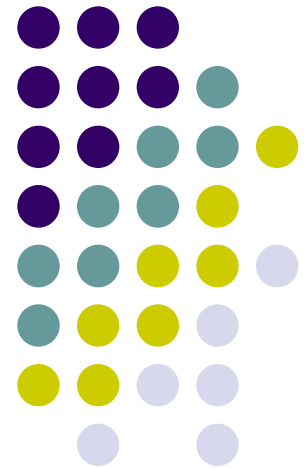


SEMICON Solutions

Thiết kế mạch tuần tự

Trình bày: Đặng Tường Dương



Mạch tổ hợp



A		0	1	AB			
		0	1				
B	0		1	CD			
	1	1	1				
				00	01	11	10
				00			
				01			
				11			
				10			

> y1?

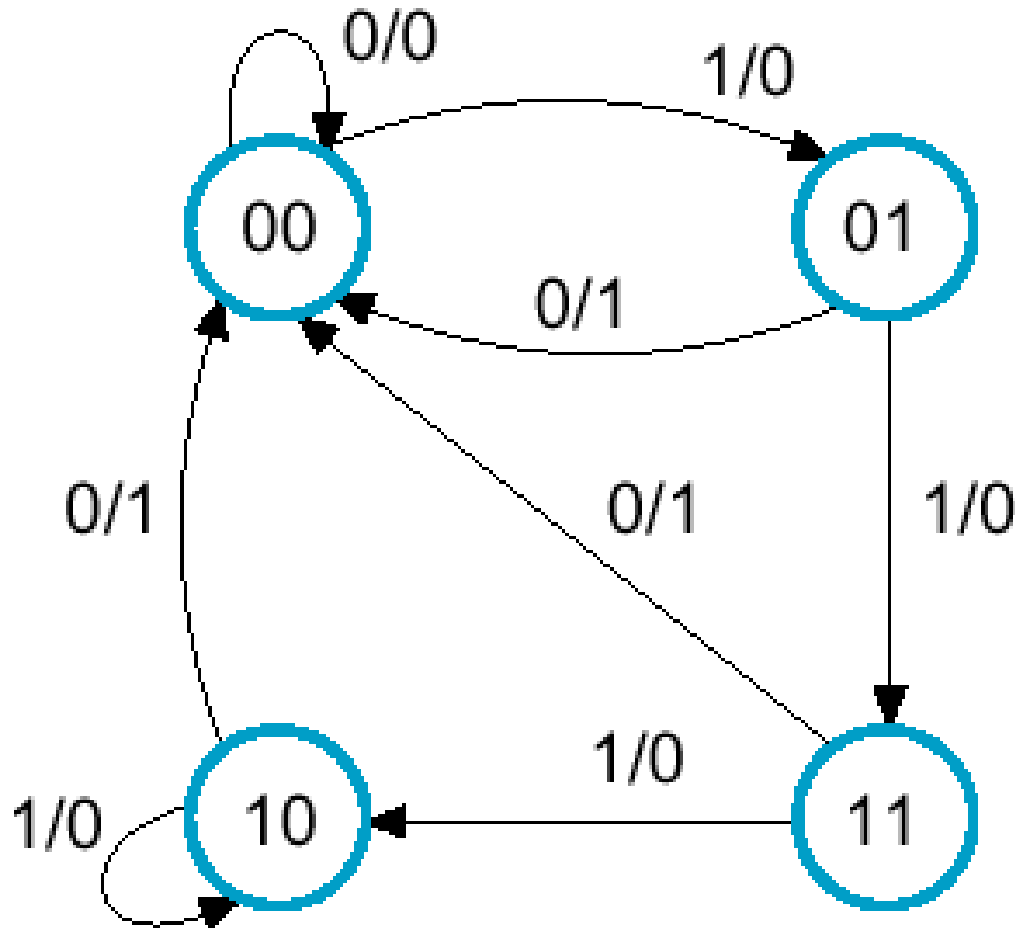
CD		ABC							
		000	001	011	010	110	111	101	100
00		1				1	1	1	
01						1	1	1	
11									
10		1							

EF		ABCD															
		0000	0001	0011	0010	0110	0111	0101	0100	1100	1101	1111	1110	1010	1011	1001	1000
00																	
01																	
11																	
10																	

Học cách tạo ra bảng K?

