Nguyên lý thiết kế mạch dãy

Nguyễn Quốc Cường – 3I

Nội dung

- Giới thiệu
- Các phần tử hai trạng thái ổn định
- Flip-Flops
- Phân tích các máy trạng thái đồng bộ bởi xung nhịp
- Thiết kế các máy trạng thái đồng bộ bởi xung nhịp

Tài liệu tham khảo

- Digital Design: Principles & Practices – John F Wakerly – Printice Hall

Giới thiệu

- Mạch logic dãy:
  - output 2 tín hiệu input tại thời điểm t_n
  - output 2 cả vào tín hiệu input trong quá khứ
- Ví dụ: mạch điều khiển chọn kênh TV sử dụng nút bấm channel–up và channel-down:
  - nếu trước đó kênh đang chọn là 9, nếu bấm channel-up thì kênh lựa chọn là 10
  - nếu trước đó kênh đang chọn là 1, nếu bấm channel-up thì kênh lựa chọn là 2
  - ...
- Việc sử dụng bảng để mô tả các output phụ thuộc vào tổ hợp các inputs đối với các mạch dãy là KHÔNG THỂ

Trạng thái

- Trong mạch dãy sử dụng khái niệm trạng thái để mô tả:
  - Trạng thái của một mạch dãy là tập hợp các biến trạng thái mà giá trị của nó tại một thời điểm chứa đầy đủ các thông tin cần thiết trong quá khứ cho phép xác định các hoạt động của mạch trong tương lai
  - Trong mạch logic các biến trạng thái chỉ có hai giá trị 0 và 1.
  - Số trạng thái của mạch có n biến trạng thái bằng 2^n trạng thái

Các phần tử 2 trạng thái ổn định

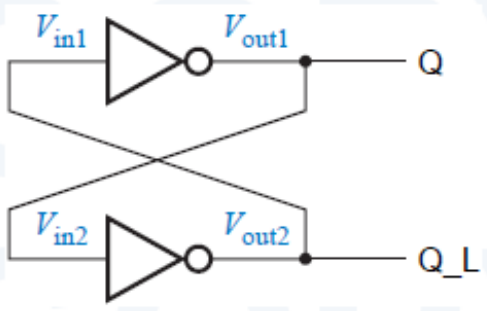


Figure 7-2
A pair of inverters forming a bistable element.

Mạch có hai trạng thái ổn định:
• Nếu Q = HIGH thì Q_L = LOW
• Nếu Q = LOW thì Q_L = HIGH

Phân tích tương tự

- Xem xét điện áp V_{out} và V_{in}

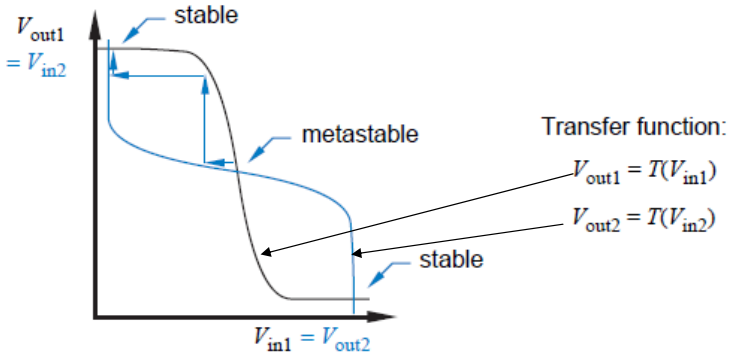


Figure 7-3
Transfer functions for inverters in a bistable feedback loop.

Giao của 2 đồ thị tại 3 điểm đó là các điểm cân bằng của mạch:
• Hai điểm ổn định (ứng với các trạng thái Q = 0 hoặc Q = 1)
• Một điểm metastable: tại đó V_{out1} và V_{out2} có giá trị điện áp nằm giữa mức 1 và 0

Metastable

- Thực tế thời gian mạch ở trạng thái metastable thường ngắn, lý do, chỉ cần một tác động đủ lớn của nhiễu sẽ kéo nó về một trong hai trạng thái stable

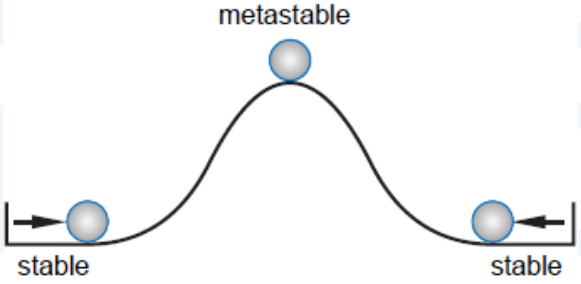
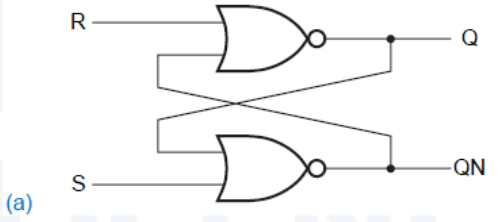


Figure 7-4
Ball and hill analogy for metastable behavior.

Latch và Flip-Flops

- Latch và Flip-flops là các phần tử cơ bản trong mạch logic dãy
- Flip-Flops: dùng để chỉ một thiết bị logic dãy có khả năng lấy mẫu tín hiệu đầu vào và thay đổi tín hiệu đầu ra tại thời điểm được xác định bởi tín hiệu xung nhịp
- Latch: dùng để chỉ thiết bị logic dãy có khả năng quan sát tín hiệu inputs một cách liên tục và có thể thay đổi đầu ra của nó tại bất kỳ thời điểm nào mà không phụ thuộc vào tín hiệu xung nhịp
- Tuy nhiên thường 2 khái niệm này có thể sử dụng như nhau

S-R Latch (Flip-flops)



(b)

S	R	Q	QN
0	0	last Q	last QN
0	1	0	1
1	0	1	0
1	1	0	0

Figure 7-5
S-R latch: (a) circuit design using NOR gates; (b) function table.

S-R flip-flop: (set-reset)
R = 1, S = 0 → Q = 0 (reset) S=1, R=0 → Q = 1(set) $\bar{Q}$
QN : thường là đầu bù của Q, trong các tài liệu còn được ký hiệu Q_L hay

Tuy nhiên trong trường hợp S=R=1 thì Q = QN = 0
Nếu R = 0, S = 0 thì mạch giống như một phần tử bistable

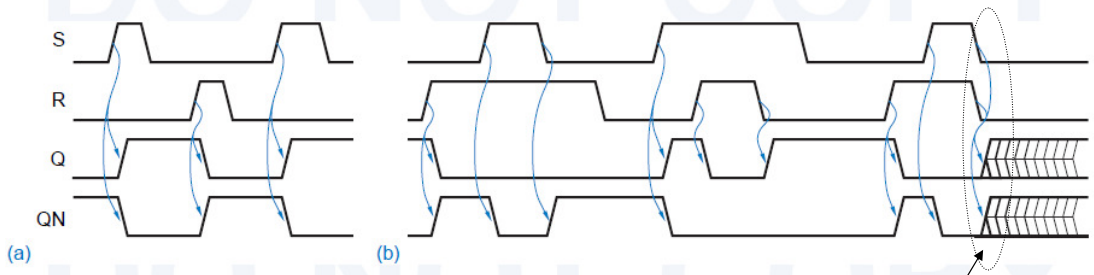
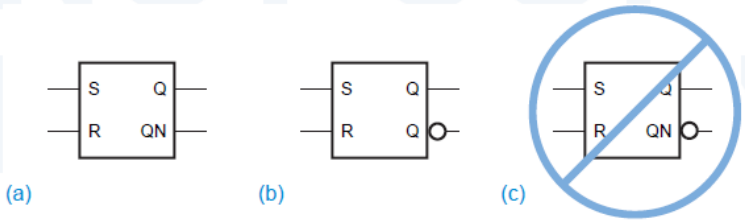


Figure 7-6 Typical operation of an S-R latch: (a) "normal" inputs; (b) S and R asserted simultaneously.

không đoán được trước giá trị của Q và QN khi cả R và S thay đổi giá trị tại cùng thời điểm

Ký hiệu

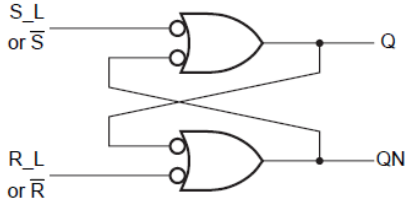
Figure 7-7
Symbols for an S-R latch:
(a) without bubble;
(b) preferred for bubble-to-bubble design;
(c) incorrect because of double negation.



$\bar{S} - \bar{R}\ latch$

Figure 7-9 $\overline{S}$-$\overline{R}$ latch: (a) circuit design using NAND gates; (b) function table; (c) logic symbol.

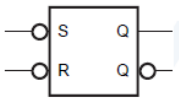
(a)



(b)

S_L	R_L	Q	QN
0	0	1	1
0	1	1	0
1	0	0	1
1	1	last Q	last QN

(c)

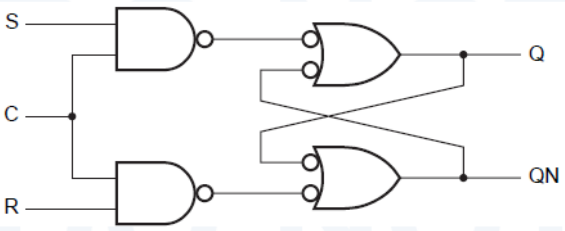


Trong công nghệ CMOS và TTL các cổng NAND thường được sử dụng hơn là cổng NOR

S – R latch với Enable

- S-R và $\bar{S} - \bar{R}\ latch$:output thay đổi phụ thuộc vào R và S input
- S-R latch với Enable: output thay đổi phụ thuộc vào R và S chỉ với điều kiện tín hiệu Enable tích cực

(a)



(b)

S	R	C	Q	QN
0	0	1	last Q	last QN
0	1	1	0	1
1	0	1	1	0
1	1	1	1	1
x	x	0	last Q	last QN

(c)

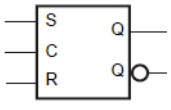
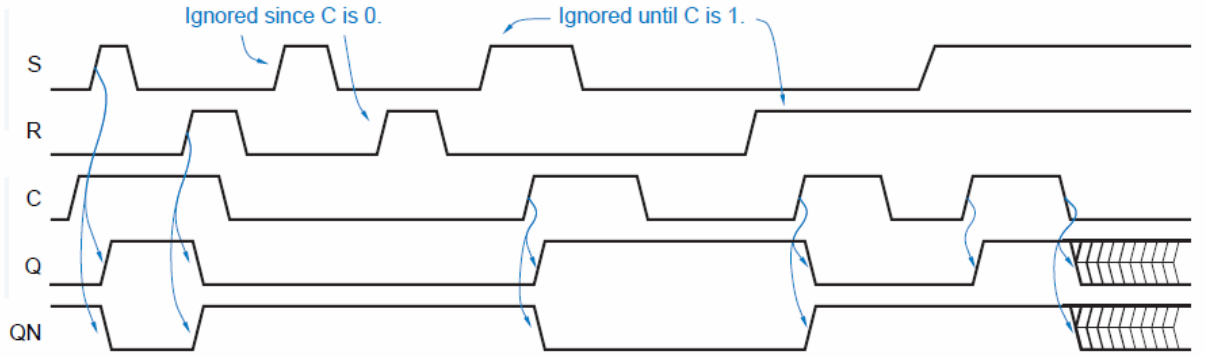
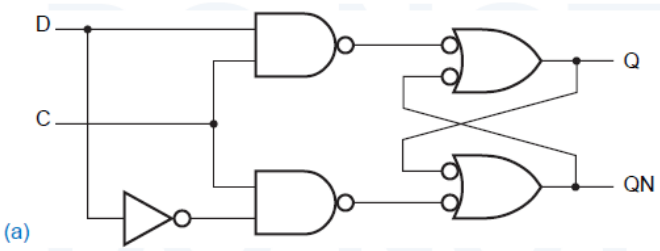


Figure 7-10 S-R latch with enable: (a) circuit using NAND gates; (b) function table; (c) logic symbol.

Figure 7-11 Typical operation of an S-R latch with enable.



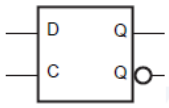
D latch (D flip-flops)



(a)

C	D	Q	QN
1	0	0	1
1	1	1	0
0	x	last Q	last QN

(b)



(c)

Figure 7-12 D latch: (a) circuit design using NAND gates; (b) function table; (c) logic symbol.

