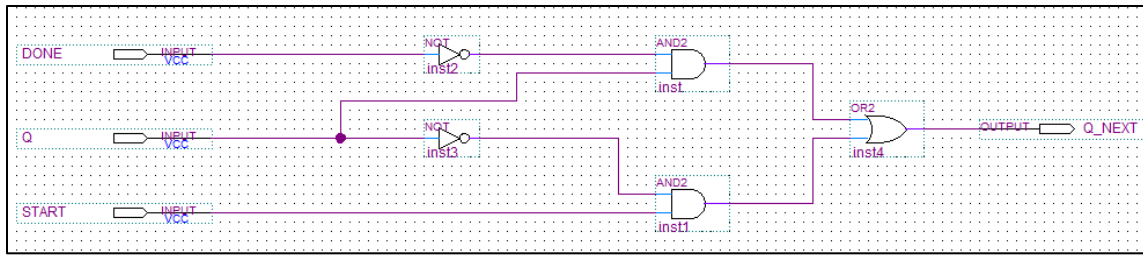
Bảng 5.4 Bảng chuyển trạng thái

S	D	Q	Q ⁺	D ₀
0	0	0	0	0
0	0	1	1	1
0	1	0	0	0
0	1	1	0	0
1	0	0	1	1
1	0	1	1	1
1	1	0	1	1
1	1	1	0	0

$$\Rightarrow D_0 = D'.Q + S.Q'$$

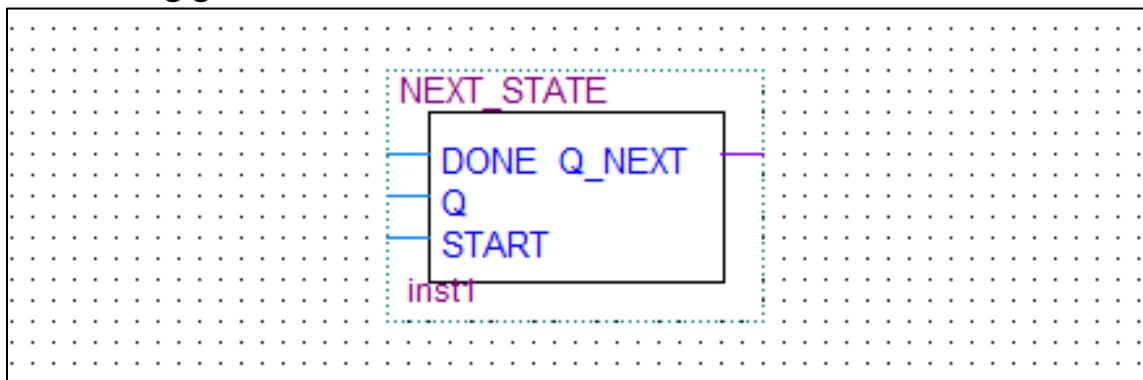
4. Thiết kế khối trạng thái kế tiếp

- Từ phương trình ngõ ra $D_0 = D'.Q + S.Q'$, ta có mạch luận lý:



Hình 5.11 Mạch miêu tả khối tạo trạng thái kế tiếp

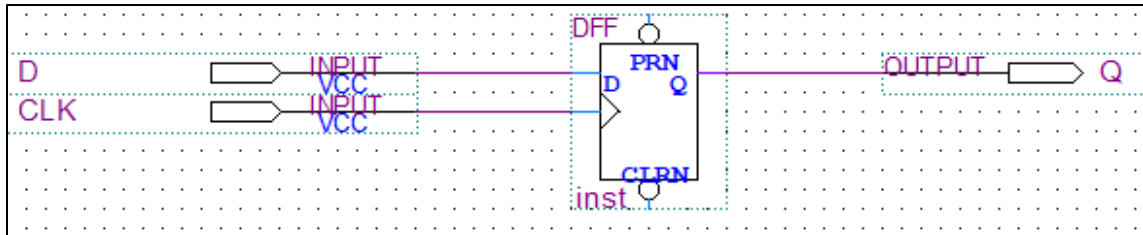
- Đóng gói mạch:



Hình 5.12 Khối trạng thái kế tiếp

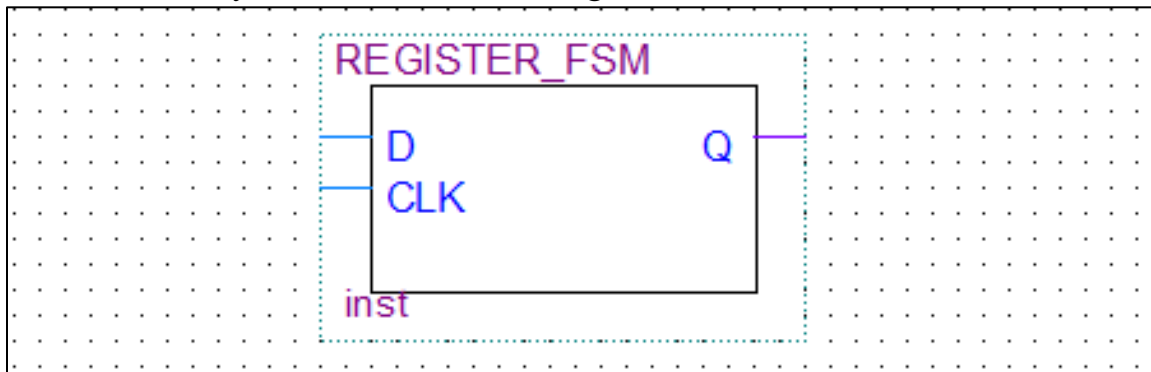
5. Thiết kế khối lưu trạng thái

Sử dụng flipflop D để lưu trạng thái theo chu kì xung clock



Hình 5.13 Mạch miêu tả khối lưu trạng thái

Từ mạch luận lý trên, ta có khối thanh ghi FSM như sau:



Hình 5.14 Khối lưu trạng thái

6. Thiết kế khối tạo tín hiệu điều khiển

a. Bảng sự thật

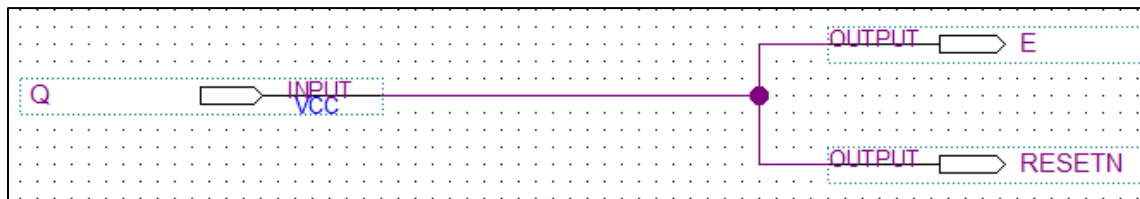
Bảng 5.5 Bảng sự thật khối tạo tín hiệu điều khiển

Q	E	RESETN
0	0	0
1	1	1

$$\Rightarrow E = Q$$

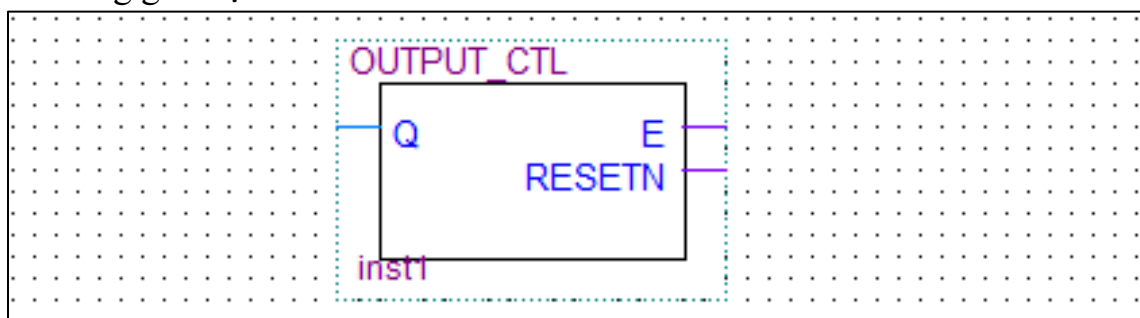
$$\Rightarrow \text{RESETN} = Q$$

b. Vẽ mạch



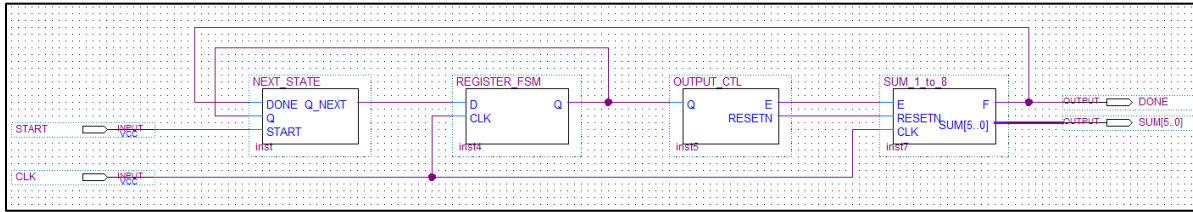
Hình 5.15 Mạch miêu tả khối tín hiệu điều khiển

c. Đóng gói mạch



Hình 5.16 Khối tín hiệu điều khiển

7. Vẽ mạch toàn bộ hệ thống



Hình 5.17 Mạch luận lý toàn bộ hệ thống

8. Mô phỏng chức năng



Hình 5.18 Mô phỏng chức năng khối tính tổng 1 đến 8

BÀI TẬP 6. THIẾT KẾ KHỐI ĐẾM SỐ LƯỢNG BIT 1 TRONG DÂY 8 BIT

I. YÊU CẦU

Thiết kế khối điều khiển và khối dữ liệu đếm số lượng bit 1 trong dây 8 bit được nhập từ ngõ vào, sau đó xuất kết quả ra màn hình mô phỏng.

II. CẤU TẠO

Để thiết kế bộ đếm số lượng bit 1 theo yêu cầu, chúng ta cần 2 phần:

- Khối dữ liệu gồm:
 - Bộ MUX 41 cho việc dịch/ nạp dữ liệu vào flipflop.
 - Thanh ghi dịch (phải) 8 bit.
 - Bộ đếm dùng để đếm số lượng bit 1.
- Khối điều khiển gồm:
 - Khối trạng thái kế tiếp.
 - Khối lưu trạng thái.
 - Khối điều khiển ngõ ra.

