Lecture 15 Sequential Circuit Design

- Example: Code converter
  - Put it all together using what we learn previously.
  - Inputs are received according to time t_0, t_1, t_2, t_3.

Table 16–1

X Input (BCD)				*Z* Output (excess-3)			
t_3	t_2	t_1	t_0	t_3	t_2	t_1	t_0
0	0	0	0	0	0	1	1
0	0	0	1	0	1	0	0
0	0	1	0	0	1	0	1
0	0	1	1	0	1	1	0
0	1	0	0	0	1	1	1
0	1	0	1	1	0	0	0
0	1	1	0	1	0	0	1
0	1	1	1	1	0	1	0
1	0	0	0	1	0	1	1
1	0	0	1	1	1	0	0

Code Converter

- State table construction: least significant bit is received first.
  - At t_1, starting with B = 0, if the network receives X = 0, we call state D. This means 00 is received. We should give Z = 1 for input X = 0. (Check Table 16-1 for all 00s, we have Z=1 at t_1).

For input = 0, output = 1 (Check T 16-1 at t_0)

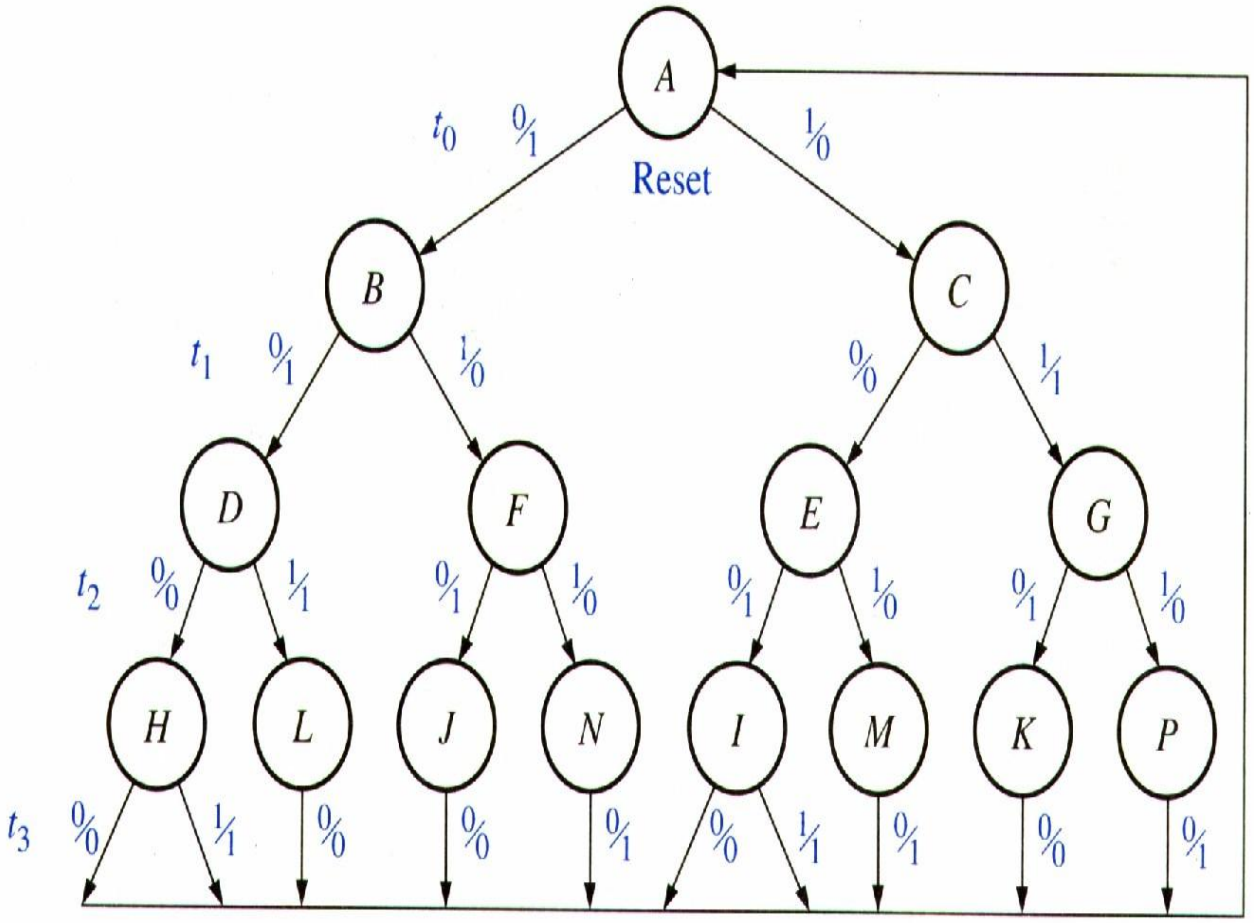
Table 16–2
State Table for Code Converter

Time	Input Sequence Received (Least Significant Bit First)	Present State	Next State $X=0$	Next State 1	Present Output (Z) $X=0$	Present Output (Z) 1
t_0	reset	A	B	C	1	0
t_1	0	B	D	F	1	0
	1	C	E	G	0	1
t_2	00	D	H	L	0	1
	01	E	I	M	1	0
	10	F	J	N	1	0
	11	G	K	P	1	0
t_3	000	H	A	A	0	1
	001	I	A	A	0	1
	010	J	A	–	0	–
	011	K	A	–	0	–
	100	L	A	–	0	–
	101	M	A	–	1	–
	110	N	A	–	1	–
	111	P	A	–	1	–

Code Converter (cont.)

– Or construct the state graph first.

- Starting at t_3, path 0000 has outputs 0011 as given in Table 16-1.



Code Converter (cont.)

– State reduction: find the equivalent states.

Table 16–2
State Table for Code Converter

Time	Input Sequence Received (Least Significant Bit First)	Present State	Next State $X=0$	Next State 1	Present Output (Z) $X=0$	Present Output (Z) 1
t_0	reset	A	B	C	1	0
t_1	0	B	D	F	1	0
	1	C	E	G	0	1
t_2	00	D	H	L	0	1
	01	E	I	M	1	0
	10	F	J	N	1	0
	11	G	K	P	1	0
t_3	000	H	A	A	0	1
	001	I	A	A	0	1
	010	J	A	–	0	–
	011	K	A	–	0	–
	100	L	A	–	0	–
	101	M	A	–	1	–
	110	N	A	–	1	–
	111	P	A	–	1	–

Table 16–3
Reduced State Table for Code Converter

Time	Present State	Next State $X=0$	Next State 1	Present Output (Z) $X=0$	Present Output (Z) 1
t_0	A	B	C	1	0
t_1	B	D	E	1	0
	C	E	E	0	1
t_2	D	H	H	0	1
	E	H	M	1	0
t_3	H	A	A	0	1
	M	A	–	1	–

Code Converter (cont.)

- State assignment
- State B and C, D and E, H and M should be given adjacent assignments.
- State A,B,E,M, state C, D, and H should be given adjacent assignments. G3. (Output)

Q_2Q_3 \ Q_1	0	1
00	A	B
01		C
11	H	D
10	M	E

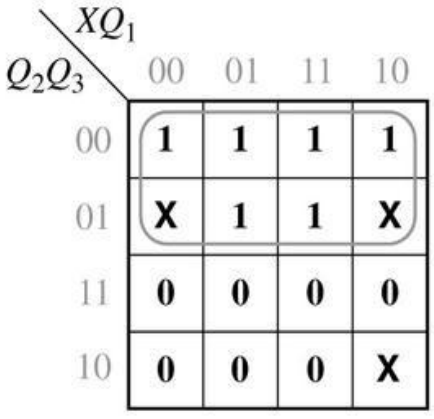
(a) Assignment map

		$Q_1^+Q_2^+Q_3^+$		Z	
	$Q_1Q_2Q_3$	$X=0$	$X=1$	$X=0$	$X=1$
A	000	100	101	1	0
B	100	111	110	1	0
C	101	110	110	0	1
D	111	011	011	0	1
E	110	011	010	1	0
H	011	000	000	0	1
M	010	000	xxx	1	x
–	001	xxx	xxx	x	x

(b) Transition table

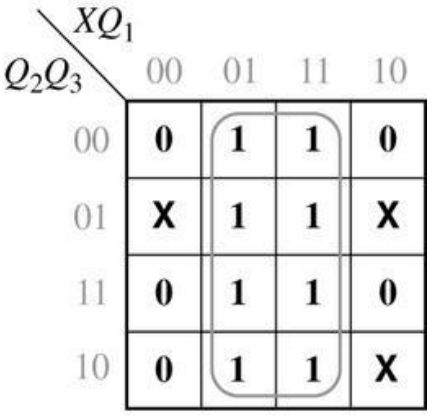
Code Converter (cont.)

