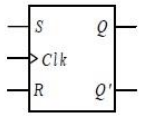
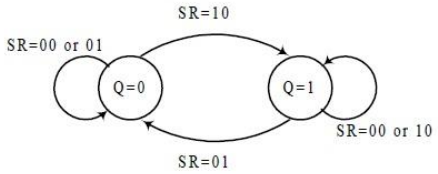
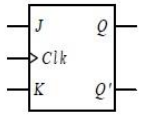
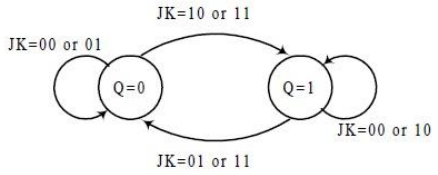
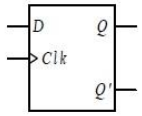
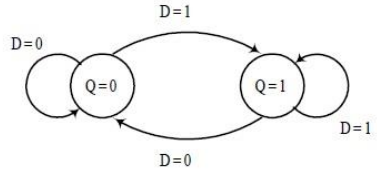
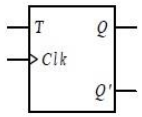
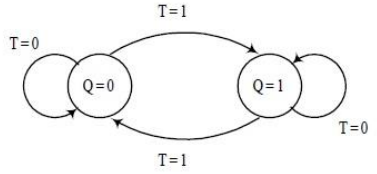
Course Name: Digital Hardware Design
Course Code: 17B1NEC741

Finite State Machine-4

Dr. Arti Noor
Dean, Academic Affairs
Electronics and Communication Engineering,
Jaypee Institute of Information Technology, Noida

Sequential Storage Units

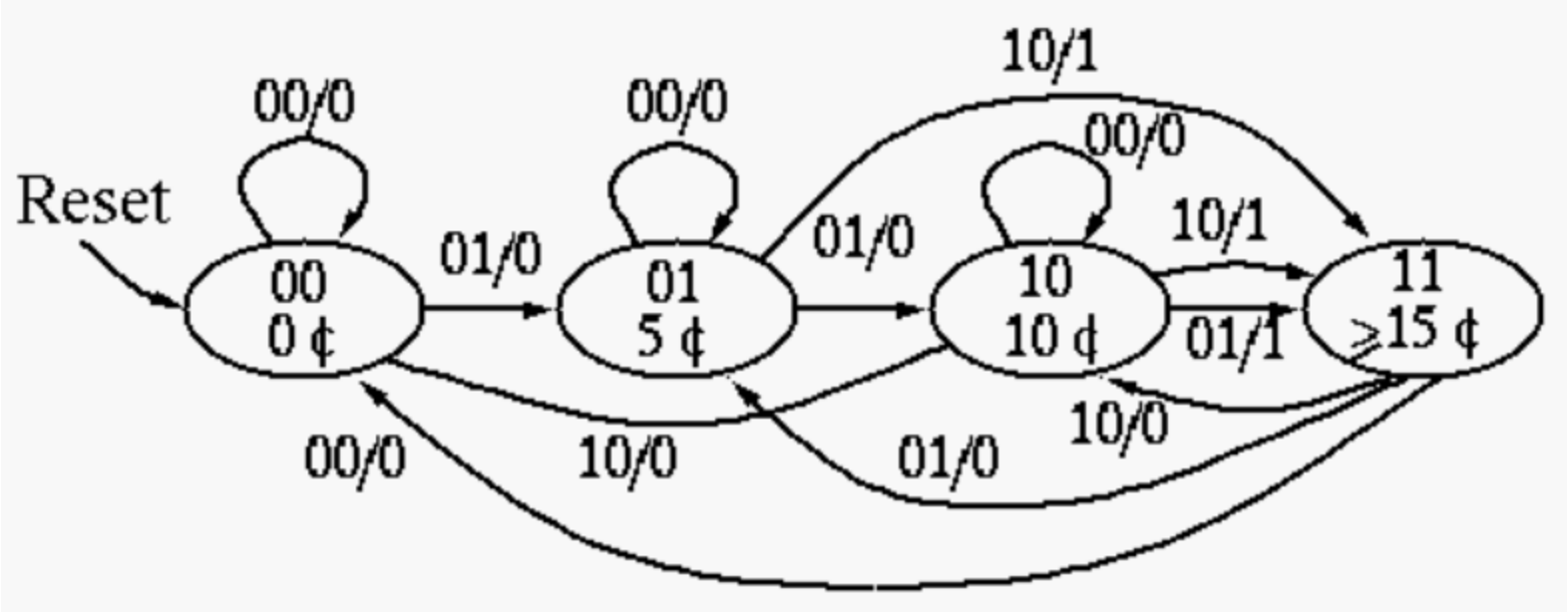
Name / Symbol	Characteristic (Truth) Table	State Diagram / Characteristic Equations	Excitation Table
SR 	$S\ R\ Q\ Q_{next}$ 0 0 0 0 0 0 1 1 0 1 0 0 0 1 1 0 1 0 0 1 1 0 1 1 1 1 0 × 1 1 1 ×	 $Q_{next} = S + R'Q$ $SR = 0$	$Q\ Q_{next}\ S\ R$ 0 0 0 × 0 1 1 0 1 0 0 1 1 1 × 0
JK 	$J\ K\ Q\ Q_{next}$ 0 0 0 0 0 0 1 1 0 1 0 0 0 1 1 0 1 0 0 1 1 0 1 1 1 1 0 1 1 1 1 0	 $Q_{next} = J'K'Q + JK' + JKQ'$ $= J'K'Q + JK'Q + JK'Q' + JKQ'$ $= K'Q(J'+J) + JQ'(K'+K)$ $= K'Q + JQ'$	$Q\ Q_{next}\ J\ K$ 0 0 0 × 0 1 1 × 1 0 × 1 1 1 × 0
D 	$D\ Q\ Q_{next}$ 0 × 0 1 × 1	 $Q_{next} = D$	$Q\ Q_{next}\ D$ 0 0 0 0 1 1 1 0 0 1 1 1
T 	$T\ Q\ Q_{next}$ 0 0 0 0 1 1 1 0 1 1 1 0	 $Q_{next} = TQ' + T'Q = T \oplus Q$	$Q\ Q_{next}\ T$ 0 0 0 0 1 1 1 0 1 1 1 0

