13

Simplification of Sequential Circuits

Simplification of Sequential Circuits

State Reduction

- In the design of a sequential circuit, we might use more states than those of necessary.
- The design must find a minimal state table to reduce the number of machine states.
- Consequently, to obtain the optimum number of the components used for the circuit.

- There are 3 methods for reducing the number of conditions:

1) **Inspection**: To observe the states that have both the same NS and O/P.

2) **Partitioning**: To arrange the groups of states that produce the groups that have got the same logical outputs.

3) **Implication Table**: the general method that make a table for comparing all states that provide the same outputs for all logical inputs.

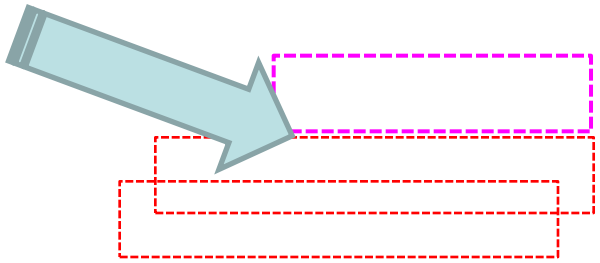
Inspection Method

1) Inspection Method [the same NS and O/P]

Example: Find the minimal state table from the state table below

PS \ X	0	1
A	B/0	C/1
B	C/0	A/1
C	D/1	B/0
D	C/0	A/1

NS/Z



- We find that state B = state D.
- Then we could merge them as the state BD.
- Then the new state will be:

  A’ = A, B’ = BD and C’ = C

- And get a new Minimal state table:

PS \ x	0	1
A’	B’/0	C’/1
B’	C’/0	A’/1
C’	B’/0	B’/0

Partitioning Method

Method Steps: [the same logical outputs]

- Let P_k be a group of the states having the same outputs, where $k = 1, 2, \ldots, i$.
- For an Input x; we can arrange a group as $P_k = (S_A S_C)(S_B S_D)(S_E)$

  which means $S_A = S_C$, $S_B = S_D$, etc.
- After that, consider the group for the NS when applying the input x,
- If any group produces the states that is in different groups,

  we must re-arrange the grouping again.
- Keep doing this until getting the group in P_k is the same as P_{k-1} .
- Then the last partitioned group will be the answer.

Partitioning Method

Example: Find the minimal state table from the state table below.

	X = 0	X = 1
A	C/1	B/0
B	C/1	E/0
C	B/1	E/0
D	D/0	B/1
E	E/0	A/1

Step 2: Consider the NS, for all input x.

Looking for the outputs that change the group, make them in a new group and arrange all groups again.

Solution:

Step1: Arrange the group of the states that has the same outputs:

The we get:

Output = 1 Output = 0

P_1: $= (A\ B\ C)^1(D\ E)^2$

x= 0: NS = $C^1\ C^1\ B^1\ D^2\ E^2$

x= 1: NS = $B^1\ E^2\ E^2\ B^1\ A^1$

P_2: $= (A)^1\ (B\ C)^2\ (D\ E)^3$

x= 0: NS = $C^2\ C^2\ B^2\ D^3\ E^3$

x= 1: NS = $B^2\ E^3\ E^3\ B^2\ A^1$

P_3: $= (A)^1\ (B\ C)^2\ (D)^3(E)^4$

Partitioning Method

Note: For the group that consists of one state, it could be ignored for comparing in the partition process.

Step 3: Redo the step 2 until we get the group partitioning is the same as the previous one, i.e. $P_k = P_{k-1}$.

P_3: $= (A)^1 (B\ C)^2 (D)^3 (E)^4$

x= 0: NS = C^2 C^2 B^2 D^3 E^4

x= 1: NS = B^2 E^4 E^4 B^2 A^1

P_4: $= (A)^1 (B\ C)^2 (D)^3 (E)^4$

$P_4 = P_3$

(B C) means B and C are in the same state.

- Then the new state will be:

A' = A, B' = BC, C' = D and D' = E .

- Get a new Minimal state table with 4 states, which requires 2FFs.