– Find the Q^+ map, and then the FF input equations.



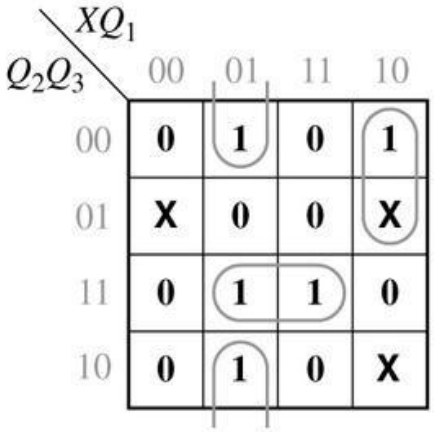
Q_2Q_3 \ XQ_1	00	01	11	10
00	1	1	1	1
01	X	1	1	X
11	0	0	0	0
10	0	0	0	X

$D_1 = Q_1^+ = Q_2'$



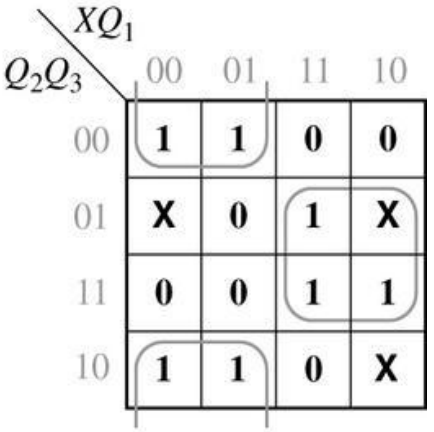
Q_2Q_3 \ XQ_1	00	01	11	10
00	0	1	1	0
01	X	1	1	X
11	0	1	1	0
10	0	1	1	X

$D_2 = Q_2^+ = Q_1$



Q_2Q_3 \ XQ_1	00	01	11	10
00	0	1	0	1
01	X	0	0	X
11	0	1	1	0
10	0	1	0	X

$D_3 = Q_3^+ = Q_1Q_2Q_3 + X'Q_1Q_3' + XQ_1'Q_2'$

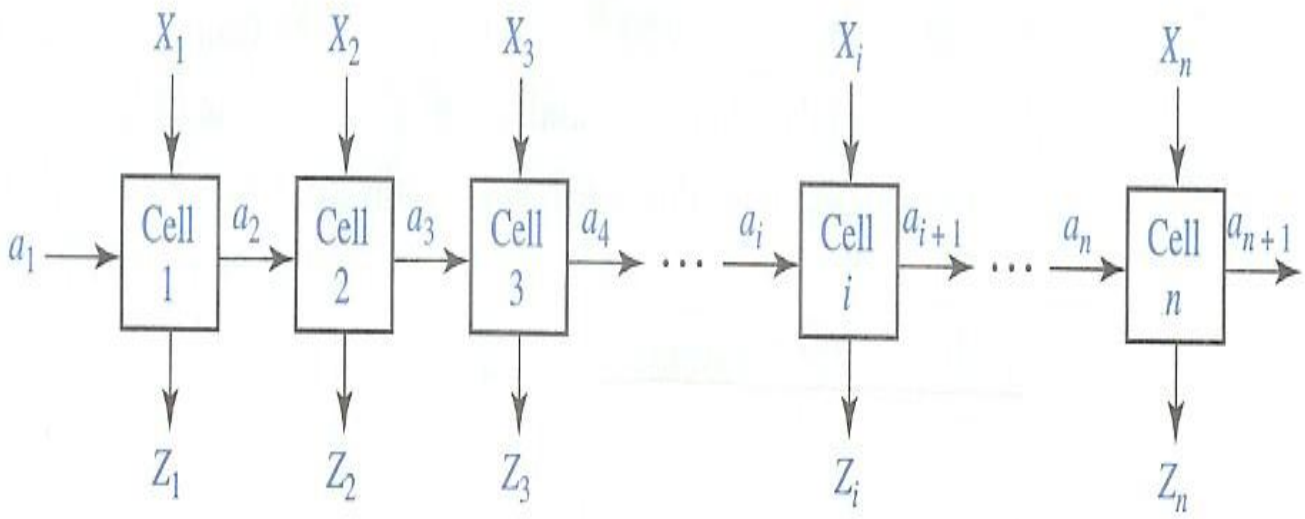


Q_2Q_3 \ XQ_1	00	01	11	10
00	1	1	0	0
01	X	0	1	X
11	0	0	1	1
10	1	1	0	X

$Z = X'Q_3' + XQ_3$

Iterative Circuits

- Same operation in each cell.
- Parallel inputs/parallel outputs
- Combinational cells
  - Primary inputs X_i
  - Primary output Z_i
  - Secondary input a_i
  - Secondary output a_{i+1}



A Comparator

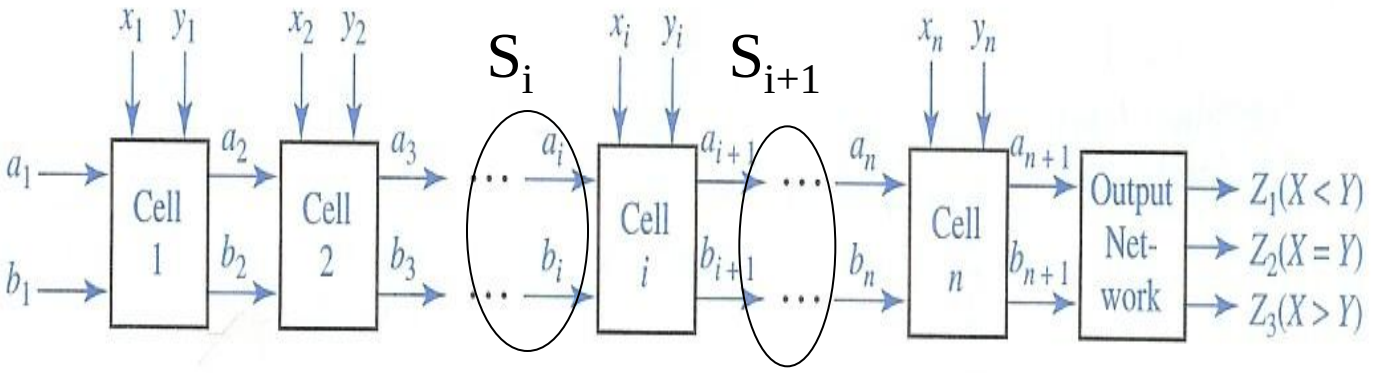
- Compare two binary numbers
  - $X = x_1x_2\ldots x_n$ and $Y = y_1y_2y_3\ldots y_n$
    - The left bit is the most significant bit.
  - Comparison proceeds from left to right.
  - To the left of cell ***i, either X=Y, X>Y, or X<Y***

    These input conditions called S_0, S_1, S_2

    S_{i+1} is output state at the right of cell i with the input x_i y_i and the input state at the left of the cell (S_i)

TABLE 16-4
State Table for Comparator

	S_i	S_{i+1} x_iy_i = 00	01	11	10	$Z_1Z_2Z_3$
$X = Y$	S_0	S_0	S_2	S_0	S_1	0 1 0
$X > Y$	S_1	S_1	S_1	S_1	S_1	0 0 1
$X < Y$	S_2	S_2	S_2	S_2	S_2	1 0 0

