Principles Of
Digital Design

Sequential logic and design

Analysis

- *State-based (Moore)*
- *Input-based (Mealy)*

FSM definition

Synthesis

- *State minimization*
- *Encoding*
- *Optimization and timing*

Analysis of sequential logic

- **Excitation equations** are Boolean expressions of the flip-flop inputs.
- **Next-state equations** are Boolean expressions representing the next value of the flip- flop outputs.
- **Next-state table** (similar to next-state equations) gives the next value of flip-flop outputs for each input value and state of flip-flops.
- **Analysis** of a sequential circuit is a procedure that produces the next-state table, state diagram and timing diagram from the logic schematic of the circuit.

  The analysis gives the answer to the following questions:

  (a) What is the next state?

  (b) What is the output?

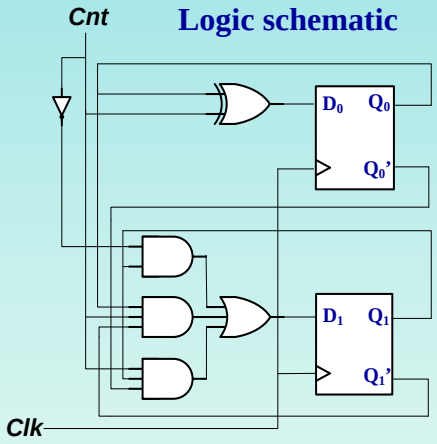
  (c) What is the function of the schematic?

Analysis of a sequential circuit

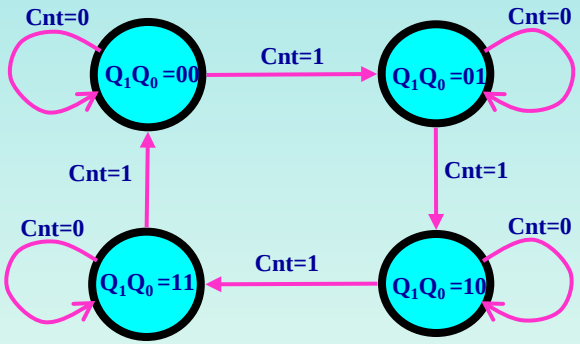
Example: Modulo-4 counter

Problem: Derive the state table & state diagram for the circuit given below.

Logic schematic



State diagram



State table

PRESENT STATE Q_1Q_0	NEXT STATE $Q_1(next)\ Q_0(next)$	
	Cnt=0	Cnt=1
0 0	0 0	0 1
0 1	0 1	1 0
1 0	1 0	1 1
1 1	1 1	0 0

$D_0 = Cnt \oplus Q_0 = Cnt' Q_0 + Cnt\ Q_0'$

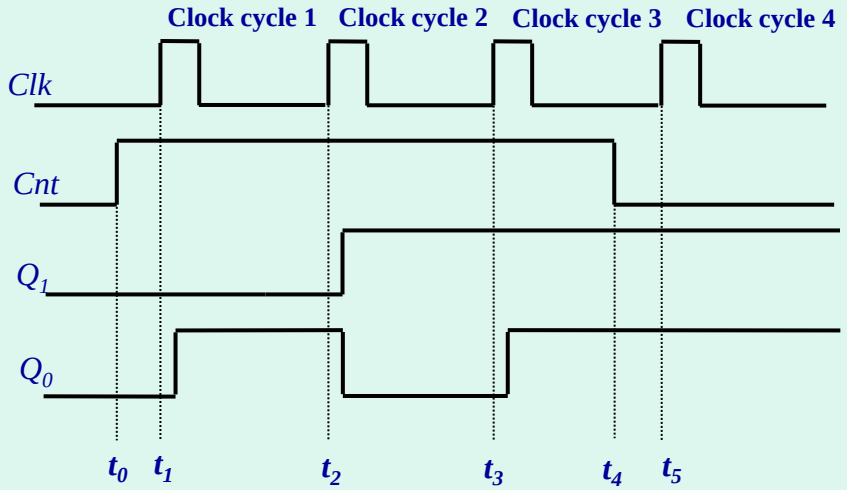
$D_1 = Cnt' Q_1 + Cnt\ Q_1' Q_0 + Cnt\ Q_1 Q_0'$

Excitation equation

$Q_0(next) = D_0 = Cnt' Q_0 + Cnt\ Q_0'$

$Q_1(next) = D_1 = Cnt' Q_1 + Cnt\ Q_1' Q_0 + Cnt\ Q_1 Q_0'$

Next-state equation

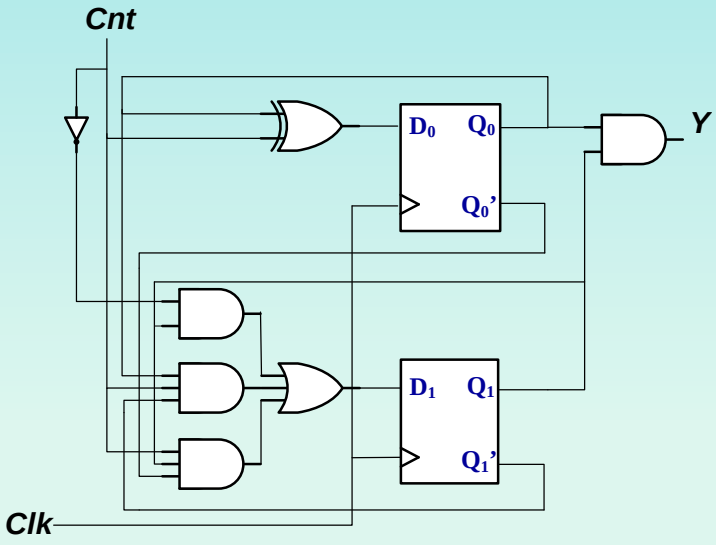


Timing diagram

Analysis of a modulo-4 counter

Example: Modulo-4 counter (state-based, Moore-type)

Problem: Derive the state, output tables and the state diagram for the circuit below.



Logic schematic

PRESENT STATE Q_1Q_0	NEXT STATE $Q_1(next)\ Q_0(next)$		OUTPUTS Y
	Cnt=0	Cnt=1	
0 0	0 0	0 1	0
0 1	0 1	1 0	0
1 0	1 0	1 1	0
1 1	1 1	0 0	1

