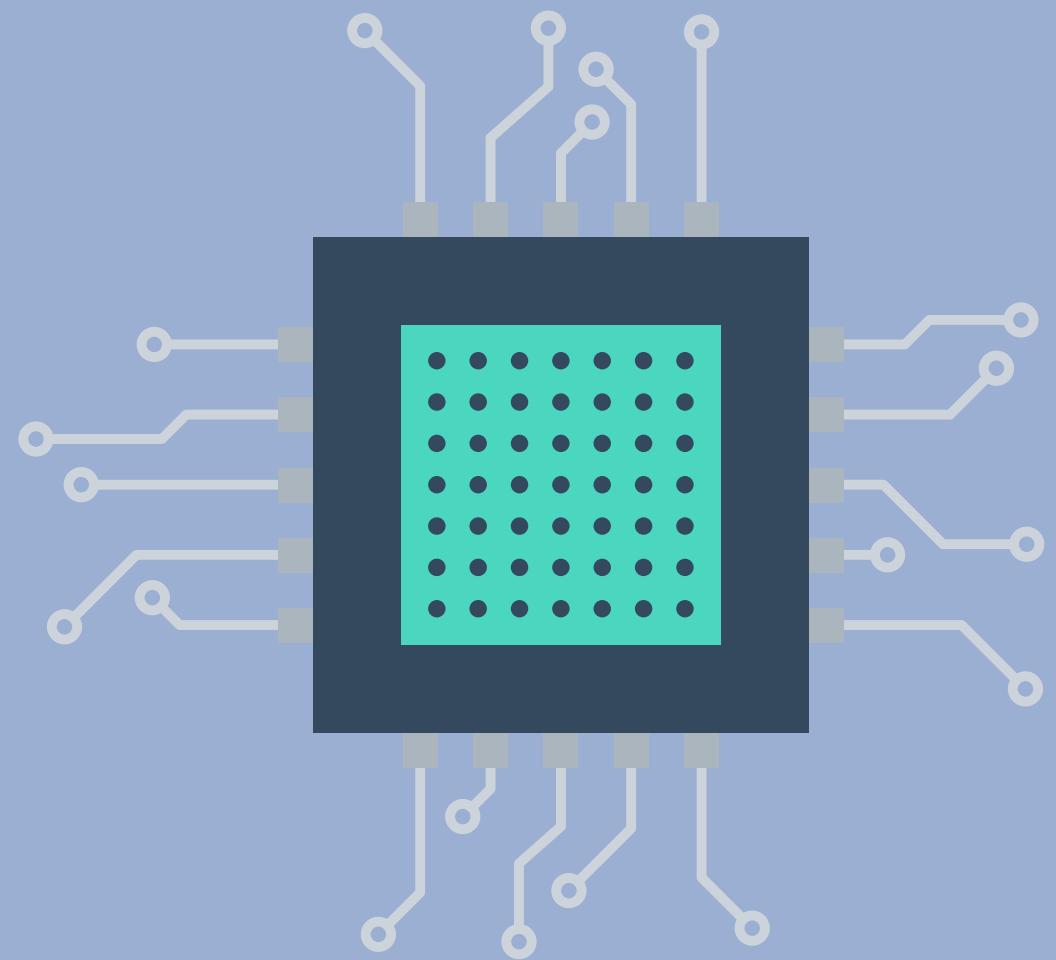


Thiết kế luận lý số CE118

Khoa Kỹ Thuật Máy Tính

# Báo cáo đồ án

THIẾT KẾ MẠCH ĐIỀU KHIỂN CHO  
PHÉP TÍNH GIAI THỪA



# Giai thừa là gì?

## Factorial

$$n! = n * (n - 1) * (n - 2) * (n - 3) * \dots * 3 * 2 * 1$$

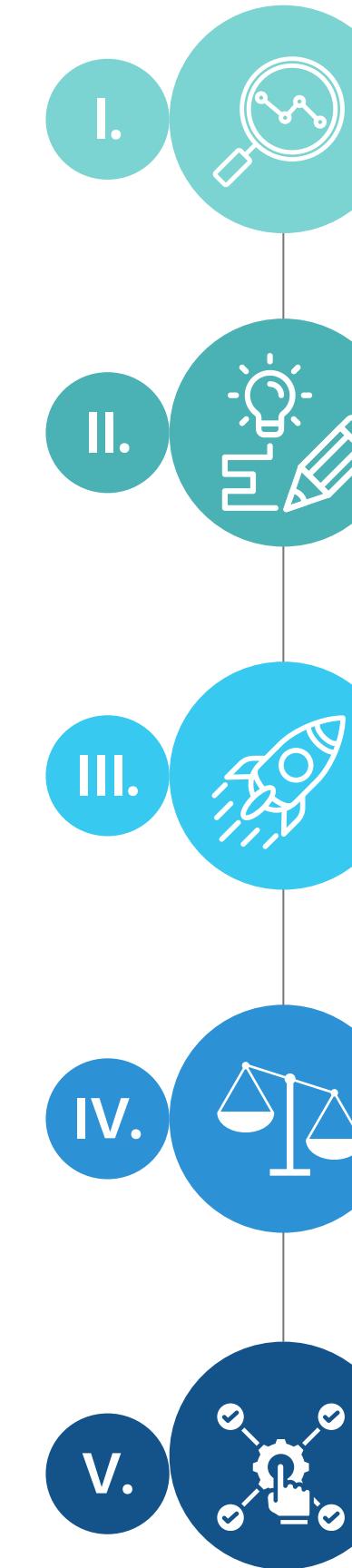
$$0! = 1$$

$$1! = 1$$

$$2! = 2 \times 1 = 2$$

$$3! = 3 \times 2 \times 1 = 6$$

# CÁC BƯỚC THỰC HIỆN



*Phân tích bài toán*

*Thiết kế Datapath và Control Unit*

*Các phương pháp tối ưu hóa*

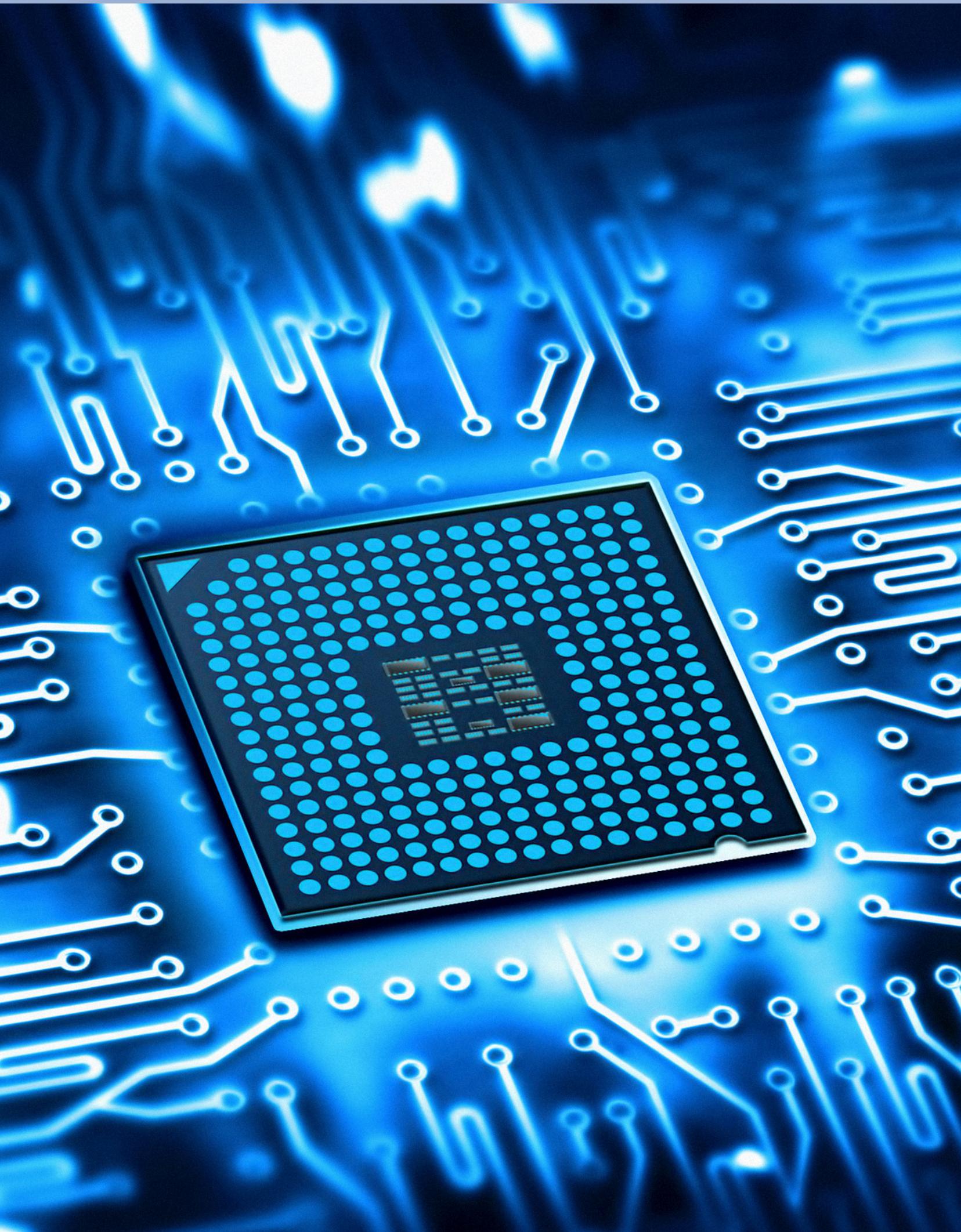
*So sánh giữa các phương pháp*

*Mô phỏng và nạp kit*



# I. Phân tích bài toán

- Ý TƯỞNG
- TẠO MÃ GIẢ
- ĐỊNH NGHĨA THANH GHI
- VẼ SƠ ĐỒ TRẠNG THÁI



# Ý tưởng thực hiện bài toán

C:

```
int a = 1;  
int b = N;  
while (b != 0) {  
    a = a * b;  
    b = b - 1;  
}  
  
cout << a;
```

=> Khởi tạo giá trị cho 2 thanh  
ghi

=> Dùng 1 bộ compare để so  
sánh

=> Dùng 1 ALU có chức năng  
thực hiện phép nhân và trừ

# Mã giả bài toán

Mã giả

```
1. Out := 1  
2. Data := Import  
3. While ( Data != 0) repeat  
    4. Out = Out * Data  
    5. Data = Data - 1  
end while  
6. Outport = Out
```