FSM Design Example-2

A vending machine will open the door to let out a pack of gum if it receives 15 cents or more in denominations of 5 (Dime) and 10 (Nickel) cents. Consider one input at a time. No return.

Input Sequence:

1. 3 Dimes(15)
2. 2 Dimes & Nickel(20)
3. Dime, Nickle(15)
4. Nickle, Dime(15)
5. 2 Nickles (20)



FSM Design Example-2

Present State		Input		Next State		Output	Signal to Trigger FFs									
Q1(t)	Q0(t)	N	D	Q1(t+1)	Q0(t+1)	Out	S1	R1	S0	R0	J1	K1	J0	K0	T1	T2
0	0	0	0	0	0	0	0	x	0	x	0	x	0	x	0	0
0	0	0	1	0	1	0	0	x	1	0	0	x	1	x	0	1
0	0	1	0	1	0	0	1	0	0	x	1	x	0	x	1	0
0	0	1	1	x	x	x	x	x	x	x	x	x	x	x	x	x
0	1	0	0	0	1	0	0	x	x	0	0	x	x	0	0	0
0	1	0	1	1	0	0	1	0	0	1	1	x	x	1	1	1
0	1	1	0	1	1	1	1	0	x	0	1	x	x	0	1	0
0	1	1	1	x	x	x	x	x	x	x	x	x	x	x	x	x
1	0	0	0	1	0	0	x	0	0	x	x	0	0	x	0	0
1	0	0	1	1	1	1	x	0	1	0	x	0	1	x	0	1
1	0	1	0	1	1	1	x	0	1	0	x	0	1	x	0	1
1	0	1	1	x	x	x	x	x	x	x	x	x	x	x	x	x
1	1	0	0	0	0	0	0	1	0	1	x	1	x	1	1	1
1	1	0	1	0	1	0	0	1	x	0	x	1	x	0	1	0
1	1	1	0	1	0	0	x	0	0	1	x	0	x	1	0	1
1	1	1	1	x	x	x	x	x	x	x	x	x	x	x	x	x

State Assignment

- Some state assignments are better than others.

- The state assignment influences the complexity of the state machine.
  - The combinational logic required in the state machine design is dependent on the state assignment.

