ME311 | Microprocessors & Automatic Control

Tutorial 1 Solutions

① Assuming,
Initially Both switches (S_1, S_2) are "off" & Light (L) is "off"

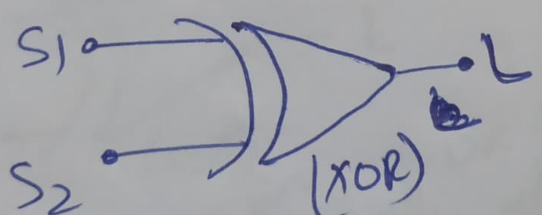
Switch(S_1)	Switch(S_2)	Light (L)
0	0	0
1	0	1
1	1	0
0	1	1

Initial →

$$S_1\bar{S_2} + S_2\bar{S_1} = L$$

(or) $L = S_1 \oplus S_2$

Logical circuit



In our homes, for a two way switch we can either use "AOI" (or) "XOR" logic. So possibly "yes."

(Method 1)

(2) We know that

(a) Assumptions

* $C_1, C_0 \neq 0,0$
* If $C_1, C_0 = 0,1$ Remainder $R_1, R_0 = 0,0$
* If $D_2 = 0$ & $C_1, C_0 = D_1, D_0$ Remainder $R_1, R_0 = 0\,0$.
* $D_2 D_1 D_0 > C_1 C_0$

Note: $(D_2 D_1 D_0)_{max} = 7 \Rightarrow 111$

$(C_1 C_0)_{max} = 3 \Rightarrow 11$

So remaining Cases possible are.

(remainder of %)	$D_2 D_1 D_0$	$C_1 C_0$	R_1 R_0
3%2	0 1 1	1 0	0 1
4%2	1 0 0	1 0	0 0
4%3	1 0 0	1 1	0 1
5%2	1 0 1	1 0	0 1
5%3	1 0 1	1 1	1 0
6%2	1 0 1	1 0	0 0
6%3	1 1 0	1 1	0 0
7%2	1 1 0	1 0	0 1
7%3	1 1 1	1 1	0 1

Cases

If $D_2 D_1 D_0 < C_1 C_0$ Remainder is always "$C_1 C_0$" i.e "$R_1 R_0$" = "$C_1 C_0$"

we can include (or) neglect for our Convenience.

To Include this, we can simply follow the method (2)

"K maps" is not must as sir mentioned (In class)

Two outputs

R_1, R_0

(b) For R_1

$D_2 D_1 D_0$	$C_1 C_0$	R_1
1 0 1	1 1	1

$R_1 = D_2 \overline{D_1} D_0 C_1 C_0$

For R_0

$D_2 D_1 D_0$	$C_1 C_0$	R_0
0 1 1	1 0	1
1 0 0	1 1	1
1 0 1	1 0	1
1 1 1	1 0	1
1 1 1	1 1	1

$R_0 = \overline{D_2} D_1 D_0 C_1 \overline{C_0}$
$+ D_2 \overline{D_1}\,\overline{D_0} C_1 C_0$
$+ D_2 \overline{D_1} D_0 C_1 \overline{C_0}$
$+ D_2 D_1 D_0 C_1 \overline{C_0}$
$+ D_2 D_1 D_0 C_1 C_0$ $\rightarrow 1$

$\boxed{\text{Mehod 2}}$

② Simply assume $C_1 C_0 \neq 00$ among ~~00~~ 01, 10, 11.

"Brute force" tech

ⓐ

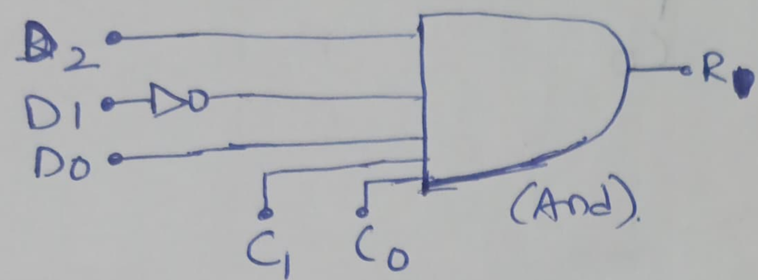
D_2	D_1	D_0	C_1	C_0	R_1	R_0
0	0	0	0	1	0	0
0	0	0	1	0	0	0
0	0	0	1	1	0	0
0	0	1	0	1	0	0
0	0	1	1	0	0	1
0	0	1	1	1	0	1
0	1	0	0	1	0	0
0	1	0	1	0	0	0
0	1	0	1	1	1	0
1	0	0	0	1	0	0
1	0	0	1	0	0	0
1	0	0	1	1	0	1
0	1	1	0	1	0	0
0	1	1	1	0	0	1
0	1	1	1	1	0	0
1	0	1	0	1	0	0
1	0	1	1	0	0	1
1	0	1	1	1	1	0
1	1	0	0	1	0	0
1	1	0	1	0	0	0
1	1	0	1	1	0	0
1	1	1	0	1	0	0
1	1	1	1	0	0	1
1	1	1	1	1	0	1