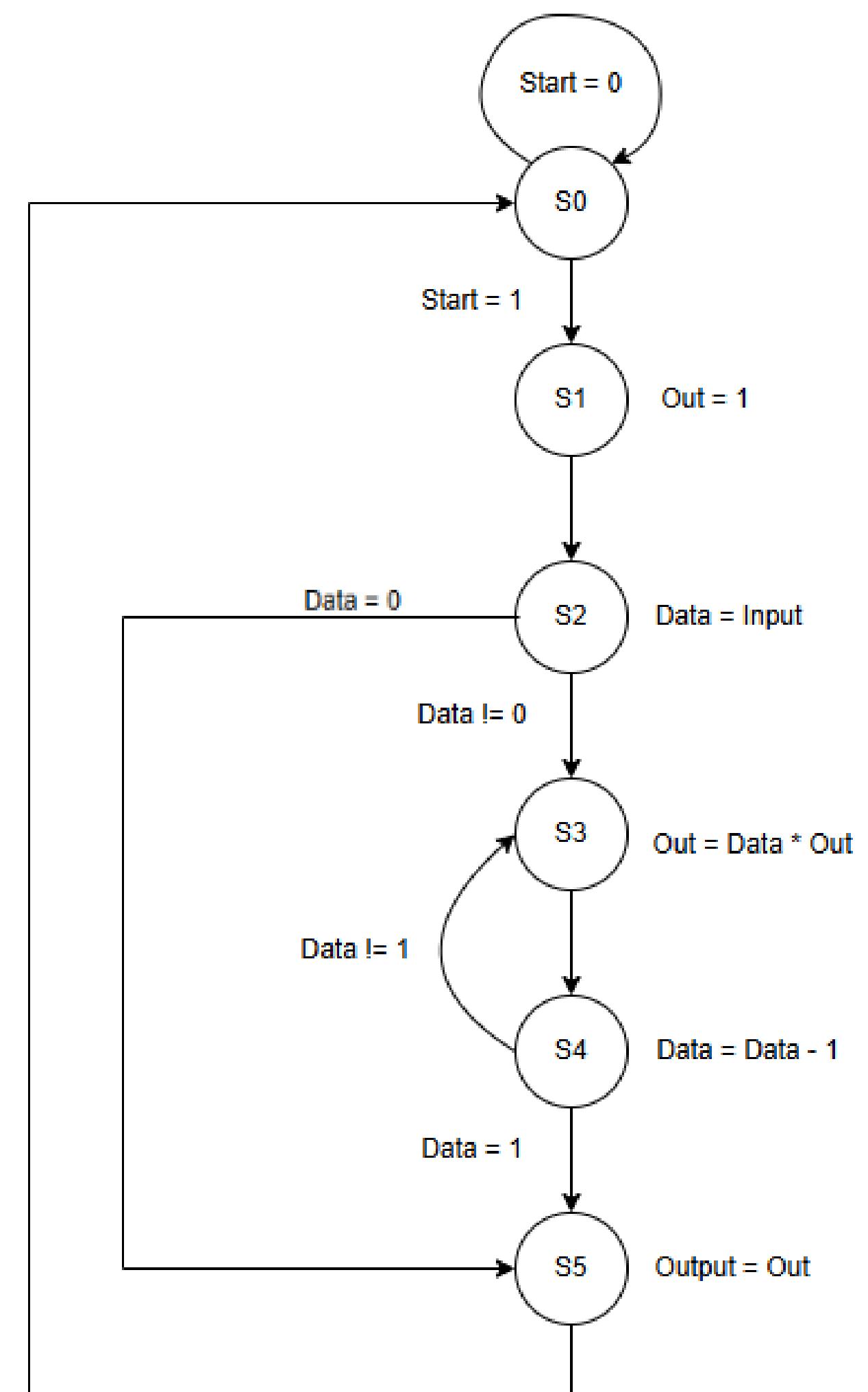
Thanh ghi

R1      Data  
R3      Out

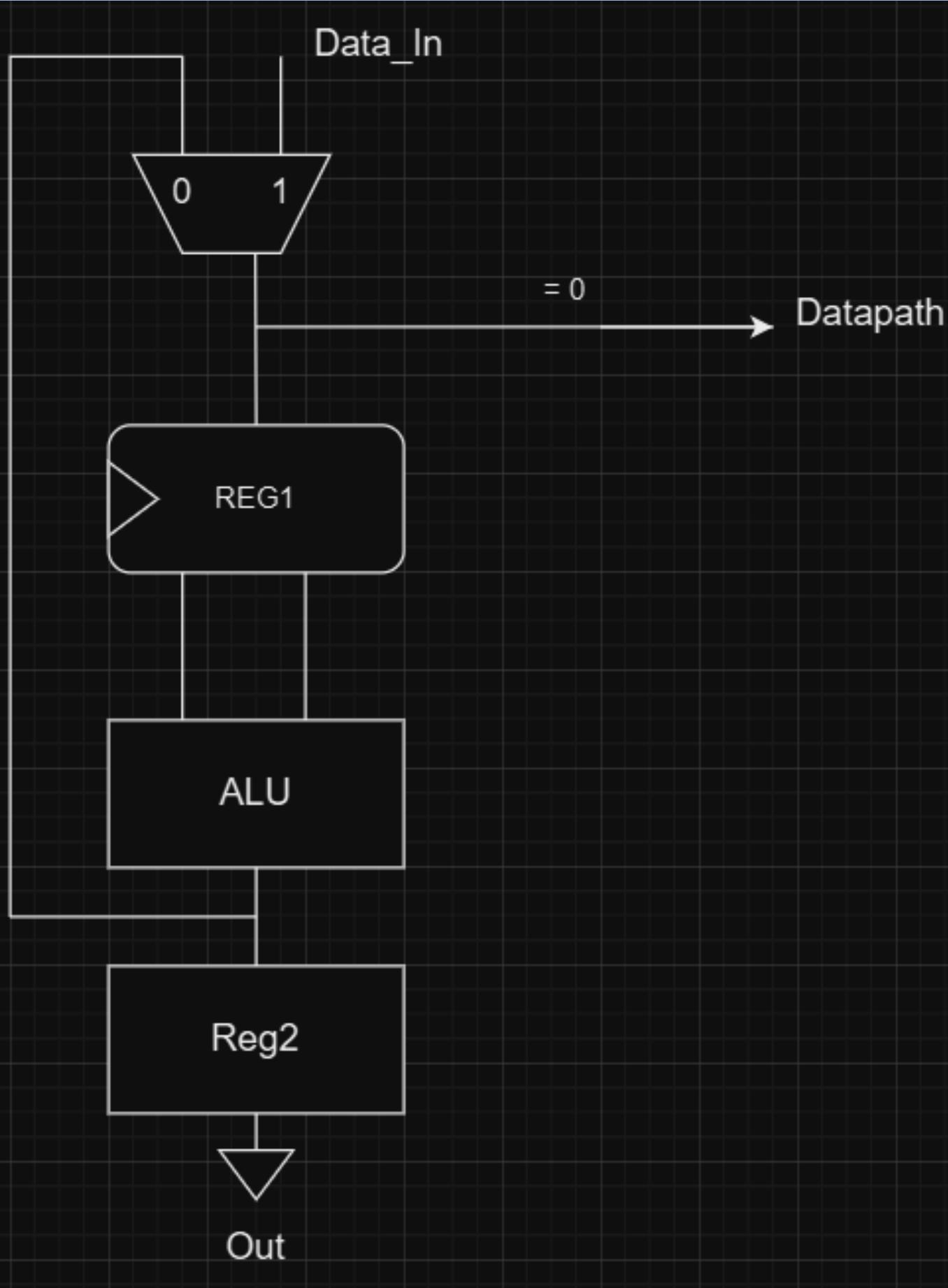
# Sơ đồ trạng thái

STATE	Q2Q1Q0	IE	WA2WA1WA0WE	RAA2RAA1RAA0REA	RAB2RAB1RAB0REB	S3S2S1S0	OE	CIN	DONE
S0	000	0	XXX0	XXX0	XXX0	XXXX	0	X	0
S1	001	0	0111	0001	XXX0	0111	0	0	0
S2	010	1	0011	XXX0	XXX0	XXXX	0	X	0
S3	011	0	0111	0011	0111	1010	0	0	0
S4	100	0	0011	0011	XXX0	0100	0	1	0
S5	101	0	XXX0	0111	0001	0101	1	0	1

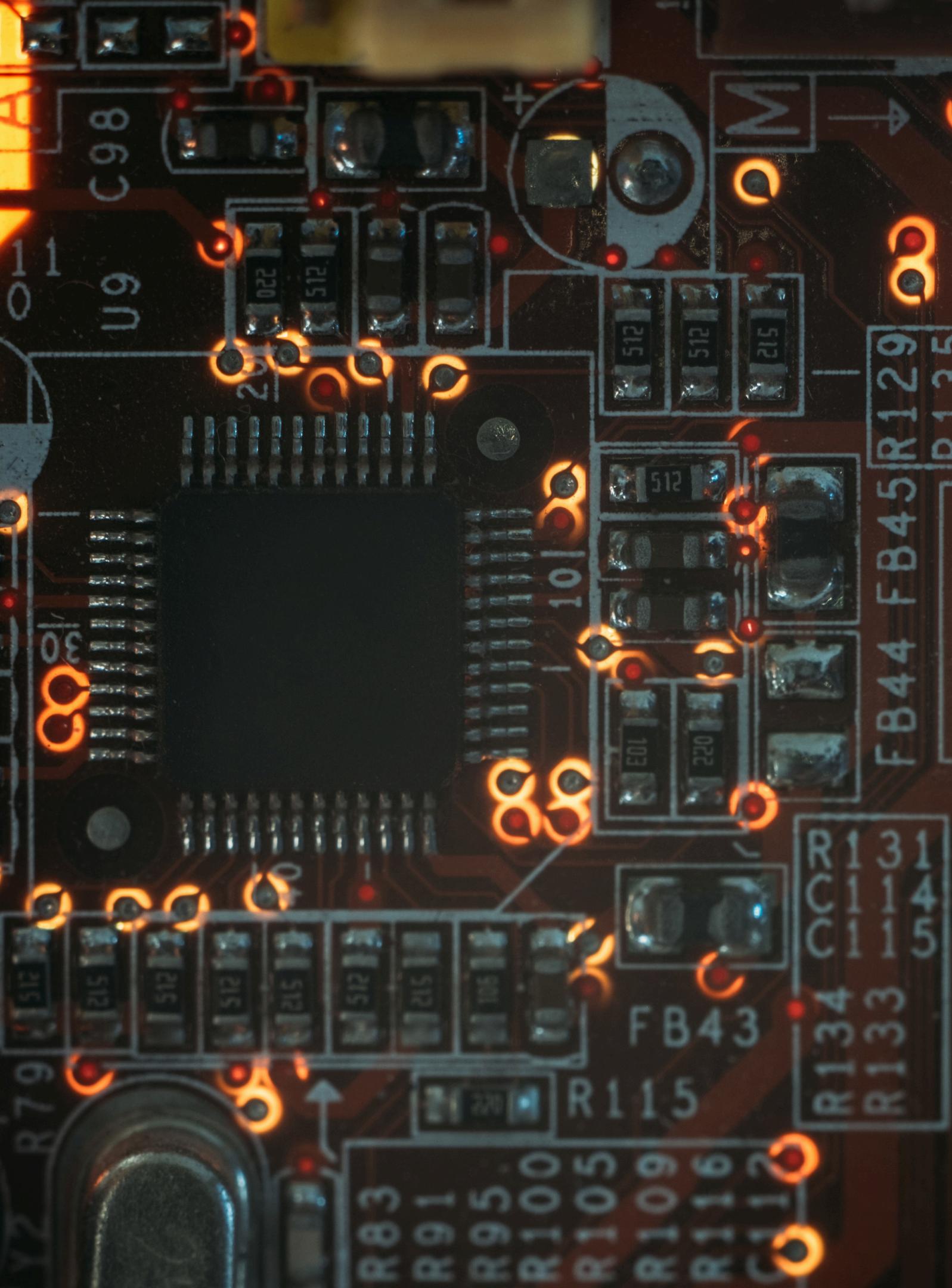
DECRE                    0100  
 INCRE                    0111  
 ADD                      0101  
 MUL                     1010