- Types of state assignment

– Binary encoding
– Gray-code encoding
– One-hot encoding

Example

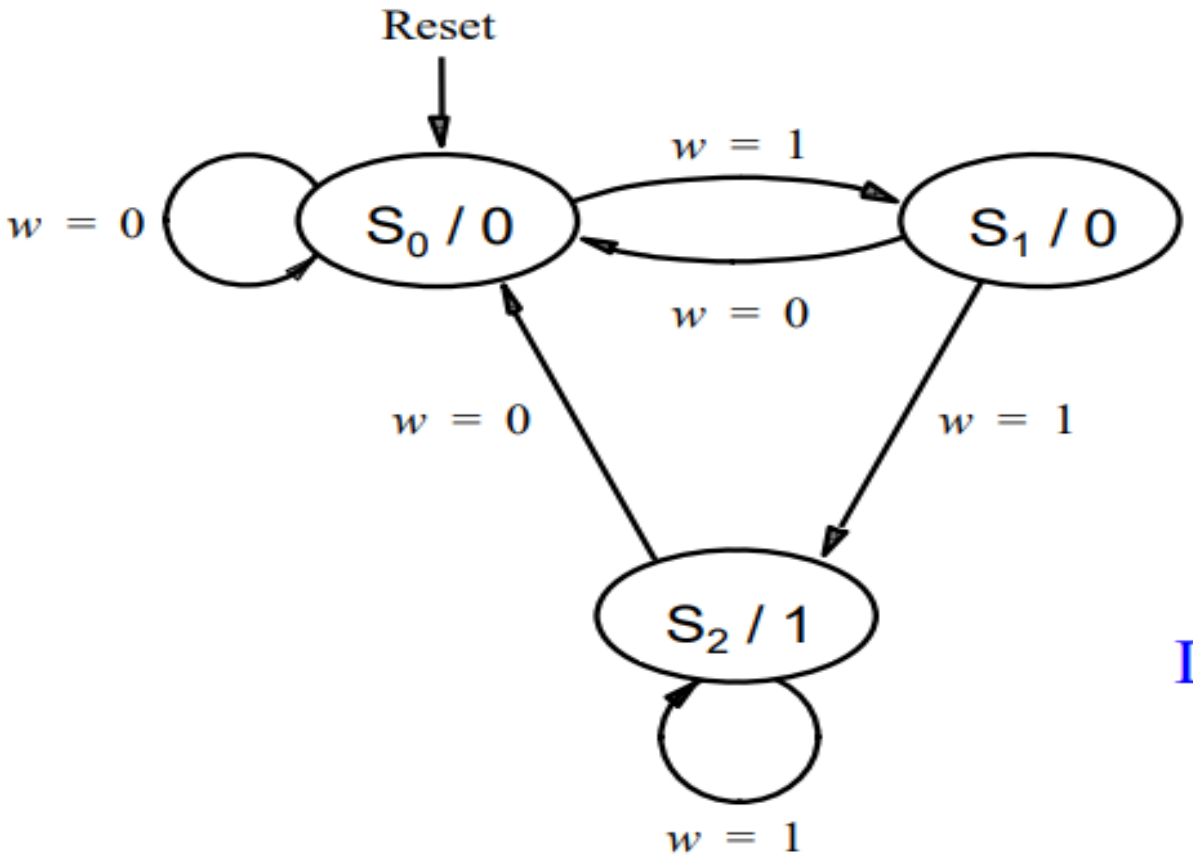
Design a FSM that detects a sequence of two or more consecutive ones on an input bit stream.

The FSM should output a 1 when the sequence is detected, and a 0 otherwise.

Input: 0 1 1 1 0 1 0 1 1 0 1 1 1 0 1 …

Output: 0 0 1 1 0 0 0 0 1 0 0 1 1 0 0 …

Example



State Diagram

Example: State Assignment 1

Present State			Next State						Output
			w = 0			w = 1			
	Q_A	Q_B		Q_A^+	Q_B^+		Q_A^+	Q_B^+	z
S_0	0	0	S_0	0	0	S_1	0	1	0
S_1	0	1	S_0	0	0	S_2	1	0	0
S_2	1	0	S_0	0	0	S_2	1	0	1
	1	1		d	d		d	d	d

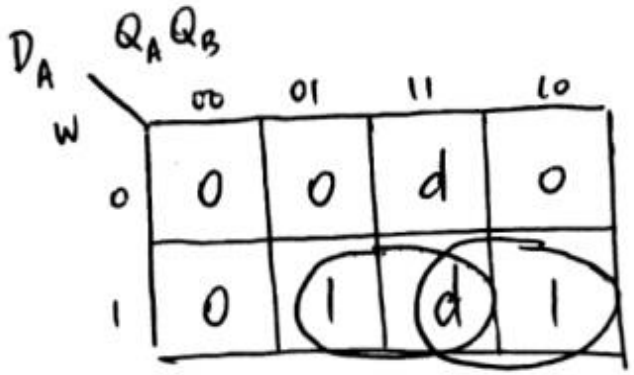
Using Binary Encoding for the State Assignment