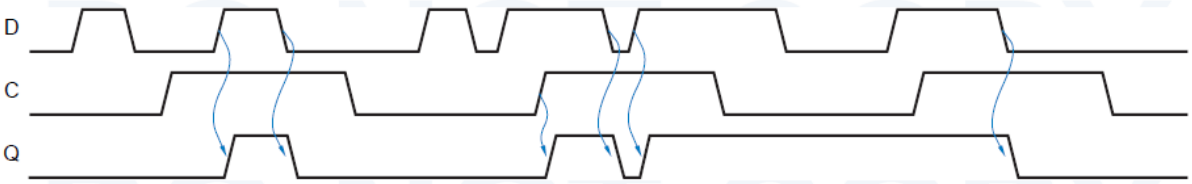
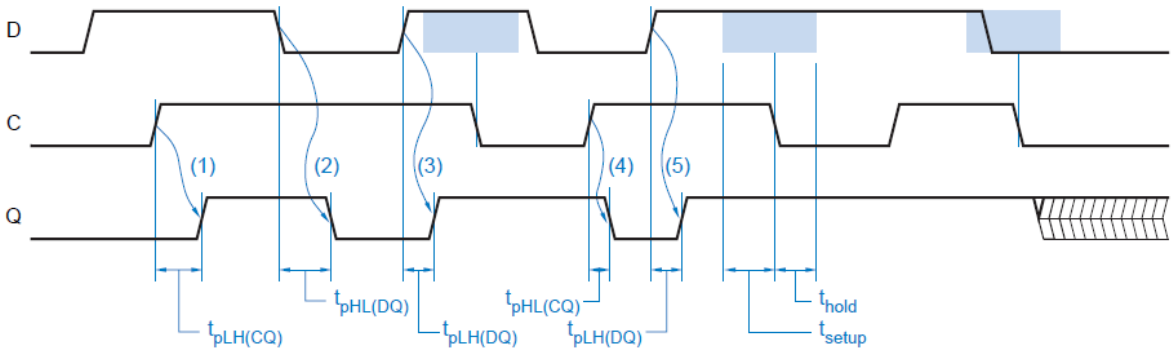


Figure 7-13 Functional behavior of a D latch for various inputs.

- D latch: giống S-R latch với R là đảo của S:
  - Tránh được trường hợp S=R=1 trong S-R latch
- Với C = 1 (tích cực):
  - D = 1 → Q = 1, QN = 0
  - D = 0 → Q = 0, QN = 1
- D latch vẫn gặp phải vấn đề về metastable khi D và C thay đổi đồng thời
- Tín hiệu C (Control) còn được ký hiệu như là E (Enable), Clk (Clock) hay G (Gate)

Figure 7-14 Timing parameters for a D latch.

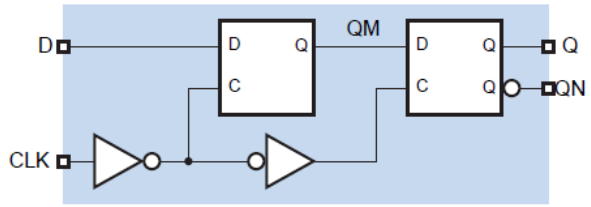


Nếu D thay đổi trong khoảng thời gian t_{setup} và t_{hold} thì D latch có thể rơi vào trạng thái metastable hoặc không xác định

D Flip-flop tác động theo sườn lên

Figure 7-15 Positive-edge-triggered D flip-flop: (a) circuit design using D latches; (b) function table; (c) logic symbol.

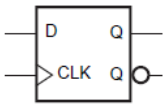
(a)



(b)

D	CLK	Q	QN
0	↑	0	1
1	↑	1	0
x	0	last Q	last QN
x	1	last Q	last QN

(c)



D flip-flop tác động theo sườn lên : sử dụng 2 D latch:

- D latch đầu tiên được gọi là master:
  - CLK = 0 → latch mở
  - CLK = 1 → latch đóng
- D latch thứ hai được gọi là slave:
  - mở trong suốt thời gian CLK = 1, tuy nhiên giá trị của nó chỉ thay đổi tại thời điểm bắt đầu khi CLK thay đổi từ 0→1 do master đã đóng và không thay đổi trong khoảng thời gian CLK = 1

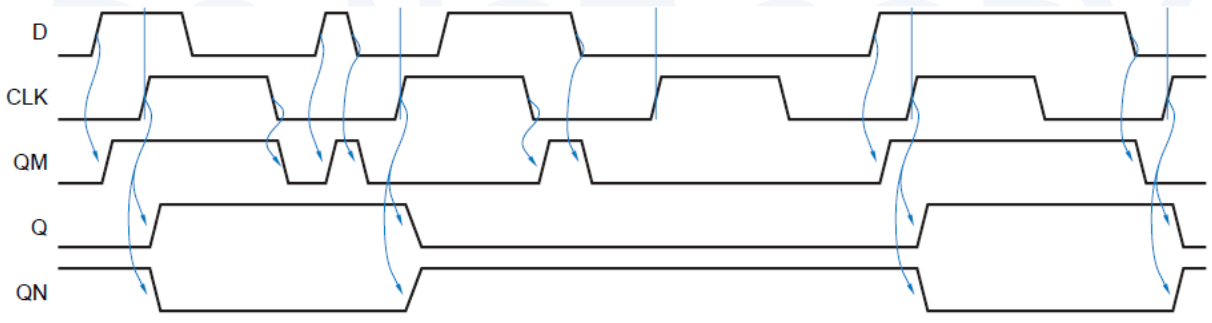
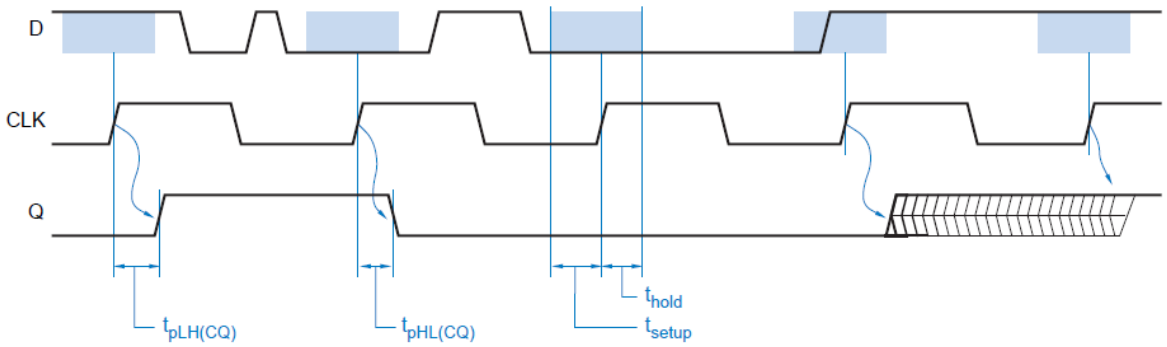


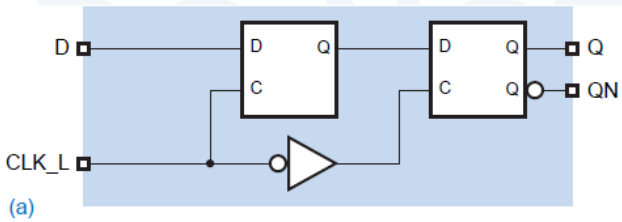
Figure 7-16 Functional behavior of a positive-edge-triggered D flip-flop.

Figure 7-17 Timing behavior of a positive-edge-triggered D flip-flop.



khi CLK thay đổi 0→1 nếu điều kiện t_{hold} và t_{setup} không thỏa mãn, D flip-flop có thể rơi vào trạng thái không xác định hoặc metasatble.

D flip-flop tác động theo sườn xuống



(b)

D	CLK_L	Q	QN
0	↓	0	1
1	↓	1	0
x	0	last Q	last QN
x	1	last Q	last QN

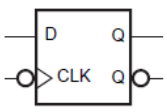
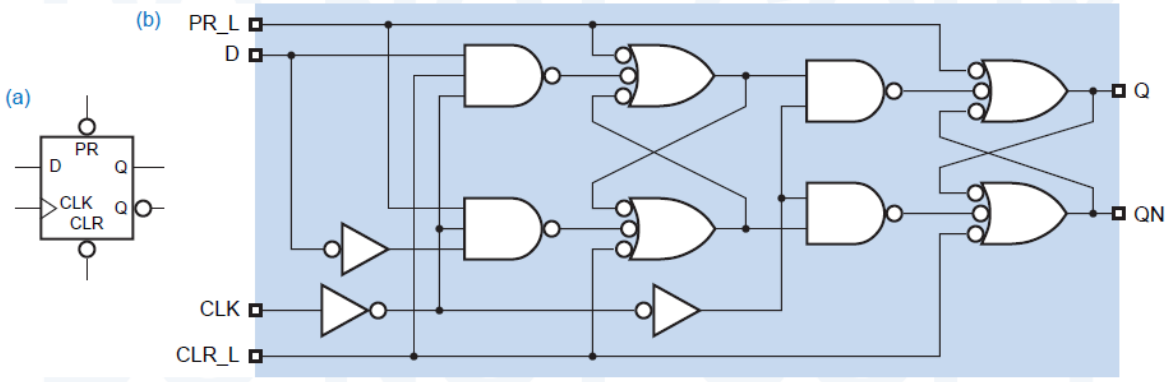


Figure 7-18 Negative-edge triggered D flip-flop: (a) circuit design using D latches; (b) function table; (c) logic symbol.

D flip-flop có đầu vào không đồng bộ

Figure 7-19 Positive-edge-triggered D flip-flop with preset and clear: (a) logic symbol; (b) circuit design using NAND gates.



Đầu vào không đồng bộ preset và clear
Chú ý: tuy nhiên sơ đồ trên không được dùng để chế tạo IC vì số gate lớn (11 gates)

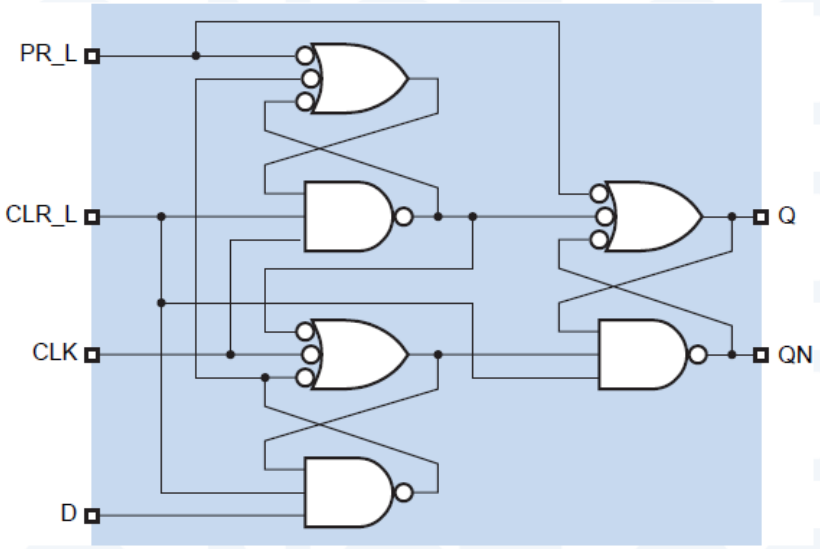


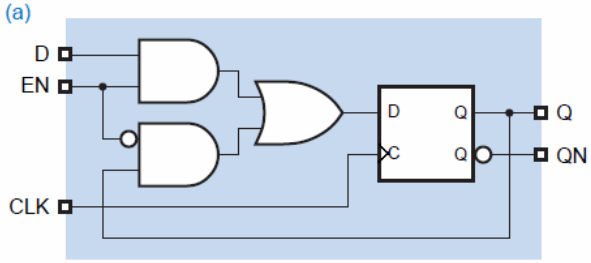
Figure 7-20 Commercial circuit for a positive-edge-triggered D flip-flop such as 74LS74.

Sơ đồ D flip-flop sử dụng 6 gates (thay vì 11 gates như đã giới thiệu trước)

D flip-flop tác động theo sườn xung với đầu vào Enable

Figure 7-21 Positive-edge-triggered D flip-flop with enable: (a) circuit design; (b) function table; (c) logic symbol.

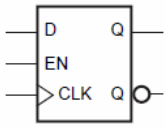
(a)



(b)

D	EN	CLK	Q	QN
0	1	↑	0	1
1	1	↑	1	0
x	0	↑	last Q	last QN
x	x	0	last Q	last QN
x	x	1	last Q	last QN

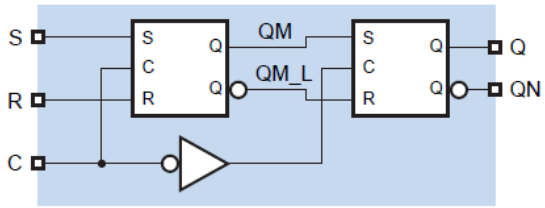
(c)



Master/Slave S-R flip-flop

Figure 7-24 Master/slave S-R flip-flop: (a) circuit using S-R latches; (b) function table; (c) logic symbol.