# Sơ đồ Datapath



- **Bộ mux:** Dùng để chọn input cho phép toán.
- **Bộ Reg1:** Dùng để lưu trữ giá trị cho các thanh ghi.
- **Bộ ALU:** Thực hiện chức năng tính toán.
- **Bộ Reg2:** Có 1 tín hiệu load enable có chức năng giống với bộ tri-state mình từng học.
- **Tín hiệu Datapath:** Kiểm tra Input có bằng 0 hay không? Nếu có, thì sẽ thực hiện xuất giá trị tại Reg2.

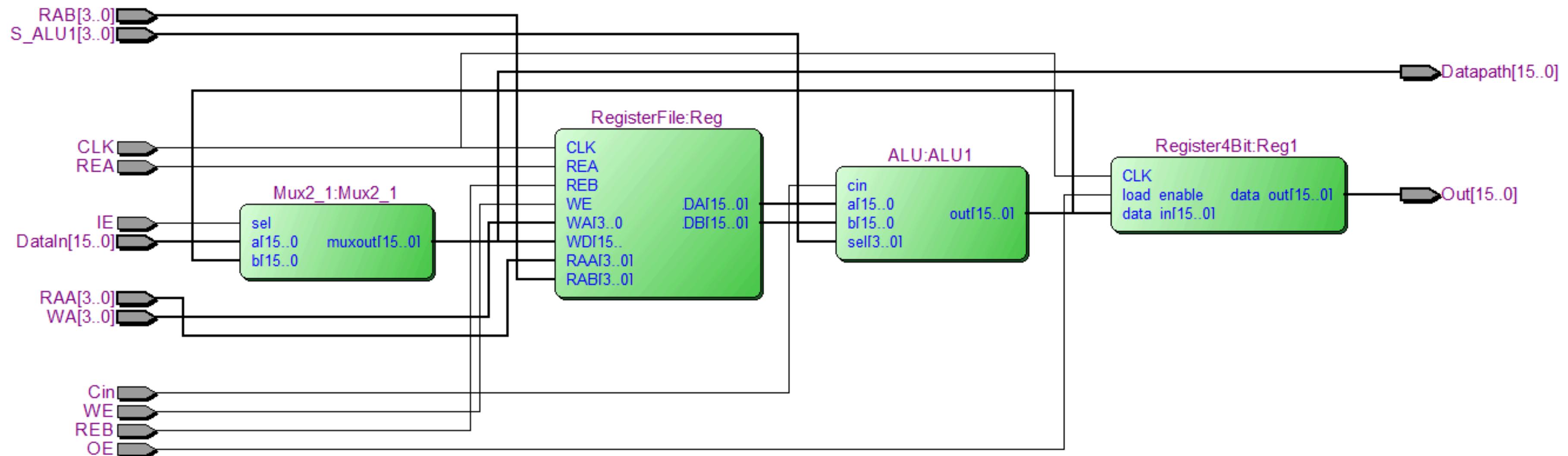


## II. Thiết kế datapath và control unit

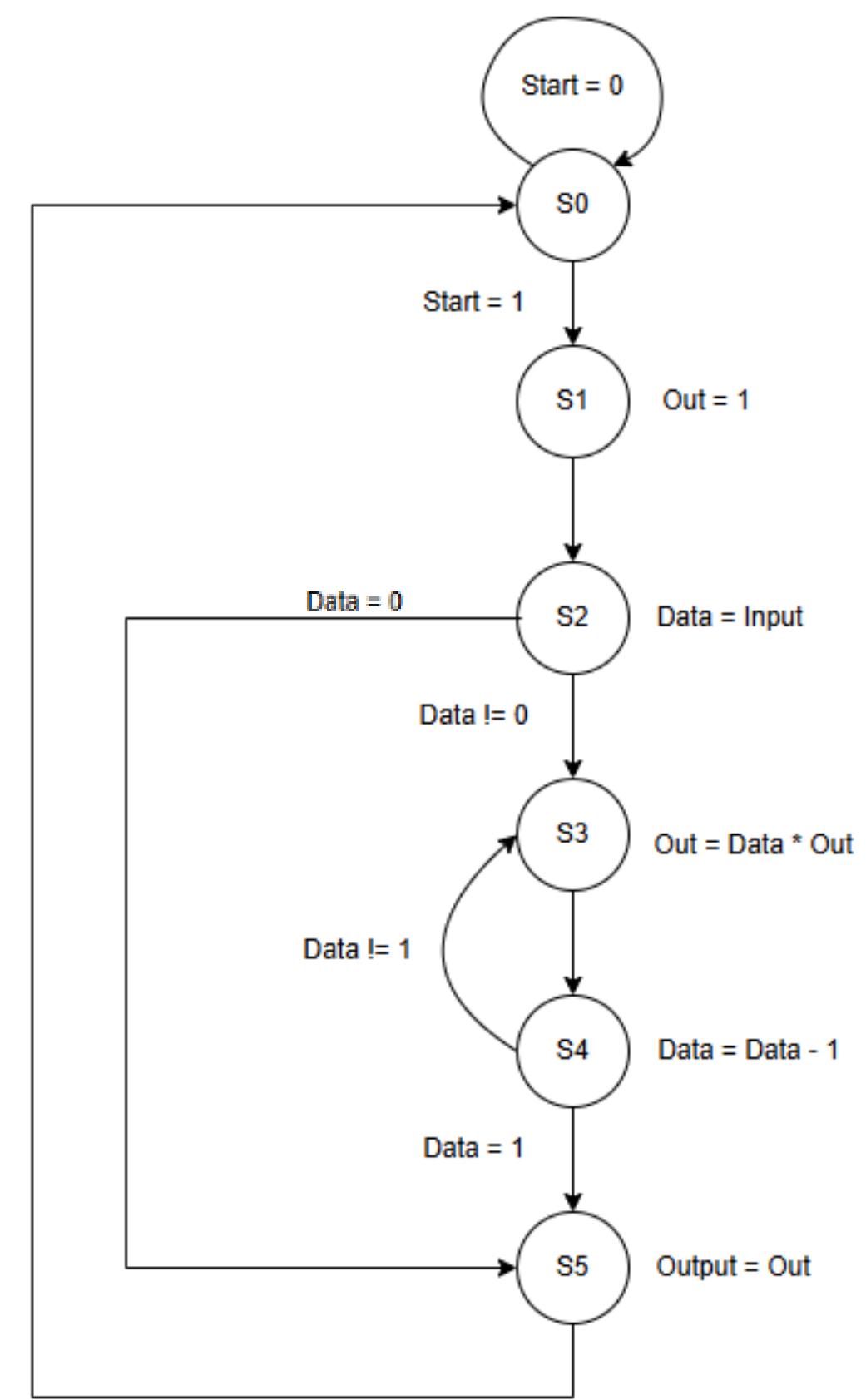
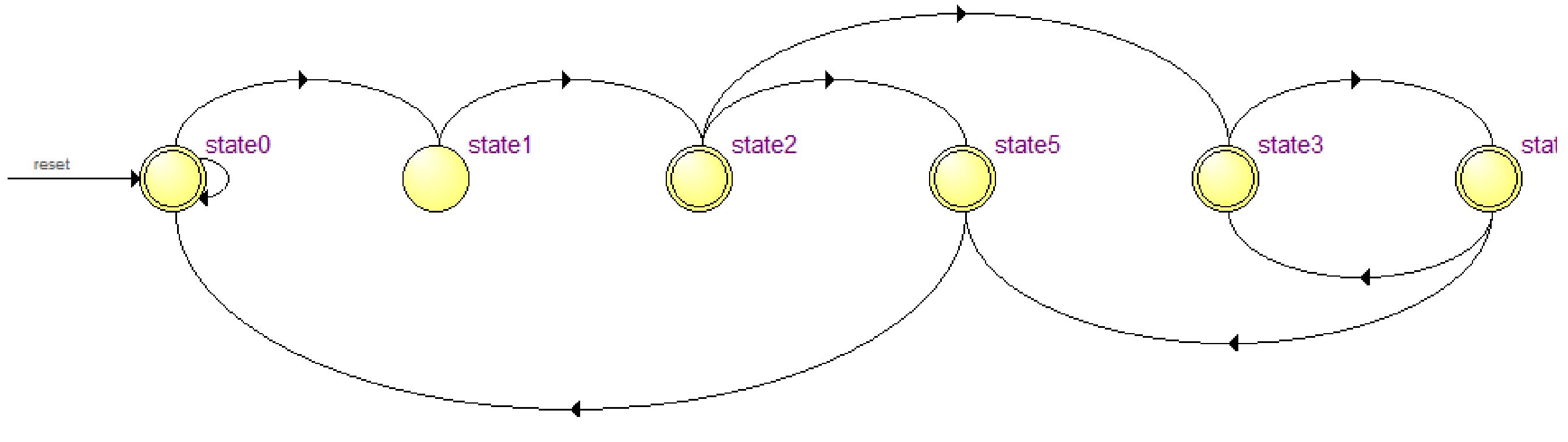
- THIẾT KẾ CÁC KHỐI CON CỦA DATAPATH
- THIẾT KẾ CONTROL UNIT
- KẾT HỢP DATAPATH & CONTROL UNIT VÀ KẾT QUẢ MÔ PHỎNG

a)