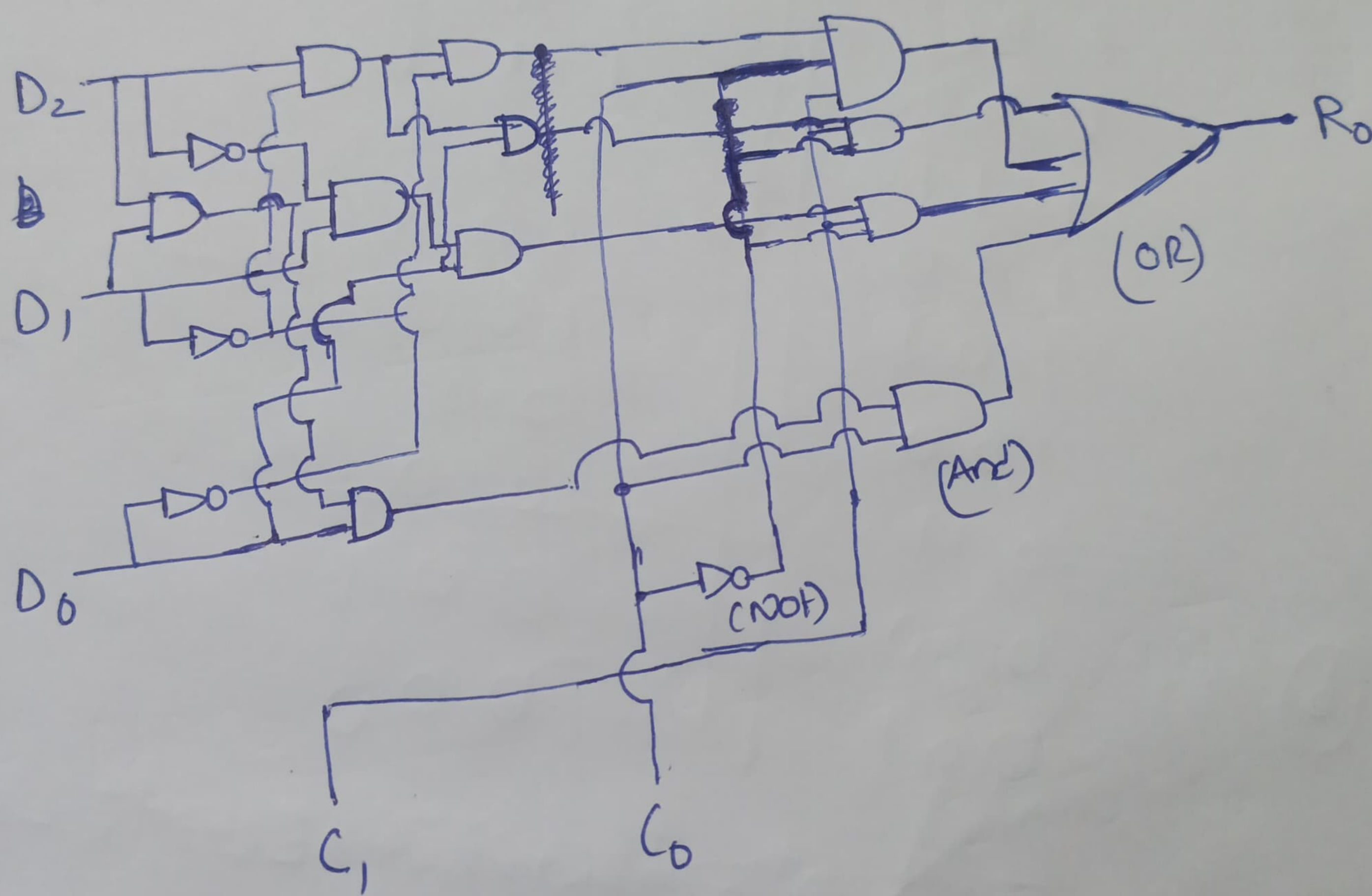
ⓑ

$$R_1 = \bar{D}_2 D_1 \bar{D}_0 C_1 C_0 + D_2 \bar{D}_1 D_0 C_0 C_1$$

$$R_0 = \bar{D}_2 \bar{D}_1 D_0 C_1 \bar{C}_0 + \bar{D}_2 \bar{D}_1 D_0 C_1 C_0 + \bar{D}_2 D_1 D_0 C_1 \bar{C}_0 + D_2 \bar{D}_1 \bar{D}_0 C_1 C_0 + D_2 \bar{D}_1 D_0 C_1 \bar{C}_0 + D_2 D_1 D_0 C_1 C_0 + D_2 D_1 D_0 C_1 \bar{C}_0$$

2(c) Method 1 Circuit Schematic

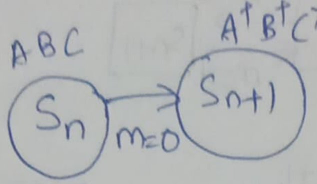


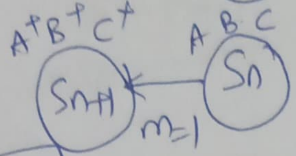


Similarly for method 2

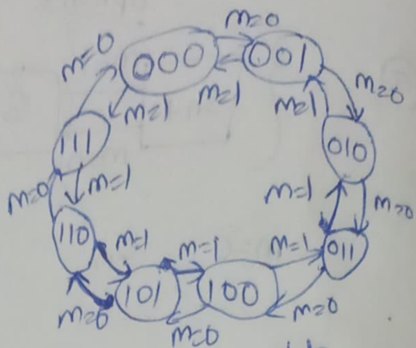
③ **States**

ⓐ





In a simpler way



Complete state diagram

Truth table

m=0

S_n (Input) A	B	C	S_{n+1} (output) A^+	B^+	C^+
0	0	0	0	0	1
0	0	1	0	1	0
0	1	0	0	1	1
0	1	1	1	0	0