Example: State Assignment 1

Present State			Next State				FF Inputs			
			w = 0		w = 1		w = 0		w = 1	
	Q_A	Q_B	Q_A^+	Q_B^+	Q_A^+	Q_B^+	D_A	D_B	D_A	D_B
S_0	0	0	0	0	0	1	0	0	0	1
S_1	0	1	0	0	1	0	0	0	1	0
S_2	1	0	0	0	1	0	0	0	1	0
	1	1	d	d	d	d	d	d	d	d

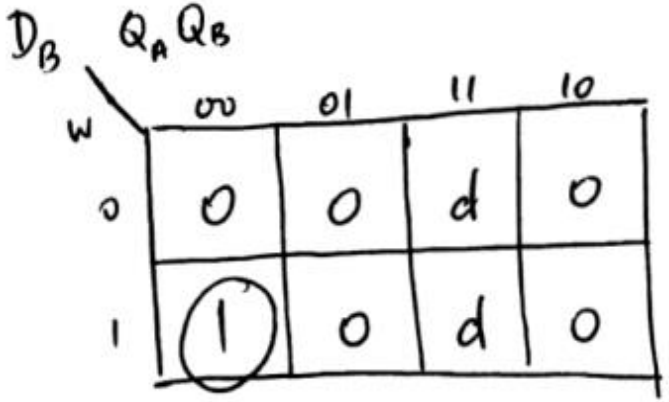
Characteristic Equation: $D = Q^+$

Example: State Assignment 1

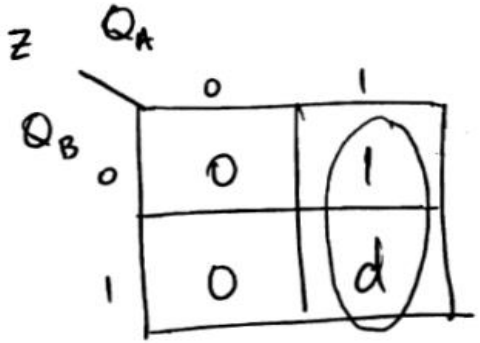


$$D_A = W \cdot Q_B + W \cdot Q_A$$

$$D_A = W \cdot (Q_A + Q_B)$$

