HMK # 11

2a)

D_1	D_0
1	0
0	1
1	1
0	0

Present State Q_1	Q_0
0	0
1	0
0	1
1	1

Next State Q_1^+	Q_0^+
1	0
0	1
1	1
0	0

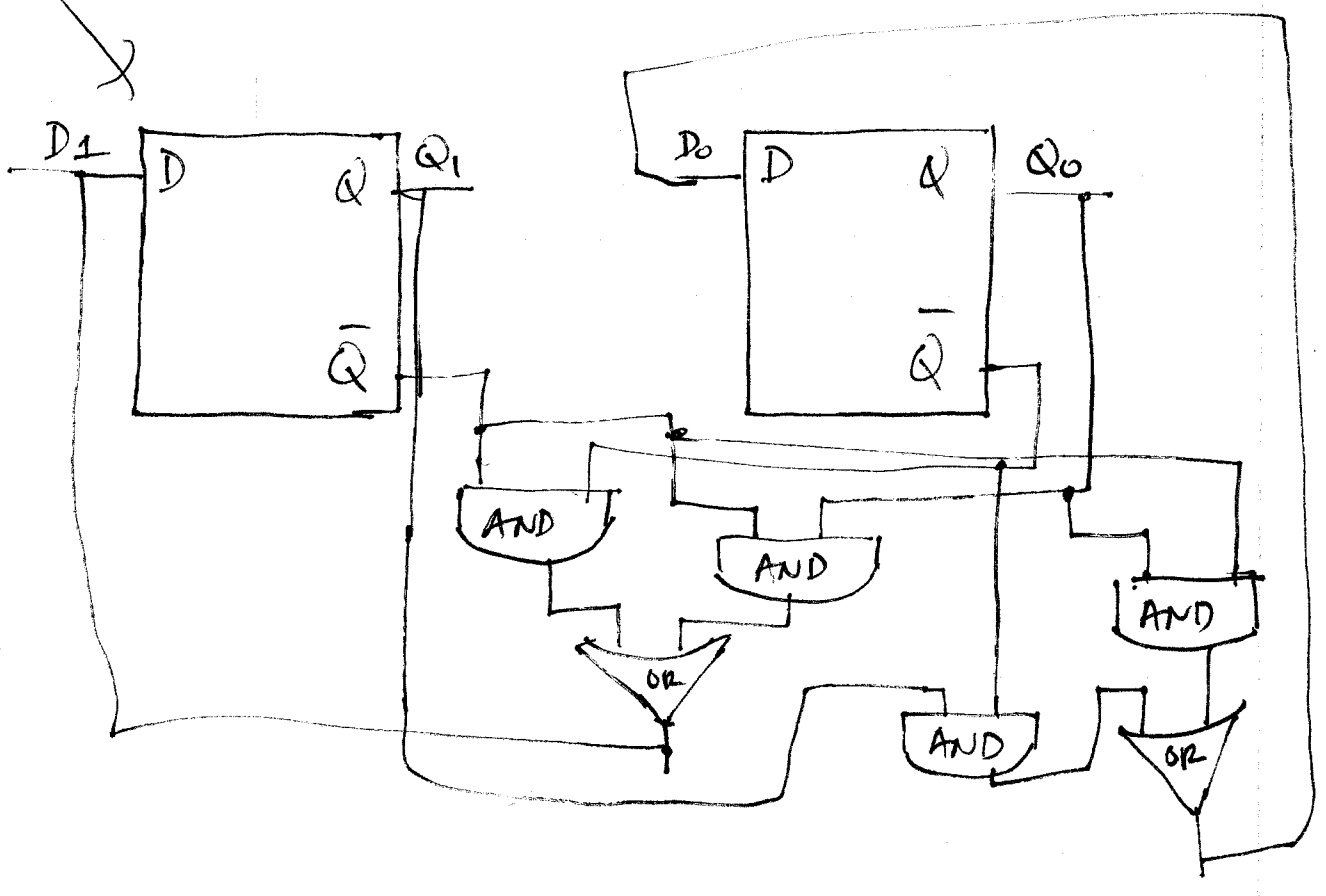
if you are asked to design anything you will need to identify design steps.