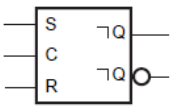
(a)



(b)

S	R	C	Q	QN
x	x	0	last Q	last QN
0	0	⎍	last Q	last QN
0	1	⎍	0	1
1	0	⎍	1	0
1	1	⎍	undef.	undef.

(c)



Giống D flip-flop: Q thay đổi tại thời điểm sườn xuống của xung Control
Khác D flip-flop: Q phụ thuộc vào các tín hiệu input trong suốt thời gian C=1 trước khi chuyển xuống 0 → flip-flop tác động theo xung

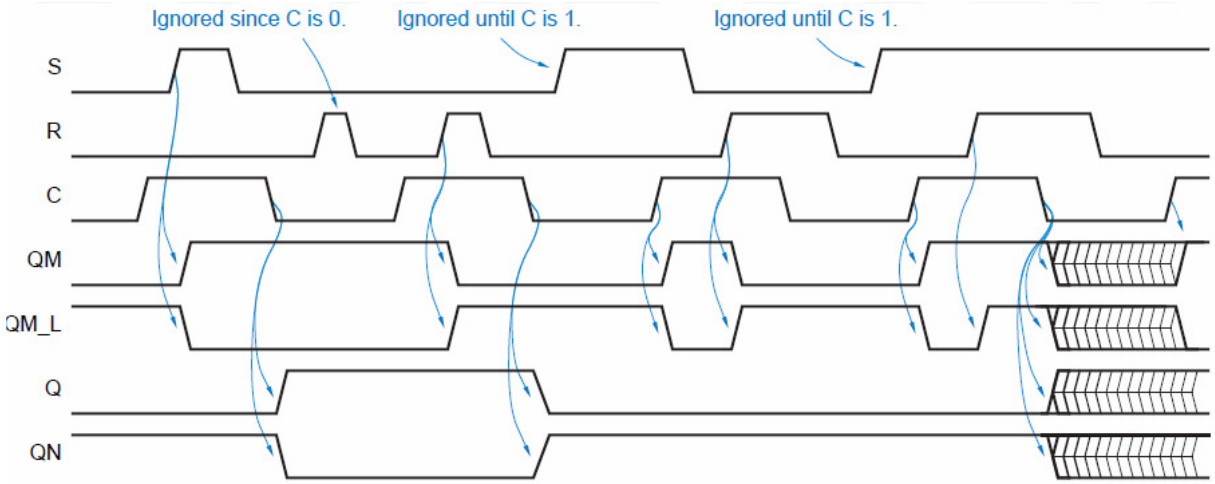


Figure 7-25 Internal and functional behavior of a master/slave S-R flip-flop.

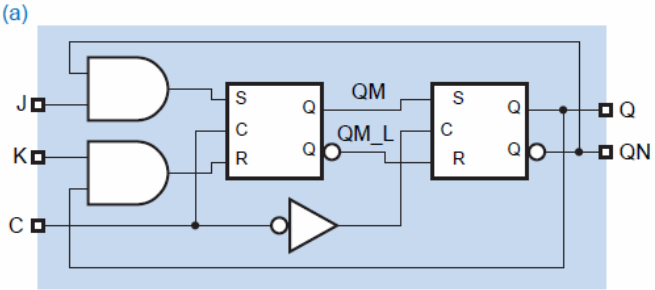
Trong trường hợp R=S=1, nếu C chuyển 1→0 các outputs sẽ rơi vào trạng thái không xác định hoặc metastable

Master-Slave J-K flip-flop

- J-K flip flop tránh được hiện tượng của R-S flip-flop khi cả hai đầu vào bằng 1

Figure 7-26 Master/slave J-K flip-flop: (a) circuit design using S-R latches; (b) function table; (c) logic symbol.

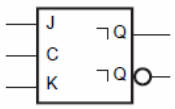
(a)



(b)

J	K	C	Q	QN
x	x	0	last Q	last QN
0	0	⎍	last Q	last QN
0	1	⎍	0	1
1	0	⎍	1	0
1	1	⎍	last QN	last Q

(c)



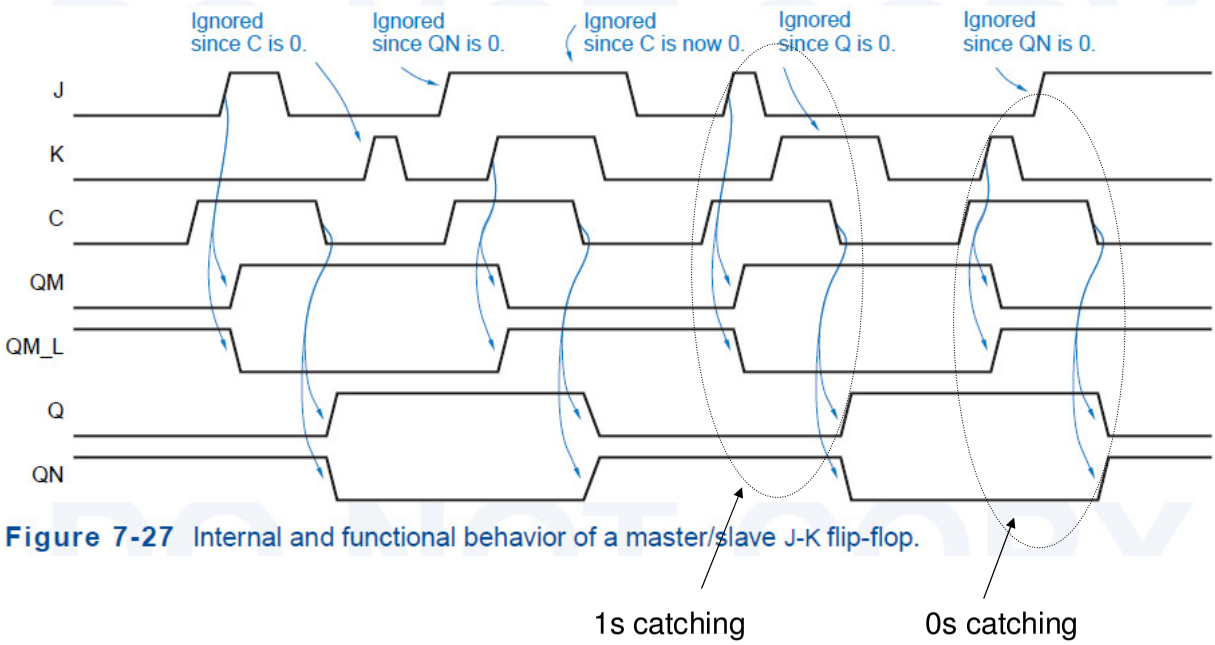


Figure 7-27 Internal and functional behavior of a master/slave J-K flip-flop.

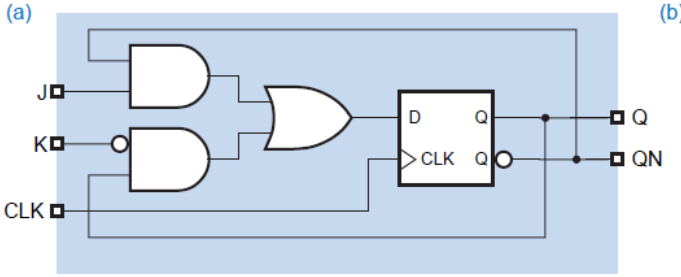
- 1s catching: tại sườn xuống của xung C:
  - J = 0, K = 1 thường Q = 0 và QN = 1
  - ***nhưng*** Q = 1, QN = 0, lý do là có một xung J = 1 tồn tại khi C = 1
- 0s catching: tại sườn xuống của xung C:
  - J = 1, K = 0 thường Q = 1 và QN = 0
  - ***nhưng*** Q = 0 và QN =1, lý do có một xung K = 1 tồn tại khi C = 1
- Để J-K flip-flop hoạt động đúng yêu cầu J và K không thay đổi trong suốt quá trình C = 1

Flip-flop J-K tác động theo sườn xung

- Hiện tượng 1s và 0 s catching có thể khắc phục sử dụng Edge-Trigerred J-K flip-flop

Figure 7-28 Edge-triggered J-K flip-flop: (a) equivalent function using an edge-triggered D flip-flop; (b) function table; (c) logic symbol.

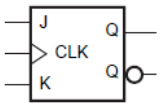
(a)



(b)

J	K	CLK	Q	QN
x	x	0	last Q	last QN
x	x	1	last Q	last QN
0	0	↓	last Q	last QN
0	1	↓	0	1
1	0	↓	1	0
1	1	↓	last QN	last Q

(c)



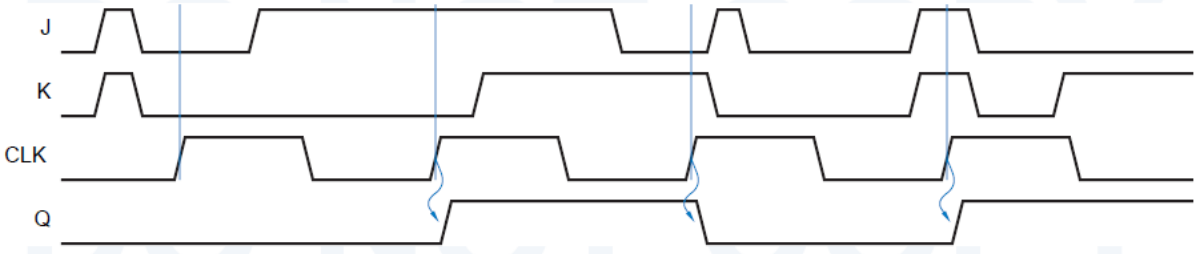
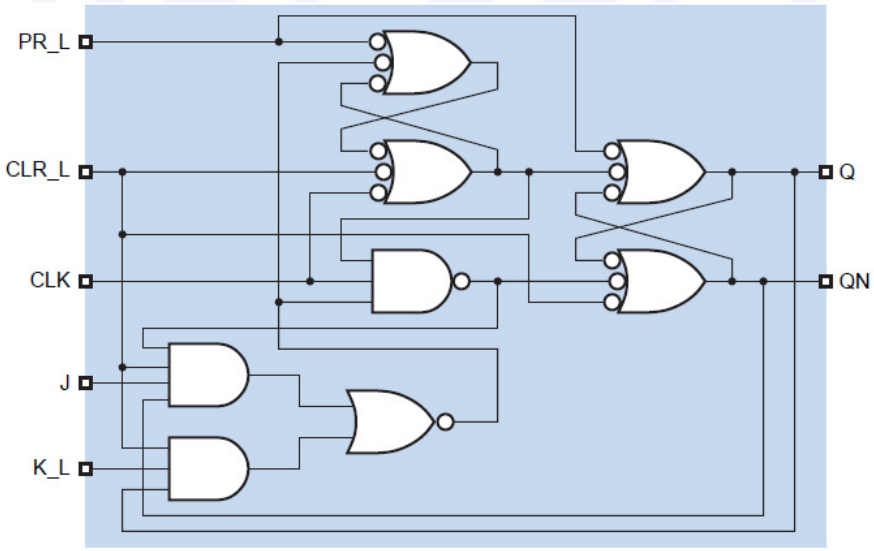


Figure 7-29 Functional behavior of a positive-edge-triggered J-K flip-flop.

Figure 7-30
Internal logic diagram for the 74LS109 positive-edge-triggered J-$\overline{K}$ flip-flop.



T (Toggle) flip-flop

- T flip-flop: thay đổi trạng thái tại mỗi xung đồng hồ

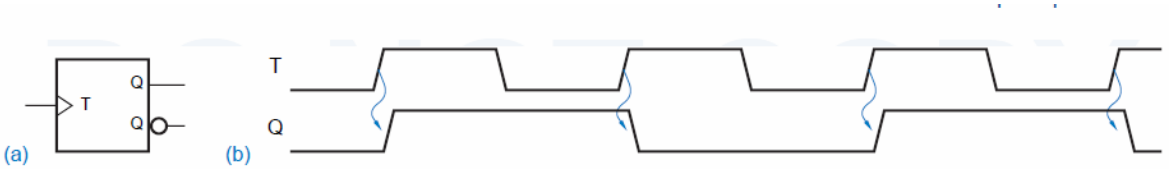


Figure 7-31 Positive-edge-triggered T flip-flop: (a) logic symbol; (b) functional behavior.