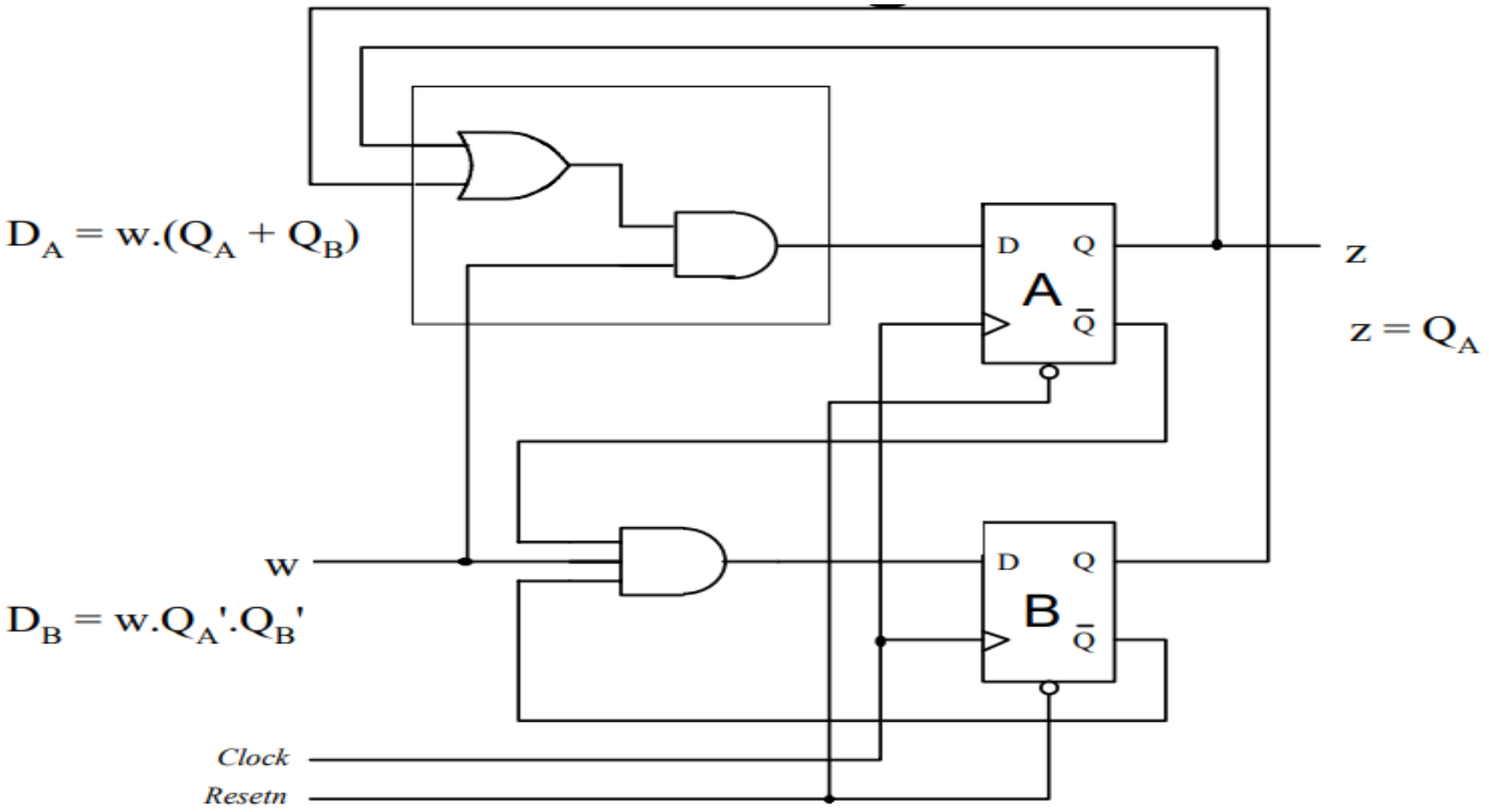


$$D_B = W \cdot \overline{Q}_A \cdot \overline{Q}_B$$



$$Z = Q_A$$

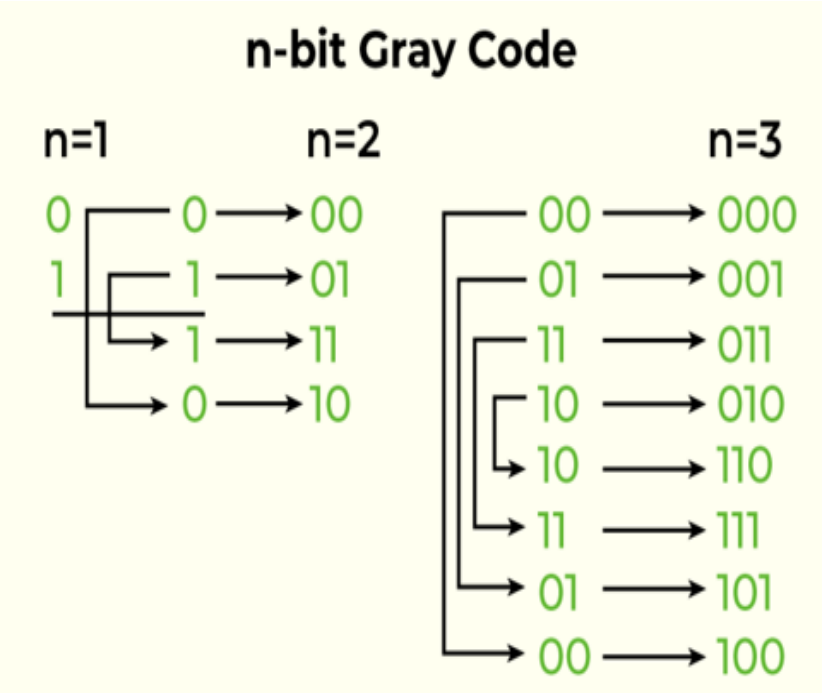
Example: State Assignment 1



Example: State Assignment 2

Gray code consists of a sequence where only one bit changes between one value and the next. In addition to also using the minimum number of bits, this encoding minimizes dynamic power consumption if the sequence of states is followed optimally.