Sequence of States = 00, 10, 01, 11

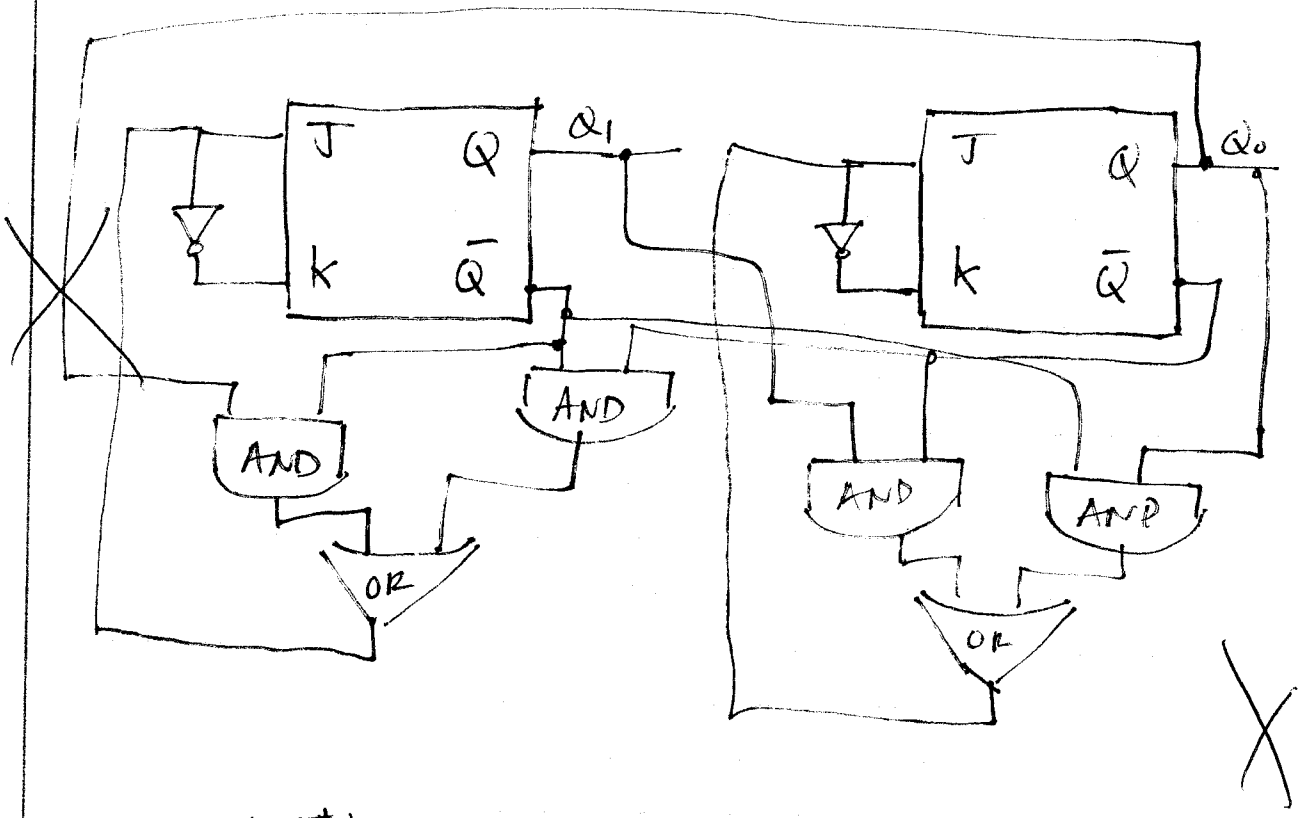
OUTPUT is G_0G_1.

$D_1 = \bar{Q}_1 \cdot \bar{Q}_0 + \bar{Q}_1 \cdot Q_0$

$D_0 = Q_1 \cdot \bar{Q}_0 + \bar{Q}_1 \cdot Q_0$



2b) By Converting JK flip flops to D-flip-flops we can use the last design



IN This problem I expect you to show that you can design a counter with J and K, not adapting the D design.