	X=0	X=1
A	C/1	B/0
B	C/1	E/0
C	B/1	E/0
D	D/0	B/1
E	E/0	A/1

PS	X=0	X=1
A'	B'/1	B'/0
B'	B'/1	D'/0
C'	C'/0	B'/1
D'	D'/0	A'/1

Minimal state table

Partitioning Method

Example: Find the minimal state table from the state table below

X	0	1
A	E/0	B/1
B	A/1	F/0
C	C/0	D/1
D	C/1	F/0
E	C/0	D/1
F	B/O	A/1

NS/Z

Solution:

Step1: Arrange the group of the states that has the same output:

The we get:

Output = 0 Output = 1

P_1: $= (A\ C\ E\ F)^1\ (B\ D)^2$

x= 0: NS = $E^1\ C^1\ C^1\ B^2\ A^1\ C^1$

x= 1: NS = $B^2\ D^2\ D^2\ A^1\ F^1\ F^1$

P_2: $= (A\ C\ E)^1\ (F)^0\ (B\ D)^2$

x= 0: NS = $E^1\ C^1\ C^1\ B^2\ A^1\ C^1$

x= 1: NS = $B^2\ D^2\ D^2\ A^1\ F^0\ F^0$

P_3: $= (A\ C\ E)^1\ (F)\ (B\ D)^2$

Partitioning Method

- $P_2 = P_3$
- (A C E) means A,C and E are in the same state.
- (B D) means B and D are in the same state.

- Then the new state will be:

A’ = (ACE) , B’ = (BD) and C’ = F .

- Get a new Minimal state table with 3 states, which requires 2 FFs.

	X = 0	X = 1
A	E/0	B/1
B	A/1	F/0
C	C/0	D/1
D	C/1	F/0
E	C/0	D/1
F	B/O	A/1

NS/Z

	X = 0	X = 1
A'	A'/0	B'/1
B'	A'/1	C'/0
C'	B'/0	A'/1

NS/Z

Partitioning Method

Example: Find the minimal state table from the state table below

PS \ x_1x_2	00	01	11	10
A	D/0	D/0	F/0	A/0
B	C/1	D/0	E/1	F/0
C	C/1	D/0	E/1	A/0
D	D/0	B/0	A/0	F/0
E	C/1	F/0	E/1	A/0
F	D/0	D/0	A/0	F/0
G	G/0	G/0	A/0	A/0
H	B/1	D/0	E/1	A/0

NS/Z

Solution:

Step1: Arrange the group of the states tha
has the same output:

The we get:

Output = 0 Output = 1

P_1: $= (A\ D\ F\ G)^1\ (B\ C\ E\ H)^2$

x= 00: NS = $D^1\ D^1\ D^1\ G^1\ C^2\ C^2\ C^2\ B^2$

x= 01: NS = $D^1\ B^2\ D^1\ G^1\ D^1\ D^1\ F^1\ D^1$

x= 11: NS = $F^1\ A^1\ A^1\ A^1\ E^2\ E^2\ E^2\ E^2$

x= 10: NS = $A^1\ F^1\ F^1\ A^1\ F^1\ A^1\ A^1\ A^1$

P_2: $= (D)^0(A\ F\ G)^1\ (B\ C\ E\ H)^2$

x= 00: NS = $D^0\ D^0\ D^0\ G^1\ C^2\ C^2\ C^2\ B^2$

x= 01: NS = $B^2\ D^0\ D^0\ G^1\ D^0\ D^0\ F^1\ D^0$

x= 11: NS = $A^1\ F^1\ A^1\ A^1\ E^2\ E^2\ E^2\ E^2$

x= 10: NS = $F^1\ A^1\ F^1\ A^1\ F^1\ A^1\ A^1\ A^1$

Partitioning Method

PS \ x_1x_2	00	01	11	10
A	D/0	D/0	F/0	A/0
B	C/1	D/0	E/1	F/0
C	C/1	D/0	E/1	A/0
D	D/0	B/0	A/0	F/0
E	C/1	F/0	E/1	A/0
F	D/0	D/0	A/0	F/0
G	G/0	G/0	A/0	A/0
H	B/1	D/0	E/1	A/0

NS/Z

P_3: $= (D)^0(A\ F)^1(G)^3(B\ C\ H)^2(E)^4$

x= 00: NS = $D^0\ D^0\ D^0\ G^3\ C^2\ C^2\ B^2\ C^2$

x= 01: NS = $B^2\ D^0\ D^0\ G^3\ D^0\ D^0\ D^0\ F^1$

x= 11: NS = $A^1\ F^1\ A^1\ A^1\ E^4\ E^4\ E^4\ E^4$

x= 10: NS = $F^1\ A^1\ F^1\ A^1\ F^1\ A^1\ A^1\ A^1$

P_4: $= (D)^0(A\ F)^1(G)^3(B\ C\ H)^2(E)^4$

- $P_3 = P_4$
- (A F) means A and F are in the same state.
- (B C H) means B, C and H are in the same state.

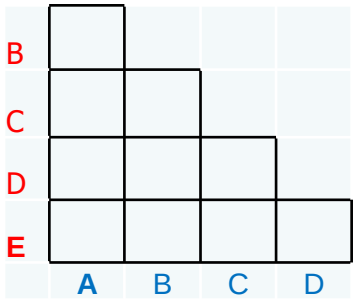
- Then the new state will be:

A' = (AF) , B' = (BCH) , C' = D, D' = E and E' = G .