III. THIẾT KẾ KHỐI DỮ LIỆU

1. Ý tưởng

Ý tưởng cho thiết kế này là chúng ta kiểm tra bit có trọng số nhỏ nhất (LSB), nếu bit này có trọng số là 1 thì sẽ làm tăng bộ Counter lên 1 đơn vị và sử dụng thanh ghi SPwPL để dịch phải dữ liệu. Sau đó tiếp tục các việc trên cho đến khi cả 8 bit dữ liệu đều bằng 0.

2. Thiết kế bộ MUX 41

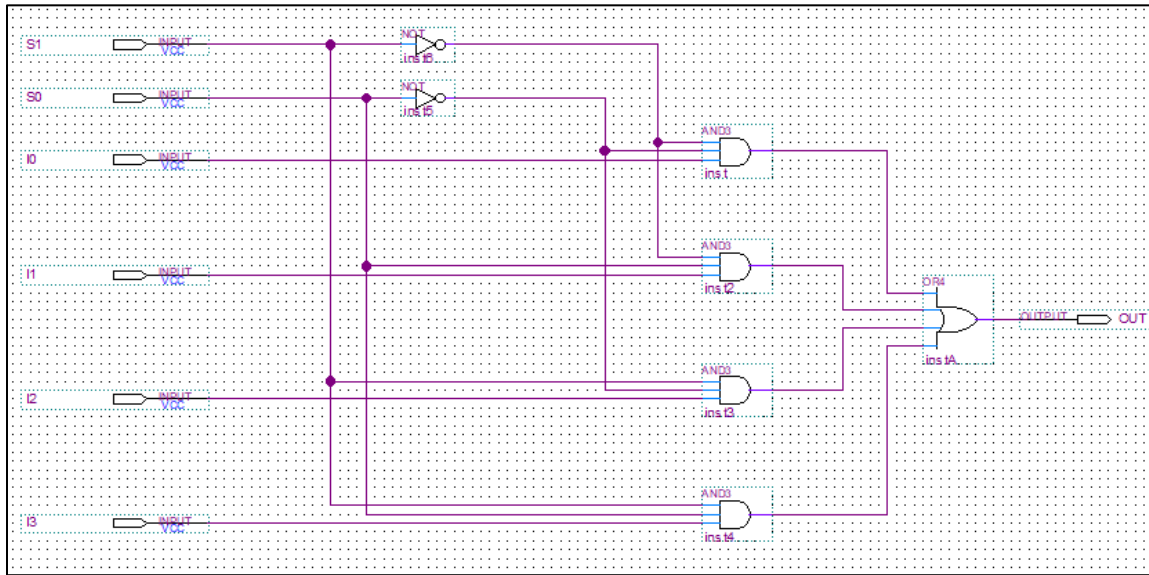
- Bảng hoạt động

Bảng 6.1 Bảng hoạt động MUX 41

S0	S1	OUT
0	0	I ₀
0	1	I ₁
1	0	I ₂
1	1	I ₃

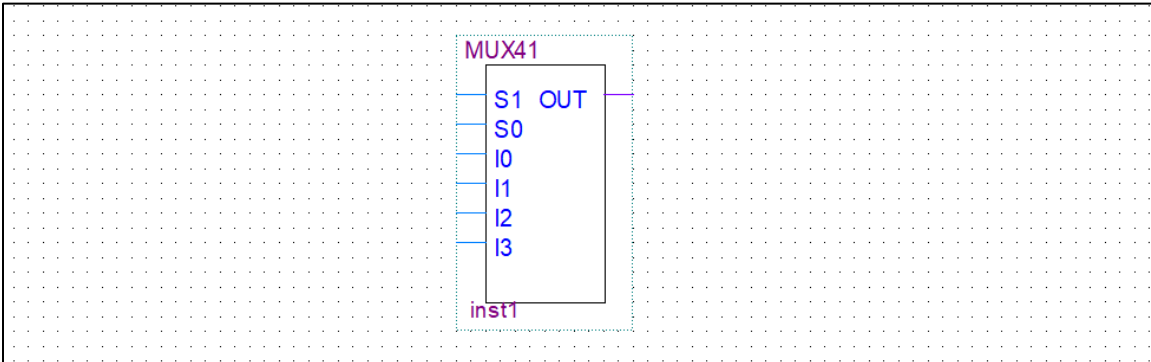
$$\Rightarrow \text{OUT} = S1'.S0'.I0 + S1'.S0.I1 + S1.S0'.I2 + S1.S0.I3$$