# Thiết kế các khối con của Datapath



## b) Thiết kế Control Unit



## b) Thiết kế Control Unit

```
module Control_Unit(CLK, Start, IE, WE, RAA, REA, RAB, REB, OE, S_ALU1, Done, Datapath, Cin);
    input CLK, Start;
    input [15:0] Datapath;
    output reg IE, OE, WE, REA, REB, Done, Cin;
    output reg [3:0] WA;
    output reg [3:0] RAA;
    output reg [3:0] RAB;
    output reg [3:0] S_ALU1;

    parameter state0 = 3'b000;
    parameter state1 = 3'b001;
    parameter state2 = 3'b010;
    parameter state3 = 3'b011;
    parameter state4 = 3'b100;
    parameter state5 = 3'b101;

    reg [2:0] c_state;
    reg [2:0] n_state;

    always@(posedge CLK) begin
        c_state <= n_state;
    end
endmodule
```

```
always@(*) begin
    case (c_state)
        state0: begin n_state = (Start) ? state1: state0;
            IE = 1'b0;
            OE = 1'b0;
            WE = 1'b0;
            REA = 1'b0;
            REB = 1'b0;
            Done = 1'b0;
        end
        state1: begin n_state = state2;
            IE = 0;
            WE = 1;
            WA = 4'b0011;
            REA = 1;
            REB = 0;
            RAA = 4'b0000;
            RAB = 4'b0000;
            S_ALU1 = 4'b0111;
            Cin = 1'b0;
            OE = 0;
            Done = 0;
        end
    endcase
end
```

## b) Thiết kế Control Unit

```
state2: begin n_state = (Datapath == 16'b0) ? state5 : state3;
    IE = 1'b1;
    WE = 1'b1;
    OE = 1'b0;
    REA = 1'b0;
    REB = 1'b0;
    WA = 4'b0001;
    Done = 1'b0;
end

state3: begin n_state = state4;
    IE = 0;
    WE = 1;
    WA = 4'b0011;
    REA = 1;
    REB = 1;
    RAA = 4'b0001;
    RAB = 4'b0011;
    S_ALU1 = 4'b1010;
    Cin = 1'b0;
    OE = 0;
    Done = 0;
end
```

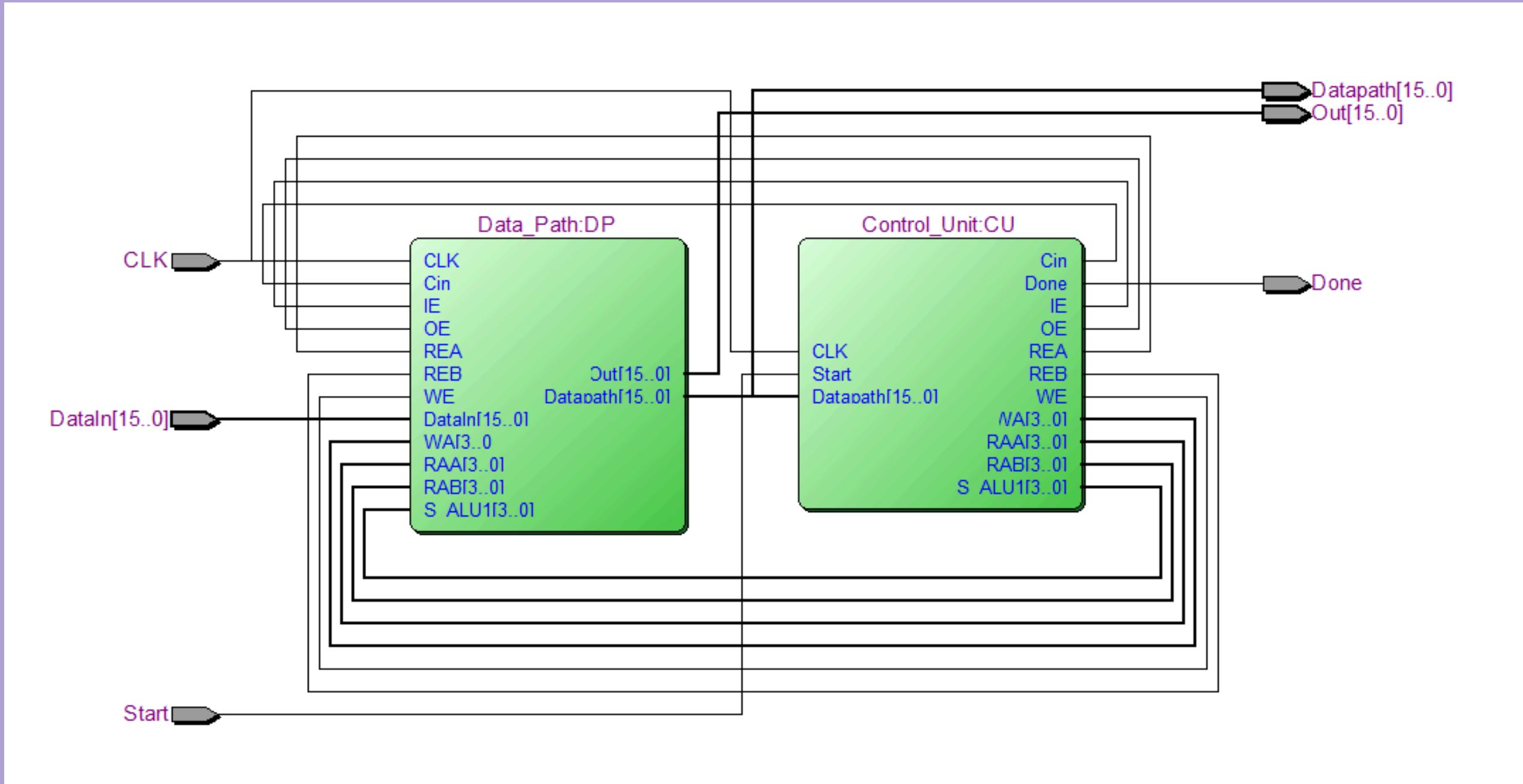
```
state4: begin n_state =(Datapath == 1) ? state5:state3;
    IE = 0;
    WE = 1;
    WA = 4'b0001;
    REA = 1;
    REB = 0;
    RAA = 4'b0001;
    RAB = 4'b0000;
    S_ALU1 = 4'b0100;
    Cin = 1'b1;
    OE = 0;
    Done = 0;
end

state5: begin n_state = state0;
    IE = 0;
    WE = 0;
    WA = 4'b0000;
    REA = 1;
    REB = 1;
    RAA = 4'b0011;
    RAB = 4'b0000;
    S_ALU1 = 4'b0101;
    Cin = 1'b0;
    OE = 1;
    Done = 1;
end
```

```
default: n_state = state0;
endcase
end

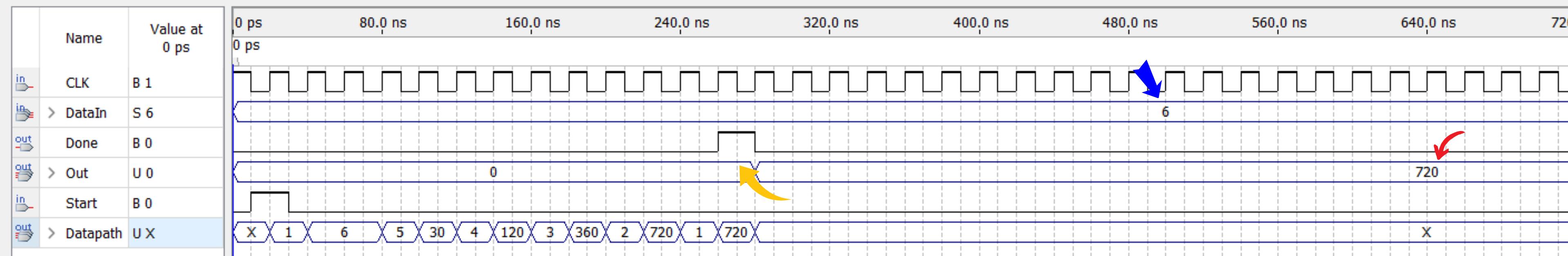
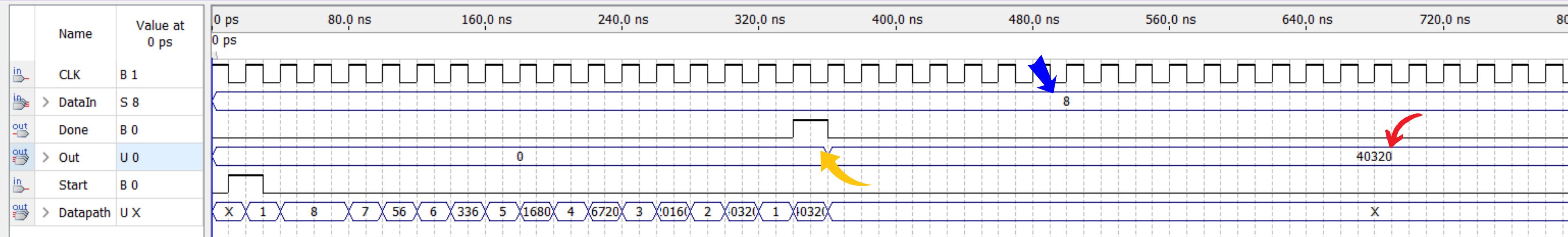
endmodule
```