- Get a new Minimal state table with 5 states, which requires 3 FFs.

Implication Table Method

Implication Table Method: is a general method for comparing every states via a table. So, it might consume more time than the others.



Implication Table is a triangle table which

- has the bottom row of the state 1 to the state (n-1) and the first column of state 2 to the state n.
- Then observe the relation between the crossing states:

• Mark X to the box if they have the different outputs for individual inputs.

• Mark different state names if they have the **same** outputs for individual inputs. Call the state names as an implied pair, or an equivalent pair if the pair is the **same** as the actual state names.

- Recheck the pairs of all marked states again.

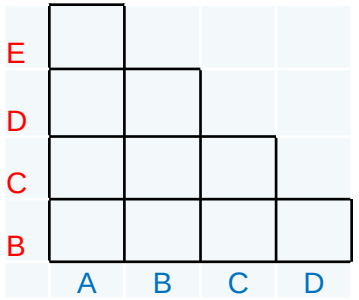
• Mark X if the pair was marked X before.

- Collect all the left the implied pair those are not marked X.

Implication Table Method

Example: Find the minimal state table from the state table below

Solution:



Step1: Make the Implication Table:

the bottom row: the state 1 to the state (*n*-1) and

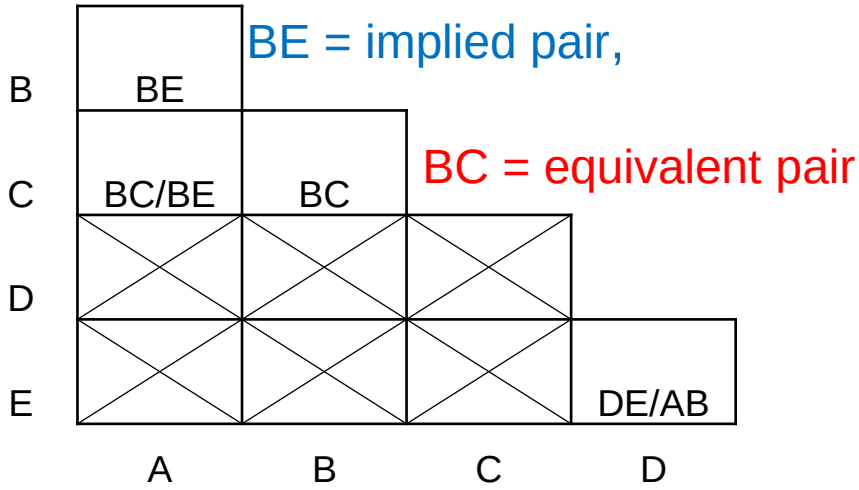
the first column: state 2 to the state *n*.

Step2: Observe the outputs between the crossing states:

- if they have the same outputs for input x = 0, 1;
  $\Rightarrow$ Write the state names for the input of 0 and 1.
- if they have the different outputs for input x = 0, 1; $\Rightarrow$ Mark X

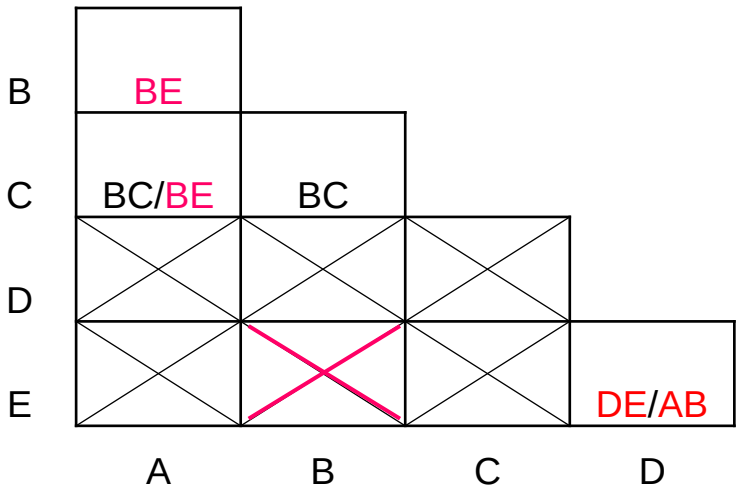
PS \ X	0	1
A	C/1	B/0
B	C/1	E/0
C	B/1	E/0
D	D/0	B/1
E	E/0	A/1