State and output table

$$D_0 = Cnt \oplus Q_0 = Cnt' Q_0 + Cnt\, Q_0'$$

$$D_1 = Cnt' Q_1 + Cnt\, Q_1' Q_0 + Cnt\, Q_1 Q_0'$$

Excitation equation

$$Q_0(next) = D_0 = Cnt' Q_0 + Cnt\, Q_0'$$

$$Q_1(next) = D_1 = Cnt' Q_1 + Cnt\, Q_1' Q_0 + Cnt\, Q_1 Q_0'$$

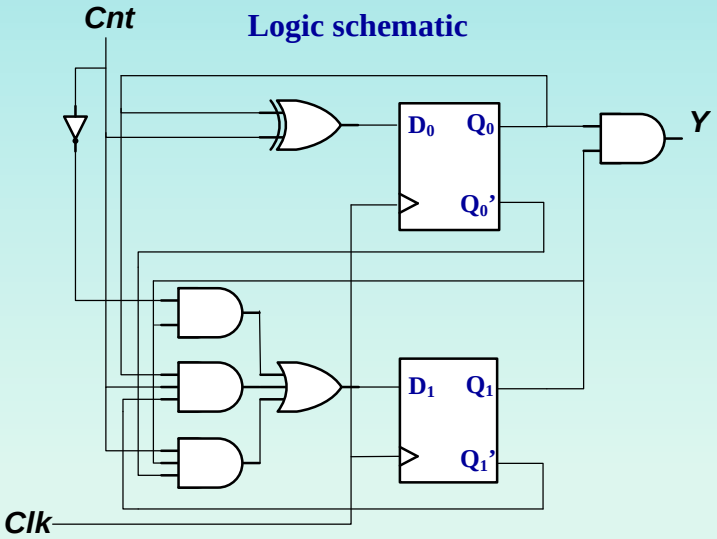
$$Y = Q_0\, Q_1$$

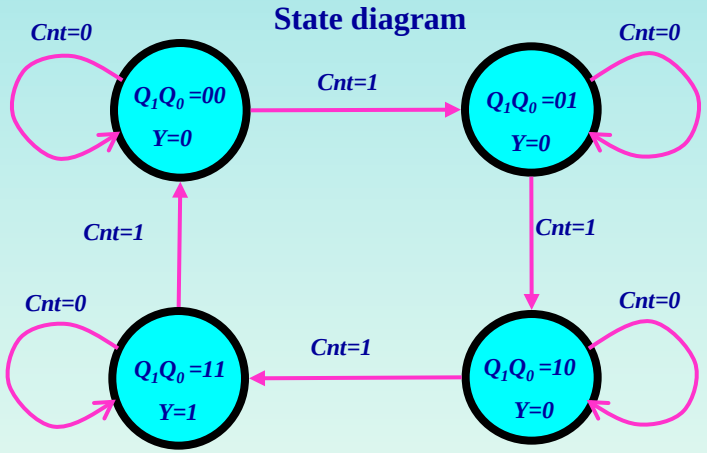
Next-state and output equation

Analysis of a modulo-4 counter

Example: Modulo-4 counter (state-based, Moore-type)

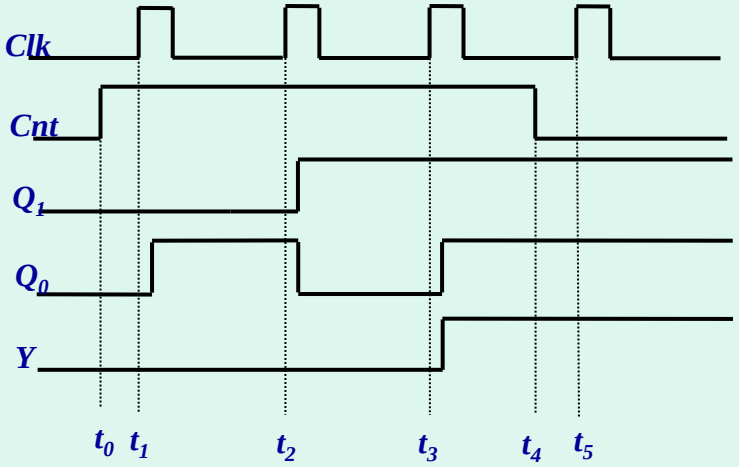
Problem: Derive the state, output tables and the state diagram for the circuit below.





PRESENT STATE Q_1Q_0	NEXT STATE $Q_1(next)\ Q_0(next)$		OUTPUTS Y
	Cnt=0	Cnt=1	
0 0	0 0	0 1	0
0 1	0 1	1 0	0
1 0	1 0	1 1	0
1 1	1 1	0 0	1

State and output table

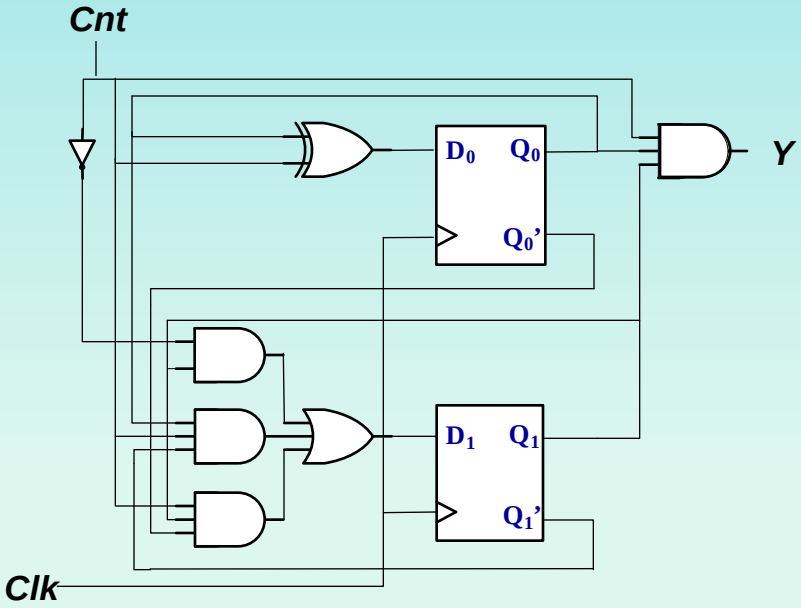


Timing diagram

Analysis of a modulo-4 counter

Example: Modulo-4 counter (input-based, Mealy-type)

Problem: Derive the state/output table and the state diagram for the circuit below.



Logic schematic

PRESENT STATE Q_1Q_0	NEXT STATE /OUTPUTS $Q_1(next)\ Q_0(next)/Y$	
	Cnt=0	Cnt=1
0 0	0 0 / 0	0 1 / 0
0 1	0 1 / 0	1 0 / 0
1 0	1 0 / 0	1 1 / 0
1 1	1 1 / 0	0 0 / 1