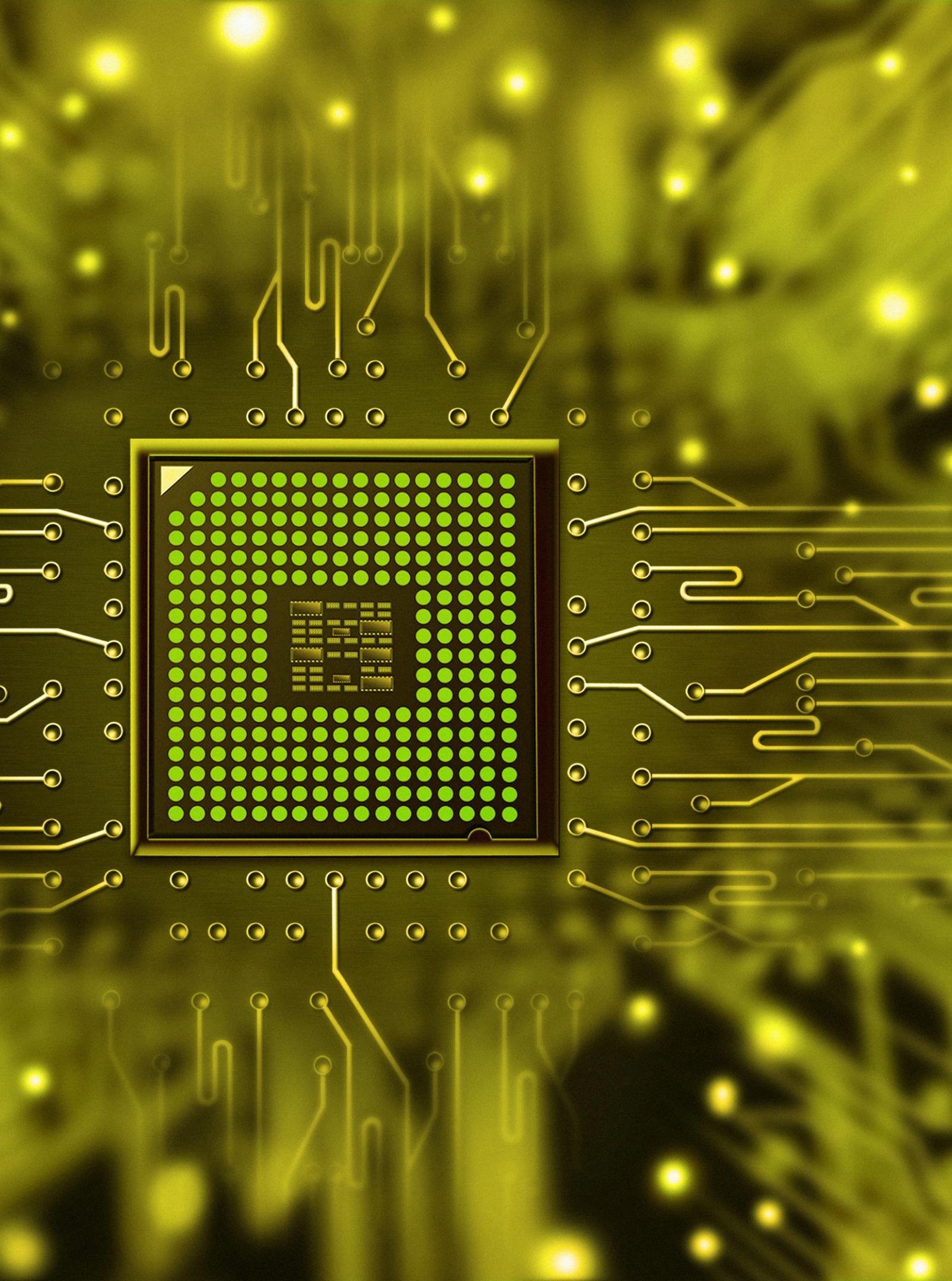
# c) Kết hợp Datapath & Control Unit



c)

# Mô phỏng kết quả





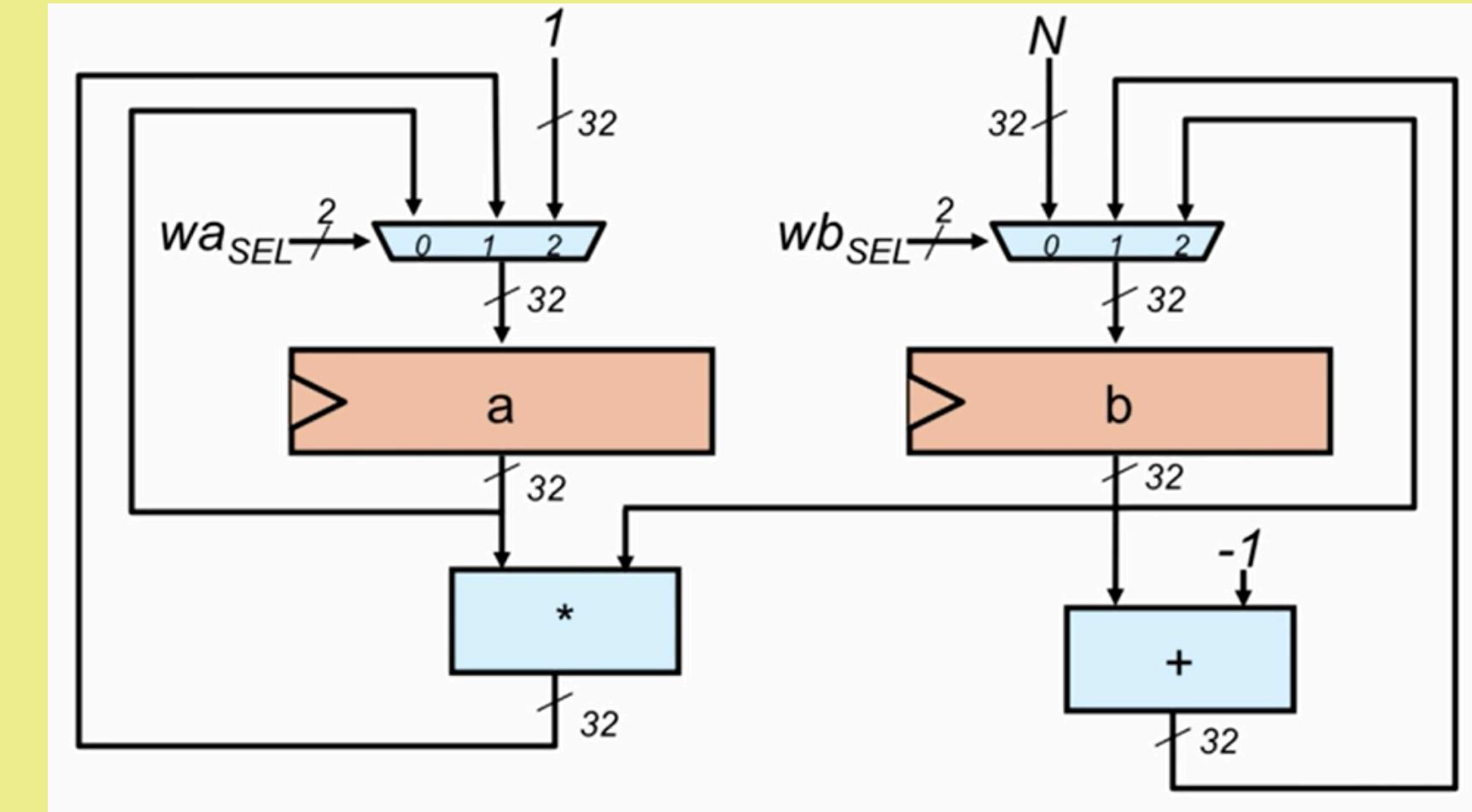
### III. Các phương pháp tối ưu hóa

- **SỬ DỤNG THIẾT KẾ SONG SONG**
- **CHIA SẺ KHỐI CHỨC NĂNG**

# 1. Thiết kế song song

## a) Ý tưởng

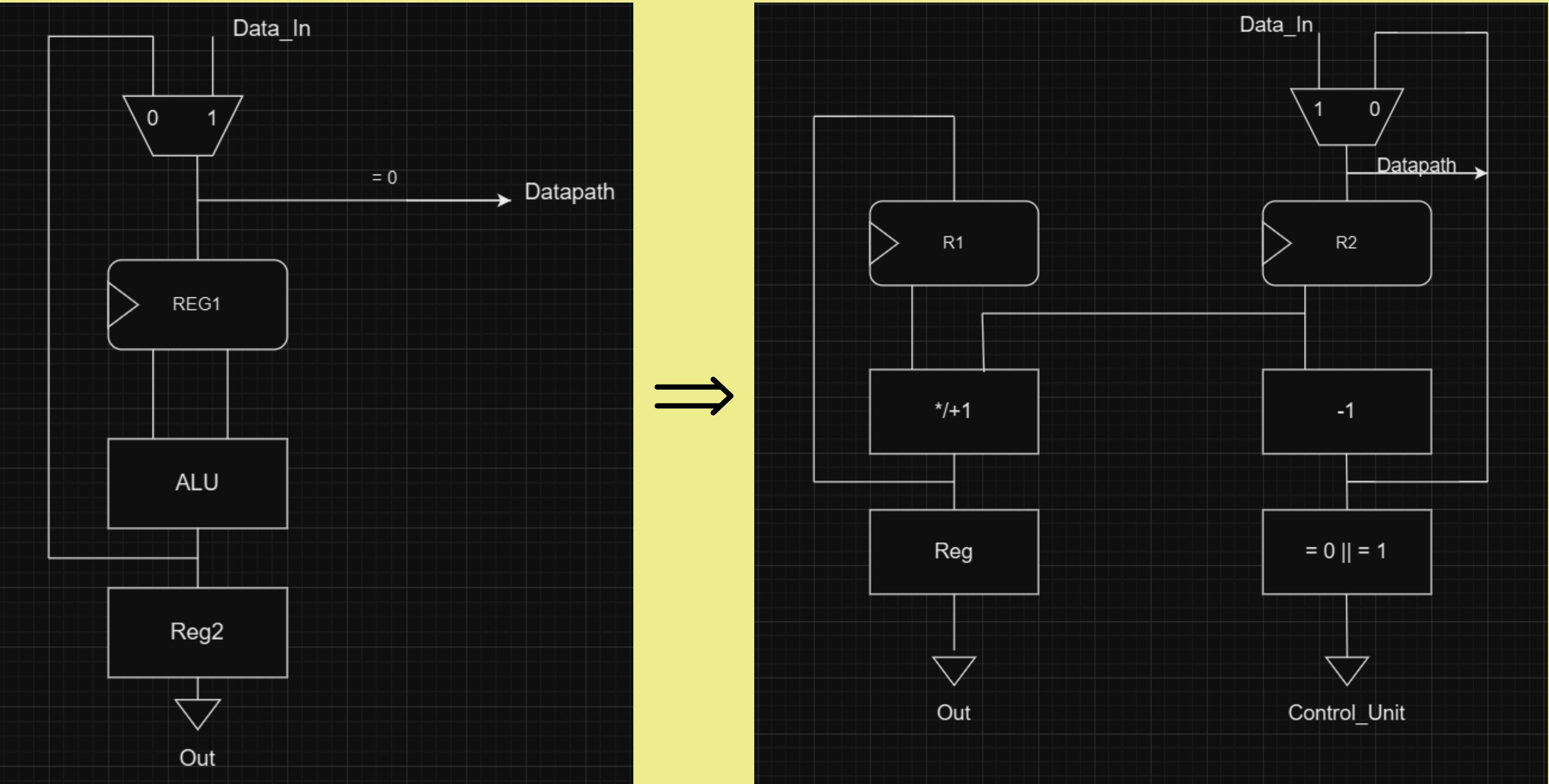
```
1. Out := 1
2. Data := Import
3. While ( Data != 0) repeat
    4. Out = Out * Data
    5. Data = Data - 1
end while
6. Outport = Out
```



Thực hiện cùng lúc bằng cách  
sử dụng 2 ALU.