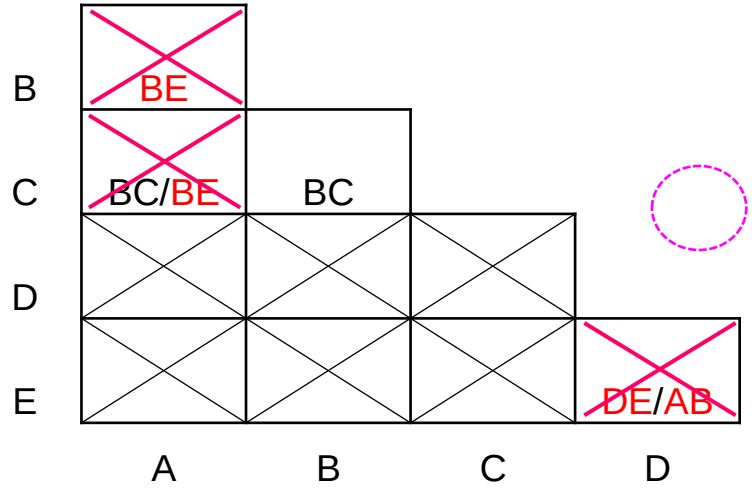
NS/Z



Implication Table Method

Step3: Recheck the implied pair of all marked states again.
Mark X if the pair are marked X before.





Step4: Collect the implied pair and equivalent pair occurring in individual column.

A: (–)(BC)

B: (BC)

C: (–)

D: (–).

- Then we get the group: $P_k = (A)(BC)(D)(E)$
- And assign again to get: A' = A, B' = (BC), C' = D and D' = E.
- Get the result the same as that of the example in Group Partitioning.

Implication Table Method

Example: Find the minimal state table from the state table below

PS \ X	0	1
A	E/0	D/0
B	A/1	F/0
C	C/0	A/1
D	B/0	A/0
E	D/1	C/0
F	C/0	D/1
G	H/1	G/1
H	C/1	B/1

NS/Z

Solution:

Step1: Make the Implication Table:

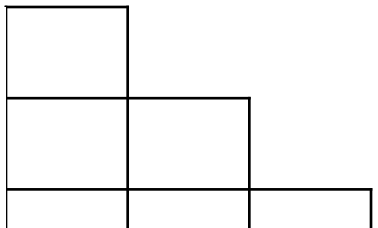
the bottom row: the state 1 to the state (*n*-1) and

the first column: state 2 to the state *n*.

Step2: Observe the relation between the crossing states:

- Mark different state with its state names if they have the same outputs for input x = 0, 1;
- Mark X state name if they have the different outputs for input x = 0, 1;

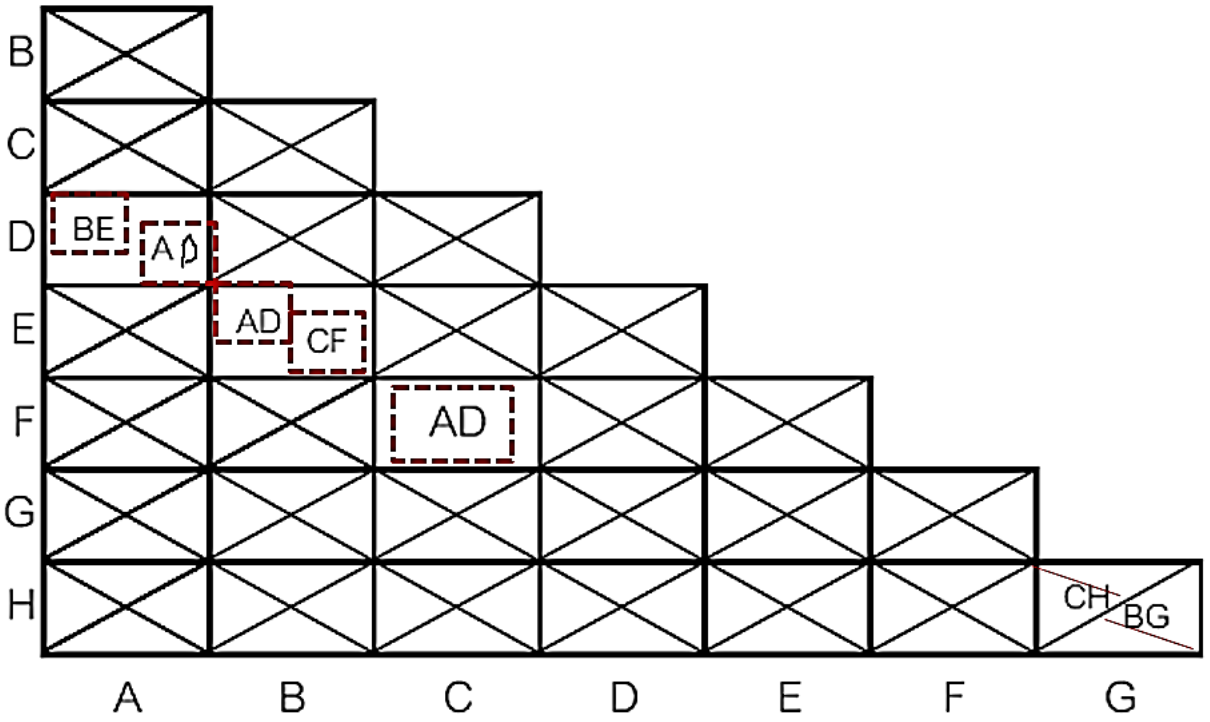
Step3: Recheck the implied pair of all marked states again. Mark X if the pair are marked X before.