State and output table

$$D_0 = Cnt \oplus Q_0 = Cnt'\,Q_0 + Cnt\,Q_0'$$

$$D_1 = Cnt'\,Q_1 + Cnt\,Q_1'Q_0 + Cnt\,Q_1Q_0'$$

Excitation equation

$$Q_0(next) = D_0 = Cnt'\,Q_0 + Cnt\,Q_0'$$

$$Q_1(next) = D_1 = Cnt'\,Q_1 + Cnt\,Q_1'Q_0 + Cnt\,Q_1Q_0'$$

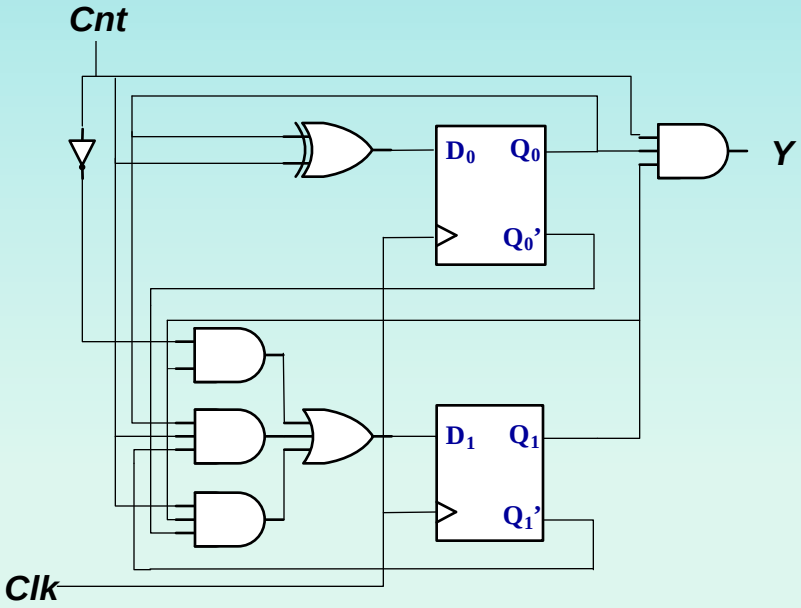
$$Y = Cnt\,Q_0\,Q_1$$

Next-state and output equation

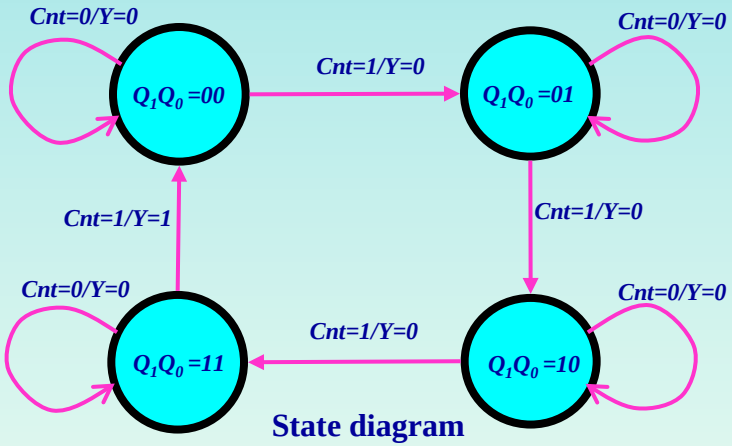
Analysis of a modulo-4 counter

Example: Modulo-4 counter (input-based, Mealy-type)

Problem: Derive the state/output table and the state diagram for the circuit below.



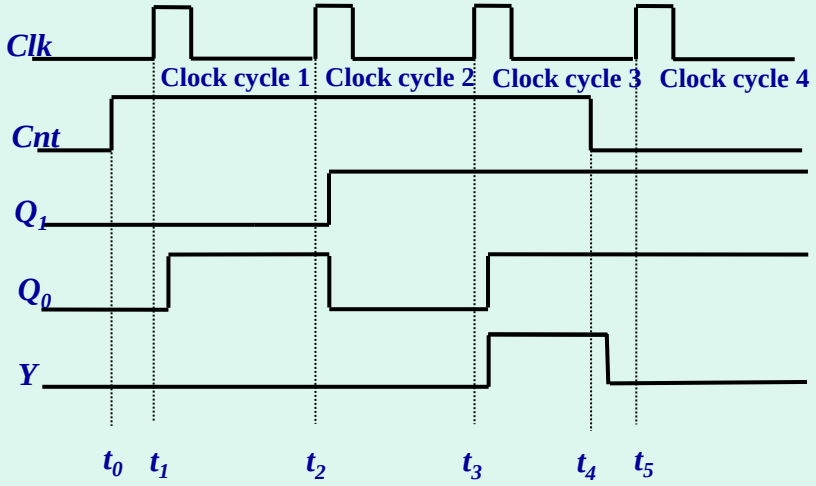
Logic schematic



State diagram

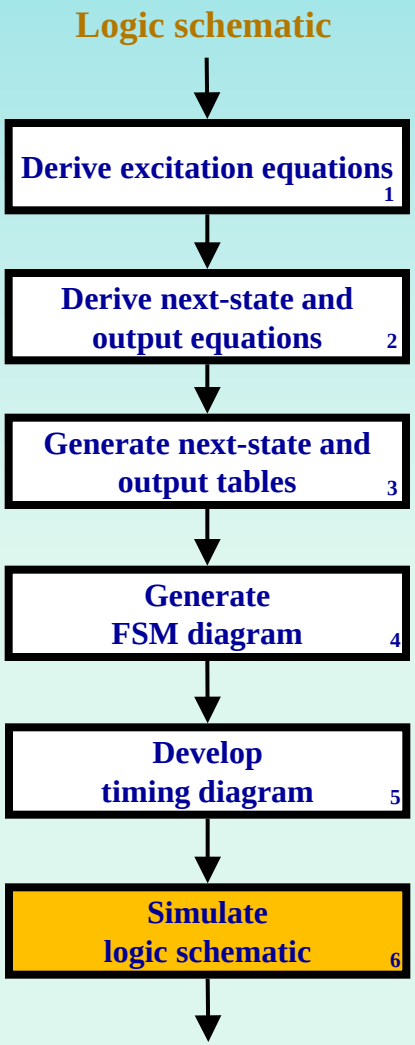
PRESENT STATE Q_1Q_0	NEXT STATE /OUTPUTS $Q_1(next)\ Q_0(next)/Y$	
	Cnt=0	Cnt=1
0 0	0 0 / 0	0 1 / 0
0 1	0 1 / 0	1 0 / 0
1 0	1 0 / 0	1 1 / 0
1 1	1 1 / 0	0 0 / 1

State and output table

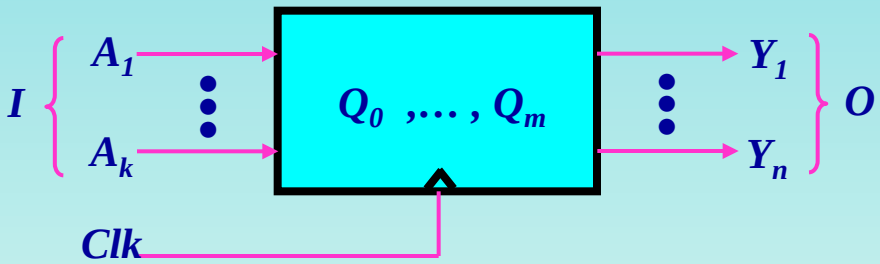


Timing diagram

Analysis procedure for sequential circuits



Finite-state machine model



The FSM can be defined abstractly as the quintuple

$$< S, I, O, f, h >$$

where *S*, *I*, and *O* represent a set of states, set of inputs and a set of outputs, respectively, and *f* and *h* represent the next-state and the output functions, that is

$$f : S \times I \longrightarrow S$$

$$h : S \times I \longrightarrow O \text{ (Mealy-type)}$$

$$S \longrightarrow O \text{ (Moore-type)}$$

where $S = Q_1 \times Q_2 \times \ldots \times Q_m$,

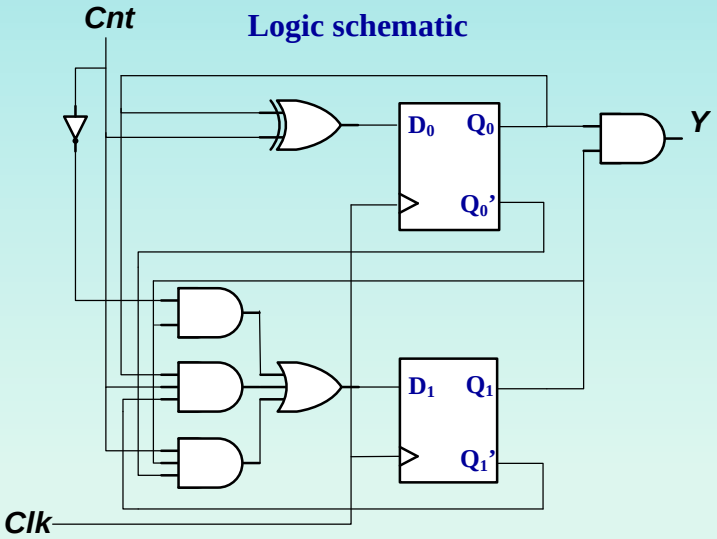
$I = A_1 \times A_2 \times \ldots \times A_k$,