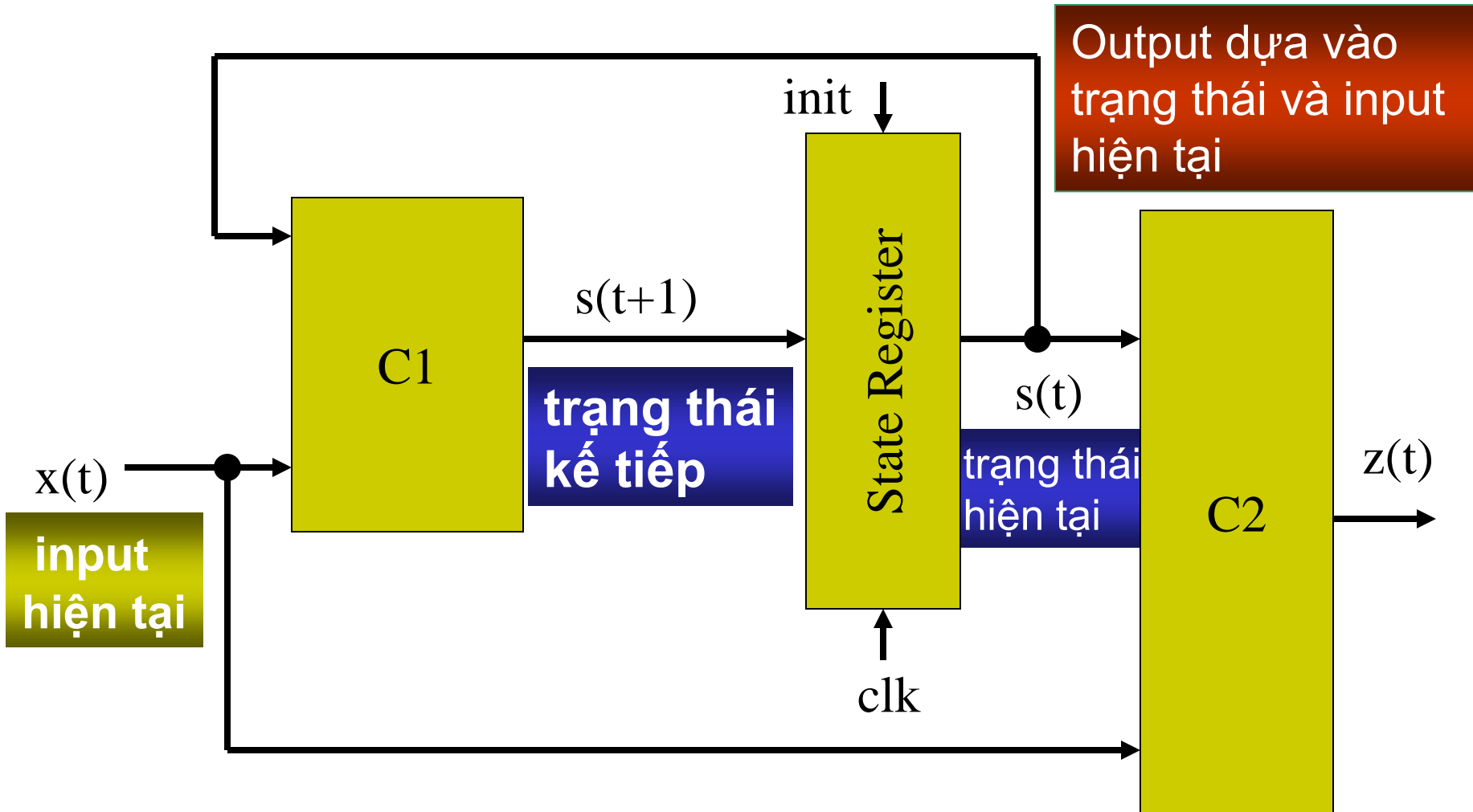
SOP	Sum of Product	Tổng các tích	OR của tất cả công AND	GOM 1	$y_1 = ABC + CDE$
POS	Product of Sum	Tích các tổng	AND của tất cả công OR	DOM 0	$y_1 = (A + B + C)(C + D + E)$