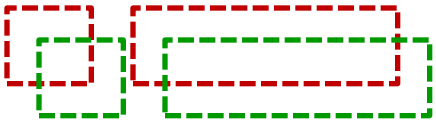


Implication Table Method

PS \ X	0	1
A	E/0	D/0
B	A/1	F/0
C	C/0	A/1
D	B/0	A/0
E	D/1	C/0
F	C/0	D/1
G	H/1	G/1
H	C/1	B/1

NS/Z





Implication Table Method

Step4: Collect the implied pair and equivalent pair occurring in individual column

A: (BE)(AD)

B: (AD)(CF)

C: (AD)

D: (–)

E: (–)

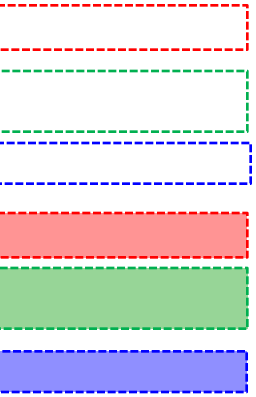
F: (–)

G: (–)....

- Then we get the group: P_k = (AD)(BE)(CF)(G)(H)
- And assign again to get:

  A’ = (AD), B’ = (BE), C’ = (CF), D’ = G and E’ = H.
- Get the minimal states as follows.

Implication Table Method



PS \ X	0	1
A	E/0	D/0
B	A/1	F/0
C	C/0	A/1
D	B/0	A/0
E	D/1	C/0
F	C/0	D/1
G	H/1	G/1
H	C/1	B/1

NS/Z

PS \ X	0	1
A'	B'/0	A'/0
B'	A'/1	C'/0
C'	C'/0	A'/1
D'	E'/1	D'/1
E'	C'/1	B'/1

NS/Z

State Assignment

State Assignment Code Guideline

Rule 1: PS that gives the same NS for each input x.

We assign them being next to each other (with 1 bit different).

Rule 2: NS of the same PS can be next to each other.

(Must apply after rule 1).

Rule 3: We usually determines the state that occurs in the table the most to be zero, and the state that rarely happens to be one. For simplicity, we generally specify the initial state as A = 000.

State Assignment

Example: Find the minimal state table from the state table below. We provide four sets of state assignment. Compare them and find which one is the best for the design with D Flip-flops.

PS \ x	0	1
A	C/0	D/0
B	C/0	A/0
C	B/0	D/0
D	A/1	B/1

NS/Z

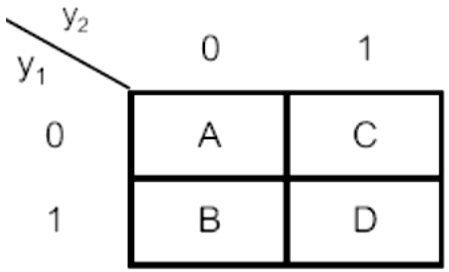
state assignments

	set 1	set 2	set 3	set 4
A	00	00	00	00
B	01	11	10	01
C	11	01	01	10
D	10	10	11	11

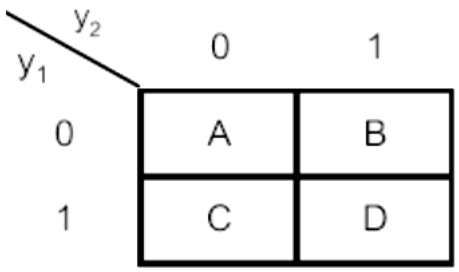
State Assignment

Solution From State Assignment Code Guideline

- Rule 1: The same NS: Give the PS of A-B and A-C.
- Rule 2: The same PS: Give the NS of C-D and B-D.



set 3



set 4

So, we must assign with one of these sets:.