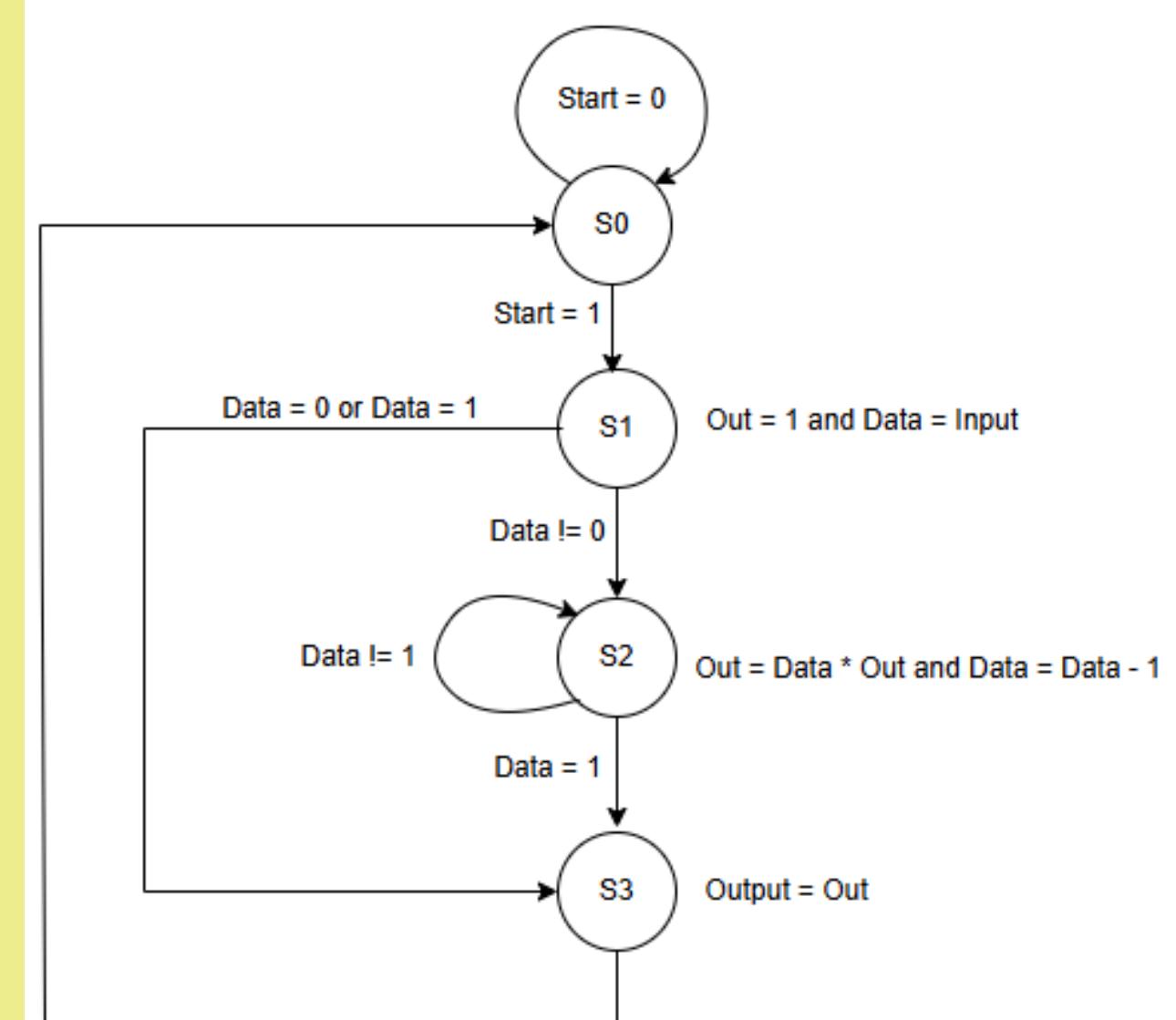
# 1. Thiết kế song song

## a) Ý tưởng



# 1. Thiết kế song song

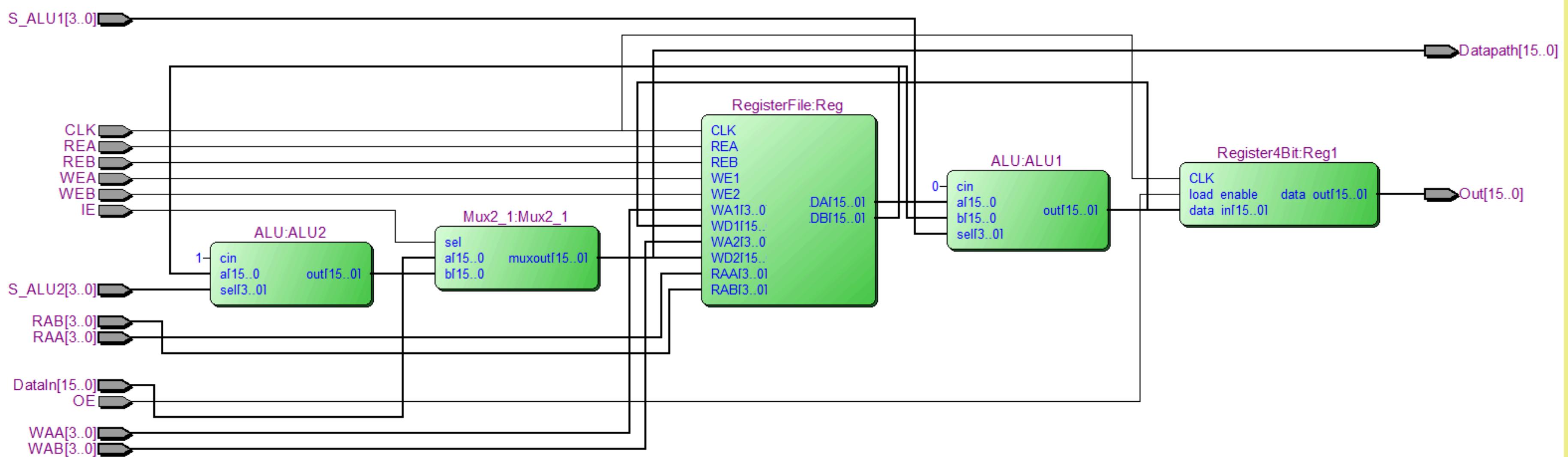
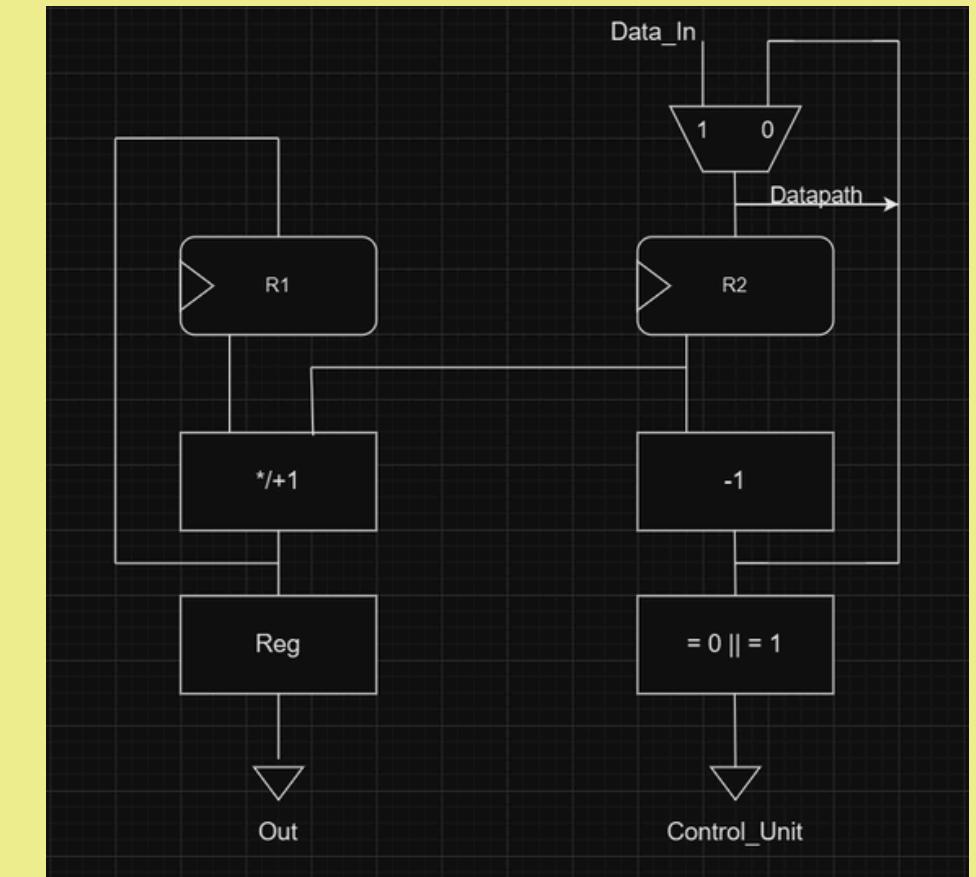
## b) Sơ đồ trạng thái