Thế nào là máy biến trạng thái



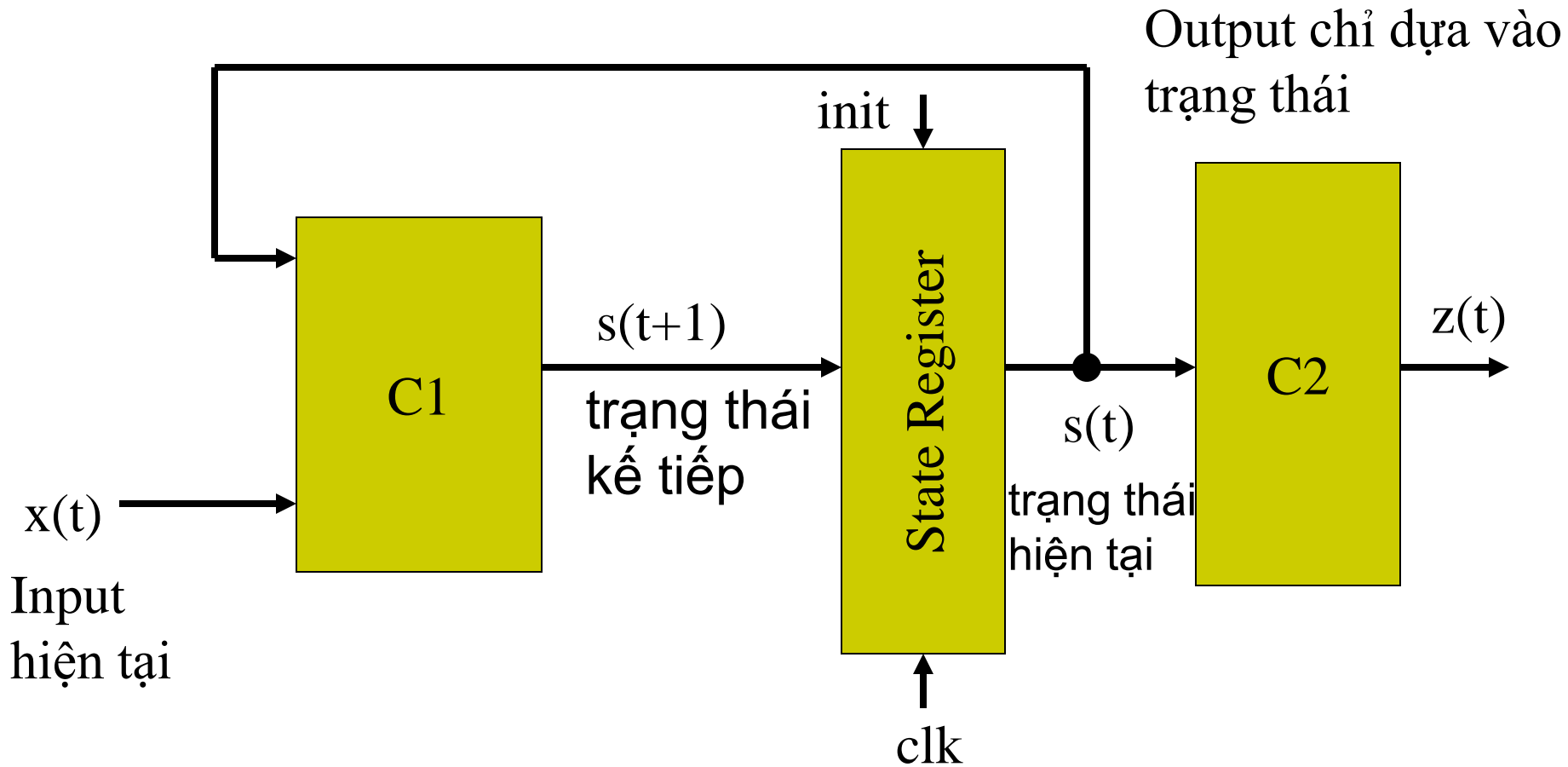


Mô hình Mealy

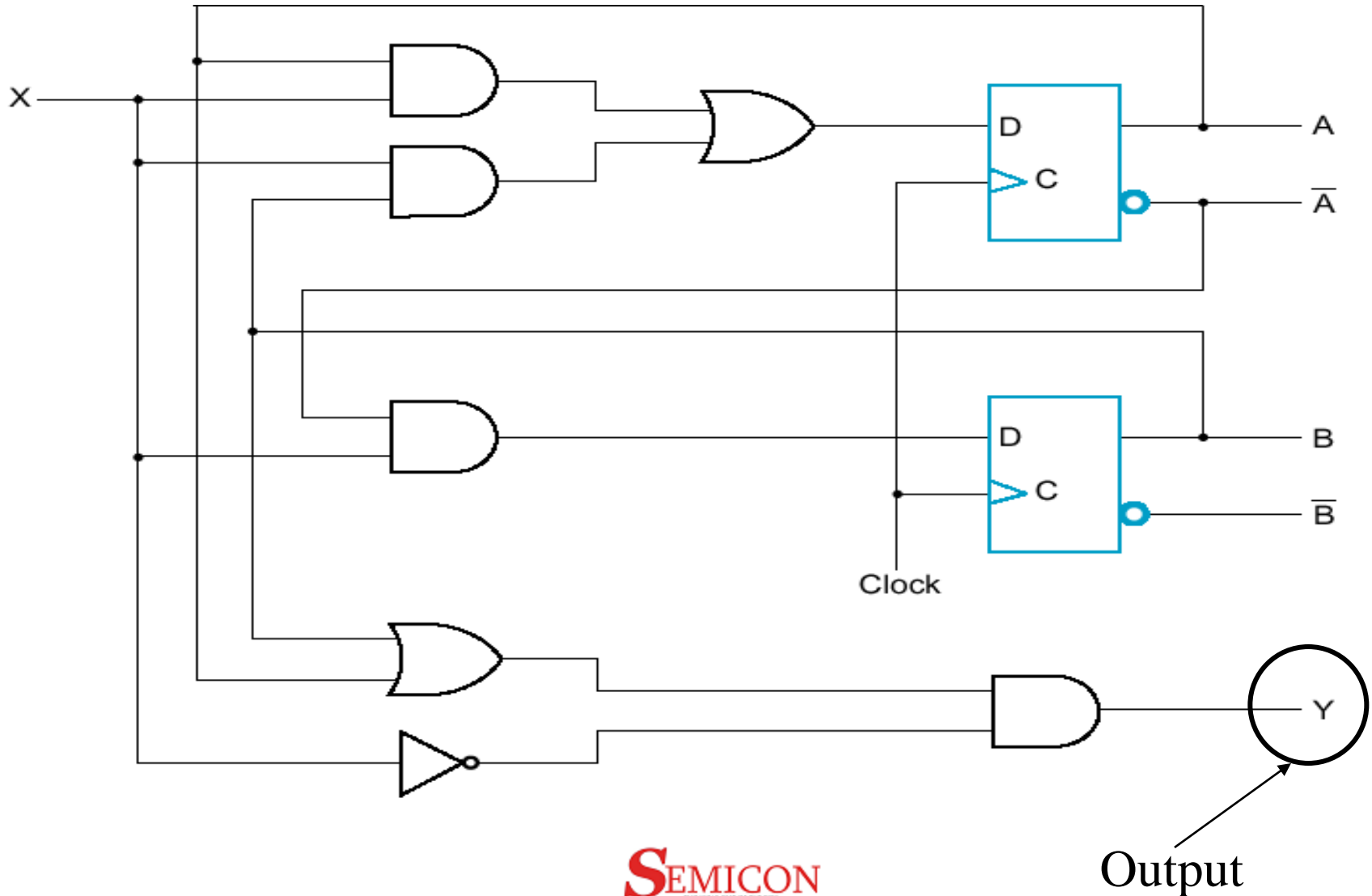
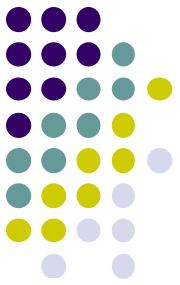