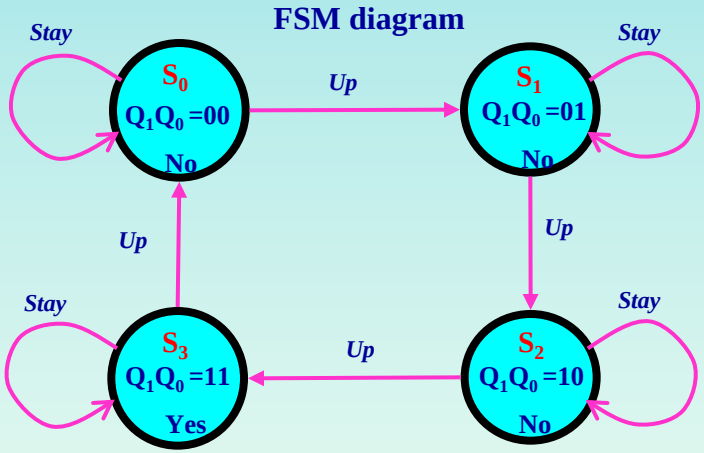
$O = Y_1 \times Y_2 \times \ldots \times Y_n$,

FSM definition of modulo-4 counter

Example: Modulo-4 counter (state-based, Moore-type)

Problem: Derive the state, output tables and the FSM diagram for the circuit below.





$S = \{S_0, S_1, S_2, S_3\}$ $I = \{Up, Stay\}$ $O = \{Yes, No\}$

PRESENT STATE Q_1Q_0	NEXT STATE $Q_1(next)\ Q_0(next)$		OUTPUTS Y
	Cnt=0	Cnt=1	
0 0	0 0	0 1	0
0 1	0 1	1 0	0
1 0	1 0	1 1	0
1 1	1 1	0 0	1

State and output table

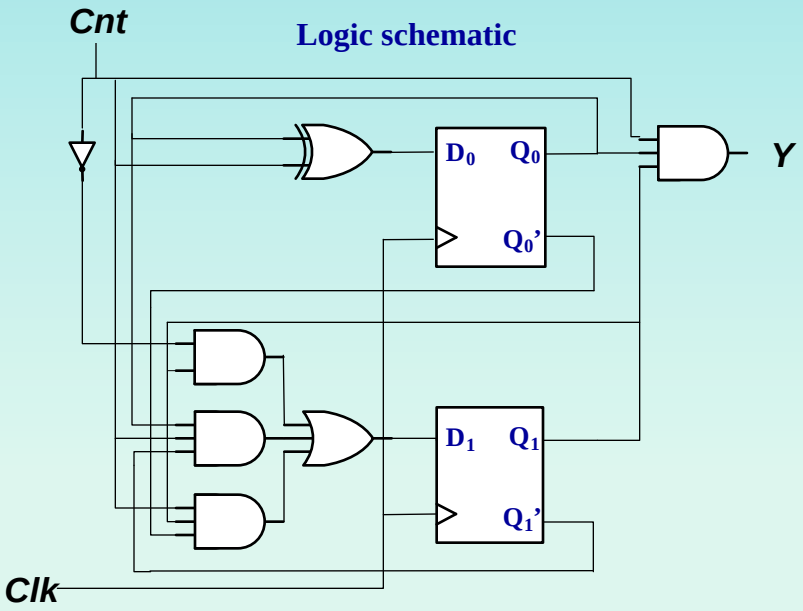
PRESENT STATE	NEXT STATE (SxI → S)		OUTPUTS (S → O)
	Stay	Up	
S_0	S_0	S_1	No
S_1	S_1	S_2	No
S_2	S_2	S_3	No
S_3	S_3	S_0	Yes

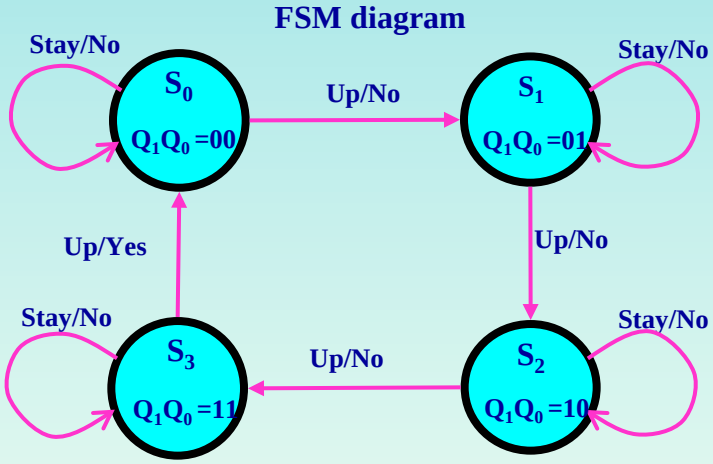
FSM table

FSM definition of modulo-4 counter

Example: Modulo-4 counter (input-based, Mealy-type)

Problem: Derive the state/output table and the FSM diagram for the circuit below.





$S = \{S_0, S_1, S_2, S_3\}$ $I = \{Up, Stay\}$ $O = \{Yes, No\}$

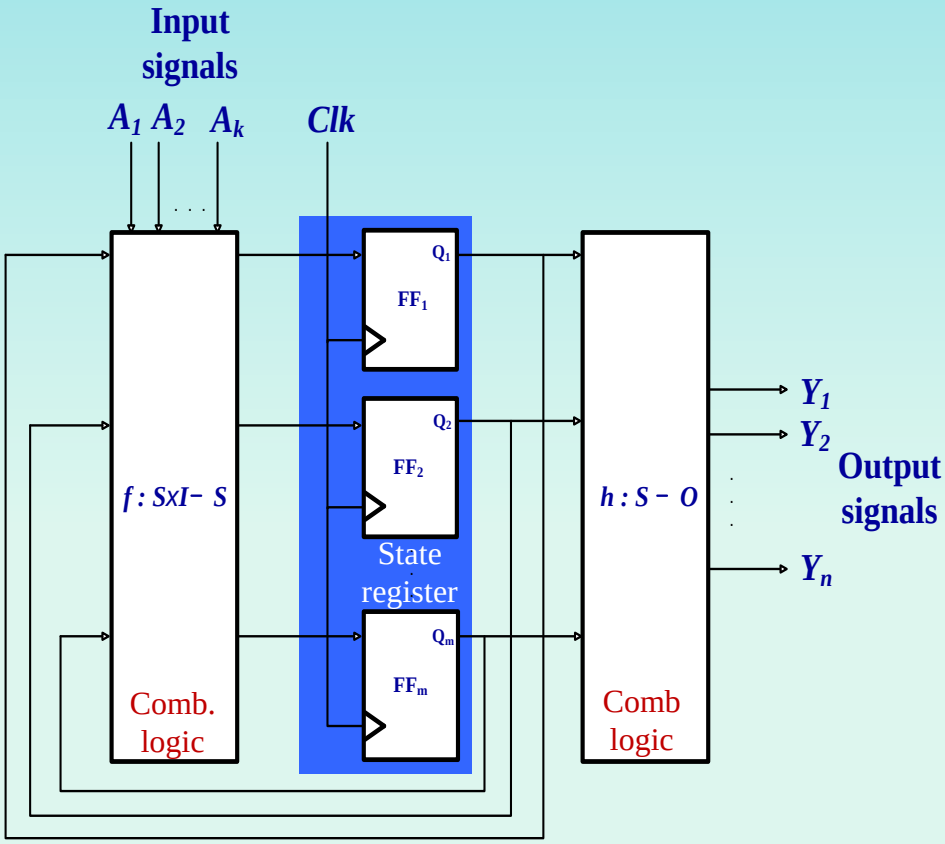
PRESENT STATE Q_1Q_0	*NEXT STATE /OUTPUTS* $Q_1(next)\ Q_0(next)/Y$	
	Cnt=0	Cnt=1
0 0	0 0 / 0	0 1 / 0
0 1	0 1 / 0	1 0 / 0
1 0	1 0 / 0	1 1 / 0
1 1	1 1 / 0	0 0 / 1