- set 3: A = 00, B = 10, C = 01, D = 11.
- set 4: A = 00, B = 01, C = 10, D = 11

If using set 3, get the Transition table:

And excitation tables for D_1, D_2 and *z* on the next page

y_1y_2 \ X	0	1
00	01/0	11/0
01	10/0	11/0
11	00/1	10/1
10	01/0	00/0

$Y_1 Y_2 / Z$

State Assignment

Transition table and excitation tables for D_1, D_2 and z using assignment set 3,

	y_1y_2 \ X	0	1
A	00	01/0	11/0
C	01	10/0	11/0
D	11	00/1	10/1
B	10	01/0	00/0

$Y_1 Y_2 / Z$

y_1y_2 \ X	0	1
00	0	1
01	1	1
11	0	1
10	0	0

$D_1 = \bar{y}_1x + \bar{y}_1y_2 + y_2x$

y_1y_2 \ X	0	1
00	1	1
01	0	1
11	0	0
10	1	0

$\therefore D_2 = \bar{y}_2\bar{x} + \bar{y}_1x$

y_1y_2 \ X	0	1
00	0	0
01	0	0
11	1	1
10	0	0

$Z = y_1y_2$

Require

- 5 AND gates
- 3 OR gates
- 15 Input lines

State Assignment

The components used for the circuit with

state assignments

Gates:	set 1	set 2	set 3	set 4
AND:	6	6	5	5
OR:	4	3	3	3
Input Lines:	20	20	15	15

So, the design with the assignment set 3 and 4 is more economic.

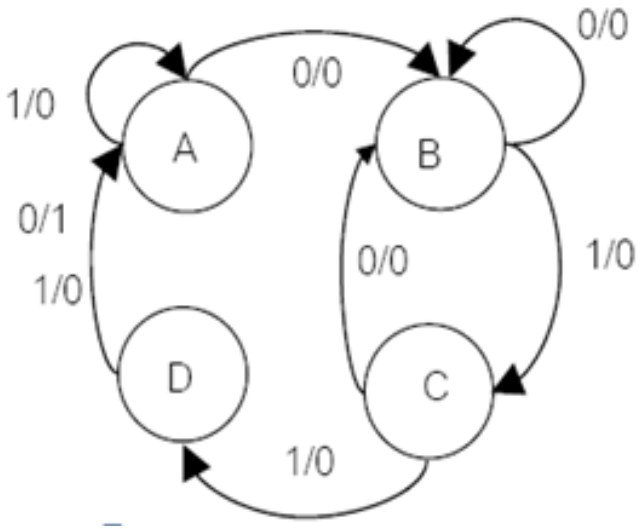
Complete Sequential Circuit Design

Example: Design the synchronous sequential circuit that detects bit stream of 0110 in the pattern of a) non-overlapping and b) overlapping in the stream.

Solution

a) Non-overlapping: If x = 0110110110
The circuit will produce z = 0001000001

1. Draw the state diagram



2. Write the state tables

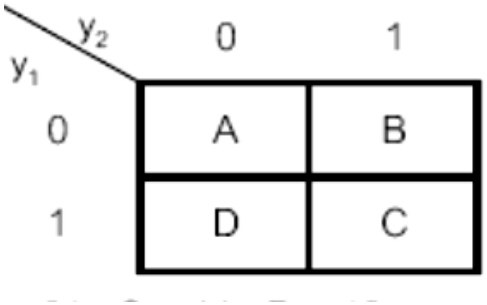
PS \ X	0	1
A	B/0	A/0
B	B/0	C/0
C	B/0	D/0
D	A/1	A/0

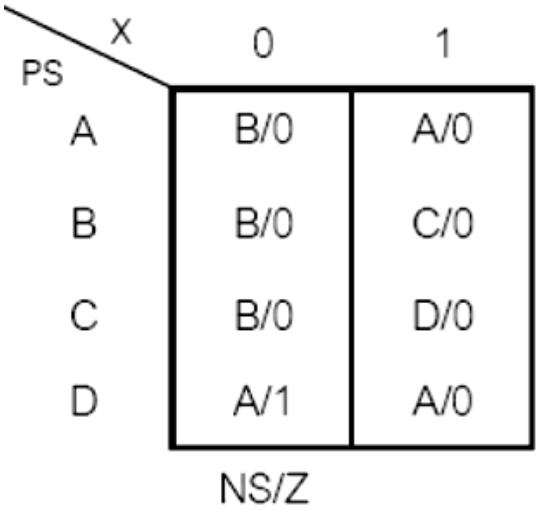
NS/Z

Complete Sequential Circuit Design

3. Do State Assignment

- Rule 1: The same NS: Give the PS of A-B, B-C, A-C and A-D.
- Rule 2: The same PS: Give the NS of B-D.





So, we have to assign the states by the codes:

A = 00, B = 01, C = 11, D = 10.