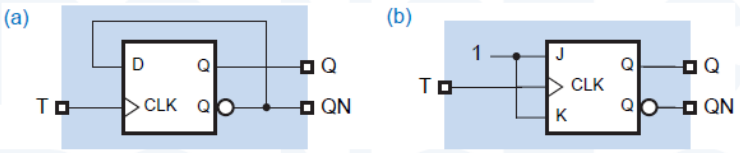


Figure 7-32
Possible circuit designs for a T flip-flop: (a) using a D flip-flop; (b) using a J-K flip-flop.

Figure 7-33 Positive-edge-triggered T flip-flop with enable: (a) logic symbol; (b) functional behavior.

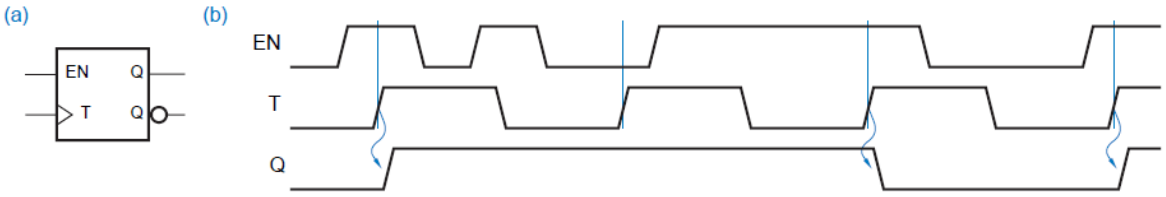
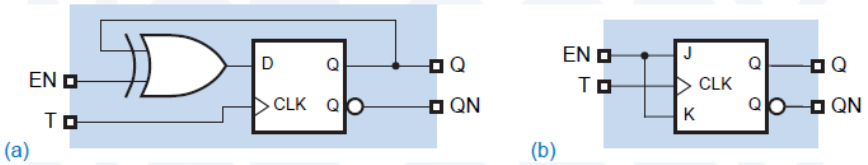


Figure 7-34
Possible circuits for a T flip-flop with enable: (a) using a D flip-flop; (b) using a J-K flip-flop.



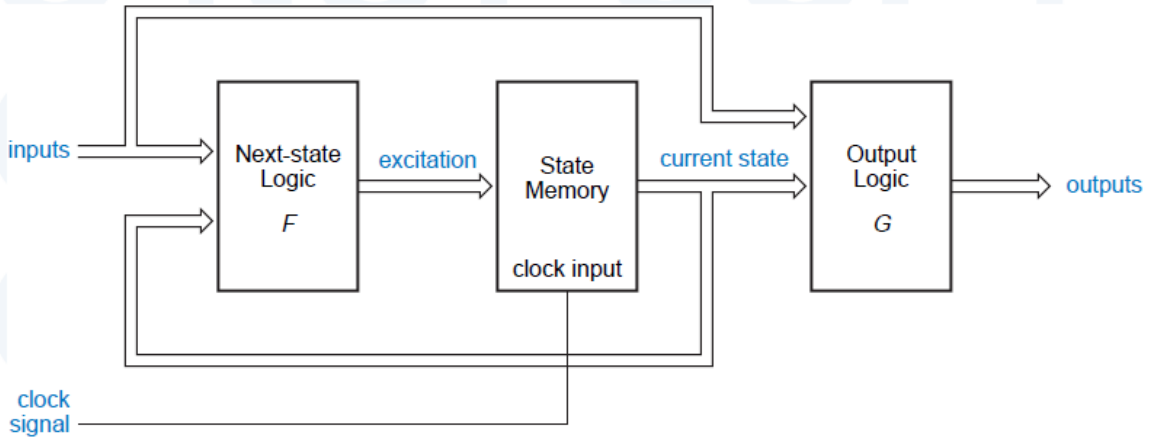
Máy trạng thái đồng bộ bởi xung nhịp

- Để hiểu phân tích máy trạng thái (state-machine), trước tiên xem xét "cloked-synchronous state machine":
  - state machine: máy trạng thái, tổng quát cho mạch logic dãy
  - clocked: các phần tử thay đổi trạng thái theo tín hiệu điều khiển
  - synchronous: các phần tử thay đổi trạng thái bởi cùng một tín hiệu clock

Cấu trúc của máy trạng thái (Mealy machine)

DO NOT COPY

Figure 7-35 Clocked synchronous state-machine structure (Mealy machine).



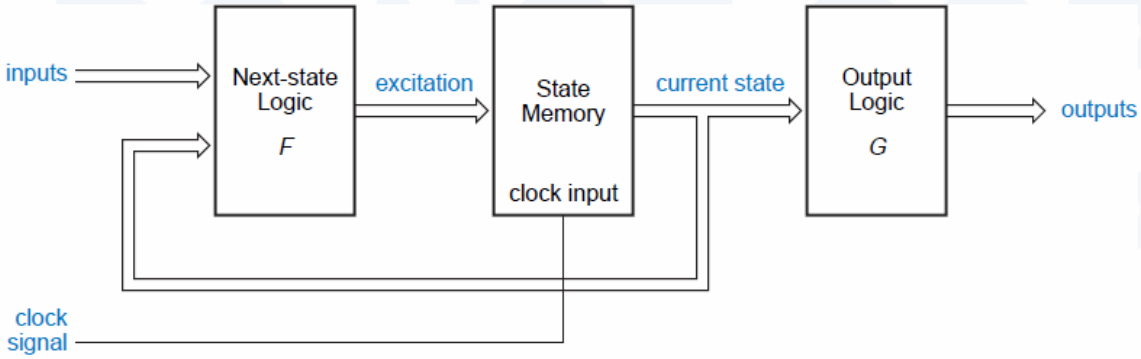
- State memory:
  - chứa n flip-flop để lưu giữ trạng thái hiện thời của máy, có 2n trạng thái khác nhau
  - các flip-flops được nối chung một nguồn Clock
- Trạng thái tiếp theo của máy được quyết định bởi mạch Next-State Logic F là một hàm của
  - các biến current state
  - các biến input
- Output logic G: là hàm của:
  - các biến current state
  - các biến input
- F và G là các mạch logic tổ hợp

$$\text{Next state} = F(\text{current state, input})$$
$$\text{Output} = G(\text{current state, input})$$

- Các flip-flop có thể sử dụng:
  - D flip-flop
  - J-K flip-flop
  - Tuy nhiên khi thiết kế mạch dãy thì D flip-flop tác động theo sườn hay được sử dụng vì việc thiết kế mạch logic ngày nay sử dụng chủ yếu là các IC logic lập trình được (được chế tạo có sẵn các D flip-flop)
- Output phụ thuộc cả vào current state và input → cấu trúc ***Mealy machine***

Moore machine

Figure 7-36 Clocked synchronous state-machine structure (Moore machine).



$$\text{Output} = G(\text{current state})$$

Pipelined output

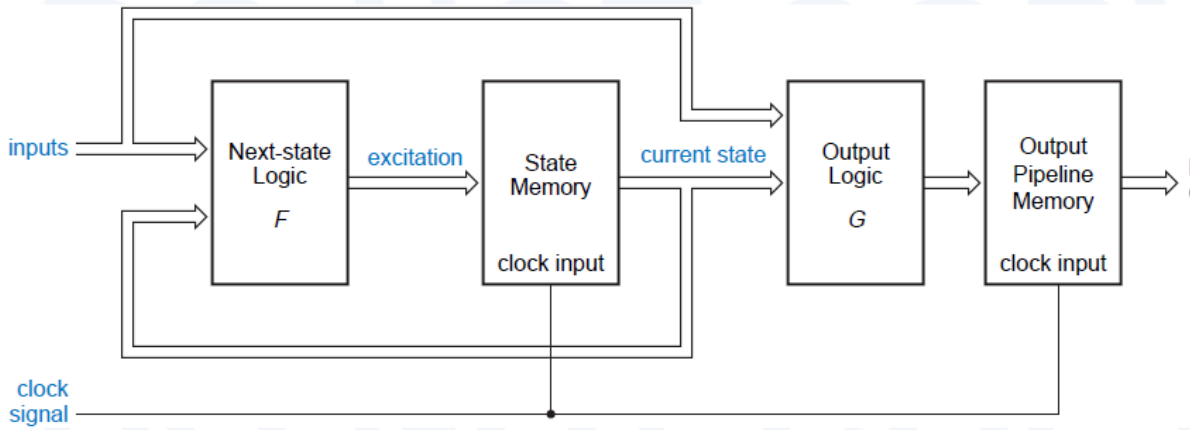


Figure 7-37 Mealy machine with pipelined outputs.

Yêu cầu: output trong một chu kỳ phụ thuộc vào giá trị state và input của chu kỳ trước → sử dụng thêm một tầng nhớ (flip-flop) đến Mealy machine

Nếu ghép Output pipeline memory như là một phần của state-memory → trở thành Moore machine

Các biểu thức đặc trưng

- Mô tả latch hay flip-flop có thể sử dụng các biểu thức đặc trưng (characteristic equation):
  - Mô tả trạng thái tiếp theo như là hàm của current state và input
  - Quy ước: Q* như là "next value of Q"
- Biểu thức đặc trưng không mô tả *chi tiết các hoạt động theo thời gian* của thiết bị (ví dụ D flip-flop tác động theo sườn lên, xuống hay mức thì đều có chung một biểu thức đặc trưng)

Device Type	Characteristic Equation
S-R latch	$Q* = S + R' \cdot Q$
D latch	$Q* = D$
Edge-triggered D flip-flop	$Q* = D$
D flip-flop with enable	$Q* = EN \cdot D + EN' \cdot Q$
Master/slave S-R flip-flop	$Q* = S + R' \cdot Q$
Master/slave J-K flip-flop	$Q* = J \cdot Q' + K' \cdot Q$
Edge-triggered J-K flip-flop	$Q* = J \cdot Q' + K' \cdot Q$
T flip-flop	$Q* = Q'$
T flip-flop with enable	$Q* = EN \cdot Q' + EN' \cdot Q$

Table 7-1 Latch and flip-flop characteristic equations.

Phân tích máy trạng thái với D flip-flop

$$\text{Next state} = F(\text{current state, input})$$
$$\text{Output} = G(\text{current state, input})$$

- Gồm 3 bước:
  - Xác định hàm F và G
  - Sử dụng F và G để xây dựng bảng state và output ứng với mỗi tổ hợp current state và current input
  - (option) Vẽ state diagram

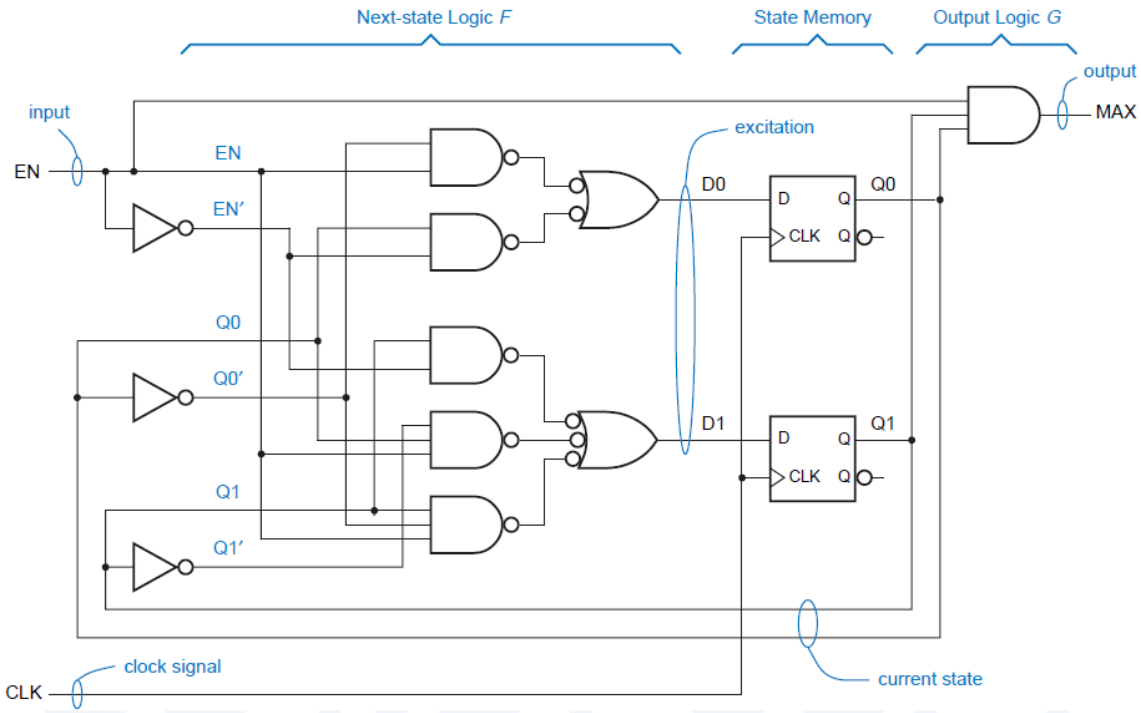


Figure 7-38 Clocked synchronous state machine using positive-edge-triggered D flip-flops.

ví dụ phân tích

- Tại mỗi xung nhịp D FF(flip-flop) sẽ sample tín hiệu tại D input và truyền đến đầu ra Q

  $Q^* = D$

- Có 2 D FF:
  - ký hiệu output là Q_0 và Q_1 là 2 biến trạng thái
  - ký hiệu input là D_0 và D_1 là hai tín hiệu kích thích (excitation)
- Biểu thức kích thích (excitation equation):

$$D0 = Q0 \cdot EN' + Q0' \cdot EN$$
$$D1 = Q1 \cdot EN' + Q1' \cdot Q0 \cdot EN + Q1 \cdot Q0' \cdot EN$$

- Sử dụng biểu thức đặc trưng của D FF

$$Q0^* = D0$$
$$Q1^* = D1$$

- Thay biểu thức kích thích:

$$Q0^* = Q0 \cdot EN' + Q0' \cdot EN$$
$$Q1^* = Q1 \cdot EN' + Q1' \cdot Q0 \cdot EN + Q1 \cdot Q0' \cdot EN$$

biểu thức này thể hiện giá trị các biến trạng thái tiếp theo như là hàm của current state và current input, được gọi là các biểu thức chuyển (transistion equation)

(a)

	EN	
Q1 Q0	0	1
00	00	01
01	01	10
10	10	11
11	11	00
	Q1* Q0*	

(b)

	EN	
S	0	1
A	A	B
B	B	C
C	C	D
D	D	A
	S*	

(c)

	EN	
S	0	1
A	A, 0	B, 0
B	B, 0	C, 0
C	C, 0	D, 0
D	D, 0	A, 1
	S*, MAX	

Table 7-2 Transition, state, and state/output tables for the state machine in Figure 7-38.