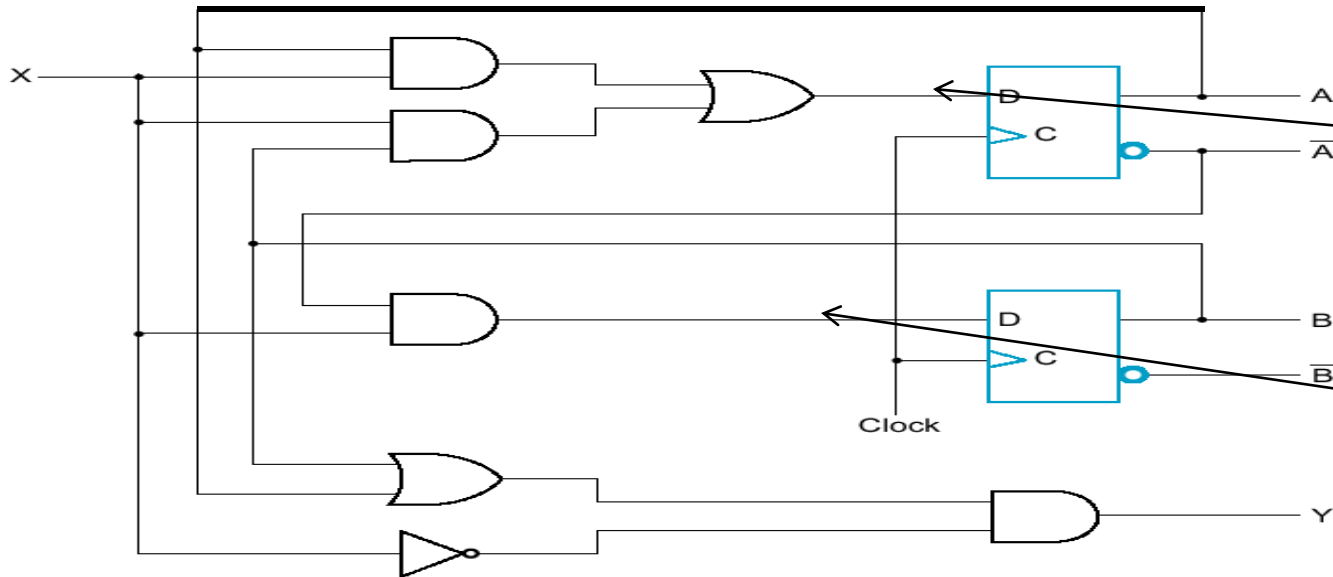
Mô hình Moore



Ví dụ về mạch tuần tự



Ví dụ về mạch tuần tự



$$A(t+1) = D_A = AX + BX$$

$$B(t+1) = D_B = \bar{A}X$$

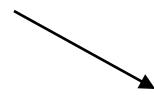
$$Y = (A + B)\bar{X}$$

Present State		Input X	Next State		Output Y
A	B		A	B	
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	0	0	1
0	1	1	1	1	0
1	0	0	0	0	1
1	0	1	1	0	0
1	1	0	0	0	1
1	1	1	1	0	0



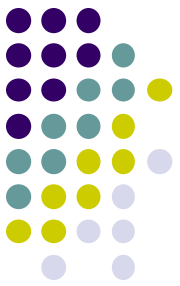
Present State		Input X	Next State		Output Y
A	B		A	B	
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	0	0	1
0	1	1	1	1	0
1	0	0	0	0	1
1	0	1	1	0	0
1	1	0	0	0	1
1	1	1	1	0	0