3a)

step 1

$D_A = \overline{T \cdot Q_A}$ $\quad D_B = \bar{Q}_A \oplus \bar{Q}_B$ $\quad D = Q_B + Q_A$

step 2

T=0

T	D_A	Q_A	$\bar{Q}_A$	Q_A^+	D_B	Q_B	$\bar{Q}_B$	Q_B^+	D
0	1	0	1	1	0	0	1	0	0
0	1	1	0	1	1	0	1	1	1
0	1	1	0	1	0	1	0	0	1
0	1	1	0	1	1	0	1	1	1

State Q_A	State Q_B
0	0
1	0
1	1
1	0
⋮	

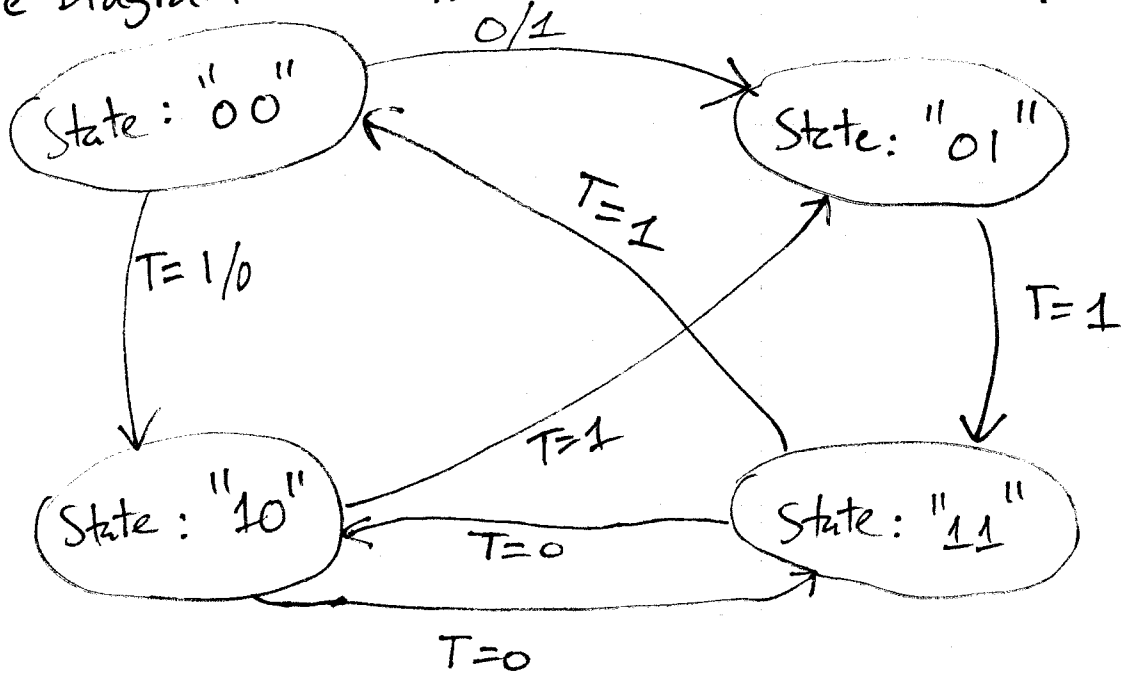
T=1

T	D_A	Q_A	$\bar{Q}_A$	Q_A^+	D_B	Q_B	$\bar{Q}_B$	Q_B^+	D
1	1	0	1	1	0	0	1	0	0
1	0	1	0	0	1	0	1	1	1
1	1	0	1	1	1	1	0	1	1
1	0	1	0	0	0	1	0	0	1

State Q_A	State Q_B
0	0
1	0
0	1
1	1
0	0
⋮	

State Diagram

step 3



3b) X.