And get the transition table:

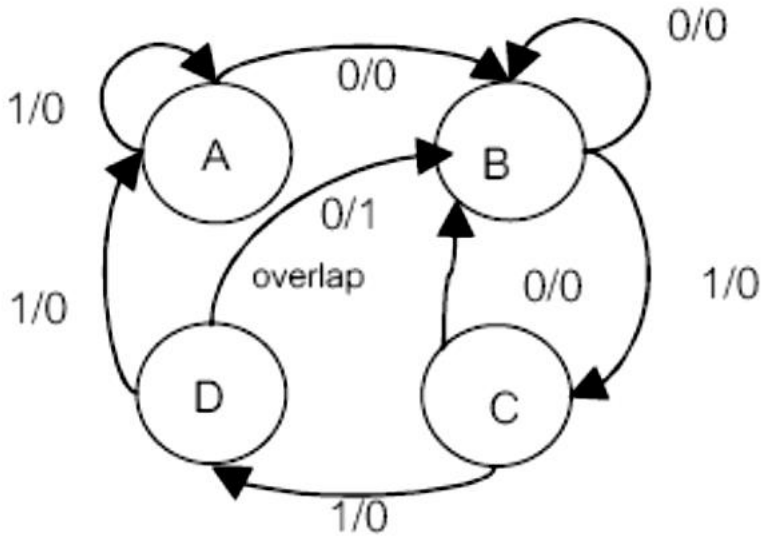
1	PS \ x	0	1
A	00	01/0	00/0
B	01	01/0	11/0
C	11	01/0	10/0
D	10	00/1	00/0

Complete Sequential Circuit Design

b) Overlapping: If x = 0110110110

The circuit will produce z = 0001001001

1. draw the state diagram



2. Write the state tables

PS \ X	0	1
A	B/0	A/0
B	B/0	C/0
C	B/0	D/0
D	B/1	A/0

NS/Z

Complete Sequential Circuit Design

3. Do State Assignment

- Rule 1: The same NS: Give the PS of A-B, B-C, C-D, A-D and B-D.
- Rule 2: The same PS: Give the NS of ----.

y_1 \ y_2	0	1
0	A	B
1	D	C

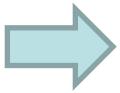
So, we must assign the states by the codes:

A = 00, B = 01, C = 11, D = 10.

And get the transition table:

PS \ X	0	1
A	B/0	A/0
B	B/0	C/0
C	B/0	D/0
D	B/1	A/0

NS/Z



1	PS \ x	0	1
A	00	01/0	00/0
B	01	01/0	11/0
C	11	01/0	10/0
D	10	01/1	00/0

Complete Sequential Circuit Design

Example: Design a Synchronous sequential Circuit used as a Controller for Robot that can travel without colliding with obstacles. It using the sensor installed in front of the robot.

- The input x = 1 when there is an obstacle,

and x = 0 when there is no obstacles.

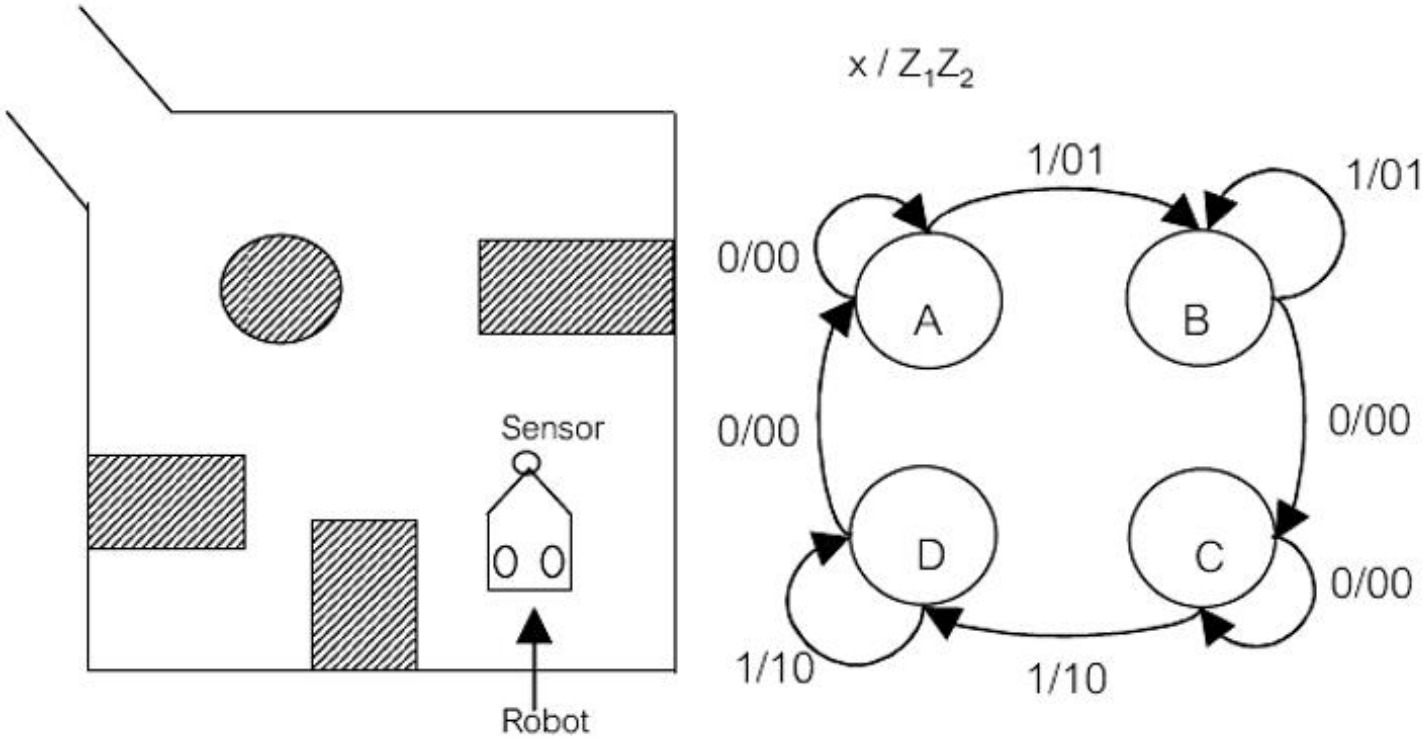
- The output $z_1 = 1$ and $z_2 = 1$ to control the left