State and output table

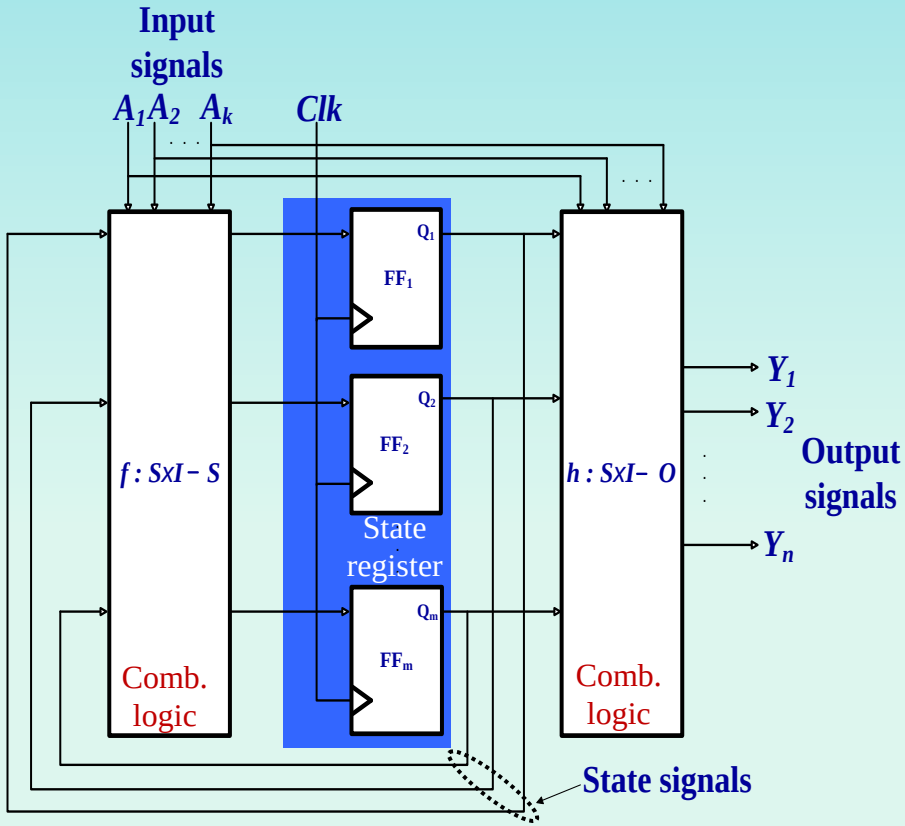
PRESENT STATE	NEXT STATE(SxI→S)/ OUTPUT(SxI→ O)	
	Stay	Up
S_0	S_0 /No	S_1 /No
S_1	S_1 /No	S_2 /No
S_2	S_2 /No	S_3 /No
S_3	S_3 /No	S_0 /Yes

FSM table

Finite-state-machine implementations

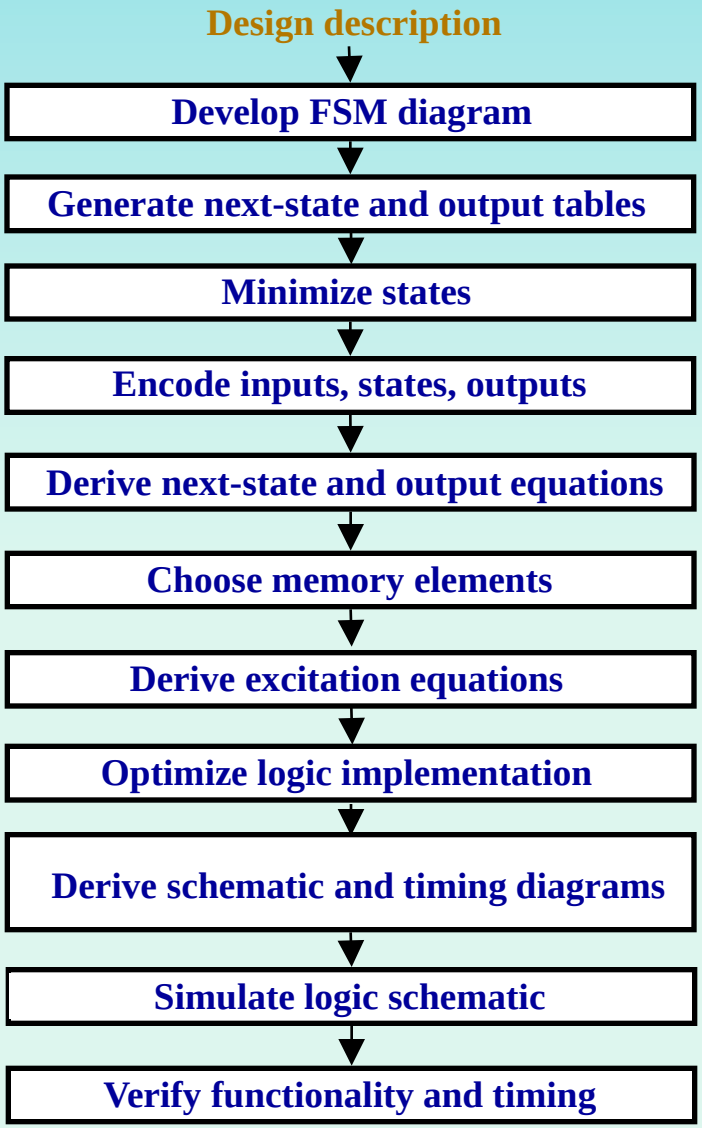


State-based (Moore)



Input-based (Mealy)

Synthesis procedure for sequential logic

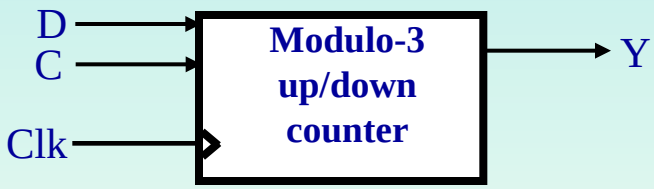


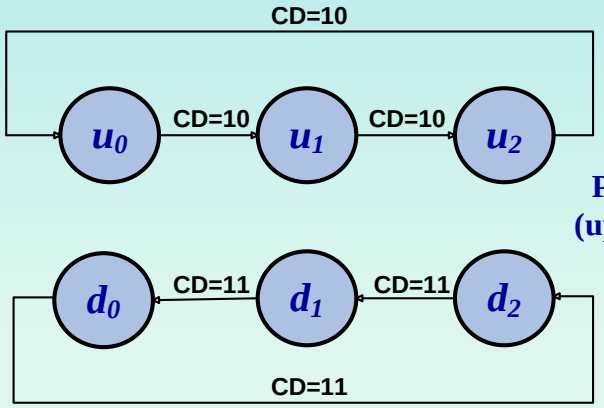
State diagram for a modulo-3 up/down counter

Example: Modulo-3 up-down counter

Problem: Derive the FSM diagram for an up-down, modulo-3 counter. The counter has two inputs: count enable (C) and count direction (D). When C=1, the counter will count in the direction specified by D, and it will stop counting when C=0. the counter will count up when D=0 and down when D=1. The counter has one output Y which will be asserted when the counter reaches 2 while counting up, or when it reaches 0 while counting down.