D_A	Q_A	$\bar{Q_A}$	Q_A^+
1	0	1	1
0	1	0	0
1	0	1	1
1	1	0	1

D_B	Q_B	$\bar{Q_B}$	Q_B^+
0	0	1	0
1	0	1	1
1	1	0	1
0	1	0	0

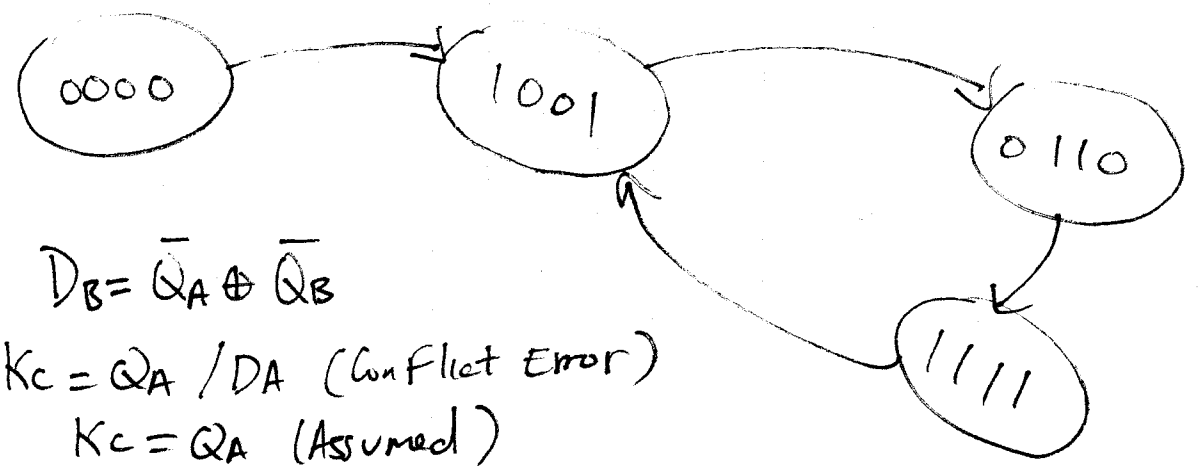
J_C	K_C	Q_C	$\bar{Q_C}$	Q_C^+
0	0	0	1	0
1	1	0	1	1
1	0	1	0	1
0	1	1	0	0

J_D	K_D	Q_D	$\bar{Q_D}$	Q_D^+
1	1	0	1	1
1	1	1	0	0
0	1	0	1	1
0	0	1	0	1

State

Q_A	Q_B	Q_C	Q_D
0	0	0	0
1	0	0	1
0	1	1	0
1	1	1	1
1	0	0	1
⋮			

State Machine.



J	K	Q_{next}
0	0	Q
0	1	0
1	0	1
1	1	$\bar{Q}$

$D_A = \overline{Q_D \cdot \bar{Q}_C}$ $\quad D_B = \bar{Q}_A \oplus \bar{Q}_B$

$J_C = D_B$ $\quad K_C = Q_A$ / D_A (Conflict Error)