■ Vẽ mạch



Hình 6.1 Mạch miêu tả MUX 41

■ Đóng gói



Hình 6.2 Khối MUX 41

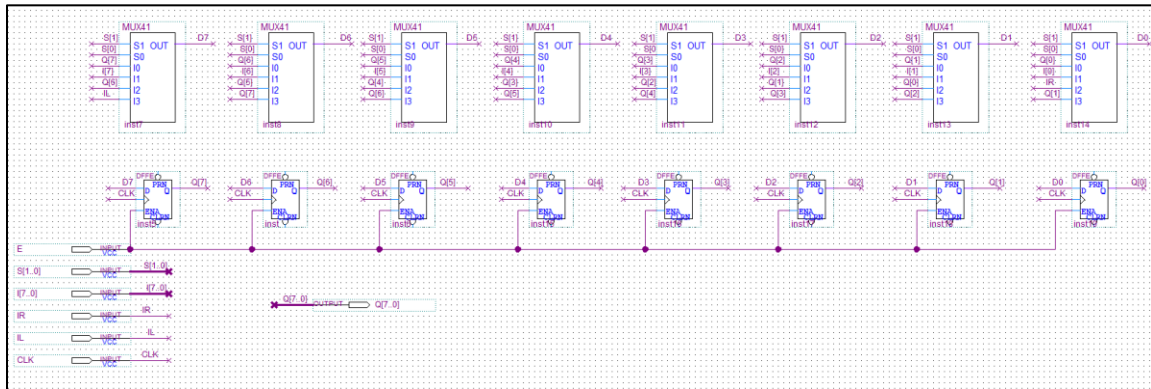
3. Thiết kế thanh ghi dịch 8 bit

■ Bảng hoạt động

Bảng 6.2 Bảng hoạt động của thanh ghi dịch

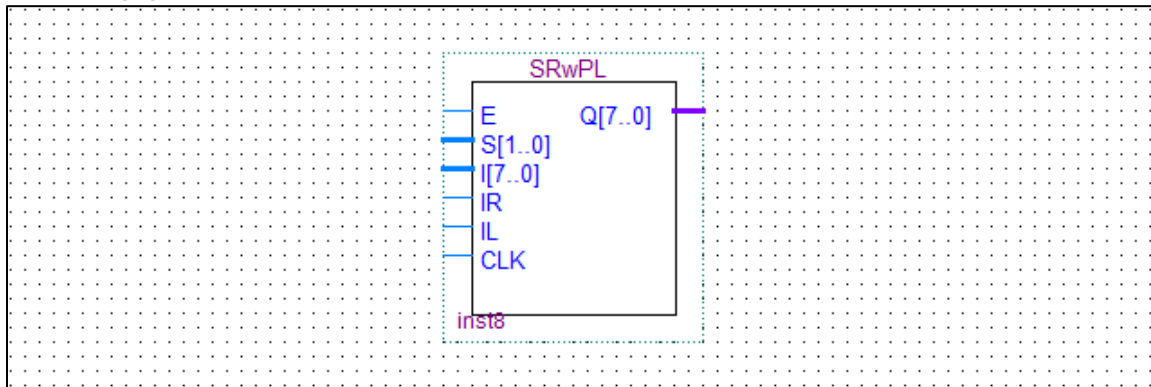
TTHT S_1S_0	Thanh ghi dịch	TTKT $Q_3Q_2Q_1Q_0$
00	Không đổi	$Q_3Q_2Q_1Q_0$
01	Load input	$I_3I_2I_1I_0$
10	Dịch trái	$Q_2Q_1Q_0I_R$
11	Dịch phải	$I_LQ_3Q_2Q_1$