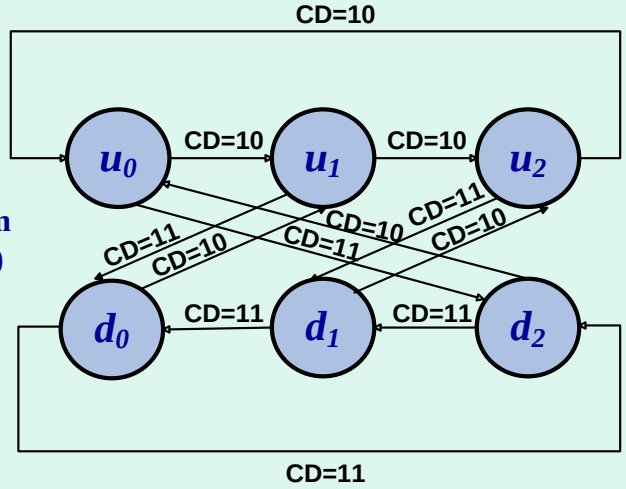
Counter symbol

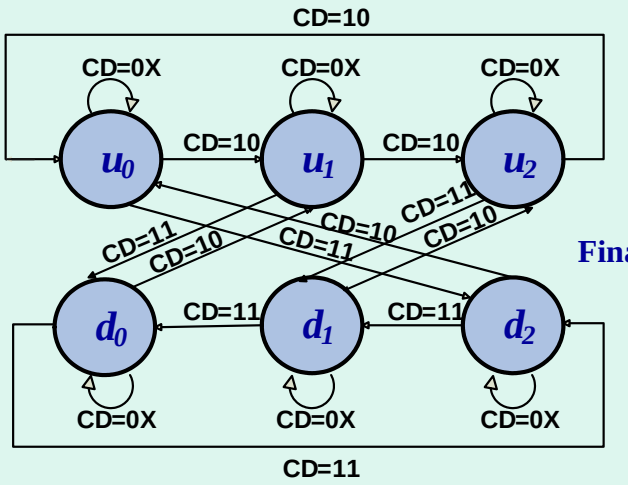




Partial state diagram (up and down counting)

Partial state diagram (changing direction)





Final state diagram

State minimization

- State minimization reduces the number of states, and therefore, number of flip-flops needed to implemented the circuit.

- State minimization is based on the concept of behavioral equivalence which states that two states are equivalent if they produce the same sequence of output symbols for every sequence of input symbols.

- More formally, two states, s_j and s_k in an FSM are said to be equivalent, $s_j \equiv s_k$, iff the following two conditions are true.

  Condition 1: Both states s_j *and* s_k produce the same output symbol for every input symbol i: that is, $h(s_j, i) = h(s_k, i)$;

  Condition 2: Both states *have equivalent next sates* for every input symbol i : that is, $f(s_j, i) = f(s_k, i)$;

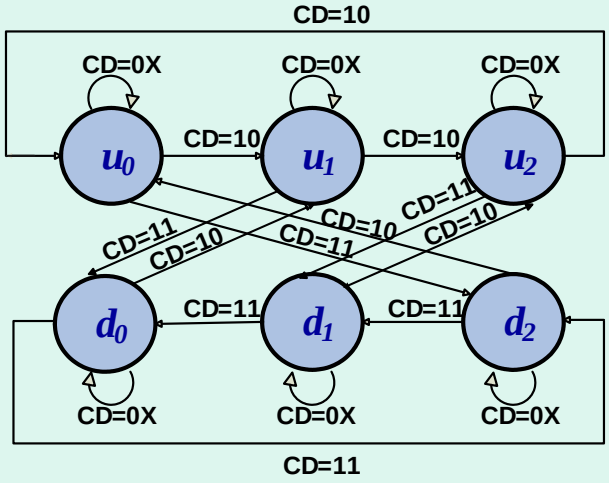
- Minimization procedure:
  1. partition states into equivalence classes
  2. construct new FSM with one state for each equivalence class

State reduction for modulo-3 counter

Example: State reduction
Problem: Derive the minimal-state FSM for the modulo-3 counter.

PRESENT STATE	NEXT STATE / OUTPUT		
	CD=0X	CD=10	CD=11
u_0	u_0 / 0	u_1 / 0	d_2 / 1
u_1	u_1 / 0	u_2 / 0	d_0 / 0
u_2	u_2 / 0	u_0 / 1	d_1 / 0
d_0	d_0 / 0	u_1 / 0	d_2 / 1
d_1	d_1 / 0	u_2 / 0	d_0 / 0
d_2	d_2 / 0	u_0 / 1	d_1 / 0

Initial next-state/output table



$(u_0, u_1, u_2, d_0, d_1, d_2)$

Partition into arrays with the same output

CD = 0X	0	0	0	0	0	0
10	0	0	1	0	0	1
11	1	0	0	1	0	0

Output values

001 000 010

$G_0 = (u_0, d_0)$ $G_1 = (u_1, d_1)$ $G_2 = (u_2, d_2)$

Partition into arrays with the same next state

CD = 0X	G_0G_0	G_1G_1	G_2G_2
10	G_1G_1	G_2G_2	G_0G_0
11	G_2G_2	G_0G_0	G_1G_1

Next states

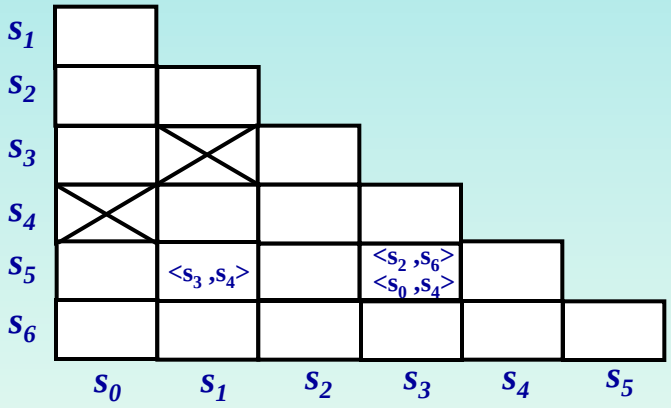
Partitioning into equivalence classes

PRESENT STATE	NEXT STATE / OUTPUT		
	CD=0X	CD=10	CD=11
G_0	G_0 / 0	G_1 / 0	G_2 / 1
G_1	G_1 / 0	G_2 / 0	G_0 / 0
G_2	G_2 / 0	G_0 / 1	G_1 / 0

Final next-state/output table

State reduction with implication table

Example: State reductions with implication table.
Problem: Find the minimal number of states for the FSM specified by the table below.