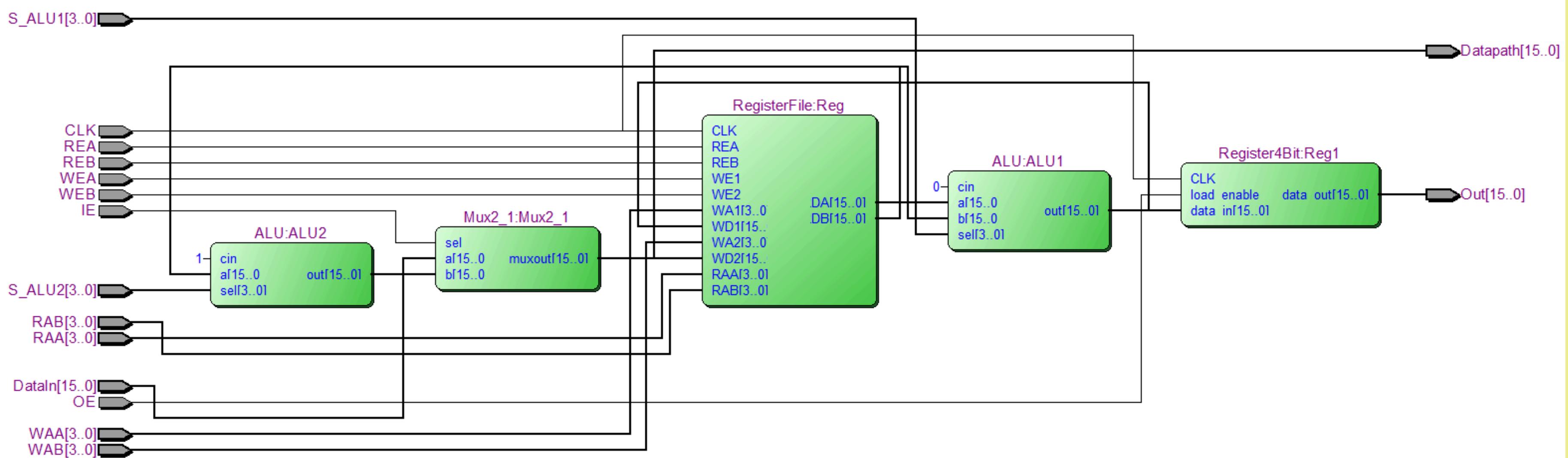
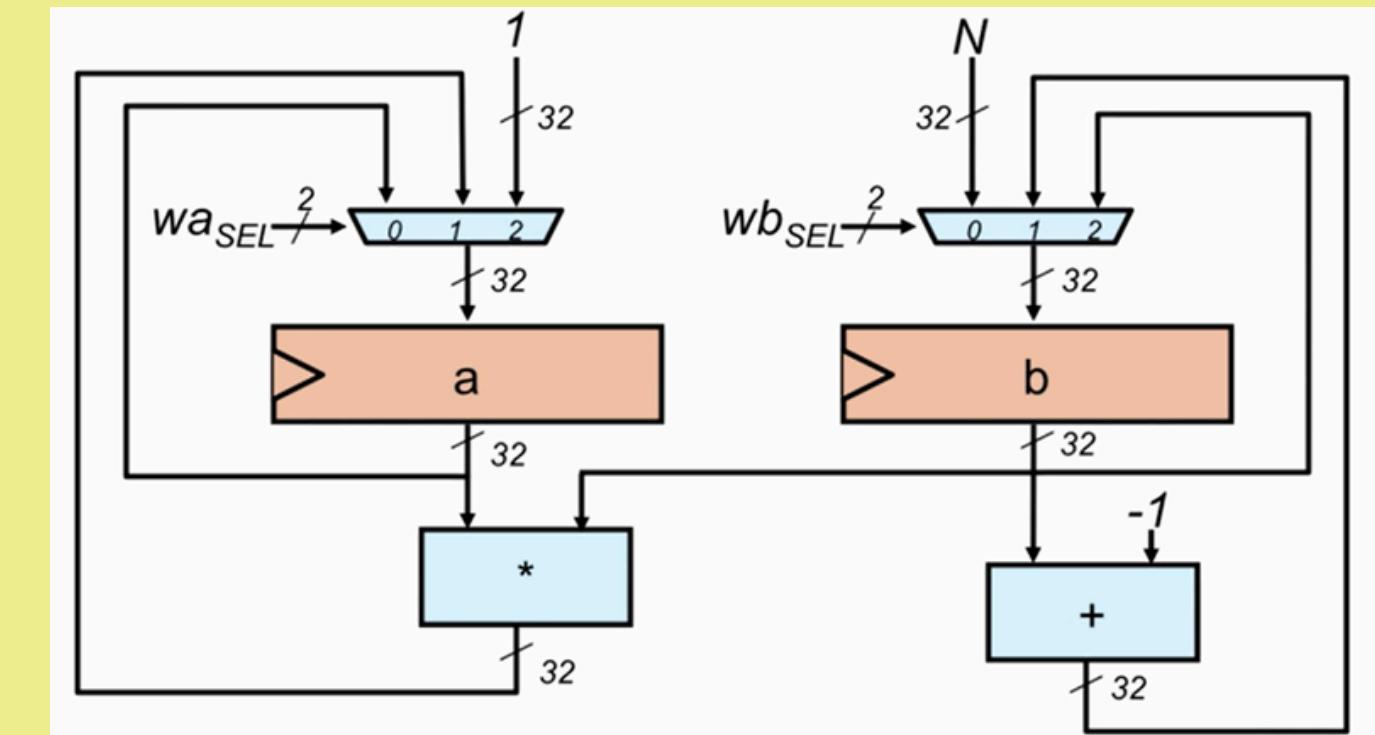
STATE	Q1Q0	IE	WAA2WAA1WAA0WEA	WAB2WAB1WAB0WEB	RAA2RAA1RAA0REA	RAB2RAB1RAB0REB	SA3SA2SA1SA0	SB3SB2SB1SB0	OE	DONE
S0	00	0	XXX0	XXX0	XXX0	XXX0	XXXX	XXXX	0	0
S1	01	1	0011	0101	0001	XXX0	0111	XXXX	0	0
S2	10	0	0011	0101	0011	0101	1010	0100	0	0
S3	11	0	0011	XXX0	0011	0001	0101	XXXX	1	1