Bảng trạng thái
2 chiều

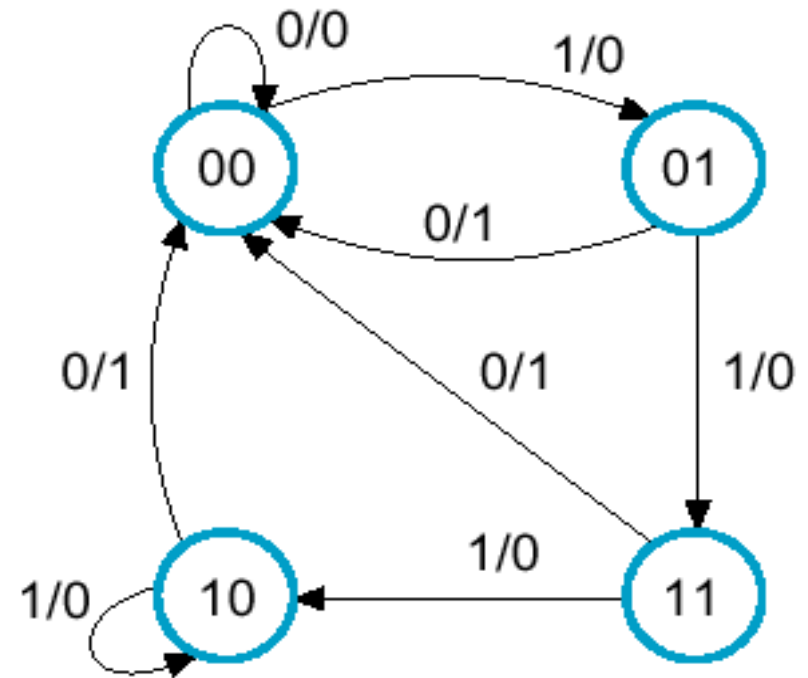


Present state		Next state				Output	
		X = 0		X = 1		X = 0	X = 1
A	B	A	B	A	B	Y	Y
0	0	0	0	0	1	0	0
0	1	0	0	1	1	1	0
1	0	0	0	1	0	1	0
1	1	0	0	1	0	1	0

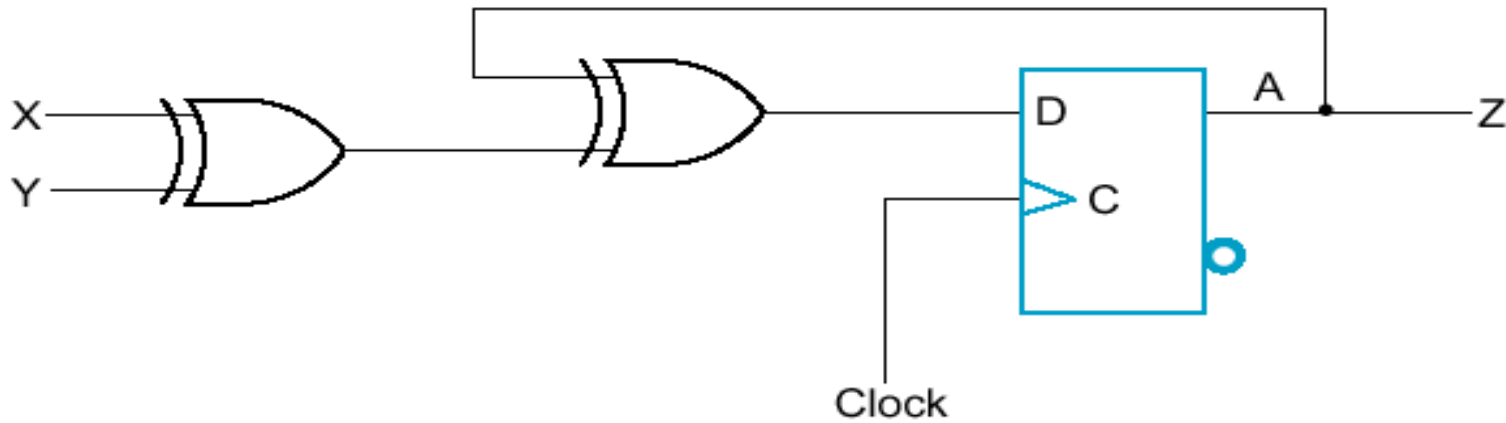
Sơ đồ trạng thái



Present state		Next state				Output	
		X = 0		X = 1		X = 0	X = 1
A	B	A	B	A	B	Y	Y
0	0	0	0	0	1	0	0
0	1	0	0	1	1	1	0
1	0	0	0	1	0	1	0
1	1	0	0	1	0	1	0



Logic Diagram and State Table for $D_A = A \oplus X \oplus Y$



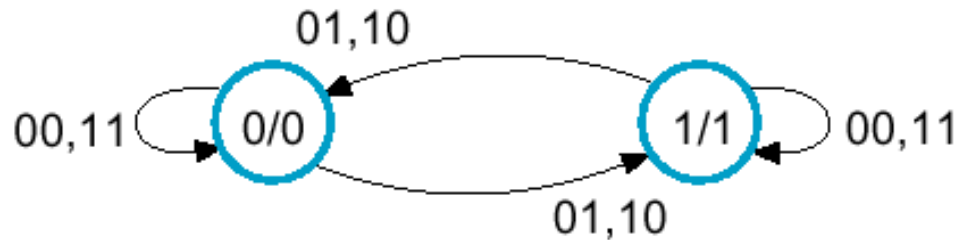
(a)

Present state	Inputs		Next state	Output
A	X	Y	A	Z
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	0
1	0	0	1	1
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

(b) State table

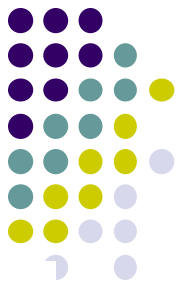


Sử dụng sơ đồ trạng thái



Present state	Inputs		Next state	Output
A	X	Y	A	Z
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	0
1	0	0	1	1
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Thiết kế máy trạng thái dùng J-K Flip Flop



Present state		Input	Next state		Flip-flop inputs			
A	B	X	A	B	J _A	K _A	J _B	K _B
0	0	0	0	1	0	0	1	0
0	0	1	0	0	0	0	0	1
0	1	0	1	1	1	1	1	0
0	1	1	1	0	1	0	0	1
1	0	0	1	1	0	0	1	1
1	0	1	1	0	0	0	0	0
1	1	0	0	0	1	1	1	1
1	1	1	1	1	1	0	0	0