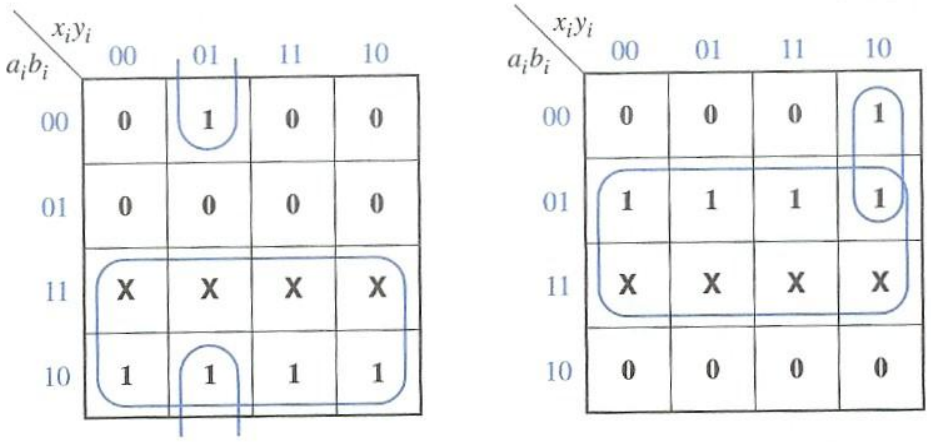


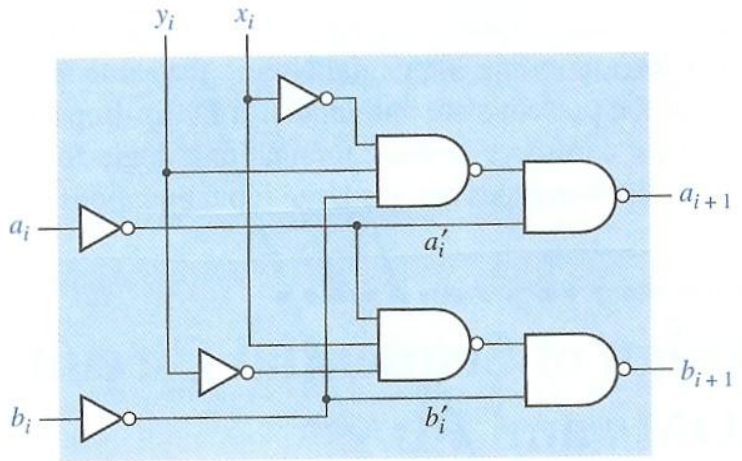
A Cell in the Comparator

TABLE 16-5
Transition Table for Comparator

a_ib_i	$a_{i+1}b_{i+1}$ $x_iy_i = 00$	01	11	10	$Z_1Z_2Z_3$
0 0	00	10	00	01	0 1 0
0 1	01	01	01	01	0 0 1
1 0	10	10	10	10	1 0 0

FIGURE 16-7
Typical Cell for Comparator





End Cells in the Comparator

- For the left end cell, $a_1 = b_1 = 0$. Can be used to simplify the equation for this cell.
- Output Z_1 (X < Y) = 1 ($S_2 = 10 = a_{n+1}\, b_{n+1}$), 11 is not used.

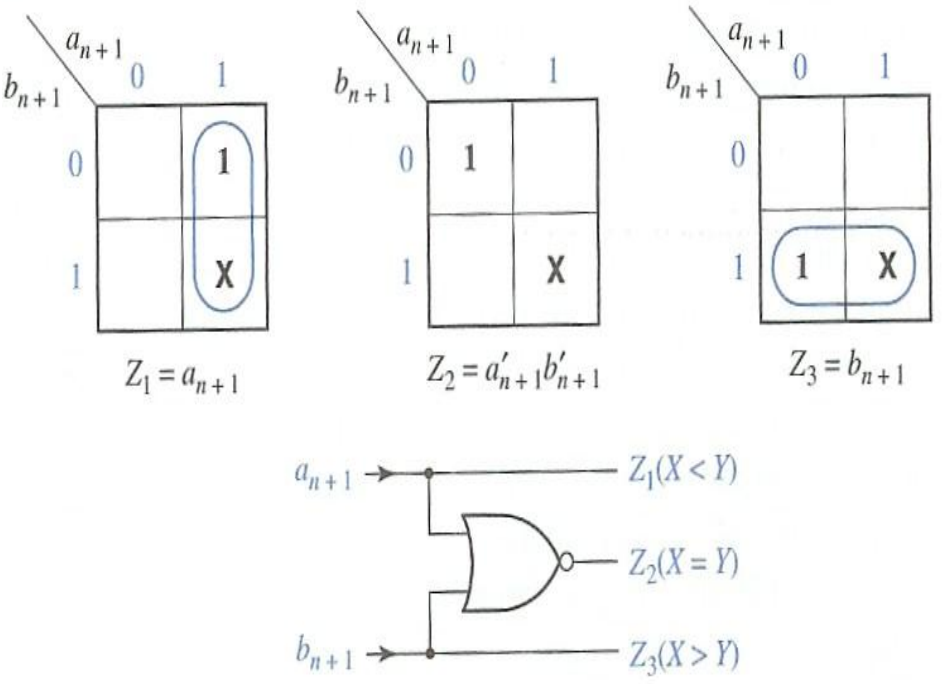
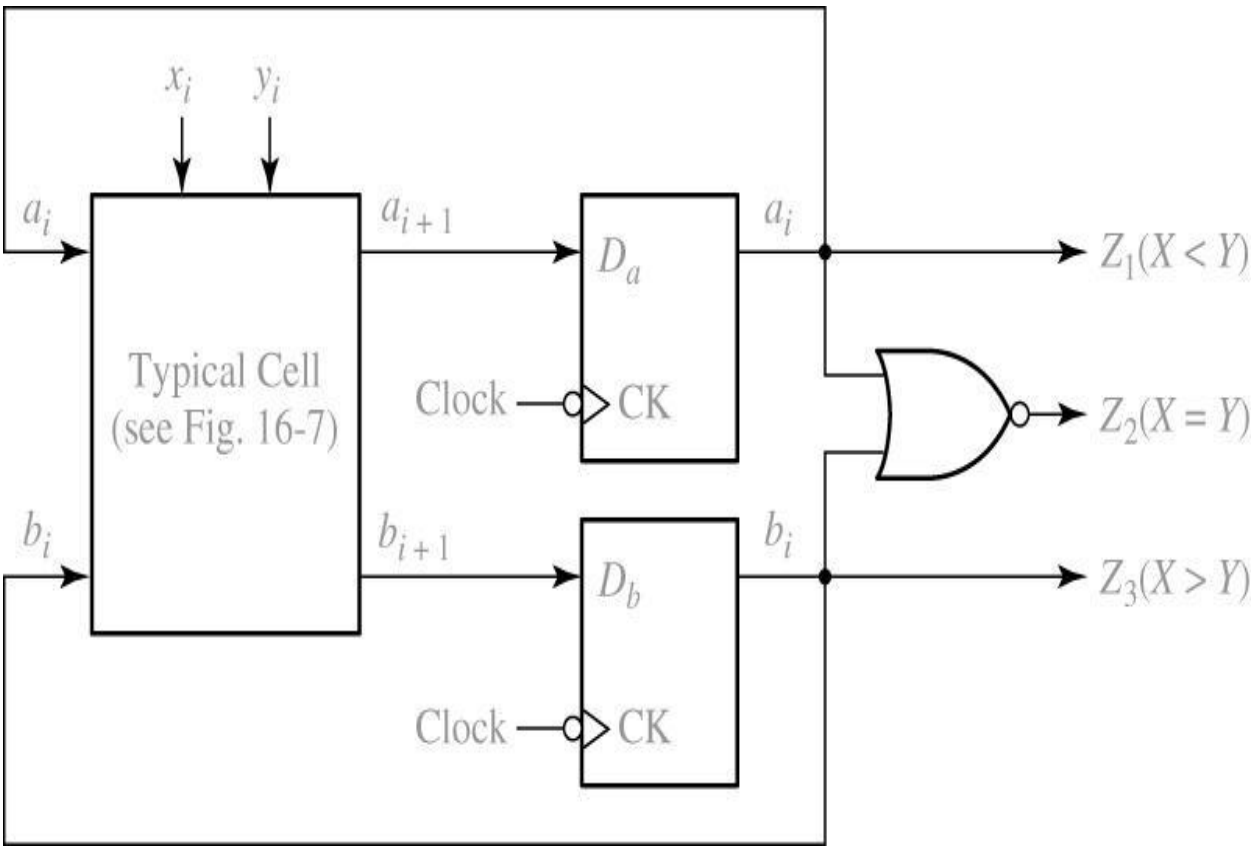


FIGURE 16-8
Output Circuit for Comparator

Sequential Circuit Version

- Inputs are received serially.
- Use the same table (state table)
- The same next state equations.