Implication table

PRESENT STATE	NEXT STATE		
	CD=0X	CD=10	CD=11
u_0	u_0 / 0	u_1 / 0	d_2 / 1
u_1	u_0 / 0	u_2 / 0	d_0 / 0
u_2	u_2 / 0	u_0 / 1	d_1 / 0
d_0	d_0 / 0	u_1 / 0	d_2 / 1
d_1	d_2 / 0	u_2 / 0	d_0 / 0
d_2	d_2 / 0	u_0 / 1	d_1 / 0

Next-state and output table

Step 1: Enter x for any pair of states that differ in output values

Step2: Enter implied equivalent states for every pair of states

Step3: Enter x for the non equivalent next-state pair

Step4: Form equivalence classes using transitivity : $s_1 \equiv s_2$ & $s_2 \equiv s_3$ => $s_1 \equiv s_3$

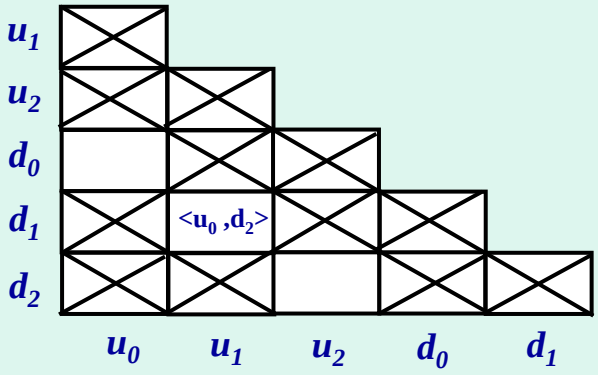
Equivalence classes:

$<u_0, d_0>$

$<u_1>$

$<d_1>$

$<u_2, d_2>$



Implication table for the table above

State encoding

ENCODING NUMBER	s_0	s_1	s_2	s_3
1	00	01	10	11
2	00	01	11	10
3	00	10	01	11
4	00	10	11	01
5	00	11	01	10
6	00	11	10	01
7	01	00	10	11
8	01	00	11	10
9	01	10	00	11
10	01	10	11	00
11	01	11	00	10
12	01	11	10	00
13	10	00	01	11
14	10	00	11	01
15	10	01	00	11
16	10	01	11	00
17	10	11	00	01
18	10	11	01	00
19	11	00	01	10
20	11	00	10	01
21	11	01	00	10
22	11	01	10	00
23	11	10	00	01
24	11	10	01	00

24 encodings of four states

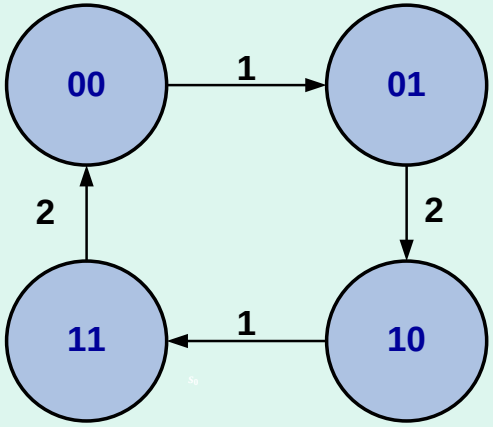
- **The cost and delay of FSM implementation depends on encoding of symbolic states.**
- **For example, four states can be encoded in 4!=24 different ways.**
- **There are more than n! different encodings for n states.**
- **Exploration of all encodings is impossible**

  Thus, we use different heuristics such as

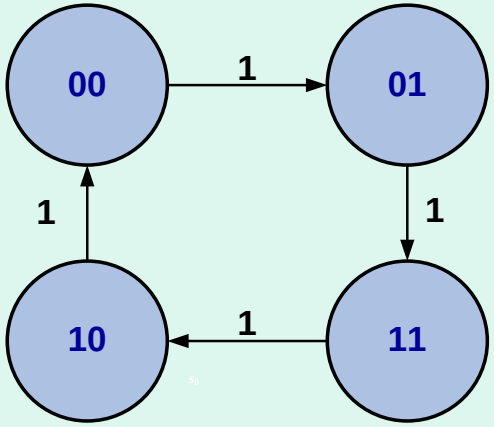
  - **minimum-bit change**
  - **prioritized adjacency**
  - **one-hot encoding**
  - **others**

Minimum-bit change

- Minimum-bit change strategy assigns codes to states so that the total number of bit changes for all state transitions is minimized.
- In other words, if every arc in the state diagram has a weight that is equal to the number of bits by which the source and destination encodings differ, then the optimal encoding would be the one that minimizes the sum of all these weights.
- Example: Two different encodings for modulo-4 binary counter.



Straightforward encoding



Minimum-bit-change encoding

Encoding example

Encoding A = Minimum-bit change
Encoding B = Simplified output logic
Encoding C = Hot-one encoding

Example: Comparison of state encodings for modulo-3 counter.

Problem: For the modulo-3 counter find the encoding that minimizes the cost and delay.

PRESENT STATE	NEXT STATE / OUTPUT		
	CD=0X	CD=10	CD=11
s_0	s_0 / 0	s_1 / 0	s_2 / 1
s_1	s_1 / 0	s_2 / 0	s_0 / 0
s_2	s_2 / 0	s_0 / 1	s_1 / 0

Final next-state/output table

STATE	ENCODING A Q_1Q_0	ENCODING B Q_1Q_0	ENCODING C $Q_2Q_1Q_0$
s_0	0 0	0 1	0 0 1
s_1	0 1	0 0	0 1 0
s_2	1 0	1 0	1 0 0

Possible state encodings for modulo-3 counter

DELAYS & COST	ENCODING A	ENCODING B	ENCODING C
C,D to Clk	8.0	8.0	7.6
Clk to Y	9.6	9.2	9.2
C,D to Y	7.6	7.2	7.2
Clk to Clk	10.0	10.0	9.6
Cost	92	90	112