■ Vẽ mạch



Hình 6.3 Mạch miêu tả thanh ghi dịch 8 bit

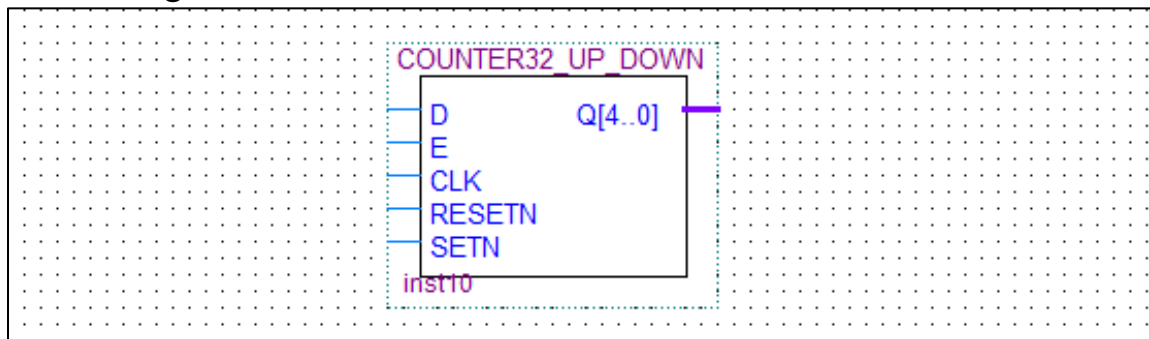
■ Đóng gói mạch



Hình 6.4 Khối thanh ghi dịch 8 bit

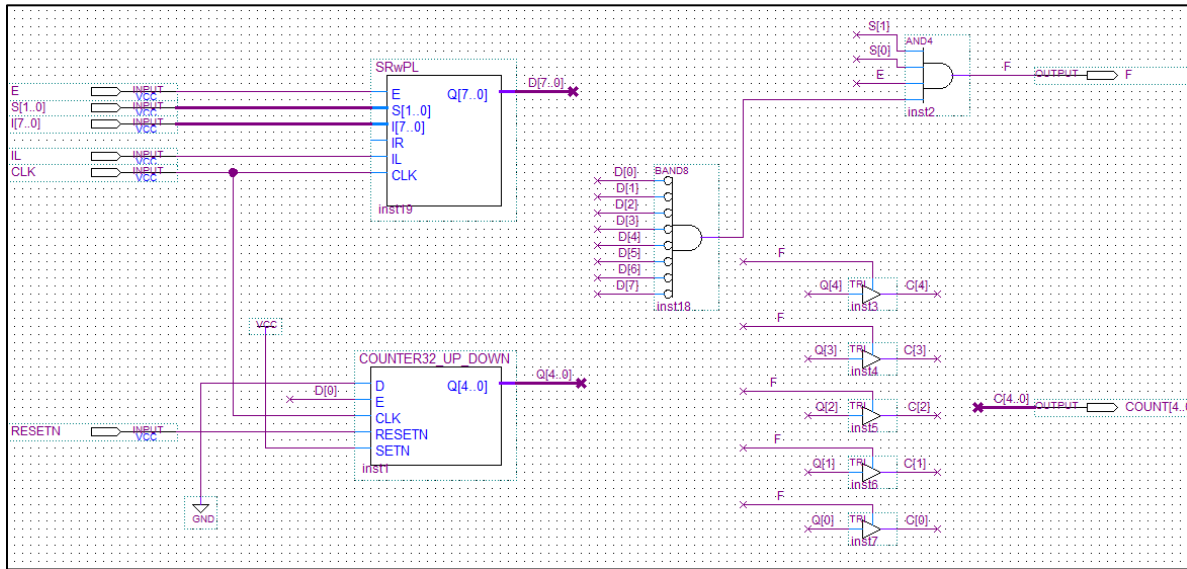
4. Thiết kế bộ đếm

Tái sử dụng bộ đếm 5 bit:



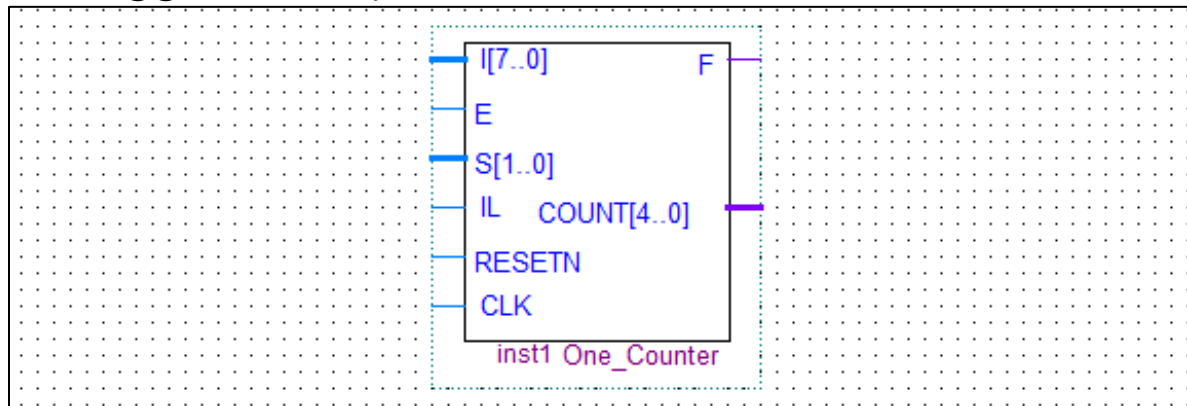
Hình 6.5 Khối đếm 5 bit

5. Tổng hợp khối dữ liệu



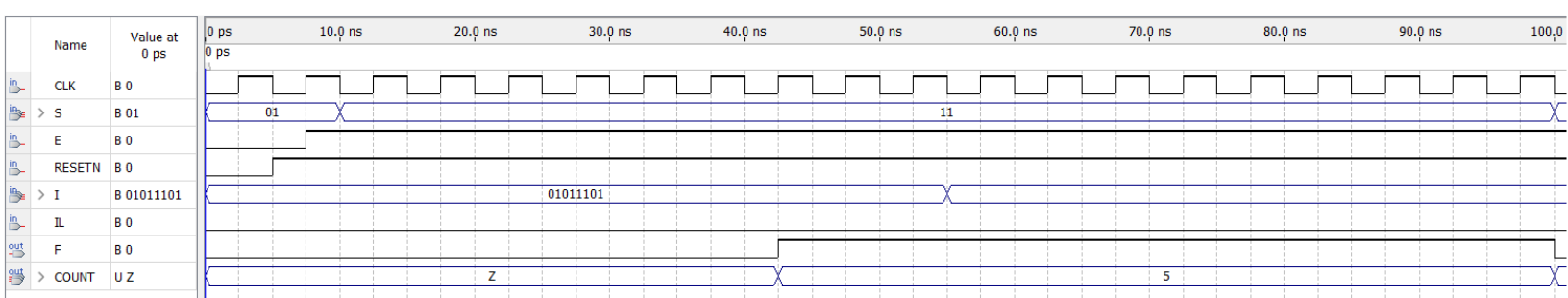
Hình 6.6 Mạch miêu tả tổng quát khối đếm số bit 1