Sequential Circuits Using ROMs

- ROM for combinational parts
- 7 states: 3 D FFs => 4 outputs/4 inputs

TABLE 16-6

(a) State table

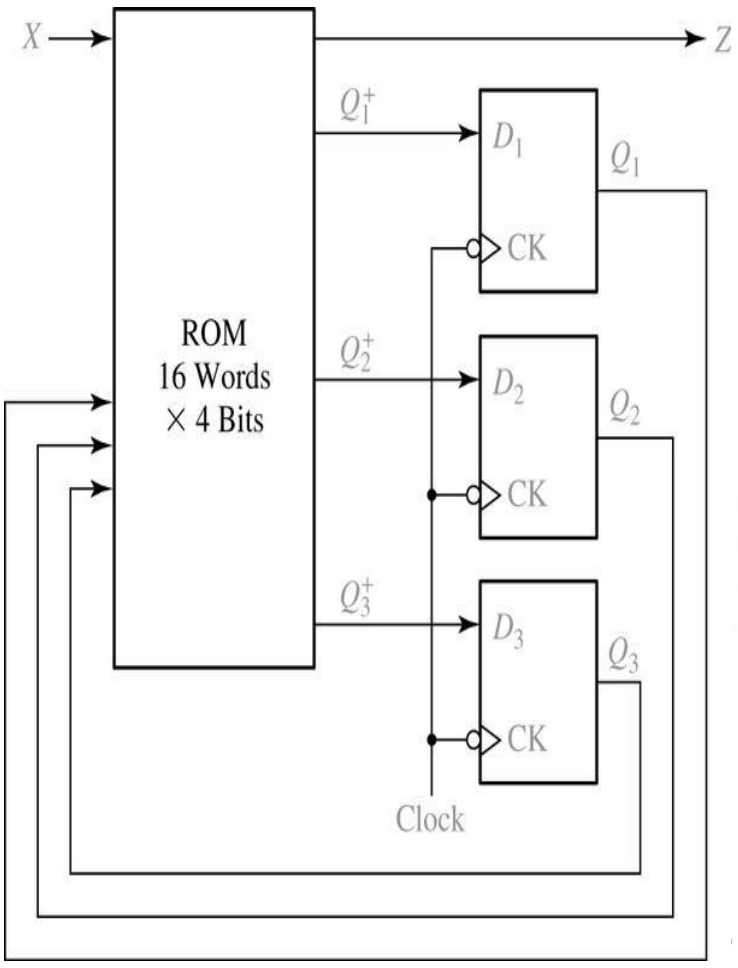
Present State	Next State X = 0	Next State X = 1	Present Output (Z) X = 0	Present Output (Z) X = 1
A	B	C	1	0
B	D	E	1	0
C	E	E	0	1
D	H	H	0	1
E	H	M	1	0
H	A	A	0	1
M	A	-	1	-

(b) Transition table

	$Q_1Q_2Q_3$	$Q_1^+Q_2^+Q_3^+$ X = 0	$Q_1^+Q_2^+Q_3^+$ X = 1	Z X = 0	Z X = 1
A	000	001	010	1	0
B	001	011	100	1	0
C	010	100	100	0	1
D	011	101	101	0	1
E	100	101	110	1	0
H	101	000	000	0	1
M	110	000	-	1	-

Sequential Circuits Using ROMs

- ROM for combinational parts
- 7 states: 3 D FFs => 4 outputs/4 inputs



Q^+

(c) Truth table

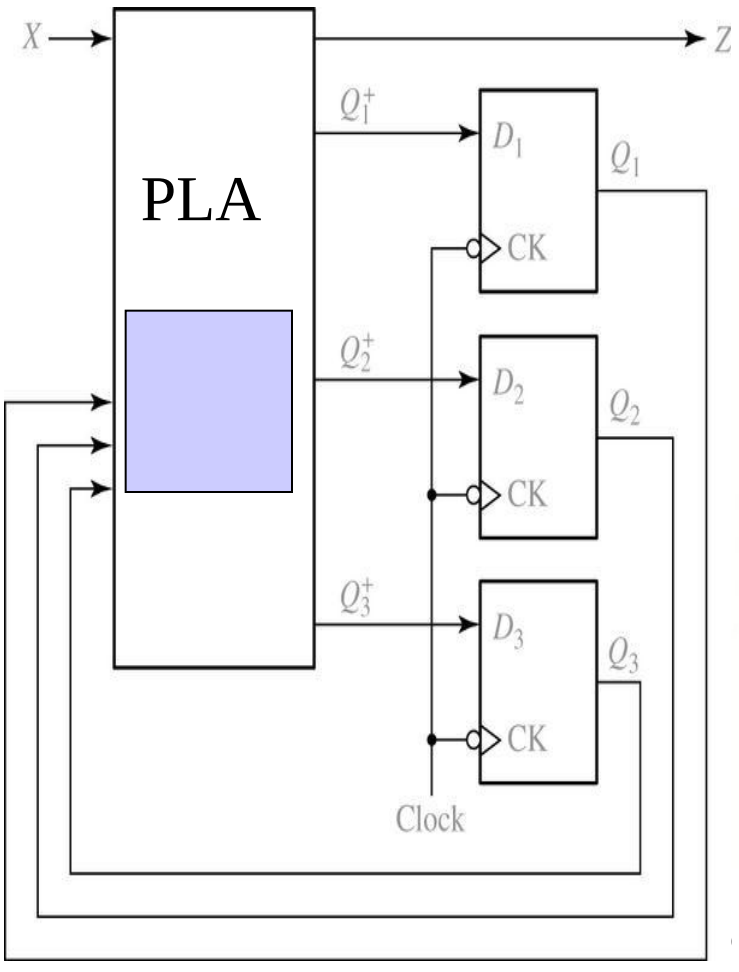
X	Q_1	Q_2	Q_3	Z	D_1	D_2	D_3
0	0	0	0	1	0	0	1
0	0	0	1	1	0	1	1
0	0	1	0	0	1	0	0
0	0	1	1	0	1	0	1
0	1	0	0	1	1	0	1
0	1	0	1	0	0	0	0
0	1	1	0	1	0	0	0
0	1	1	1	x	x	x	x
1	0	0	0	0	0	1	0
1	0	0	1	0	1	0	0
1	0	1	0	1	1	0	0
1	0	1	1	1	1	0	1
1	1	0	0	0	1	1	0
1	1	0	1	1	0	0	0
1	1	1	0	x	x	x	x
1	1	1	1	x	x	x	x

Address bus

Data bus

Sequential Circuits Using PLAs

- For PLA, this table uses 4 inputs/4 outputs, 13 products terms (excluding xxxx)



(c) Truth table

X	Q_1	Q_2	Q_3	Z	D_1	D_2	D_3
0	0	0	0	1	0	0	1
0	0	0	1	1	0	1	1
0	0	1	0	0	1	0	0
0	0	1	1	0	1	0	1
0	1	0	0	1	1	0	1
0	1	0	1	0	0	0	0
0	1	1	0	1	0	0	0
0	1	1	1	x	x	x	x
1	0	0	0	0	0	1	0
1	0	0	1	0	1	0	0
1	0	1	0	1	1	0	0
1	0	1	1	1	1	0	1
1	1	0	0	0	1	1	0
1	1	0	1	1	0	0	0
1	1	1	0	x	x	x	x
1	1	1	1	x	x	x	x

Sequential Circuits Using PLAs

- For PLA, this table uses 4 inputs/4 outputs and 7 product terms. (Fig 16-2 assignment)