Using J-K Flip Flops

$$J_A = B$$

$$K_A = B\bar{X}$$

$$J_B = X$$

$$K_B = A\bar{X} + \bar{A}X$$

Thiết kế mạch tuần tự

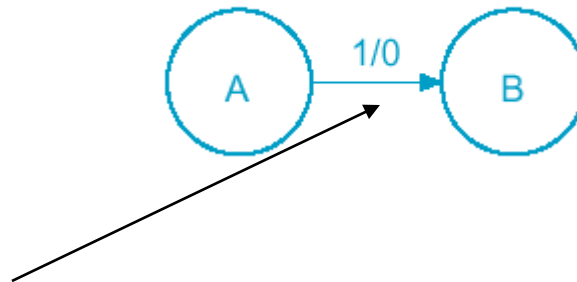


- Thực hiện sơ đồ trạng thái của bảng trạng thái từ yêu cầu của mạch trạng thái.
- Nếu chỉ có 1 biểu đồ trạng thái sẵn sàng, thì thực hiện bảng trạng thái.
- Dùng mã nhị phân cho các trạng thái.
- Phương trình input Flip-Flop bắt nguồn trong mục trạng thái kế tiếp trong bảng mã hóa trạng thái.
- Phương trình output Flip-Flop bắt nguồn từ mục output trong bảng mã hóa trạng thái.
- Đơn giản hóa phương input và output .
- Vẽ sơ đồ logic với DFF và các cổng , cụ thể hóa các input và output trong DFF.

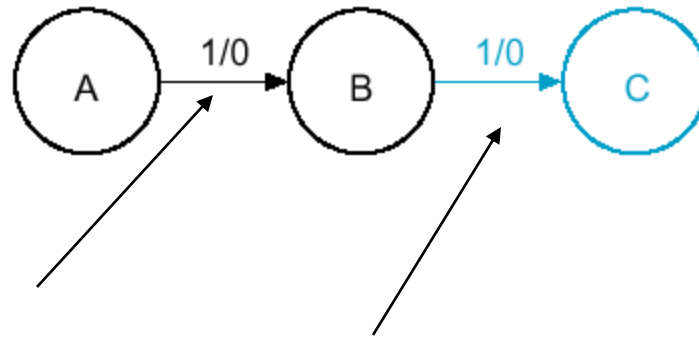
VD: sắp xếp tuần tự



Chúng ta cần để ý những bit ngõ vào lần lượt “1101”.

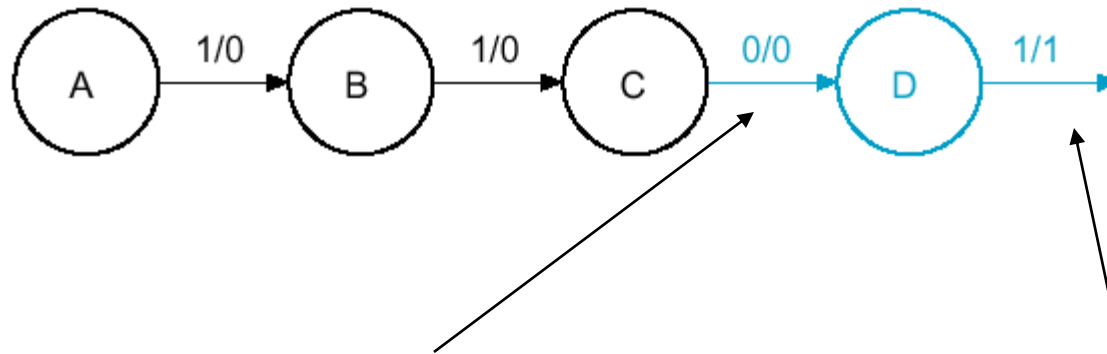


Nếu đầu vào là A = '1' trạng thái A chuyển sang trạng thái B và output của A là '0' (không phát hiện được bit “1101”)



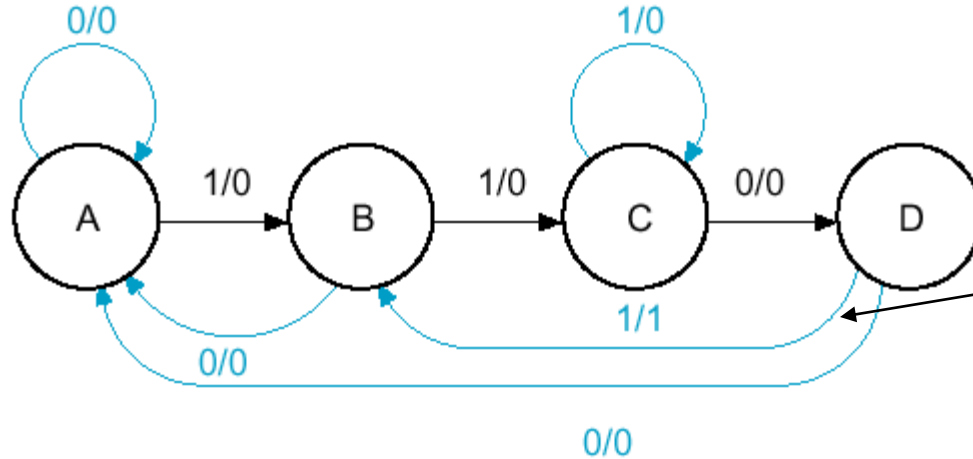
bit đầu tiên
Là A là '1'