1	0	0	1	0	1
1	0	1	1	1	0
1	1	0	1	1	1
1	1	1	0	0	0

m=1

S_n (Input) A	B	C	S_{n+1} (output) A^+	B^+	C^+
0	0	0	1	1	1
1	1	1	1	1	0
1	1	0	1	0	1
1	0	1	1	0	0
1	0	0	0	1	1
0	1	1	0	1	0
0	1	0	0	0	1
0	0	1	0	0	0

ⓑ

For $m=0 \Rightarrow \bar{m}$

$A^+ = \bar{A}BC + A\bar{B}\bar{C} + A\bar{B}C + AB\bar{C} = A\bar{B} + B(A\oplus C)$

$B^+ = \bar{A}\bar{B}C + \bar{A}B\bar{C} + A\bar{B}C + AB\bar{C} = B\bar{C} + \bar{B}C$

$C^+ = \bar{A}\bar{B}\bar{C} + \bar{A}B\bar{C} + A\bar{B}\bar{C} + AB\bar{C} = A\bar{C} + \bar{A}\bar{C}$

$= \bar{C}$

For $m=1$

$$A^+ = \bar{A}\bar{B}\bar{C} + ABC + AB\bar{C} + AC\bar{B} = AB + \bar{B}$$

$$B^+ = \bar{A}\bar{B}\bar{C} + ABC + A\bar{B}\bar{C} + \bar{A}BC = \bar{B}\bar{C} + BC$$