(a): transistion table

4 trạng thái (Q1,Q0) = (0,0) (0,1) (1,0) (1,1)

1 tín hiệu input EN = 0 , 1 → có 8 tổ hợp state/input

(b): state table: bằng cách gán tên cho các trạng thái

(0,0) = A, (0,1) = B, (1,0) = C và (1,1) = D ta có bảng trạng thái (b)

S ký hiệu cho current state, S^* ký hiệu cho next state

- Từ sở đồ, xây dựng hàm logic cho output

$$MAX = Q1 \cdot Q0 \cdot EN$$

- Từ đó xây dựng bảng (c): state/output table

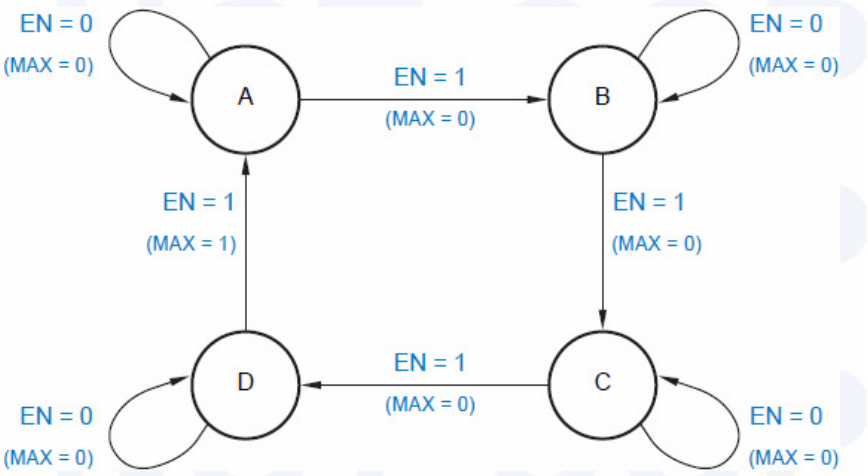


Figure 7-39
State diagram corresponding to the state machine of Table 7-2.

Mỗi vòng tròn (hay nút) ký hiệu cho một state. Tên vòng tròn là tên của state

Các state liên kết bởi các mũi tên chỉ chiều chuyển trạng thái và điều kiện chuyển

các bước phân tích chi tiết

excitation equations
transition equations
transition table
output equations

1. Determine the excitation equations for the flip-flop control inputs.
2. Substitute the excitation equations into the flip-flop characteristic equations to obtain transition equations.
3. Use the transition equations to construct a transition table.
4. Determine the output equations.

5. Add output values to the transition table for each state (Moore) or state/input combination (Mealy) to create a transition/output table. — *transition/output table*
6. Name the states and substitute state names for state-variable combinations in the transition/output table to obtain a state/output table. — *state names*, *state/output table*
7. (Optional) Draw a state diagram corresponding to the state/output table. — *state diagram*

Ví dụ (bài tập về nhà)

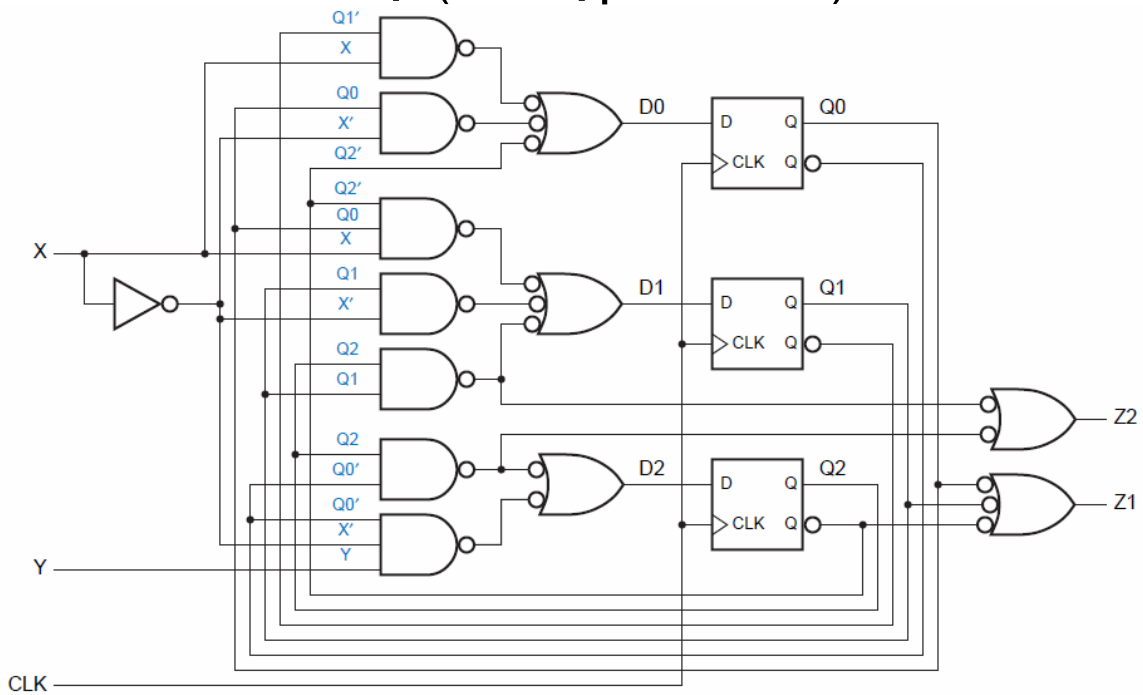


Figure 7-43 A clocked synchronous state machine with three flip-flops and eight states.

- excitation equation

$$D0 = Q1' \cdot X + Q0 \cdot X' + Q2$$
$$D1 = Q2' \cdot Q0 \cdot X + Q1 \cdot X' + Q2 \cdot Q1$$
$$D2 = Q2 \cdot Q0' + Q0' \cdot X' \cdot Y$$

- excitation equation

$$Q0* = Q1' \cdot X + Q0 \cdot X' + Q2$$
$$Q1* = Q2' \cdot Q0 \cdot X + Q1 \cdot X' + Q2 \cdot Q1$$
$$Q2* = Q2 \cdot Q0' + Q0' \cdot X' \cdot Y$$

Transistion table

(a)

	XY				
Q2 Q1 Q0	00	01	10	11	Z1 Z2
000	000	100	001	001	10
001	001	001	011	011	10
010	010	110	000	000	10
011	011	011	010	010	00
100	101	101	101	101	11
101	001	001	001	001	10
110	111	111	111	111	11
111	011	011	011	011	11
	Q2* Q1* Q0*				

- output equation

$$Z1 = Q2 + Q1' + Q0'$$
$$Z2 = Q2 \cdot Q1 + Q2 \cdot Q0'$$

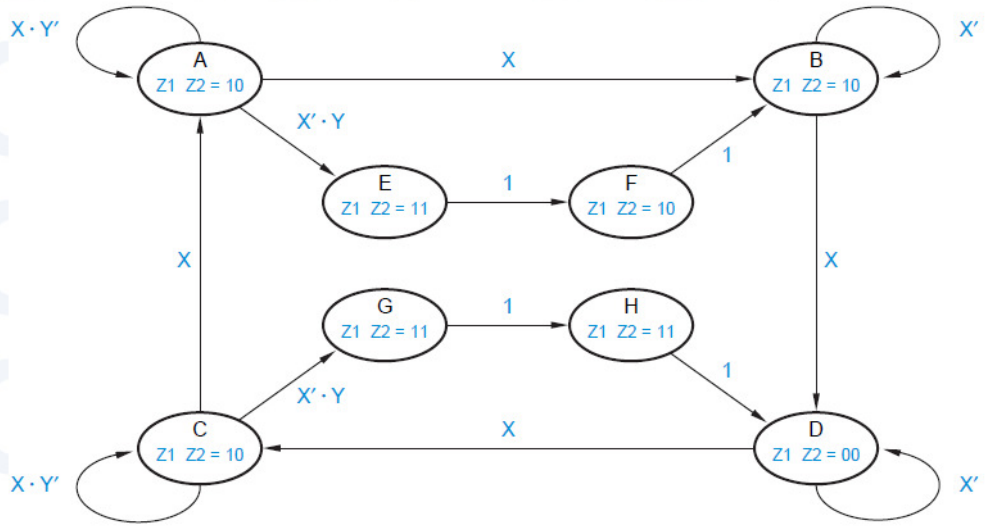
(b)

	XY				
S	00	01	10	11	Z1 Z2
A	A	E	B	B	10
B	B	B	D	D	10
C	C	G	A	A	10
D	D	D	C	C	00
E	F	F	F	F	11
F	B	B	B	B	10
G	H	H	H	H	11
H	D	D	D	D	11
	S*				

state/output table

- Biểu đồ trạng thái

Figure 7-44 State diagram corresponding to Table 7-4.



- Mỗi liên kết được gán với một biểu thức → transistion expression
- Chuyển trạng thái xảy ra khi tổ hợp các giá trị input sao cho transistion expression = 1
- Nếu liên kết được gán “1” có nghĩa là luôn xảy ra

Phân tích state machine với J-K flip-flop

- Các mạch clocked-synchronous state machine với J-K ff có thể phân tích giống như là vơi D ff với lưu ý:
  - Biểu thức kích thích cần phải viết cho 2 input J và K
  - Biểu thức đặc trưng của J-K ff là $Q^* = J ¢ Q' + K' ¢ Q$

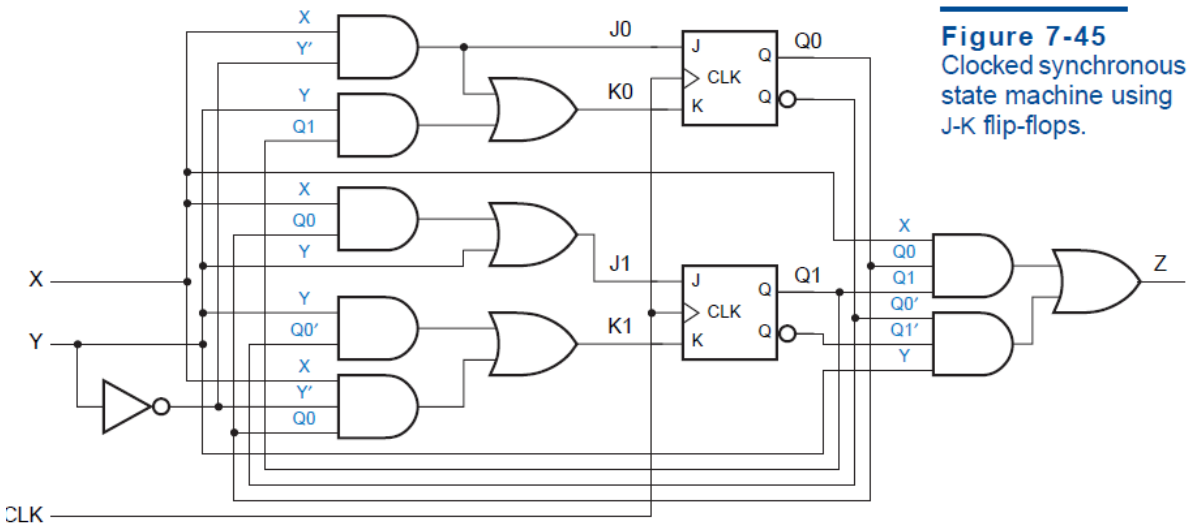


Figure 7-45
Clocked synchronous state machine using J-K flip-flops.