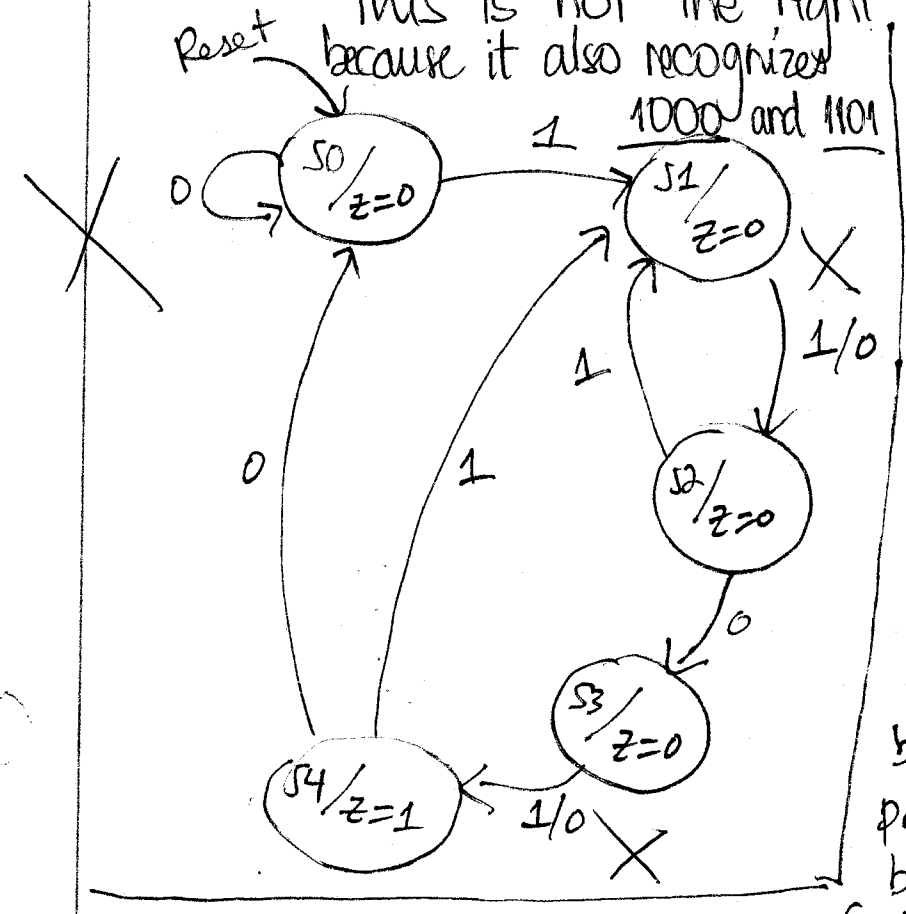
$J_D = D_A \oplus Q_D$ $\quad K_C = Q_A$ (Assumed)

$K_D = \bar{J}_D$

Analysis: you need to show 3 step process

(4) Word → 1100 or 1001

This is not the right state machine because it also recognizes 1000 and 1101




```
Entity Sequence is
port(Clock, Reset, w: in std_logic;
     z: out std_logic);
end sequence

Architecture behavior of sequence is
type state_type is
(S0, S1, S2, S3, S4);
signal y: state_type
begin
process (Reset, Clock)
begin
if Reset = '1' then y <= S0;
elsif (Clock'Event and Clock = '1') then
case y is
 when S0 => if w='0' then y<=S0; else y<=S1; end if
 when S1 =>  y <= S2;
 when S2 => if w='0' then y<=S3; else y<=S1; end if
 when S3 =>  y<=S4;
when S4 => if w='0' then y<=S0; else y<=S1; end if
  end case
end if;
end process;
 z <= '1' when (y=S4) else '0';
end behavior;
```

5a) (SIN)

$$D_A = \overline{(Q_E \oplus Q_D)} \cdot (Q_D \oplus Q_C) \cdot Q_B$$

$$D_B = Q_A \quad D_C = Q_B \quad D_D = Q_C \quad D_E = Q_D$$

State shifting

look at the sequence of 4 1's

$D_A D_B D_C D_D D_E$	$D_A Q_A Q_A^+$	$D_B Q_B Q_B^+$	$D_C Q_C Q_C^+$	$D_D Q_D Q_D^+$	$D_E Q_E Q_E^+$
1 1 1 1 1	0 1 0	1 1 ϕ	1 1 ϕ	1 1 1	1 1 1
0 1 1 1 1	0 0 0	0 1 0	1 1 1	1 1 1	1 1 1
0 0 1 1 1	0 0 0	0 0 0	0 1 0	1 1 1	1 1 1
0 0 0 1 1	0 0 0	0 0 0	0 0 0	0 1 0	1 1 1
0 0 0 0 1	0 0 0	0 0 0	0 0 0	0 0 0	0 1 0
0 0 0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0

Assuming all Q's = '1' as initial state

Based on the equation for D_A (SIN) IF All Q's = '0' then D_A will never become '1'. ~~Please explain.~~

This is a cycle shift register.