and the right turning respectively.

- When Robot finds an obstacle, it will turn right until no obstacle was found.
- If it subsequently found another obstacle, it would turn left until no obstacle was found, as in Figure.
- The robot controller uses 4 total states:

- State A = No obstacle is detected with the last turn is left;
- State B = Obstacle is found and going to turn right;
- State C = No obstacles found with the last turn is right;
- State D = Obstacle is found and going to turn left.

Complete Sequential Circuit Design

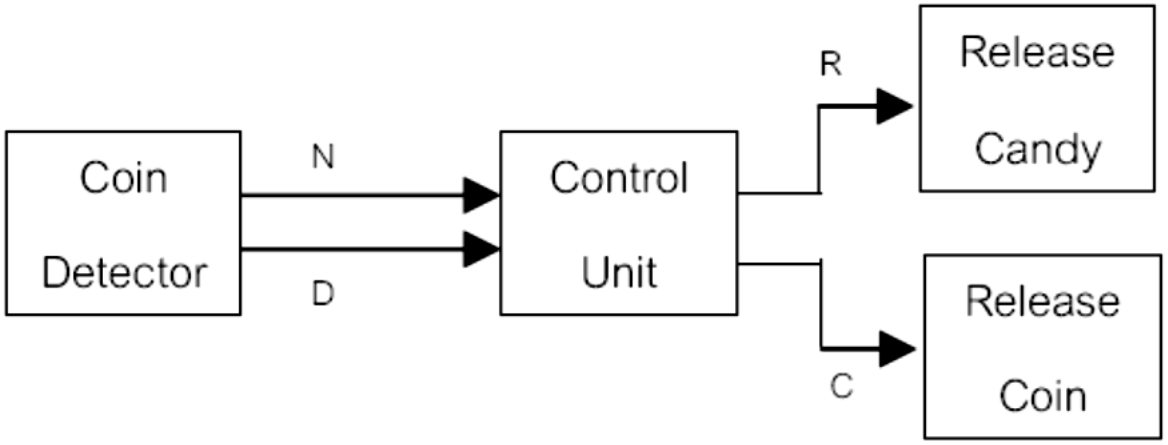


- The controller start to work at state A.
- It does not turn until an obstacle is found, then go to state B
- It would turn right until no obstacles are detected,
- then go to state C and stop turning right.
- The controller stays in state C until it encounters another obstacle.
- Then go to state D and turn left until no obstacles are detected.
- Then go to state A.

Complete Sequential Circuit Design

Example: Design a Synchronous sequential circuit that controls a can vending machine.

- The can price is 20 Baht.
- This vending machine will accept only 5Baht coins and 10Baht coins .
- The machine can give the change if we insert more than 20 Baht
- and the maximum number of coins per purchase is no more than 25 Baht.
- Then there will be a change of 5 Baht as in the Figure.



Complete Sequential Circuit Design

- When insert 5Baht Coin, N = 1.
- When insert 10Baht Coin, D = 1.
- And the signal N, D will Reset = 0 automatically when the next clock comes (assuming that N, D cannot be added at the same time).
- N, D signals do not occur during a same single clock.
- The control unit have an R and C signals for selling candy and giving the change money, respectively.
- The number of the states of the control unit will be the same as the amount of the money inserting in the machine i.e., 0, 5,10 and 15 Baht.
- When the amount of the money reaches to 20 or 25 Baht, the device will return to state 0 and release the candy, or giving the change, respectively.
- The stale diagram is illustrated below.