- Biểu thức kích thích

$$J0 = X \cdot Y'$$
$$K0 = X \cdot Y' + Y \cdot Q1$$
$$J1 = X \cdot Q0 + Y$$
$$K1 = Y \cdot Q0' + X \cdot Y' \cdot Q0$$

- Biểu thức chuyển trạng thái

$$Q0* = J0 \cdot Q0' + K0' \cdot Q0$$
$$= X \cdot Y' \cdot Q0' + (X \cdot Y' + Y \cdot Q1)' \cdot Q0$$
$$= X \cdot Y' \cdot Q0' + X' \cdot Y' \cdot Q0 + X' \cdot Q1' \cdot Q0 + Y \cdot Q1' \cdot Q0$$
$$Q1* = J1 \cdot Q1' + K1' \cdot Q1$$
$$= (X \cdot Q0 + Y) \cdot Q1' + (Y \cdot Q0' + X \cdot Y' \cdot Q0)' \cdot Q1$$
$$= X \cdot Q1' \cdot Q0 + Y \cdot Q1' + X' \cdot Y' \cdot Q1 + Y' \cdot Q1 \cdot Q0' + X' \cdot Q1 \cdot Q0 + Y \cdot Q1 \cdot Q0$$

- Biểu thức output

$$Z = X \cdot Q1 \cdot Q0 + Y \cdot Q1' \cdot Q0'$$

Table 7-5 Transition/output and state/output tables for the state machine in Figure 7-45.

(a)

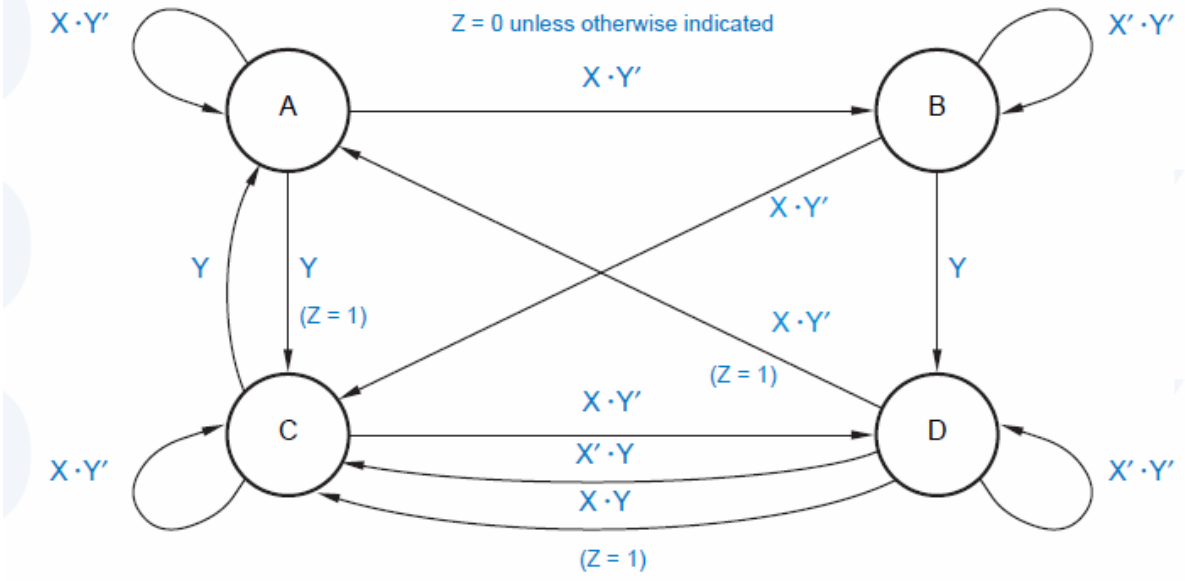
	XY			
Q1 Q0	00	01	10	11
00	00, 0	10, 1	01, 0	10, 1
01	01, 0	11, 0	10, 0	11, 0
10	10, 0	00, 0	11, 0	00, 0
11	11, 0	10, 0	00, 1	10, 1

Q1* Q0*, Z

(b)

	XY			
S	00	01	10	11
A	A, 0	C, 1	B, 0	C, 1
B	B, 0	D, 0	C, 0	D, 0
C	C, 0	A, 0	D, 0	A, 0
D	D, 0	C, 0	A, 1	C, 1

S*, Z



Các bước thiết kế mạch logic dãy đồng bộ

1. Construct a state/output table corresponding to the word description or specification, using mnemonic names for the states. (It's also possible to start with a state diagram; this method is discussed in \secref{diagdsgn}.)	*state/output table*
2. (Optional) Minimize the number of states in the state/output table.	*state minimization*
3. Choose a set of state variables and assign state-variable combinations to the named states.	*state assignment*
4. Substitute the state-variable combinations into the state/output table to create a transition/output table that shows the desired next state-variable combination and output for each state/input combination.	*transition/output table*
5. Choose a flip-flop type (e.g., D or J-K) for the state memory. In most cases, you'll already have a choice in mind at the outset of the design, but this step is your last chance to change your mind.	
6. Construct an excitation table that shows the excitation values required to obtain the desired next state for each state/input combination.	*excitation table*
7. Derive excitation equations from the excitation table.	*excitation equations*
8. Derive output equations from the transition/output table.	*output equations*
9. Draw a logic diagram that shows the state-variable storage elements and realizes the required excitation and output equations. (Or realize the equations directly in a programmable logic device.)	*logic diagram*

Ví dụ

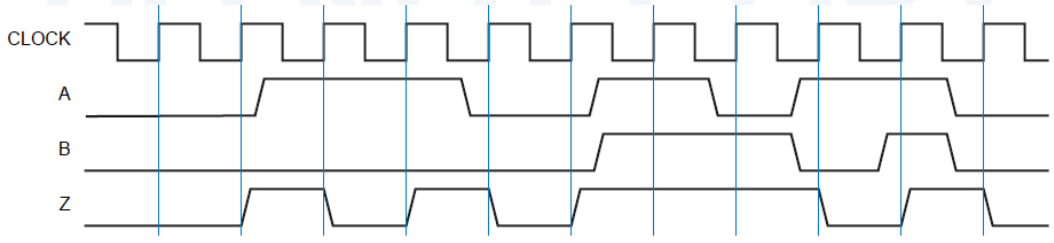
Design a clocked synchronous state machine with two inputs, A and B, and a single output Z that is 1 if:

- A had the same value at each of the two previous clock ticks, *or*
- B has been 1 since the last time that the first condition was true.

Otherwise, the output should be 0.

Biểu diễn dạng tín hiệu theo thời gian

Figure 7-47 Timing diagram for example state machine.



Phân tích yêu cầu

- z_k bằng 1 nếu:
  - $A_k = 0$ và $A_{k-1} =$ 0 hoặc
  - $A_k = 1$ và $A_{k-1} = 1$ hoặc
  - B =1 bắt đầu từ thời điểm (trong quá khứ) mà tại đó A bằng nhau tại 2 xung nhịp liên tiếp (trong trường hợp này z=1 không phụ thuộc vào A)
- Ngược lại z sẽ bằng 0

Trạng thái (1)

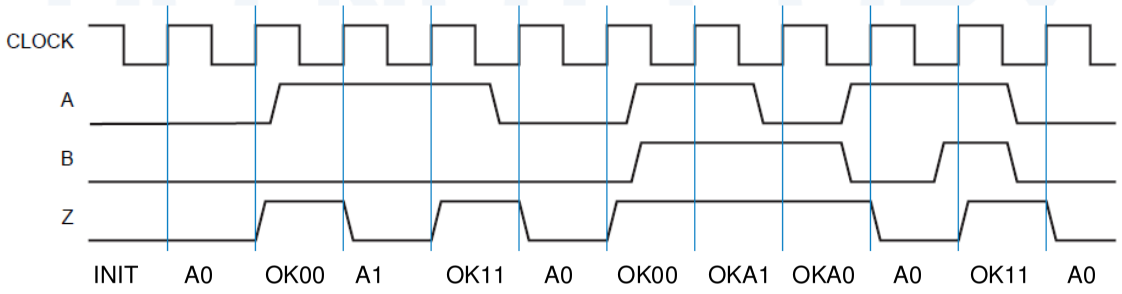
- Trạng thái (trong khoảng thời gian từ k đến k+1)
- A0 (Z = 0)
  - $A_k = 0$ và $A_{k-1} = 1$
  - và B = 0 tại thời điểm mà trước đó đã có một cặp giá trị A bằng nhau (trong quá khứ)
- A1 (Z = 0)
  - $A_k = 1$ và $A_{k-1} = 0$
  - và B = 0 tại thời điểm bất kỳ (như vậy tại thời điểm k, B có thể = 1 hoặc 0) mà trước đó đã có một cặp giá trị A bằng nhau (trong quá khứ)
- OK00 (Z = 1)
  - $A_k = 0$ và $A_{k-1} = 0$
  - B bất kỳ
- OK11 (Z = 1)
  - $A_k = 1$ và $A_{k-1} = 1$
  - B bất kỳ
- OKA0 (Z= 1)
  - $A_k = 0$ và $A_{k-1} = 1$
  - B = 1 kể từ thời điểm gần nhất có có cặp A có giá trị bằng nhau
- OKA1 (Z=1)
  - $A_k = 1$ và $A_{k-1} = 0$
  - B = 1 kể từ thời điểm gần nhất có có cặp A có giá trị bằng nhau

Bảng chuyển trạng thái (1)

(a)

Meaning	S	A B: 00	01	11	10	Z
Initial state	INIT	A0	A0	A1	A1	0
Got a 0 on A	A0	OK00	OK00	A1	A1	0
Got a 1 on A	A1	A0	A0	OK11	OK11	0
Got 00 on A	OK00	OK00	OK00	OKA1	A1	1
Got 11 on A	OK11	A0	OKA0	OK11	OK11	1
OK, got a 0 on A	OKA0	OK00	OK00	OKA1	A1	1
OK, got a 1 on A	OKA1	A0	OKA0	OK11	OK11	1
		S*				

Figure 7-47 Timing diagram for example state machine.



Trạng thái (2)

- A0 và A1 tương tự như trên
- OK0 (Z=1)
  - $A_k = 0$ và $A_{k-1} = 0$ và B bất kỳ
  - hoặc $A_k = 0$ và $A_{k-1} = 1$ và B = 1 kể từ thời điểm gần nhất có có cặp A có giá trị bằng nhau
- OK1 (Z=1)
  - $A_k = 1$ và $A_{k-1} = 1$ và B bất kỳ
  - hoặc $A_k = 1$ và $A_{k-1} = 0$ và B = 1 kể từ thời điểm gần nhất có có cặp A có giá trị bằng nhau

Bảng chuyển trạng thái (2)

		A B				
Meaning	S	00	01	11	10	Z
Initial state	INIT	A0	A0	A1	A1	0
Got a 0 on A	A0	OK0	OK0	A1	A1	0
Got a 1 on A	A1	A0	A0	OK1	OK1	0
Two equal, A=0 last	OK0	OK0	OK0	OK1	A1	1
Two equal, A=1 last	OK1	A0	OK0	OK1	OK1	1
		S*				

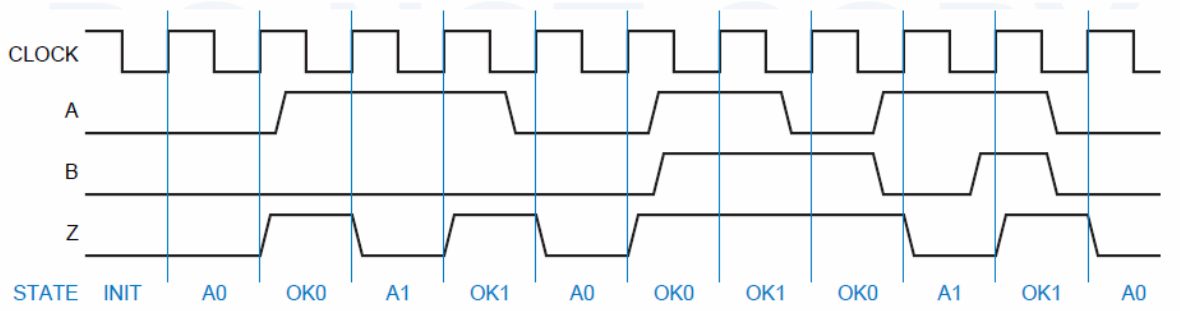


Figure 7-50 Timing diagram and state sequence for example state machine.