$Q_1(next) = Q_1C'+Q_0CD'+Q_1'Q_0'CD$

$Q_0(next) = Q_0C'+Q_1CD+Q_1'Q_0'CD'$

$Y = Q_1CD'+Q_1'Q_0'CD$

$Q_1(next) = Q_1C'+Q_0CD+Q_1'Q_0'C'D'$

$Q_0(next) = Q_0C'+Q_1CD'+Q_1'Q_0'CD$

$Y = Q_1CD+Q_1CD'$

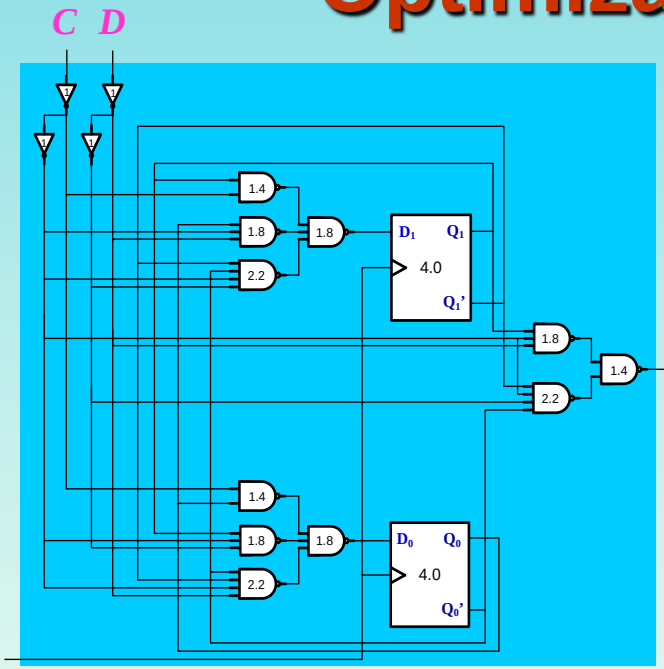
$Q_2(next) = Q_2C'+Q_0CD+Q_1CD'$

$Q_1(next) = Q_1C'+Q_2CD+Q_0CD'$

$Q_0(next) = Q_0C'+Q_2CD'+Q_1CD'$

$Y = Q_0CD+Q_2CD'$

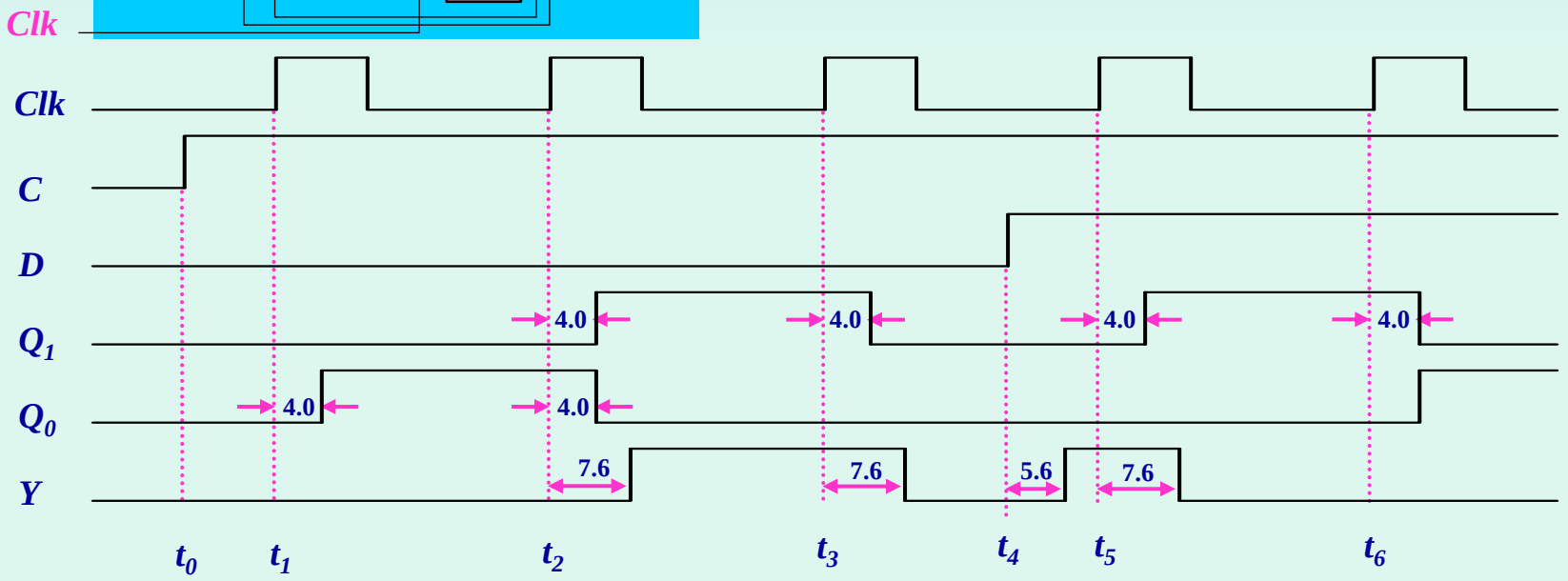
Optimization and timing



Logic schematic

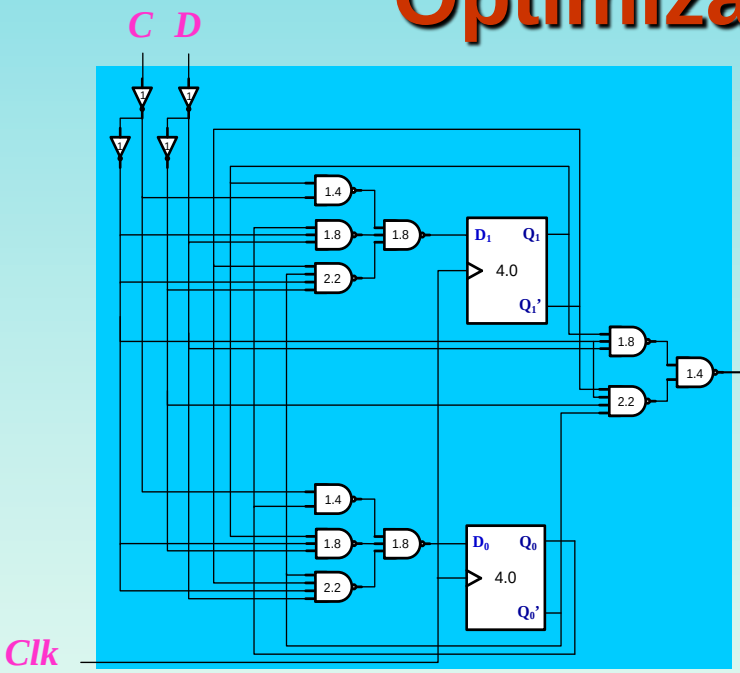
Input-output delays	
C, D to Clk	6.0
Clk to Y	7.6
C,D to Y	5.6
Clk to Clk	8.0

Delay table



Timing diagram

Optimization and timing

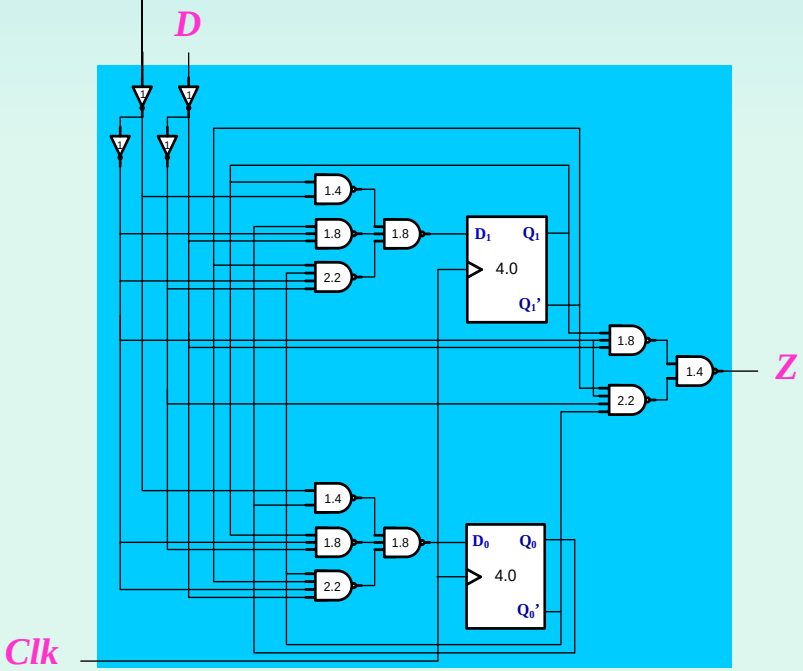


Input-output delays	
C, D to Clk	6.0
Clk to Y	7.6
C,D to Y	5.6
Clk to Clk	8.0

Delay table

Input-output delays	
C, D to Clk	6.0
Clk to Y	7.6
C,D to Z	11.2
Clk to Clk	13.6

Delay table



Summary

- We described procedures for sequential logic
  - Analysis
  - Synthesis with
    - FSM capture
    - state minimization
    - state encoding
    - optimization and timing
- We defined the FSM model and synthesis from FSM