Example: State Assignment 2



Present State			Next State						Output
			w = 0			w = 1			
	Q_A	Q_B		Q_A^+	Q_B^+		Q_A^+	Q_B^+	z
S_0	0	0	S_0	0	0	S_1	0	1	0
S_1	0	1	S_0	0	0	S_2	1	1	0
S_2	1	1	S_0	0	0	S_2	1	1	1
	1	0		d	d		d	d	d

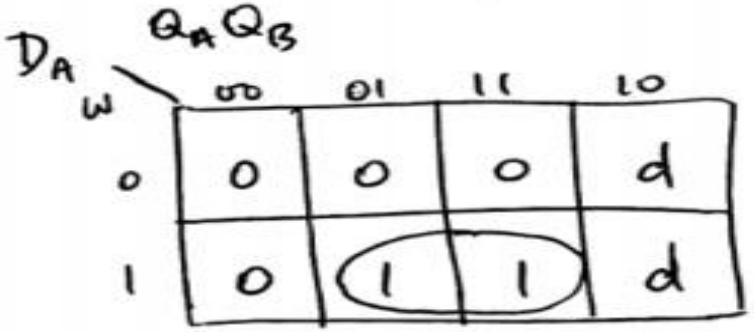
Using Gray-code Encoding for the State Assignment

Example: State Assignment 2

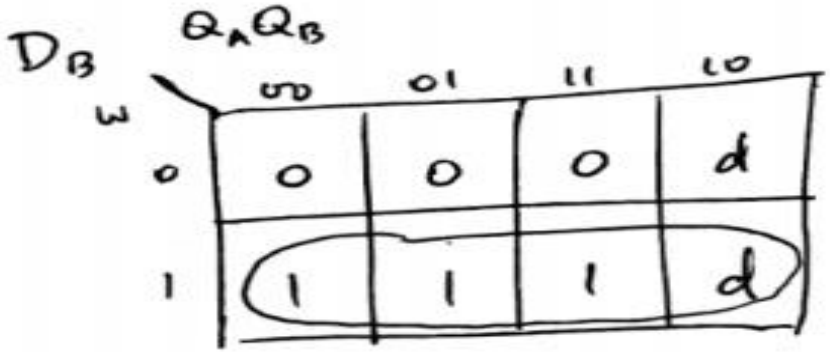
Present State			Next State				FF Inputs			
			w = 0		w = 1		w = 0		w = 1	
	Q_A	Q_B	Q_A^+	Q_B^+	Q_A^+	Q_B^+	D_A	D_B	D_A	D_B
S_0	0	0	0	0	0	1	0	0	0	1
S_1	0	1	0	0	1	1	0	0	1	1
S_2	1	1	0	0	1	1	0	0	1	1
	1	0	d	d	d	d	d	d	d	d