Tối thiểu hóa số trạng thái

- Ý tưởng giảm số trạng thái là dựa trên việc xác định trạng thái tương đương
- Hai trạng thái S_1 và S_2 được coi là tương đương nếu thỏa mãn 2 điều kiện:
  - S_1 và S_2 cần tạo ra output giống nhau với tất cả các tổ hợp input
  - Với mỗi tổ hợp input, S_1 và S_2 cần tạo ra next state giống nhau hoặc tương đương

(a)

Meaning	S	A B = 00	A B = 01	A B = 11	A B = 10	Z
Initial state	INIT	A0	A0	A1	A1	0
Got a 0 on A	A0	OK00	OK00	A1	A1	0
Got a 1 on A	A1	A0	A0	OK11	OK11	0
Got 00 on A	OK00	OK00	OK00	OKA1	A1	1
Got 11 on A	OK11	A0	OKA0	OK11	OK11	1
OK, got a 0 on A	OKA0	OK00	OK00	OKA1	A1	1
OK, got a 1 on A	OKA1	A0	OKA0	OK11	OK11	1
		S*				

OK00 và OKA0 là tương đương
OK11 và OKA1 là tương đương

Biến trạng thái

- n flip-flop có thể mô tả 2^n trạng thái
- với s trạng thái cần *ít nhất* ($\log_2 s$) flip-flop → có thể có một số trạng thái không sử dụng
- Trong ví dụ với 5 trạng thái sẽ cần ít nhất 3 flip-flop (dư 3 trạng thái không sử dụng)
- Chú ý: việc lựa chọn số biến trạng thái ít nhất *không* đảm bảo rằng:
  - các biểu thức kích thích là đơn giản nhất
  - các biểu thức output là đơn giản nhất
  - mạch là rẻ nhất

- Làm cách nào để lựa chọn số biến trạng thái và tổ hợp các biến trạng thái tổ tối ưu ???
- Câu trả lời là: phải tiến hành thử tất cả các trường hợp có thể → tốn rất nhiều thời gian:

CAUTION: MATH The number of different ways to choose m coded states out of a set of n possible states is given by a *binomial coefficient*, denoted $\binom{n}{m}$, whose value is $\frac{n!}{m! \cdot (n-m)}$. (We used binomial coefficients previously in Section 2.10, in the context of decimal coding.) In our example, there are $\binom{8}{5}$ different ways to choose five coded states out of eight possible states, and 5! ways to assign the five named states to each different choice. So there are $\frac{8!}{5! \cdot 3!} \cdot 5!$ or 6,720 different ways to assign the five states of our example machine to combinations of three binary state variables. We don't have time to look at all of them.

	Assignment			
State name	Simplest Q1–Q3	Decomposed Q1–Q3	One-hot Q1–Q5	Almost one-hot Q1–Q4
INIT	000	000	00001	0000
A0	001	100	00010	0001
A1	010	101	00100	0010
OK0	011	110	01000	0100
OK1	100	111	10000	1000

Table 7-7 Possible state assignments for the state machine in Table 7-6.

DO NOT COPY

Các trạng thái không sử dụng

We promised earlier to consider the disposition of *unused states* when the number of states available with n flip-flops, 2^n, is greater than the number of states required, s. There are two approaches that make sense, depending on the application requirements:

- *Minimal risk.* This approach assumes that it is possible for the state machine somehow to get into one of the unused (or "illegal") states, perhaps because of a hardware failure, an unexpected input, or a design error. Therefore, all of the unused state-variable combinations are identified, and explicit next-state entries are made so that, for any input combination, the unused states go to the "initial" state, the "idle" state, or some other "safe" state. This is an automatic consequence of some design methodologies if the initial state is coded 00. . . 00.
- *Minimal cost.* This approach assumes that the machine will never enter an unused state. Therefore, in the transition and excitation tables, the next-state entries of the unused states can be marked as "don't-cares." In most cases, this simplifies the excitation logic. However, the machine's behavior if it ever does enter an unused state may be pretty weird.

Tổng hợp sử dụng D flip-flop

- Nhắc lại: sử dụng D flip-flop có ưu điểm:
  - tồn tại ở dạng IC rời cũng như trong các thiết bị lập trình được
  - Dễ sử dụng (hơn so với J-K flip-flop) vì biểu thức đặc trưng đơn giản Q* = D
- Do (Q* = D) do bảng transistion/output = excitation/output

	A B				
Q1 Q2 Q3	00	01	11	10	Z
000	100	100	101	101	0
100	110	110	101	101	0
101	100	100	111	111	0
110	110	110	111	101	1
111	100	110	111	111	1
	Q1* Q2* Q3*				

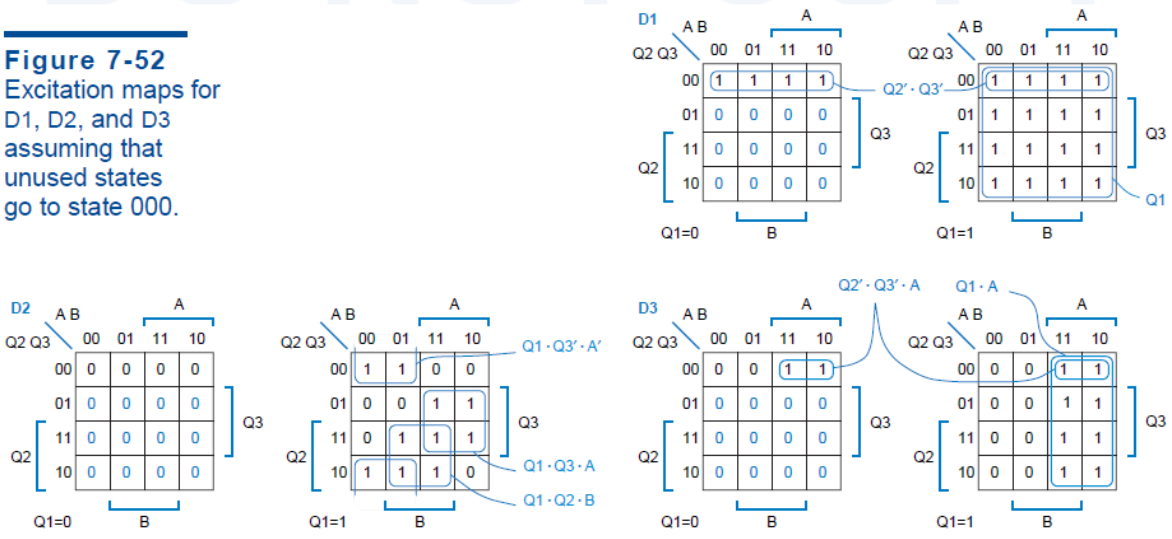
Table 7-8 Transition and output table for example problem.

Table 7-9
Excitation and output table for Table 7-8 using D flip-flops.

	AB				
Q1 Q2 Q3	00	01	11	10	Z
000	100	100	101	101	0
100	110	110	101	101	0
101	100	100	111	111	0
110	110	110	111	101	1
111	100	110	111	111	1
	D1 D2 D3				

Bảng excitation giống như bảng chân lý với các hàm logic D1, D2, D3 là hàm của 5 biến (A,B,Q1,Q2,Q3) → sử dụng phương pháp tổng hợp hàm logic "tổng của các tích hoặc tích của các tổng"
Nếu số biến ít → có thể sử dụng phương pháp bìa Karnaugh để tổng hợp hàm

Figure 7-52
Excitation maps for D1, D2, and D3 assuming that unused states go to state 000.



- Chú ý excitation bảng và bảng chân lý có sự khác nhau:
  - Bảng excitation không chỉ ra hàm logic của tất cả các tổ hợp input (các unused states)
- Trong ví dụ trên, sử dụng quy tắc minimal-risk: khi hệ thống rơi vào unused state, thì nex-state sẽ là trạng thái 000:
  - Với Q1 = 0 thì 3 hàng cuối sẽ là 0

- Với bìa Karnaugh trên thu được biểu thức kích thích

$$
\begin{aligned}
D1 &= Q1 + Q2' \cdot Q3' \\
D2 &= Q1 \cdot Q3' \cdot A' + Q1 \cdot Q3 \cdot A + Q1 \cdot Q2 \cdot B \\
D3 &= Q1 \cdot A + Q2' \cdot Q3' \cdot A
\end{aligned}
$$

- Tương tự ta có thể xây dựng hàm logic cho output

$$
\begin{aligned}
Z &= Q1 \cdot Q2 \cdot Q3' + Q1 \cdot Q2 \cdot Q3 \\
&= Q1 \cdot Q2
\end{aligned}
$$

MINIMAL-COST SOLUTION

If we choose in our example to derive minimal-cost excitation equations, we write "don't-cares" in the next-state entries for the unused states. The colored d's in Figure 7-53 are the result of this choice. The excitation equations obtained from this map are somewhat simpler than before:

$$D1 = 1$$
$$D2 = Q1 \cdot Q3' \cdot A' + Q3 \cdot A + Q2 \cdot B$$
$$D3 = A$$

For a minimal-cost output function, the value of Z is a "don't-care" for the unused states. This leads to an even simpler output function, $Z = Q2$. The logic diagram for the minimal-cost solution is shown in Figure 7-54.

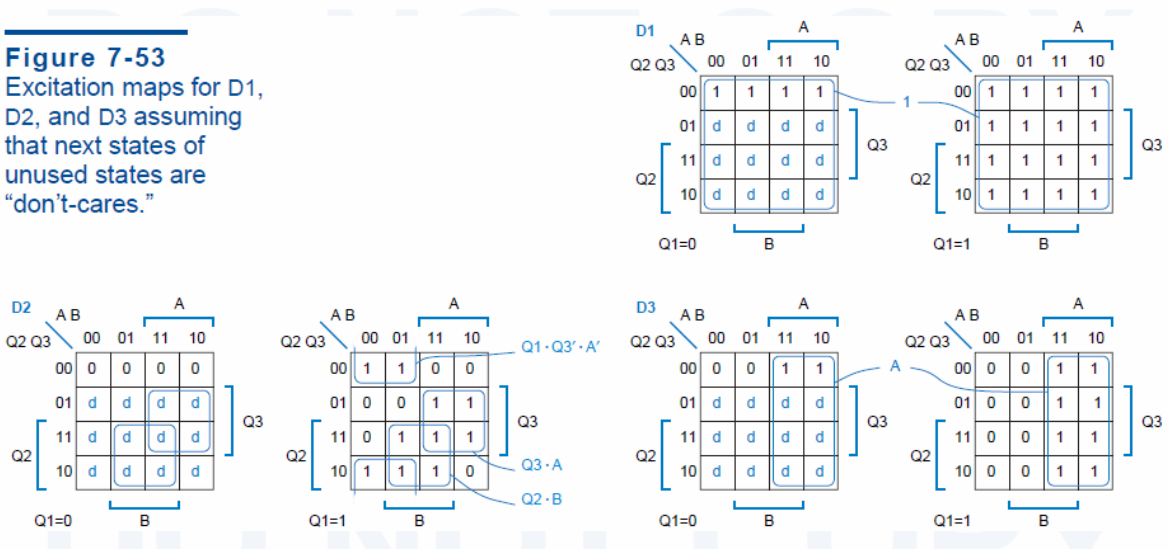


Figure 7-53
Excitation maps for D1, D2, and D3 assuming that next states of unused states are "don't-cares."

sử dụng tiêu chuẩn minimal-cost, next-state của các unused-state là don't-care → hàm logic tổng hợp sẽ đơn giản hơn

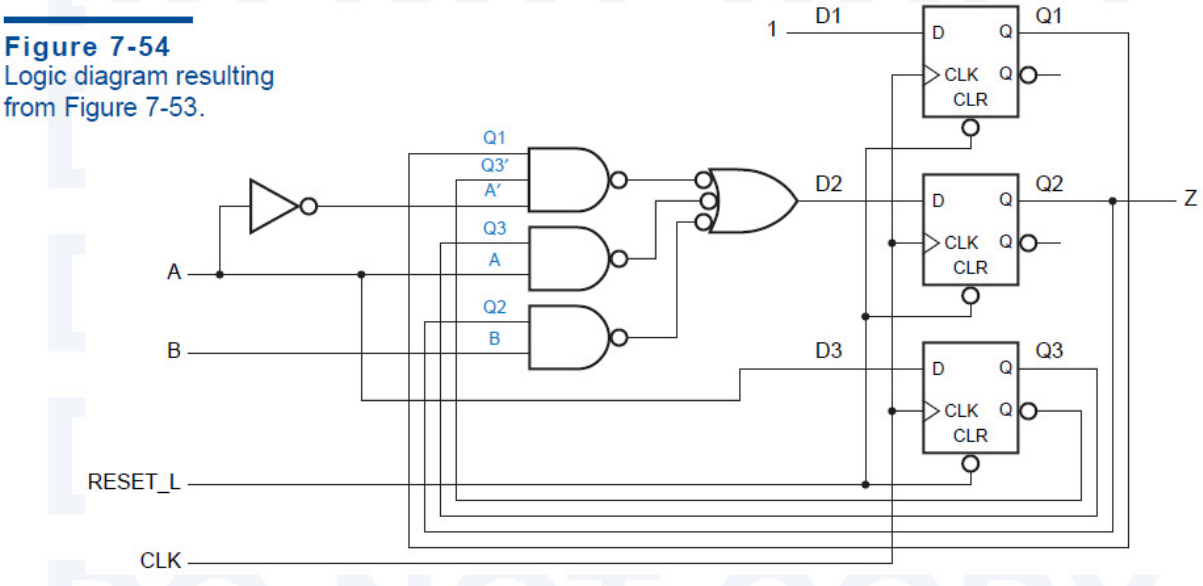


Figure 7-54
Logic diagram resulting from Figure 7-53.