6. Đóng gói khối dữ liệu



Hình 6.7 Đóng gói khối dữ liệu

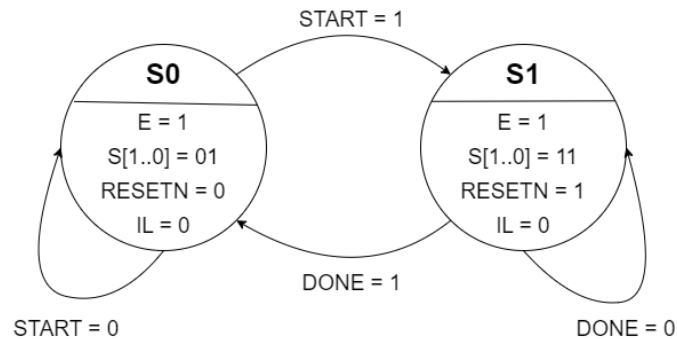
7. Mô phỏng chức năng



Hình 6.8 Kết quả mô phỏng khối dữ liệu

IV. THIẾT KẾ KHỐI ĐIỀU KHIỂN

1. Sơ đồ trạng thái



Hình 6.9 Sơ đồ trạng thái

- Khối điều khiển có 2 trạng thái:
 - Trạng thái S0: Là trạng thái bắt đầu, ngõ ra trạng thái này gồm 4 tín hiệu điều khiển:
 - E (Enable) = 1: thanh ghi được phép dịch.
 - S[1..0] = 01: cho phép nạp 8 bit giá trị đầu vào.
 - RESETN = 0: reset giá trị trong thanh ghi về 0.
 - IL = 0: khi dịch phải, bit MSB được load bằng 0.
 - Trạng thái S1: Là trạng thái diễn ra việc tính toán ra kết quả, ngõ ra trạng thái này gồm 4 tín hiệu điều khiển:
 - E (Enable) = 1: thanh ghi được phép dịch.
 - S[1..0] = 11: dịch phải giá trị trong thanh ghi dịch.
 - RESETN = 1: tắt reset để thanh ghi được phép lưu các giá trị tính toán.
 - IL = 0: khi dịch phải, bit MSB được load bằng 0.
- Khối điều khiển có 2 điều kiện để chuyển trạng thái:
 - START: Tín hiệu cho phép bắt đầu việc tính toán.
 - DONE: Tín hiệu cho biết đã hoàn thành việc tính toán.

2. Mã hóa trạng thái

Bảng 6.3 Bảng mã hóa trạng thái

Trạng thái	Mã hóa
S0	0
S1	1

3. Chọn flipflop và lập bảng chuyển trạng thái

Từ sơ đồ trạng thái, ta lập được bảng trạng thái như sau:

Bảng 6.4 Bảng trạng thái

Trạng thái	SD			
	00	01	10	11
S0	S0	S0	S1	S1
S1	S1	S0	S1	S0

Để hiện thực khối Controller này, ta cần sử dụng 1 flipflop (loại flipflop D) để hiện thực hóa, khi đó ta có bảng chuyển trạng thái như sau:

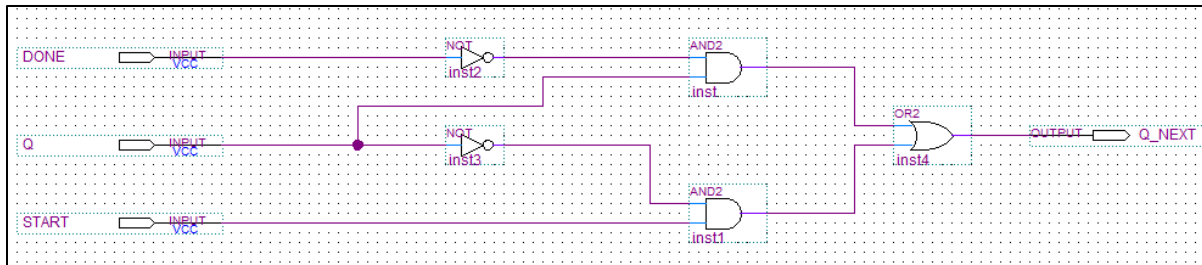
Bảng 6.5 Bảng chuyển trạng thái

S	D	Q	Q ⁺	D ₀
0	0	0	0	0
0	0	1	1	1
0	1	0	0	0
0	1	1	0	0
1	0	0	1	1
1	0	1	1	1
1	1	0	1	1
1	1	1	0	0

$$\Rightarrow D_0 = D'.Q + S.Q'$$

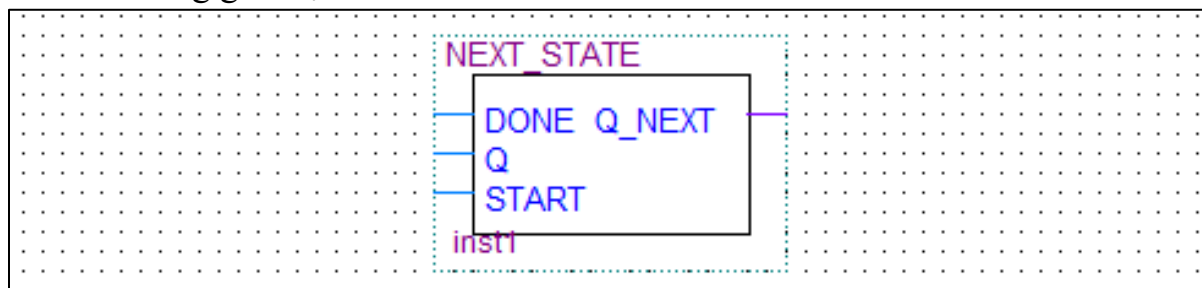
4. Vẽ mạch khối tạo trạng thái kế tiếp

- Vẽ mạch luận lý miêu tả khối tạo trạng thái kế tiếp:



Hình 6.10 Mạch miêu tả khối tạo trạng thái kế tiếp

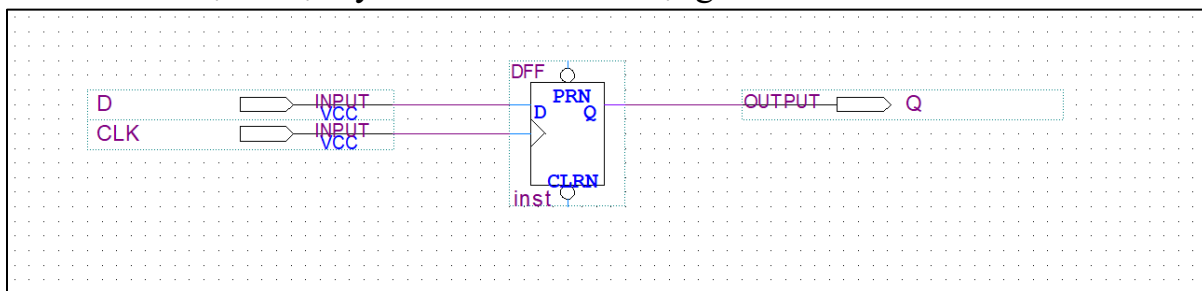
- Đóng gói mạch:



Hình 6.11 Khối trạng thái kế tiếp

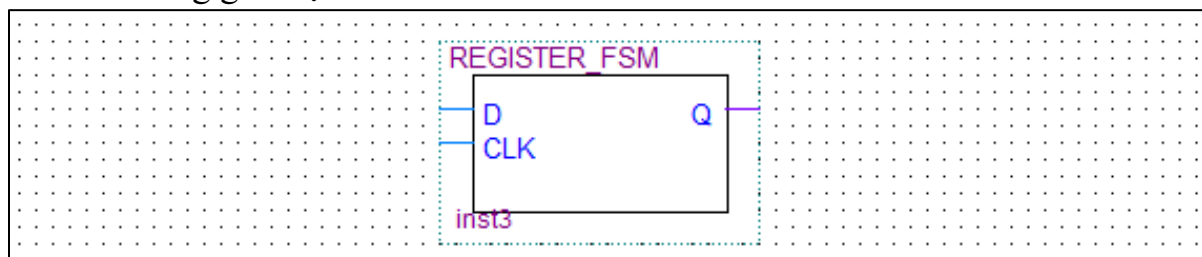
5. Khối lưu trạng thái

- Vẽ mạch luận lý miêu tả khối lưu trạng thái



Hình 6.12 Mạch miêu tả khối lưu trạng thái

- Đóng gói mạch



Hình 6.13 Khối lưu trạng thái

6. Khởi tạo ra tín hiệu điều khiển

- Bảng sự thật

Bảng 6.6 Bảng sự thật khởi tạo tín hiệu điều khiển

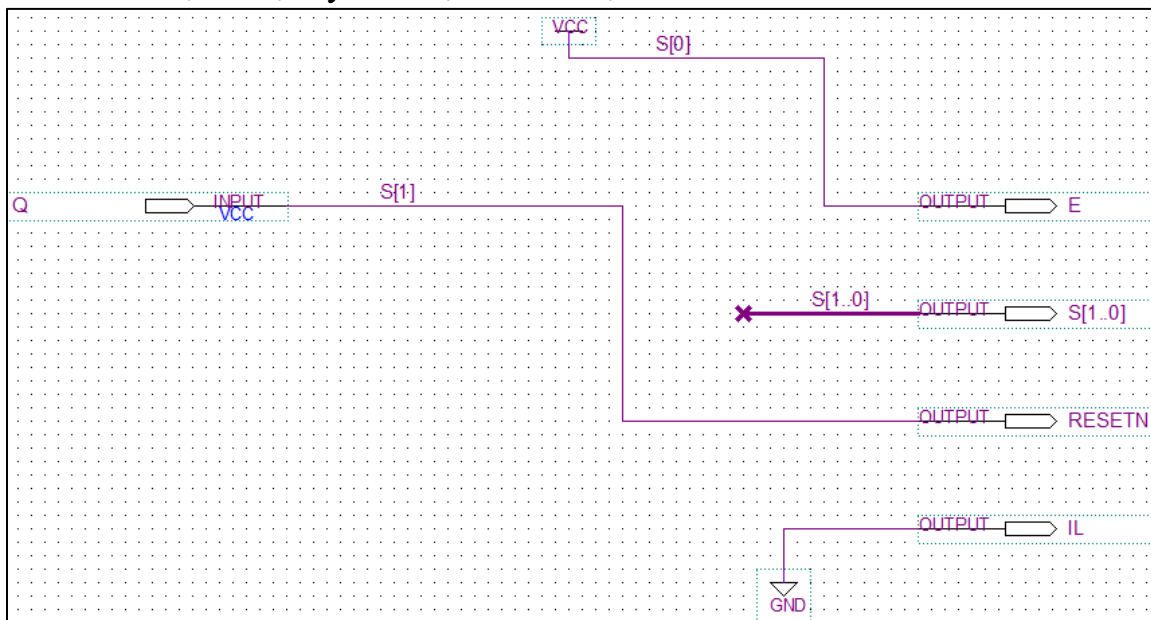
Trạng thái	Q	E	S[1]	S[0]	IL	RESETN
S0	0	1	0	1	0	0
S1	1	1	1	1	0	1