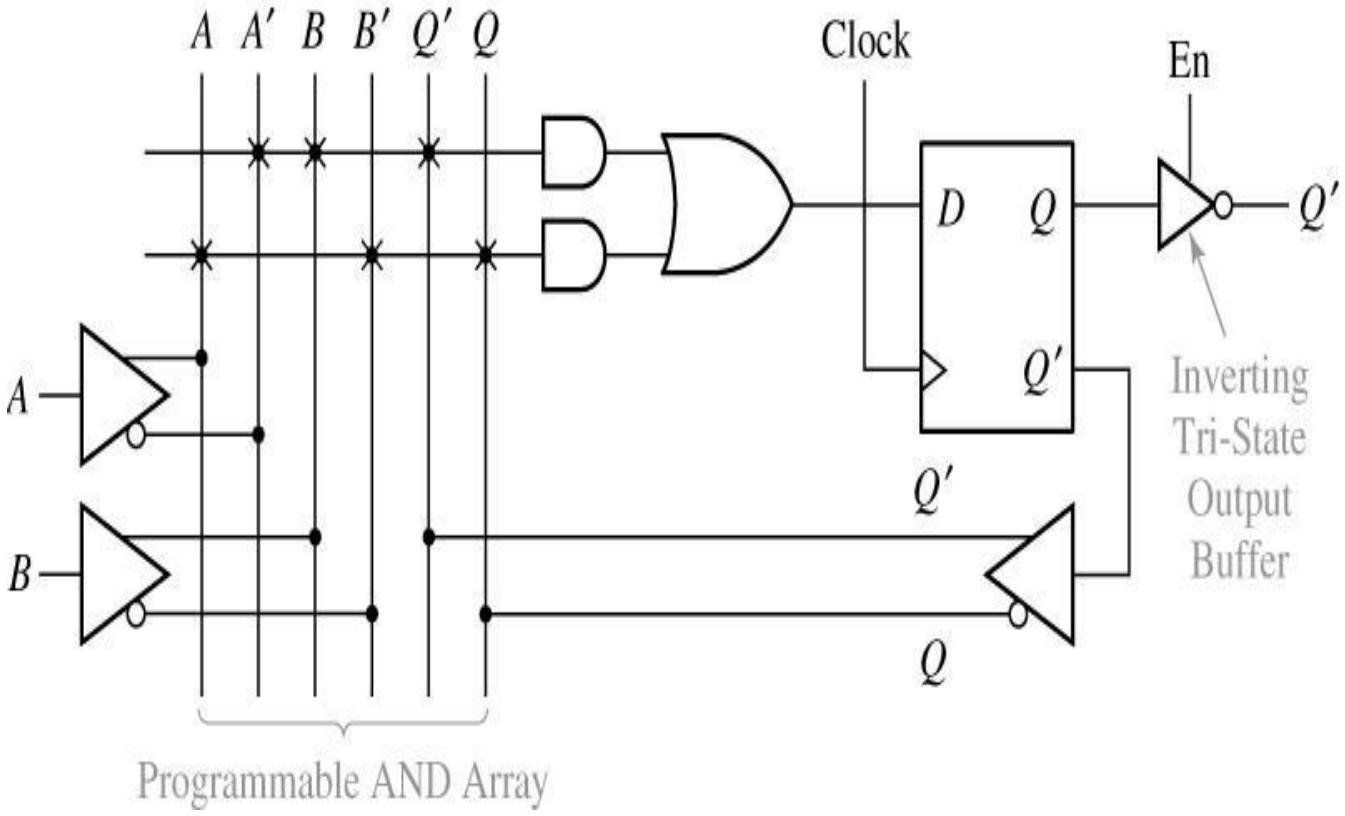
$$D_1 = Q_1^+ = Q_2'$$
$$D_2 = Q_2^+ = Q_1$$
$$D_3 = Q_3^+ = Q_1Q_2Q_3 + X'Q_1Q_3' + XQ_1'Q_2'$$
$$Z = X'Q_3' + XQ_3$$

TABLE 16-7

X	Q_1	Q_2	Q_3	Z	D_1	D_2	D_3
–	–	0	–	0	1	0	0
–	1	–	–	0	0	1	0
–	1	1	1	0	0	0	1
0	1	–	0	0	0	0	1
1	0	0	–	0	0	0	1
0	–	–	0	1	0	0	0
1	–	–	1	1	0	0	0