Tổng hợp mạch dãy sử dụng J-K flip-flop

- J-K flip-flop có biểu thức đặc trưng phức tạp hơn D flip-flop: $Q^* = J ¢ Q' + K' ¢ Q$
- Với nhiều input hơn, sử dụng J-K cho phép mạch điều khiển tín hiệu excitation *có thể* (không chắc chắn) đơn giản hơn
- Thực tế:
  - sử dụng J-K flip-flop phù hợp cho các thiết kế với các IC loại SSI (Small-Scale Integration) hơn là các IC khả trình loại MSI hoăck LSI
  - Hiện nay trong các thiết kế với mạch logic khả trình sử dụng chủ yếu là D flip-flop

- Từ bảng transistion *không thể* chuyển qua *trực tiếp* bảng excitation như đối với D flip-flop:
  - Để xây dựng J-K excitation table, cần xem xét trạng thái hiện tại và cả next-state (khác với D flip-flop chỉ cần quan tâm next-state để xây dựng biểu thức kích thích)
  - Sử dụng bảng hoạt động của J-K flip-flop

Q	Q*	J	K
0	0	0	d
0	1	1	d
1	0	d	1
1	1	d	0

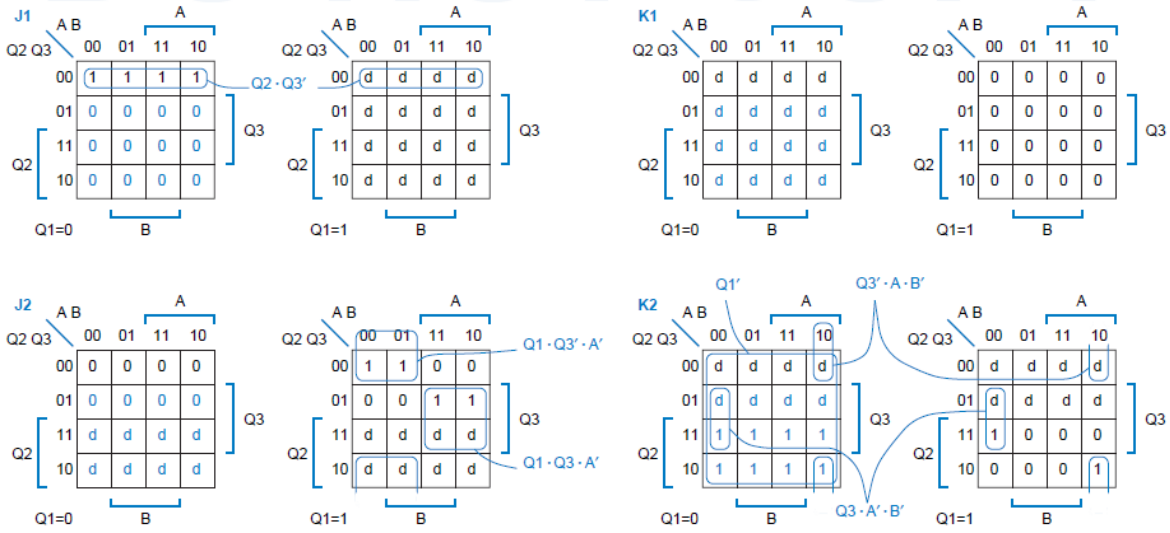
Table 7-10
Application table for J-K flip-flops.

d: don't-care

	AB				
Q1 Q2 Q3	00	01	11	10	Z
000	1d, 0d, 0d	1d, 0d, 0d	1d, 0d, 1d	1d, 0d, 1d	0
100	d0, 1d, 0d	d0, 1d, 0d	d0, 0d, 1d	d0, 0d, 1d	0
101	d0, 0d, d1	d0, 0d, d1	d0, 1d, d0	d0, 1d, d0	0
110	d0, d0, 0d	d0, d0, 0d	d0, d0, 1d	d0, d1, 1d	1
111	d0, d1, d1	d0, d0, d1	d0, d0, d0	d0, d0, d0	1
	J1 K1, J2 K2, J3 K3				

Table 7-11
Excitation and output table for the state machine of Table 7-8, using J-K flip-flops.

Figure 7-55 Excitation maps for J1, K1, J2, K2, J3, and K3, assuming that unused states go to state 000.



Sử dụng phương châm minimal-risk: các trạng thái không sử dụng sẽ được chuyển về trạng thái 000

- Biểu thức kích thích

$$J1 = Q2' \cdot Q3' \qquad K1 = 0$$
$$J2 = Q1 \cdot Q3' \cdot A' + Q1 \cdot Q3 \cdot A \qquad K2 = Q1' + Q3' \cdot A \cdot B' + Q3 \cdot A' \cdot B'$$
$$J3 = Q2' \cdot A + Q1 \cdot A \qquad K3 = Q1' + A'$$

(so với biểu thức kích thích sử dụng D flip-flop thì KHÔNG đơn giản hơn)

MINIMAL-COST SOLUTION

In the preceding design example, excitation maps for the minimal-cost approach would have been somewhat easier to construct, since we could have just put d's in all of the unused state entries. Sum-of-products excitation equations obtained from the minimal-cost maps (not shown) are as follows:

$$J1 = 1 \qquad K1 = 0$$
$$J2 = Q1 \cdot Q3' \cdot A' + Q3 \cdot A \qquad K2 = Q3' \cdot A \cdot B' + Q3 \cdot A' \cdot B'$$
$$J3 = A \qquad K3 = A'$$

The state encoding for the J-K circuit is the same as in the D circuit, so the output equation is the same, $Z = Q1 \cdot Q2$ for minimal risk, $Z = Q2$ for minimal cost.

A logic diagram corresponding to the minimal-cost equations is shown in Figure 7-56. This circuit has two more gates than the minimal-cost D circuit in Figure 7-54, so J-K flip-flops still didn't save us anything.

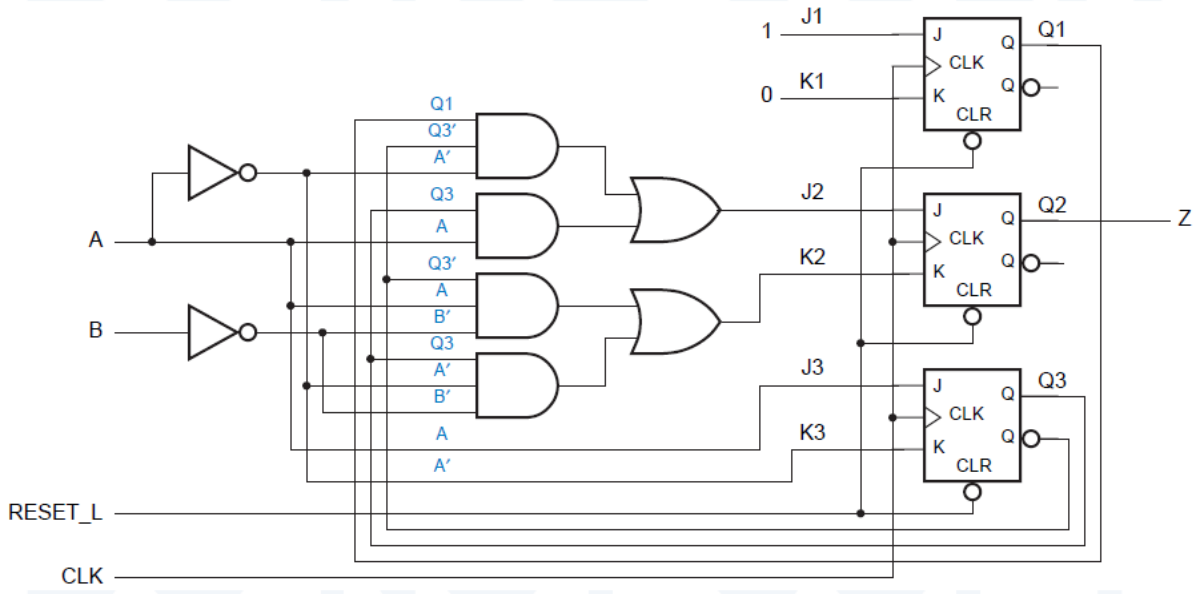


Figure 7-56 Logic diagram for example state machine using J-K flip-flops and minimal-cost excitation logic.

Ví dụ thiết kế sử dụng D flip-flop (bài tập về nhà)

Ví dụ 1:

Design a clocked synchronous state machine with two inputs, X and Y, and one output, Z. The output should be 1 if the number of 1 inputs on X and Y since reset is a multiple of 4, and 0 otherwise.

- Tại thời điểm xét nếu số bit 1s đếm tại X và Y là bội số của 4 thì giá trị output sẽ bằng 1 → sử dụng 4 trạng thái:
  - S0: trạng thái mà tại X và Y có 4n bits 1
  - S1: trạng thái mà tại X và Y có (4n+1) bits 1
  - S2: trạng thái mà tại X và Y có (4n+2) bits 1
  - S3: trạng thái mà tại X và Y có (4n+3) bits 1

Table 7-12 State and output table for 1s-counting machine.

		XY				
Meaning	***S***	***00***	***01***	***11***	***10***	***Z***
Got zero 1s (modulo 4)	S0	S0	S1	S2	S1	1
Got one 1 (modulo 4)	S1	S1	S2	S3	S2	0
Got two 1s (modulo 4)	S2	S2	S3	S0	S3	0
Got three 1s (modulo 4)	S3	S3	S0	S1	S0	0
		S*				

	XY				
Q1 Q2	00	01	11	10	Z
00	00	01	11	01	1
01	01	11	10	11	0
11	11	10	00	10	0
10	10	00	01	00	0
	Q1* Q2* or D1 D2				

Table 7-13
Transition/excitation and output table for 1s-counting machine.

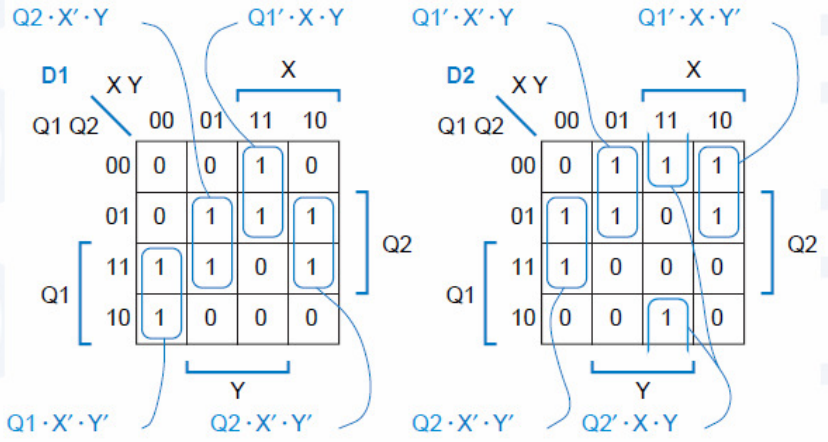


Figure 7-57
Excitation maps for D1 and D2 inputs in 1s-counting machine.

$$D1 = Q2 \cdot X' \cdot Y + Q1' \cdot X \cdot Y + Q1 \cdot X' \cdot Y' + Q2 \cdot X \cdot Y'$$
$$D2 = Q1' \cdot X' \cdot Y + Q1' \cdot X \cdot Y' + Q2 \cdot X' \cdot Y' + Q2' \cdot X \cdot Y$$
$$Z = Q1' \cdot Q2'$$