Characteristic Equation: $D = Q^+$

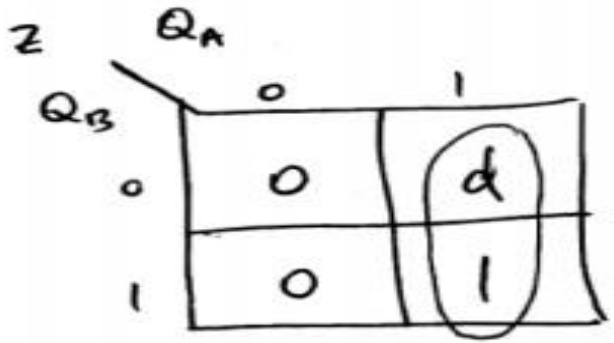
Example: State Assignment 2



$D_A = w \cdot Q_B$

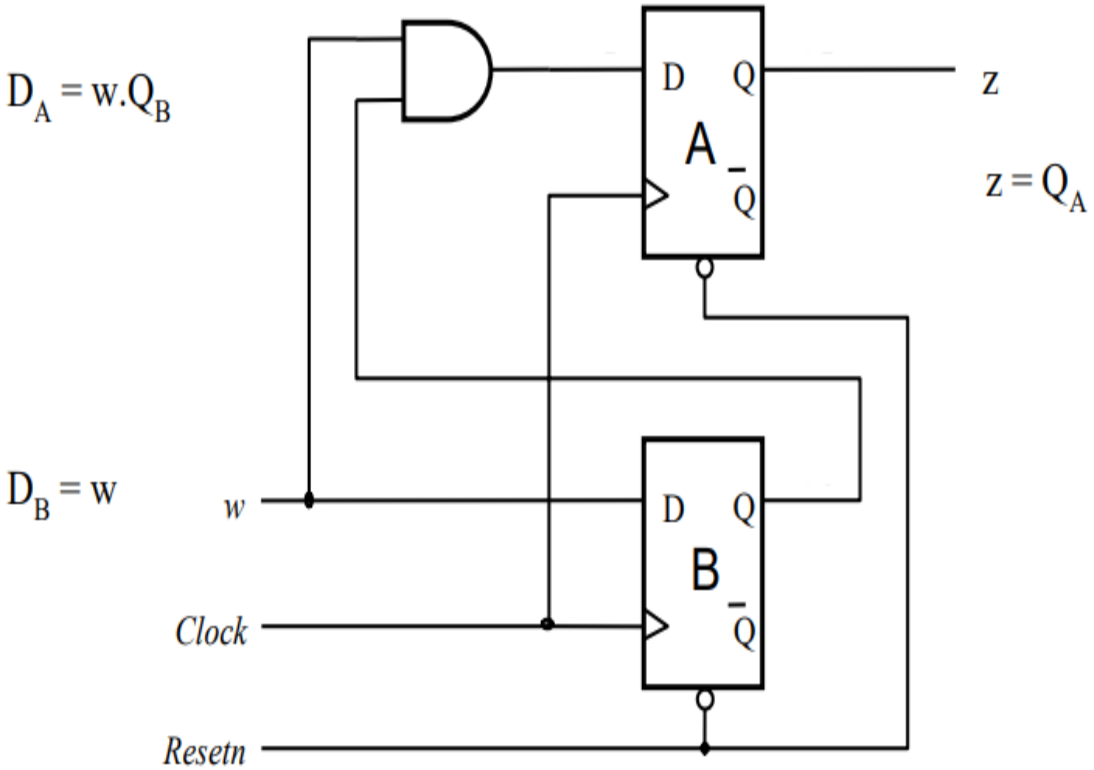


$D_B = w$

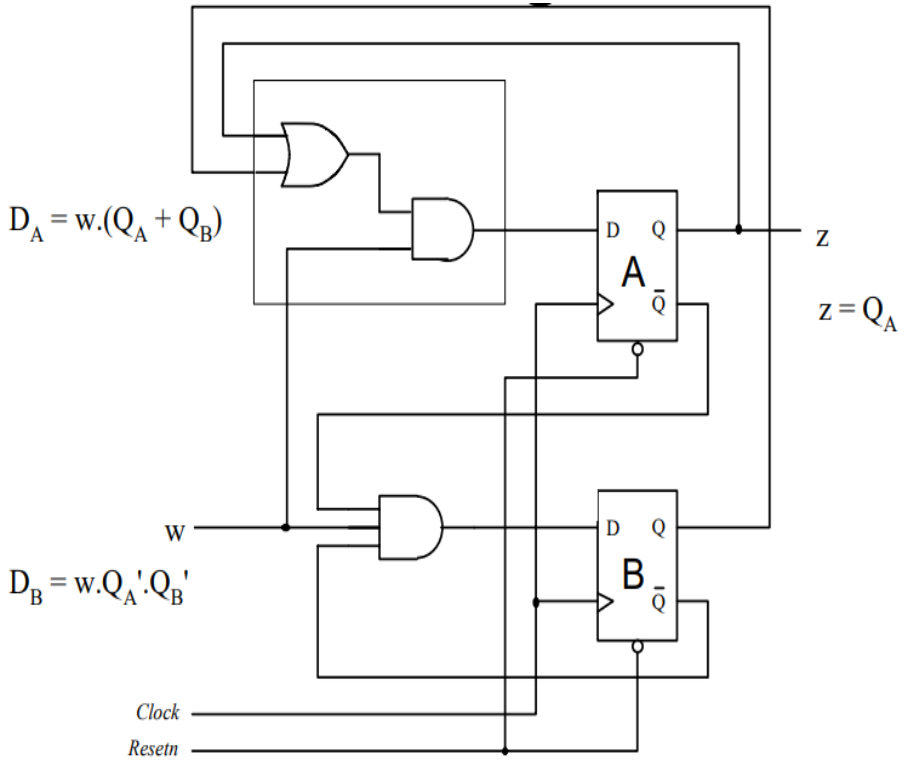


$z = Q_A$

Example: State Assignment 2



Gray Code



Binary Code

Example: State Assignment 3

One-Hot coding

- Encode n states using n flip-flops.
- Assign a single “1” for each state. Example: 0001 0010 0100 1000.
- Propagate a single “1” from one flip-flop to the next .All other flip-flop outputs are “0”

Example: State Assignment 3

Present State				Next State							
				w = 0				w = 1			
	Q_A	Q_B	Q_C		Q_A^+	Q_B^+	Q_C^+		Q_A^+	Q_B^+	Q_C^+
S_0	0	0	1	S_0	0	0	1	S_1	0	1	0
S_1	0	1	0	S_0	0	0	1	S_2	1	0	0
S_2	1	0	0	S_0	0	0	1	S_2	1	0	0

Using One-hot Encoding for the State Assignment

For each state only one flip-flop is set to 1.
The remaining combination of state variables are not used.