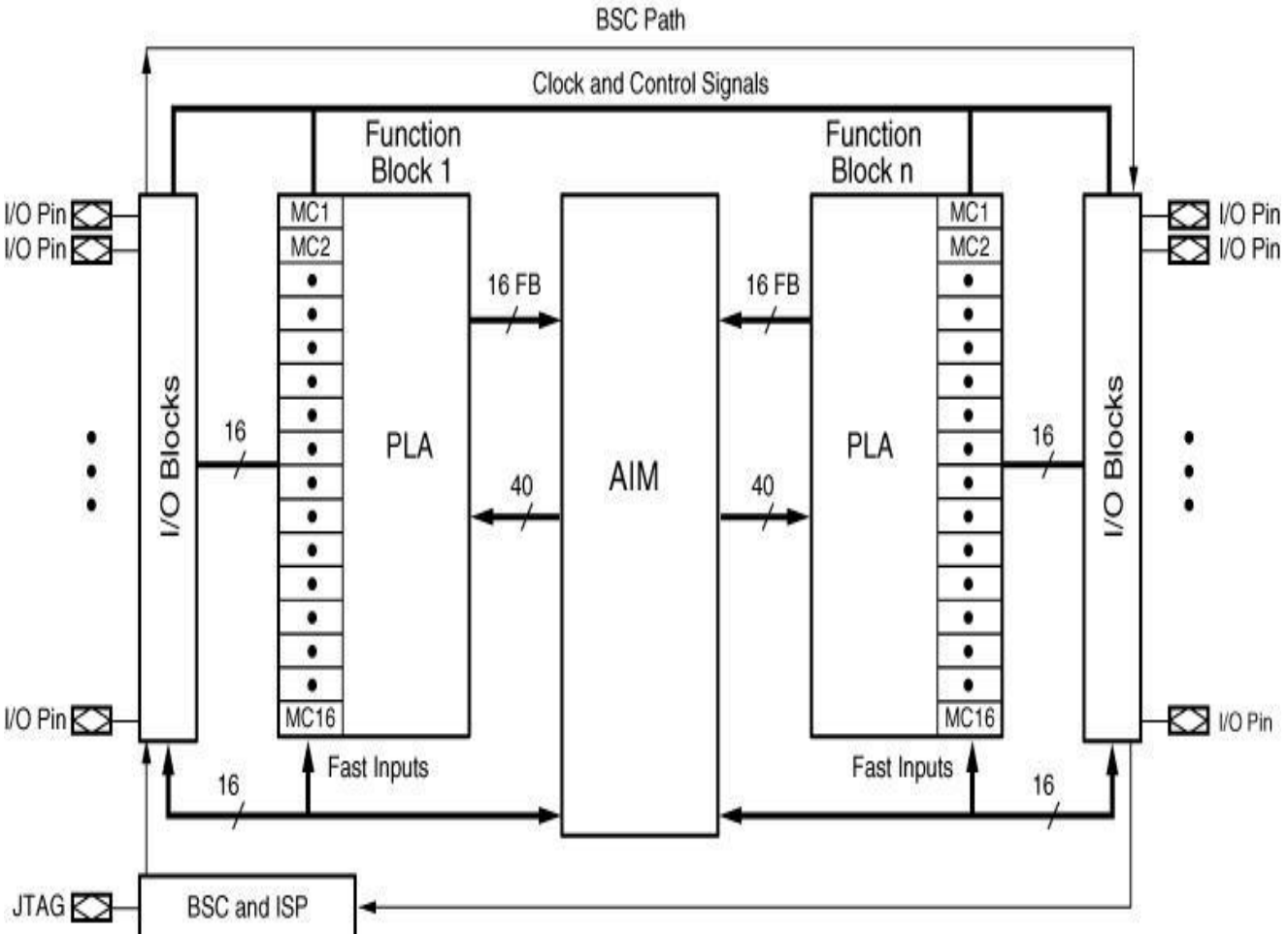
PALs for Sequential Circuit

- Programmable AND array.
- $Q^+ = D = A'BQ' + AB'Q$



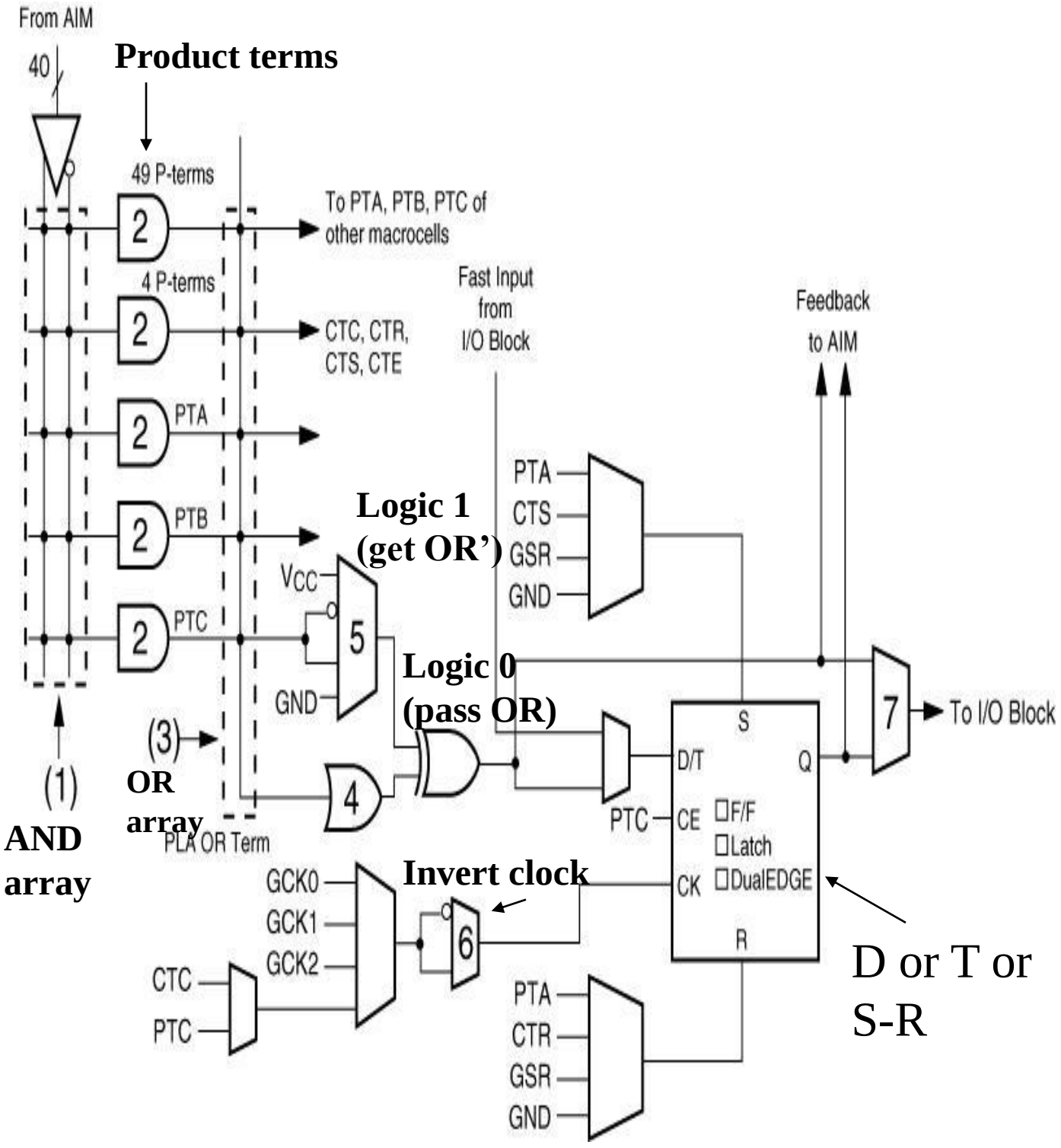
CPLDs for Sequential Circuit

- CPLD = function blocks + MCs + AIM + I/Os + etc
  - PLA in each function block
  - AIM: advanced interconnection matrix
  - Macocell = MUXs + FFs (or latches)
    - Dual-edge triggered FF