# 1. Thiết kế song song

## b) Thiết kế khối song song:

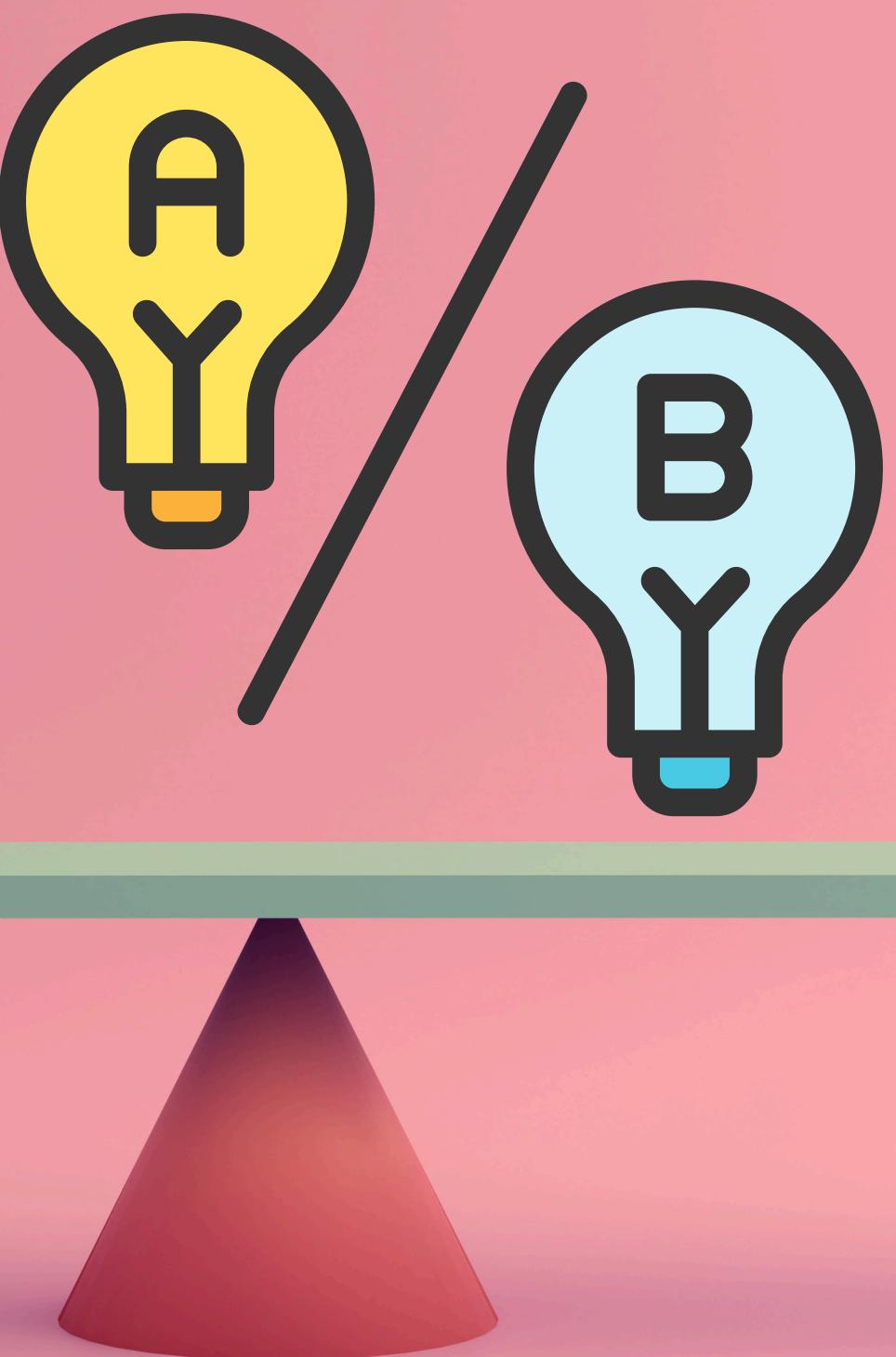


## 2. Chia sẻ khối chức năng



## IV. So sánh giữa các phương pháp

- THIẾT KẾ KHÔNG SONG SONG
- THIẾT KẾ SONG SONG

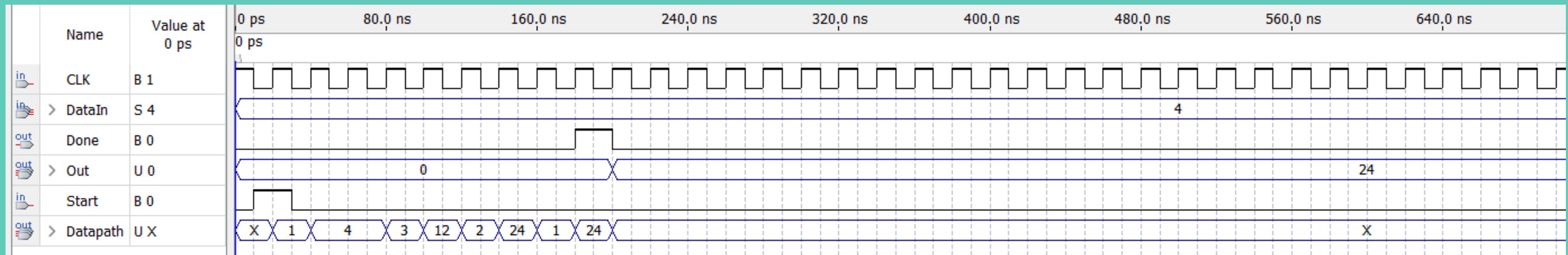


# V. So sánh giữa các phương pháp

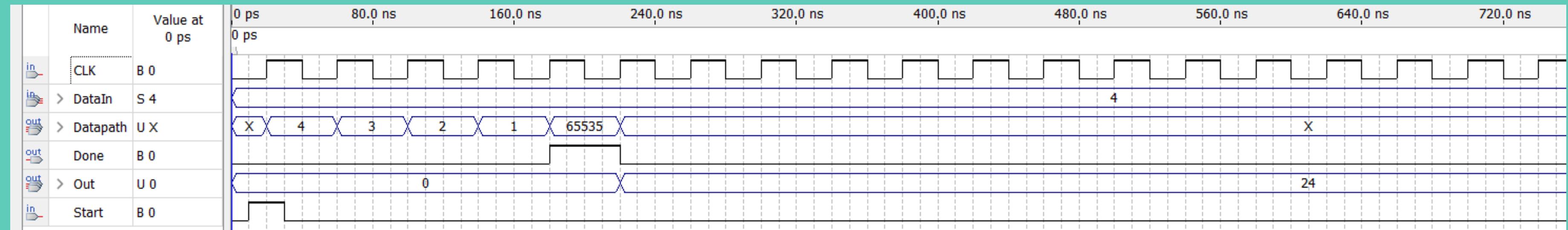
	Mạch không song song	Mạch song song
Chu kỳ Clock (T)	$3 + 2(n-1)$	$2 + (n-1)$
Số Input	1	1
Số Output	1	1

1. Out := 1
2. Data := Import
3. While ( Data != 0) repeat
  - 4. Out = Out \* Data
  - 5. Data = Data - 1
- end while
6. Outport = Out

# V. So sánh giữa các phương pháp

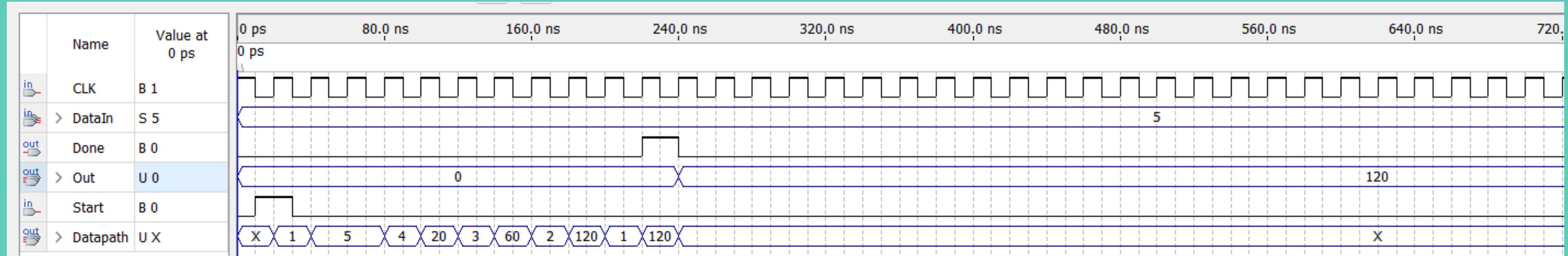


Với  $N = 4$ , cần 9 chu kỳ clock, gồm 3 clock chuẩn bị và 6 clock tính toán

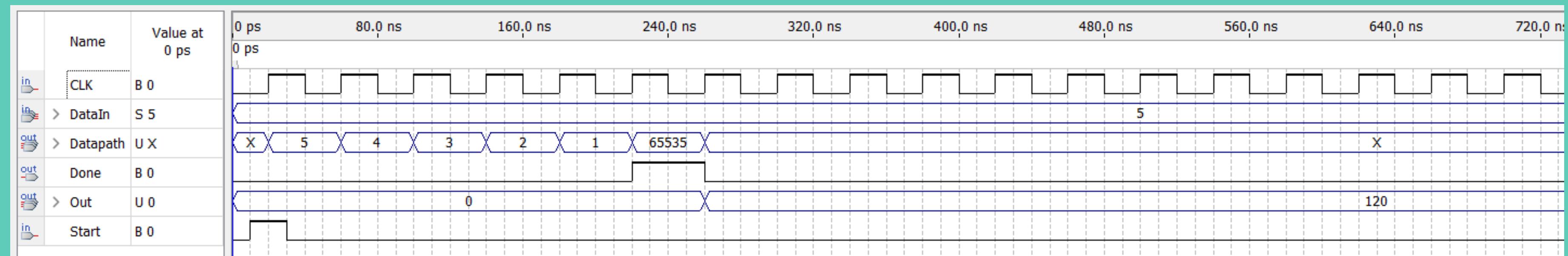


Với  $N = 4$ , cần 5 chu kỳ clock, gồm 2 clock chuẩn bị và 3 clock tính toán

# V. So sánh giữa các phương pháp



Với  $N = 5$ , cần 11 chu kỳ clock, gồm 3 clock chuẩn bị và 8 clock tính toán



Với  $N = 5$ , cần 6 chu kỳ clock, gồm 2 clock chuẩn bị và 4 clock tính toán

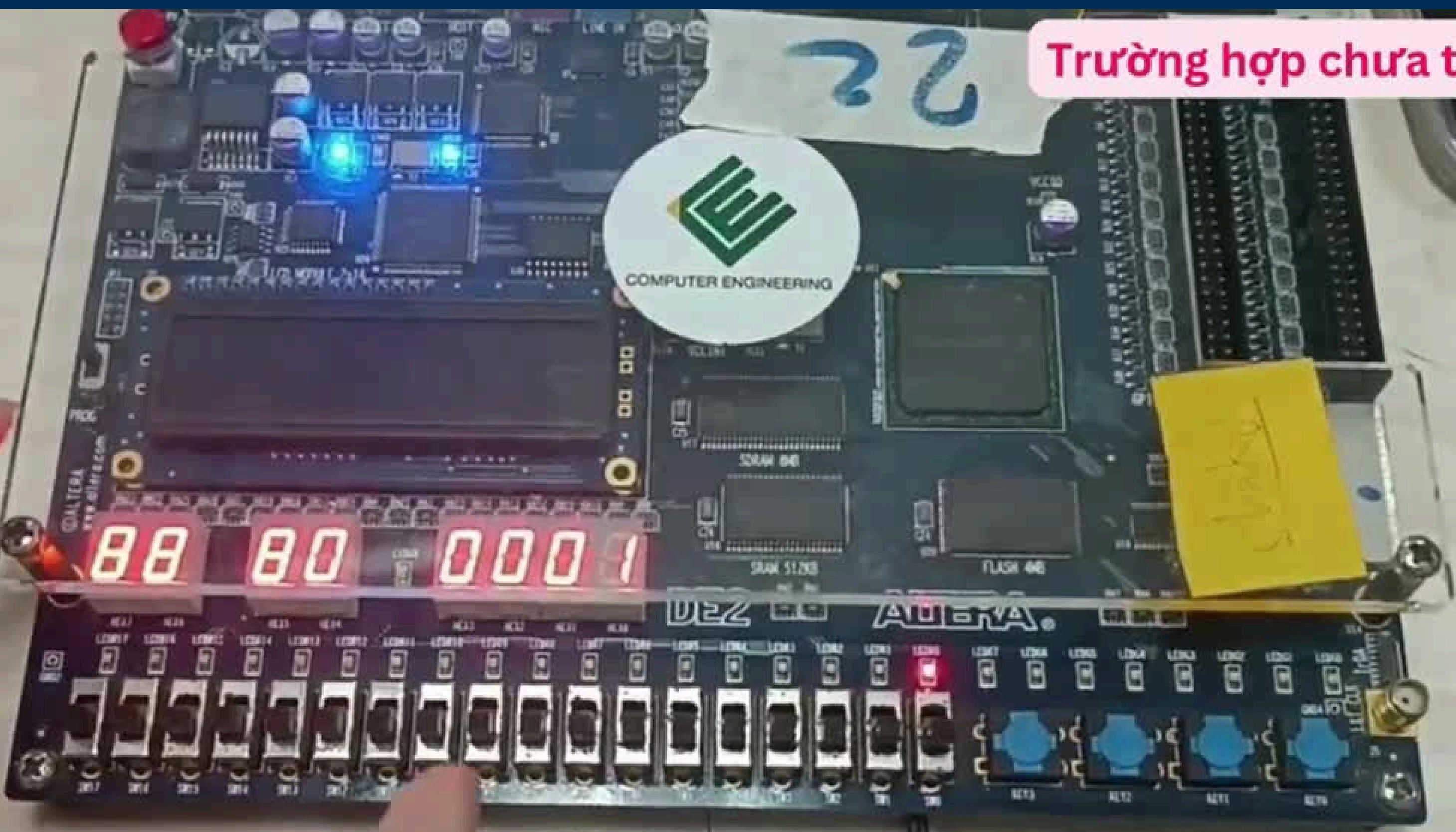
# V. Video nạp kit mô phỏng và so sánh



Vui lòng hãy click vào ĐÂY để xem video.



Trường hợp chưa tối ưu



Cảm ơn cô cùng các bạn đã lắng nghe bài thuyết trình của nhóm chúng em. Hi vọng rằng các bạn có thể nắm được ý tưởng để giải quyết bài toán này.

Thank  
you! :)