$$\Rightarrow E = S[0] = 1$$

$$\Rightarrow S[1] = \text{RESETN} = Q$$

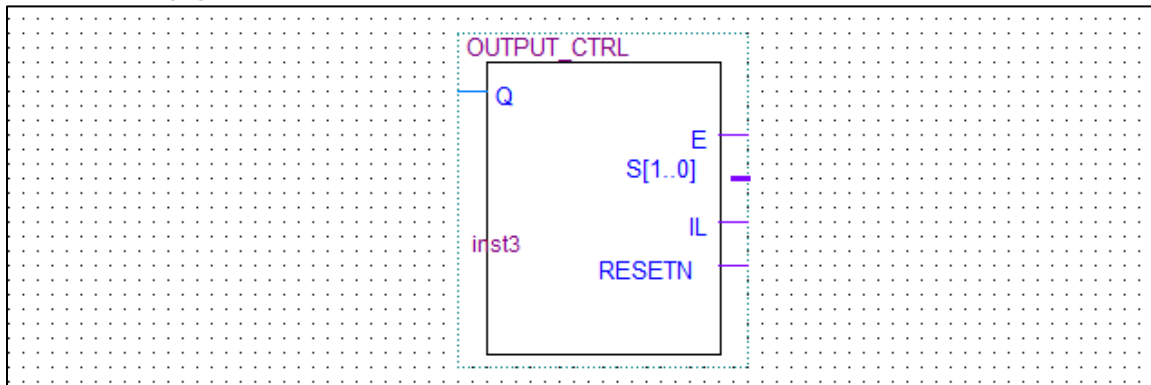
$$\Rightarrow \text{IL} = 0$$

- Vẽ mạch luận lý khởi tạo ra tín hiệu điều khiển



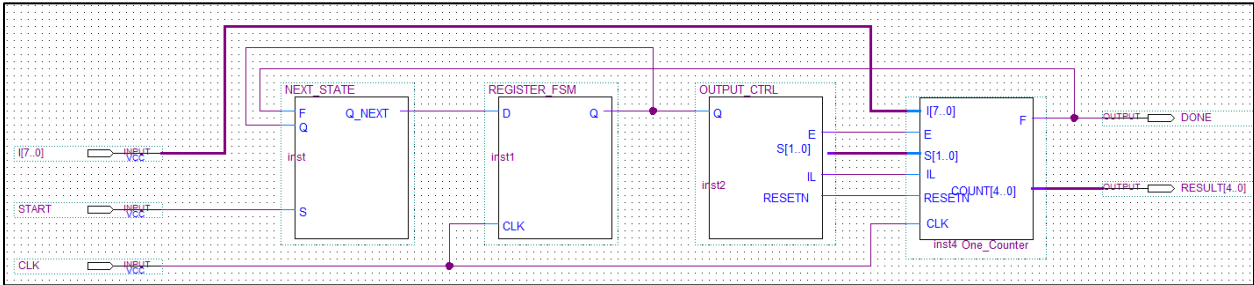
Hình 6.14 Mạch luận lý khởi tạo tín hiệu điều khiển

- Đóng gói mạch



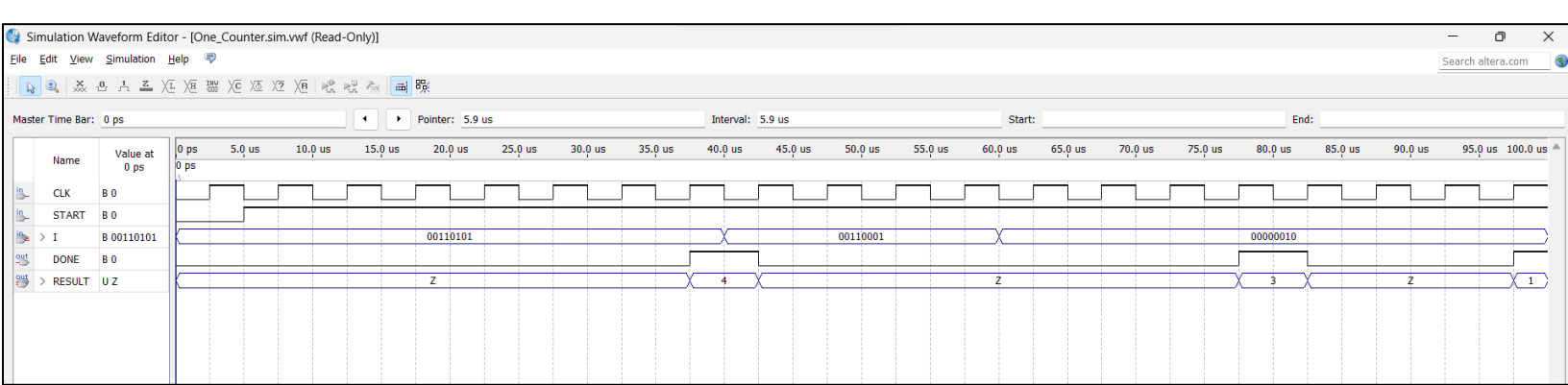
Hình 6.15 Đóng gói khởi tạo tín hiệu điều khiển

7. Vẽ mạch toàn bộ hệ thống



Hình 6.16 Mạch luận lý toàn bộ hệ thống

8. Mô phỏng chức năng



Hình 6.17 Kết quả chạy mô phỏng