Characteristic Equation: $D = Q^+$

Example: State Assignment 3

D_A

wQ_A \ Q_BQ_C	00	01	11	10
00	d	0	d	0
01	0	d	d	d
11	1	d	d	d
10	d	0	d	1

$D_A = w \cdot \bar{Q}_C$

D_B

wQ_A \ Q_BQ_C	00	01	11	10
00	d	0	d	0
01	0	d	d	d
11	0	d	d	d
10	d	1	d	0

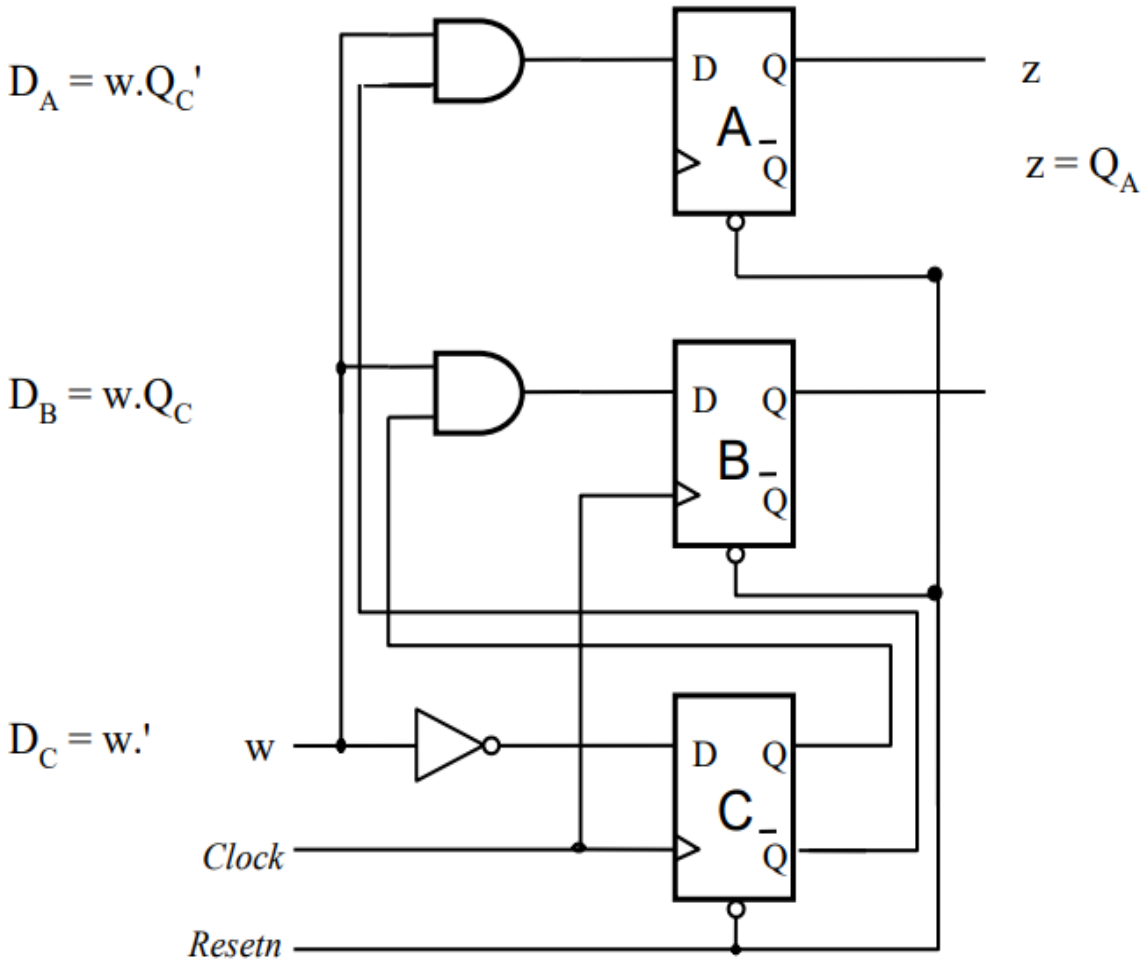
$D_B = w \cdot Q_C$

D_C

wQ_A \ Q_BQ_C	00	01	11	10
00	d	1	d	1
01	1	d	d	d
11	0	d	d	d
10	d	0	d	0

$D_C = \bar{w}$.

Example: State Assignment 3



- This may not seem very efficient at first because of the number of bits used, and the excessive number of invalid states.
- However, one-hot encoding is very good at simplifying the stimulus logic for the flip flops because there's no need to decode the states. The bits are the states.