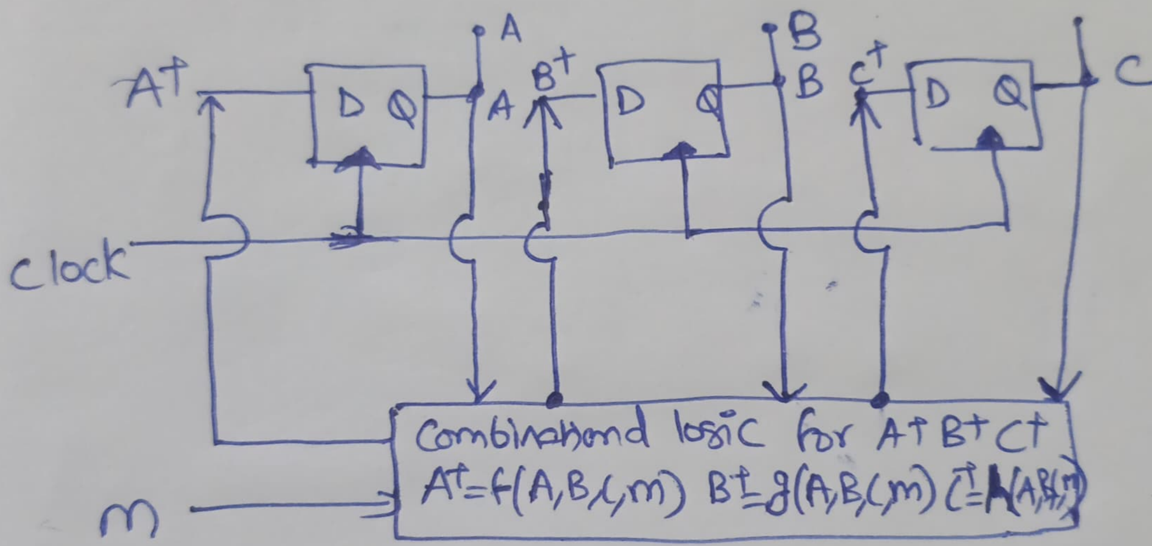
$$C^+ = \bar{A}\bar{B}\bar{C} + AB\bar{C} + \bar{A}\overline{BC} + A\bar{B}\bar{C} = \bar{C}$$

$$\Rightarrow A^+ = \bar{m}\left(A\bar{B} + B(A \oplus C)\right) + m(AB + \bar{B})$$

$$B^+ = \bar{m}(B\bar{C} + \bar{B}C) + m(\bar{B}\bar{C} + BC)$$

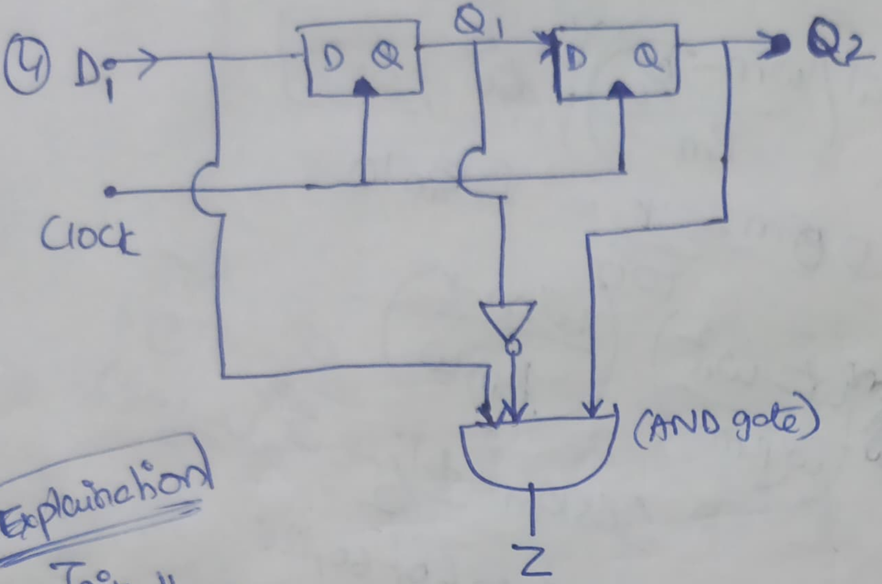
$$C^+ = \bar{m}(\bar{C}) + m(\bar{C}) = \bar{C}$$

(c)



A^+, B^+, C^+ are evaluated using combinational logic from current values of "A, B, C" & "m"

(4)



"D_1" is the input comes continuously for every clock cycle

Explanation

Initially (Let's assume sequence to be exactly as "101") (ie $D_1 \to 1 \to 0 \to 1$)

Clock	D_1	Q_1	Q_2	Z
	1	0	0	0
0	0	1	0	0
1	1	0	1	1

$Z = D_1 Q_1' Q_2$

In order to get our output high, we need to use current value of "D_1" & remember earlier "2" values of "D_1", So two flipflops are needed to remember.