Nếu chúng ta đang ở trạng thái B (điều này có nghĩa là rằng chúng ta đã đọc $a = '1'$ ngay lập tức trước và input kế tiếp là $a = '1'$ Sau đó chúng ta tìm cách làm cho đọc thành công bit "1101" sau đó chuyển sang trạng thái C



Bit tiếp theo, chúng ta đọc chuỗi “1101” tiếp theo là chuỗi “0” nếu chúng ta đọc là “0” thì chuyển sang trạng thái D --nếu báo rằng vẫn bằng “0” , chúng ta chưa đọc được chuỗi

Sau trạng thái D, chúng ta thành công nếu A = '1' được đọc và tiến hành. Đầu ra sẽ là mức cao hoặc '1'.



Chúng ta đặt “output mức cao sau đó chuyển sang trạng thái B

Chúng ta không cần tiến hành sang trạng thái E, cho dù, nếu chúng ta nhận ra “1101”, chúng ta không những phát hiện 1 chuỗi bit mà còn biết cách phát hiện chuỗi “1101” khác. dạng như “1101101”.

2 chuỗi

Present State		Input	Next State		Output
A	B	X	A	B	Y
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	0	0	1
0	1	1	1	1	0
1	0	0	0	0	1
1	0	1	1	0	0
1	1	0	0	0	1
1	1	1	1	0	0

A	B	0	1	CD	AB	00	01	11	10
0	0	0	1	00	00				
0	1	1	1	01	01				
1	0	0	0	10	10				
1	1	1	1	11	11				

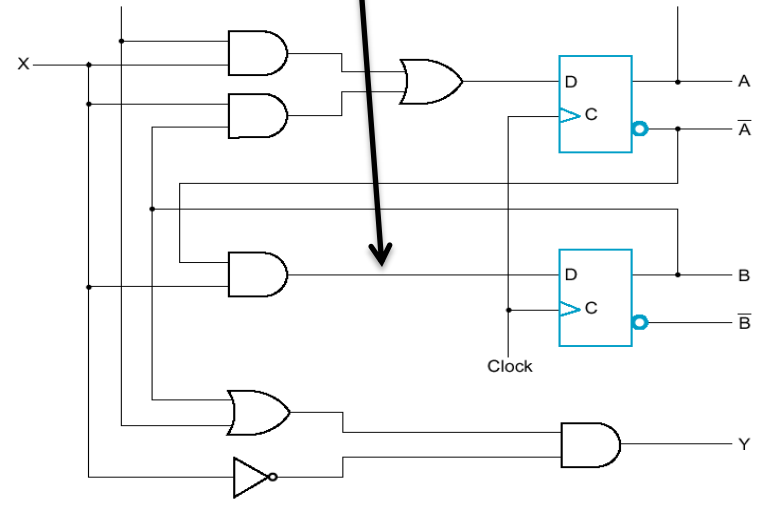
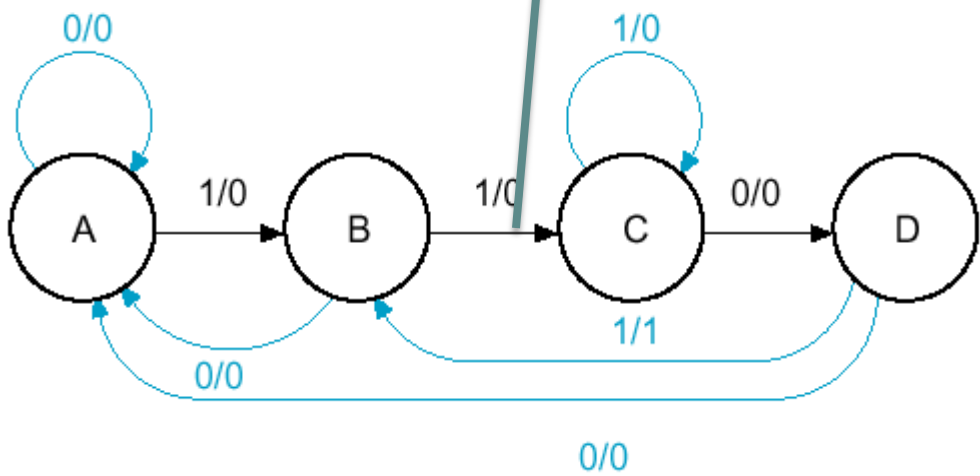
ABC	000	001	011	010	110	111	101	100
00	1					1	1	1
01							1	1
11							1	1
10	1							

ABCD	0000	0001	0011	0010	0110	0111	0101	0100	1100	1101	1111	1110	1010	1011	1001	1000
00																
01																
11																
10																

Hoc cach tao ra bang K?