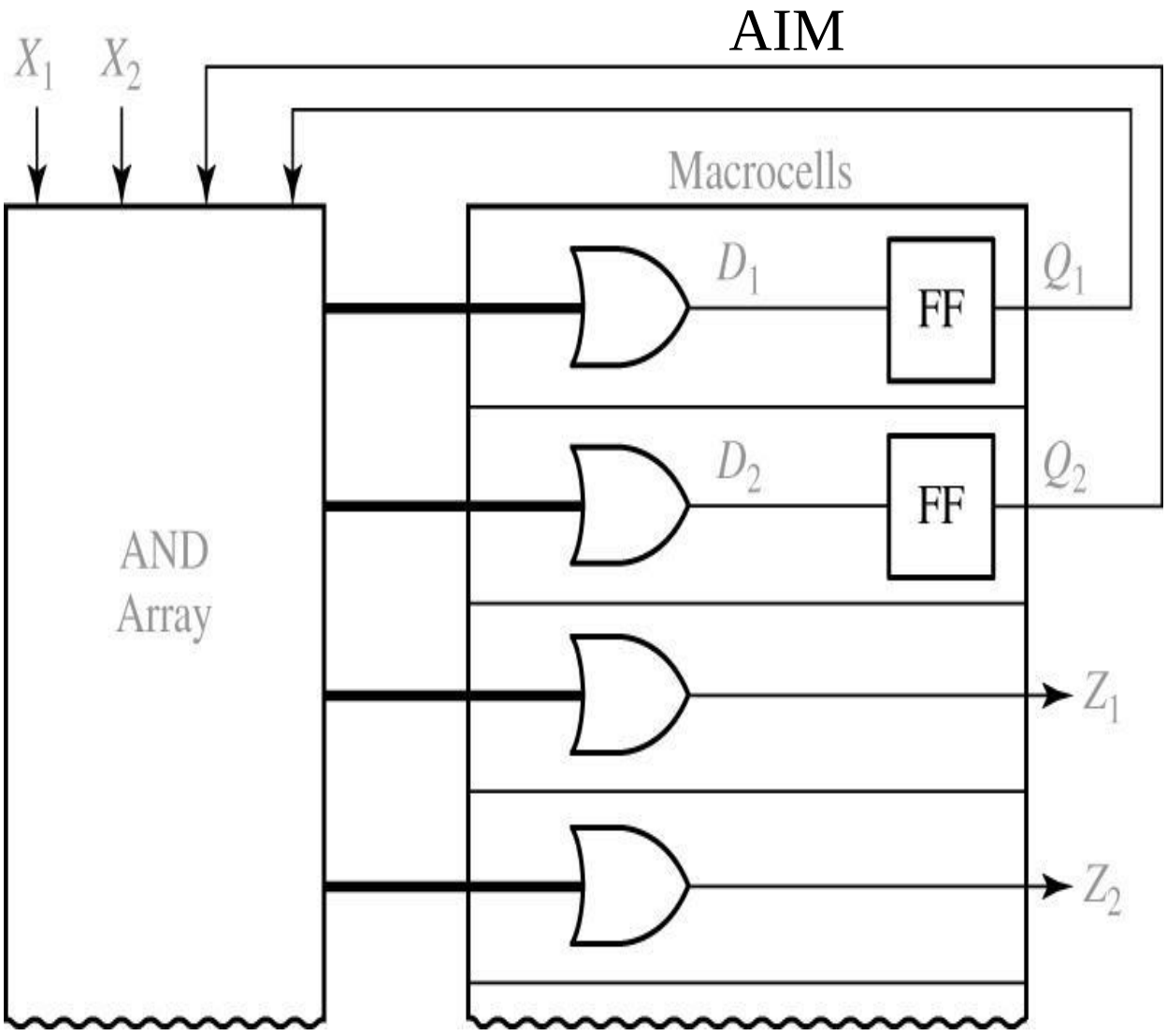
A Macrocell

- CPLD



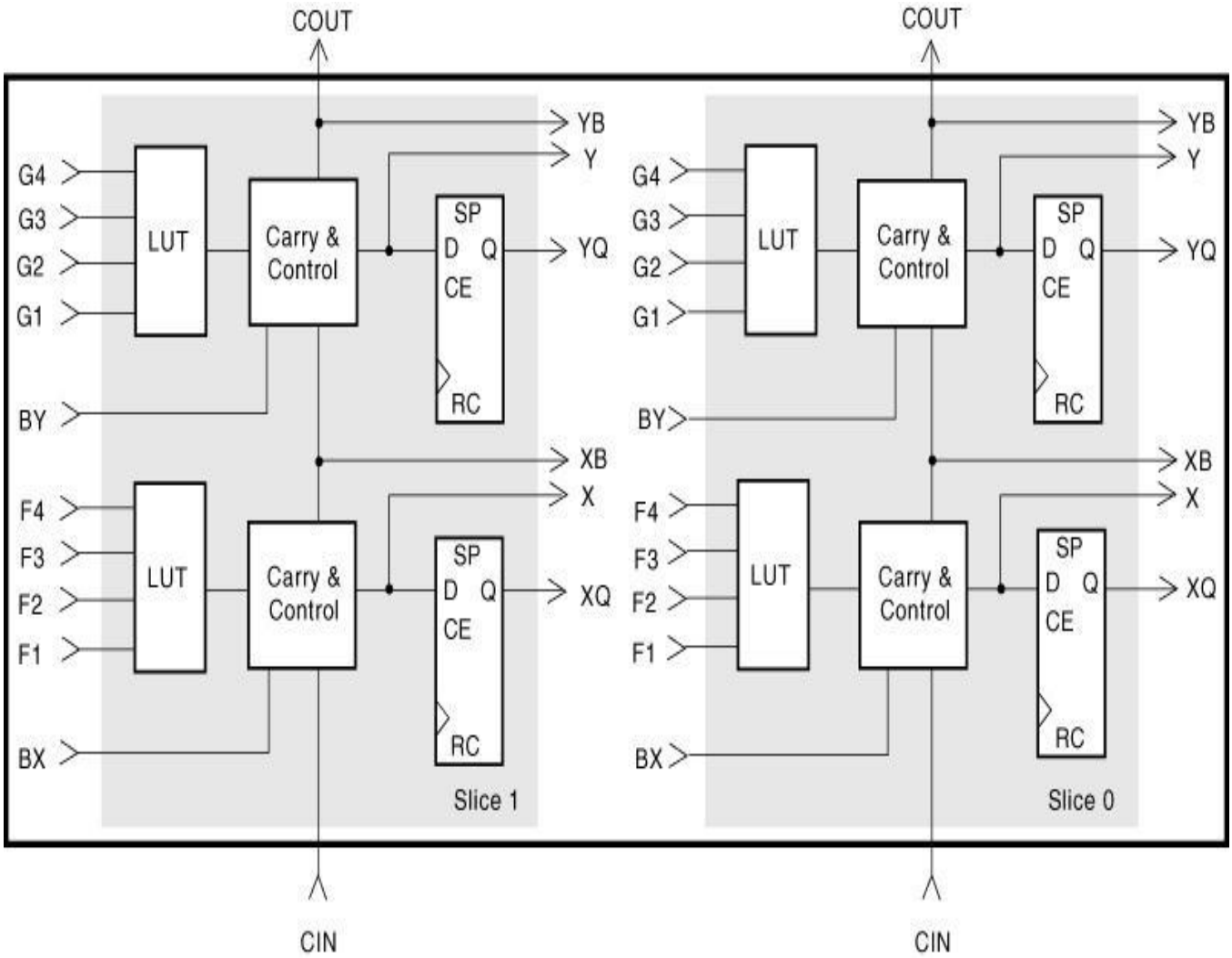
CPLDs for Sequential Circuit

- CPLD for a Mealy machine
- 4 macrocells
  - Two for D FF inputs
  - Two for output Z



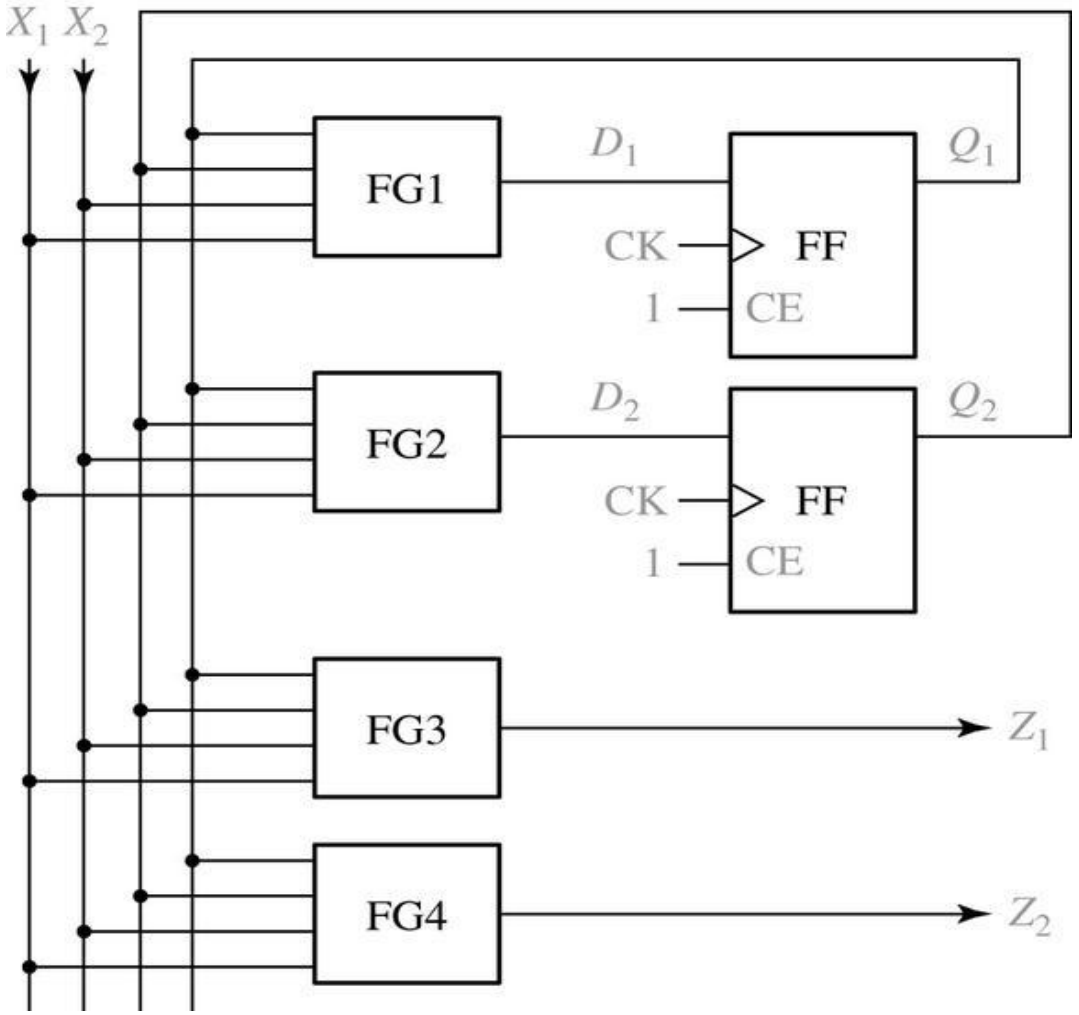
FPGAs for Sequential Circuit

- FPGA = configurable logic blocks (CLB) + I/O blocks + (memory blocks + CLK generators + tri-state buffers)
- CLB = LUTs + MUXes + D-CE FFs
- I/O block = FFs (for I/O) + tri-state buffers