SOP	Sum of Product	Tong cac tich	OR của tat ca cong AND	GOM1	$y1 = ABC + CDE$
POS	Product of Sum	Tich cac tong	AND của tat ca cong OR	DOM0	$y1 = (A + B + C)(C + D + E)$

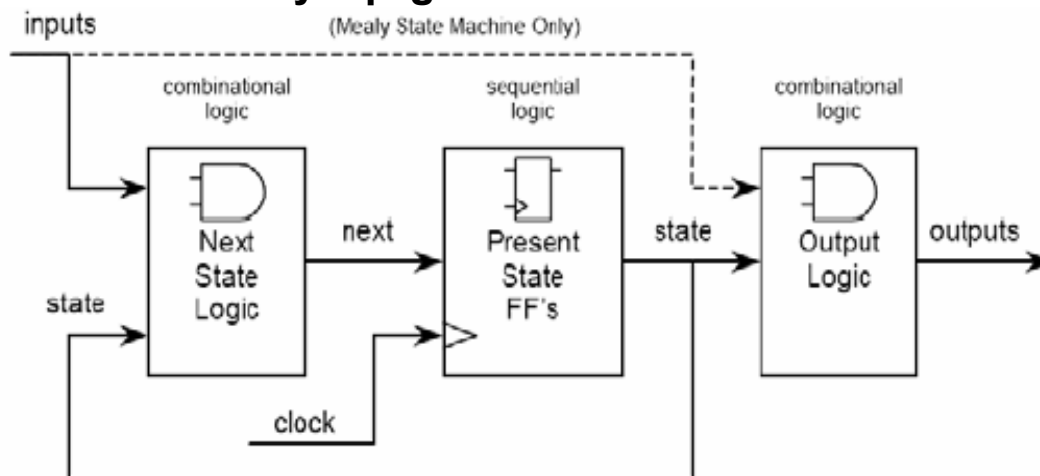
Present state		Next state				Output	
		X = 0		X = 1		X = 0	X = 1
A	B	A	B	A	B	Y	Y
0	0	0	0	0	1	0	0
0	1	0	0	1	1	1	0
1	0	0	0	1	0	1	0
1	1	0	0	1	0	1	0



Coding máy trạng thái



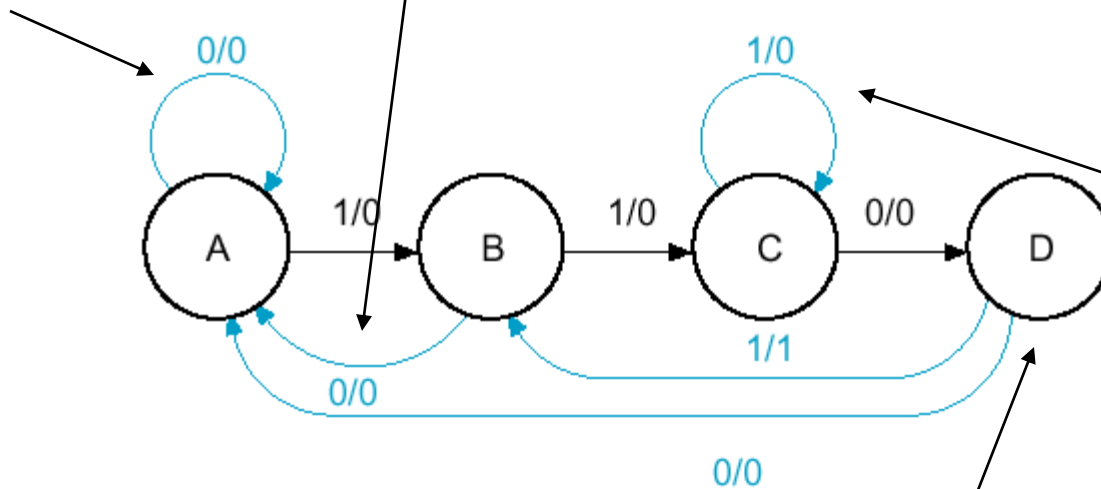
- Hướng dẫn
 - Tách rời diễn tả máy trạng thái thành 2 quy trình
 - Mạch tổ hợp
 - Mạch tuần tự
 - Dùng diễn tả `define để định nghĩa vector trạng thái.
 - Giữ logic FSM và logic non-FSM trong những module tách rời.
 - Gán giá trị mặc định cho máy trạng thái.



```
.....
reg [1:0] state, next_state; // FSM state
//-----
parameter IDLE = 2'h0;
parameter FOUND = 2'h1;
parameter MODULUS = 2'h2;
parameter DONE = 2'h3;
//-----
always @ (state or input)
  case (state)
    IDLE: begin
      ...
      next_state <= FOUND;
    end end // IDLE
    FOUND: begin
      ...
      next_state <= DONE;
    end // FOUND
  endcase
always @ (posedge clk or negedge rst_b)
  if (rst_b == 1'b0) state <= IDLE;
  else state <= next_state;
endmodule
```

bắt đầu '0'

bit thứ 2 là 0



bit thứ 3 là '1'
nghĩa là
Chúng ta đã
đọc chuỗi
"111"
. Điều này chờ
đợi
chúng ta đặt 1
bit '0'

A '0' là bit cuối cùng (A '0' is the last bit ("1100") và trở lại ban đầu

Chúng ta còn phải đặt trạng thái "thất bại", khi không đọc được bit "1101"

Bài Tập



- Thiết kế sơ đồ mạch số dùng DFF
- Viết Verilog cho mạch vừa thiết kế



Câu Hỏi & Trả Lời