FPGAs for Sequential Circuit

- FPGA for a Mealy machine
  - FG: functional generator

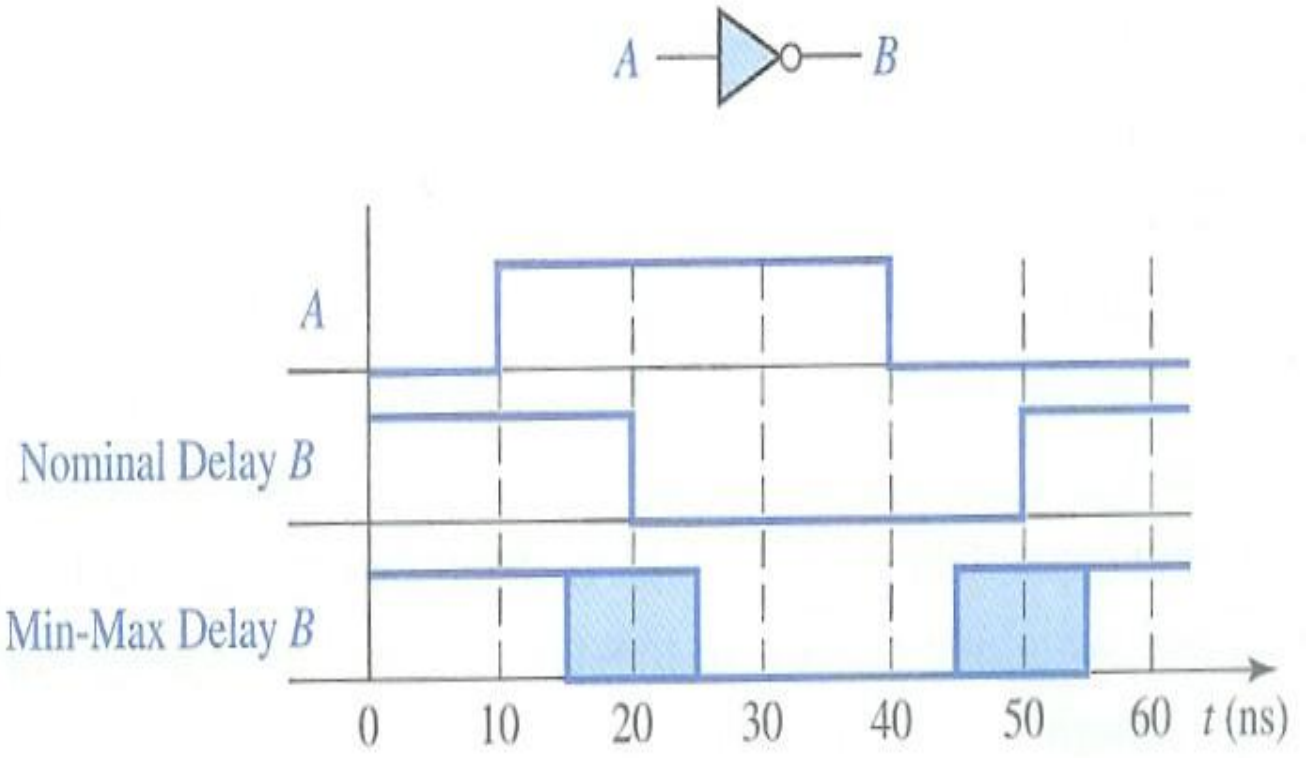


Simulations

- Functional level
  - Sequence of transfers btw registers, ALU, memories.
  - Verify high level system design
- Logic level
  - Logic gates, FF, and interconnection
  - Verify logic design and analyze timing
- Circuit level
  - Each gate is represented in transistors, R, C.
  - Information about voltage level and switching speed.

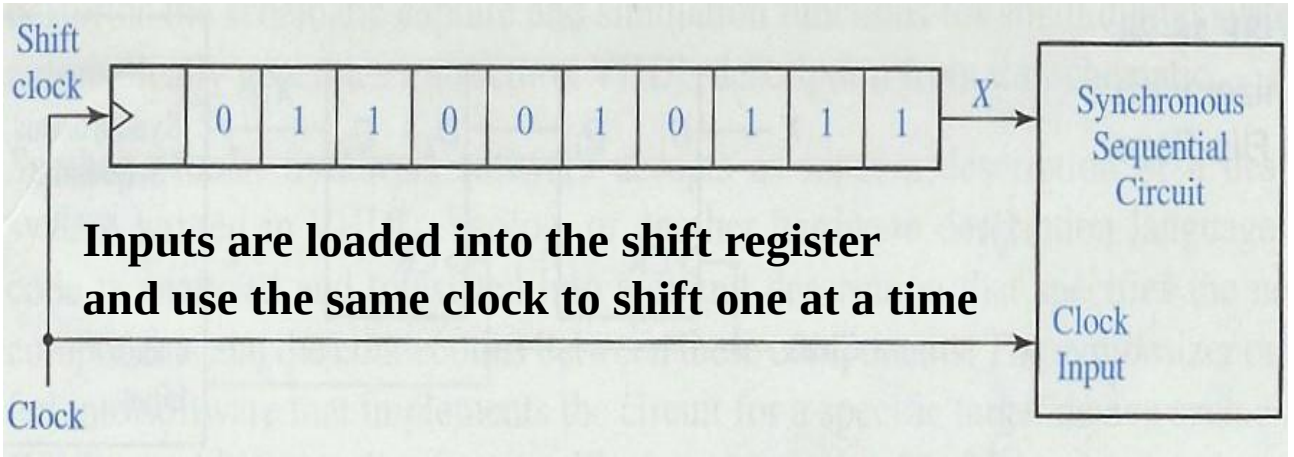
Delays

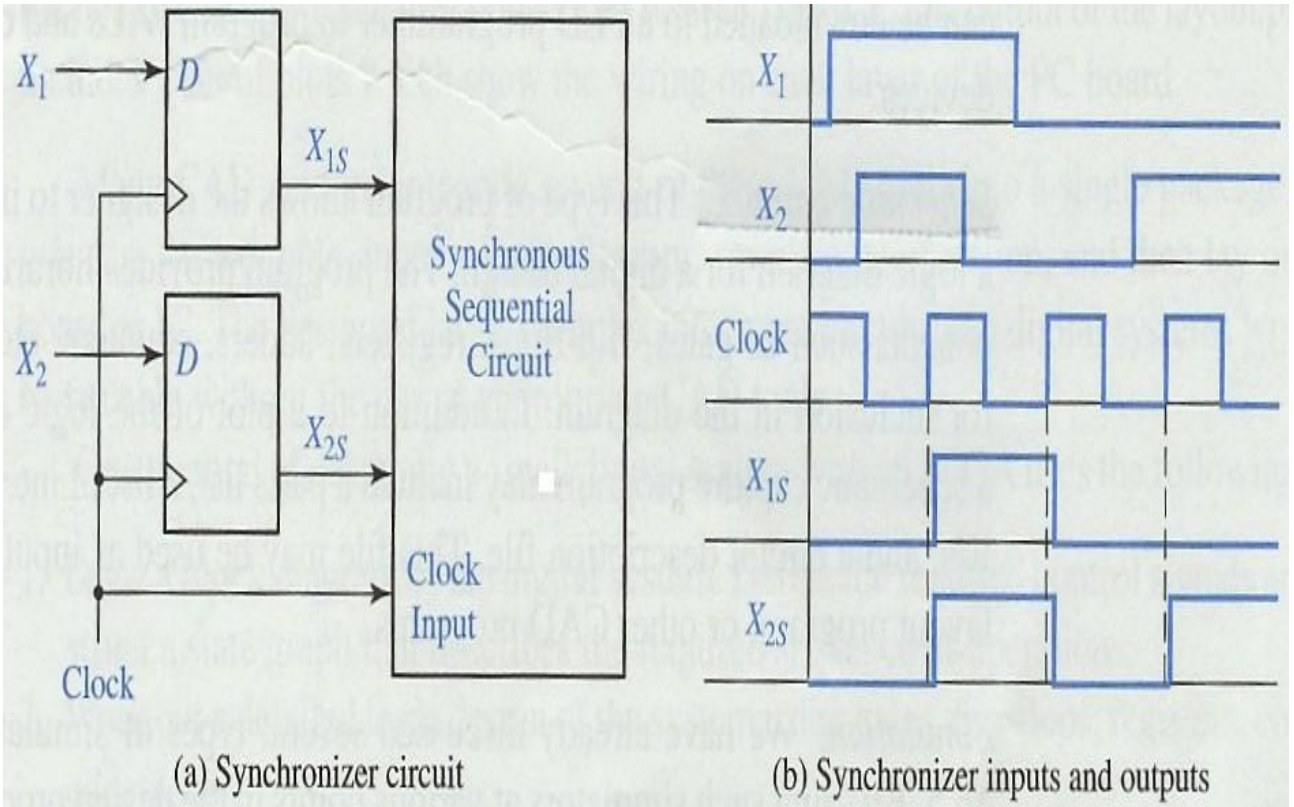
- Unit delay model for simulation at first.
- Minimum – Nominal- Maximum delay
- Shaded area indicates that B may change at any time during this interval.



Synchronizer Circuits

- Purpose: Synchronize the inputs w.r.t. clock.
  - X_{1s} and X_{2s} always change immediately following the clock pulse.





Synchronizer Circuits

- This synchronizer may fail if the FF enters the metastable state.
  - When the signal that is sampled is not stable for the required set-up time and hold time. The FF may go into a metastable state where the output will not have a legitimate high or low value, but in an indeterminate region between them.
  - The FF can not be guaranteed to exit the metastable in any bounded time. But the probability of the FF in the metastable state decreases exponentially with time.
  - So, the solution is to wait more time until the output is stable.

Synchronizer Circuits

- This synchronizer will work properly if the period of metastability is less than the clock period.
  - The first output of the D FF may be metastable, it will not be seen by other logic element until the second clock, when the second D FF samples the signal, which by that time should no longer be in a metastable state. It does not matter whether X_{1s} is delayed for one or two clocks.