University of Tehran
College of Engineering
School of Electrical and Computer Engineering

Digital Systems 1

Dr.Navabi

Homework 9

Soroush Mesforush Mashhad

SN:810198472

Khordad 00

Question 1

1. We want to make a 11101 Moore Machine Detector.

State Diagram:

$$\begin{tikzcd}
A/0 \arrow[loop above, "0"] \arrow[r, bend left, "1"] & B/0 \arrow[l, bend left, "0"] \arrow[r, "1"] & C/0 \arrow[r, "1"] \arrow[ll, bend left=50, "0"] & D/0 \arrow[loop above, "1"] \arrow[r, "0"] & E/0 \arrow[r, "1"] \arrow[llll, bend left=60, "0"] & F/1 \arrow[lll, bend right=40, "1"'] \arrow[lllll, bend left=70, "0"]
\end{tikzcd}$$

		0	1	W
000	A	A	B	0
001	B	A	C	0
010	C	A	D	0
011	D	E	D	0
100	E	A	F	0
101	F	A	C	1

State table

→

	$V_2 V_1 V_0$	0	1	W
A	0 0 0	000	001	0
B	0 0 1	000	010	0
C	0 1 0	000	011	0
D	0 1 1	100	011	0
E	1 0 0	000	101	0
F	1 0 1	000	010	1
		$V_2^+ V_1^+ V_0^+$		

transition table

Excitation table

$V_2 V_1 V_0$	0 J	1	W
0 0 0	000	001	0
0 0 1	000	010	0
0 1 0	000	011	0
0 1 1	100	011	0
1 0 0	000	101	0
1 0 1	000	001	1
	$D_2 D_1 D_0$		

K-maps

$V_1V_0 \backslash JV_2$	00	01	11	10
00	0	0	1	0
01	0	0	0	0
11	1	–	–	0
10	0	–		0

D_2

$D_2 = JV_2V_0 + \bar{J}V_1V_0$

$V_1V_0 \backslash JV_2$	00	01	11	10
00	0	0	0	0
01	0	0	1	1
11	0	–	–	1
10	0	–	–	1

D_1

$D_1 = J V_0 + JV_1$

$V_1V_0 \backslash JV_2$	00	01	11	10
00	0	0	1	1
01	0	0	0	0
11	0	–	–	1
10	0	–	–	1

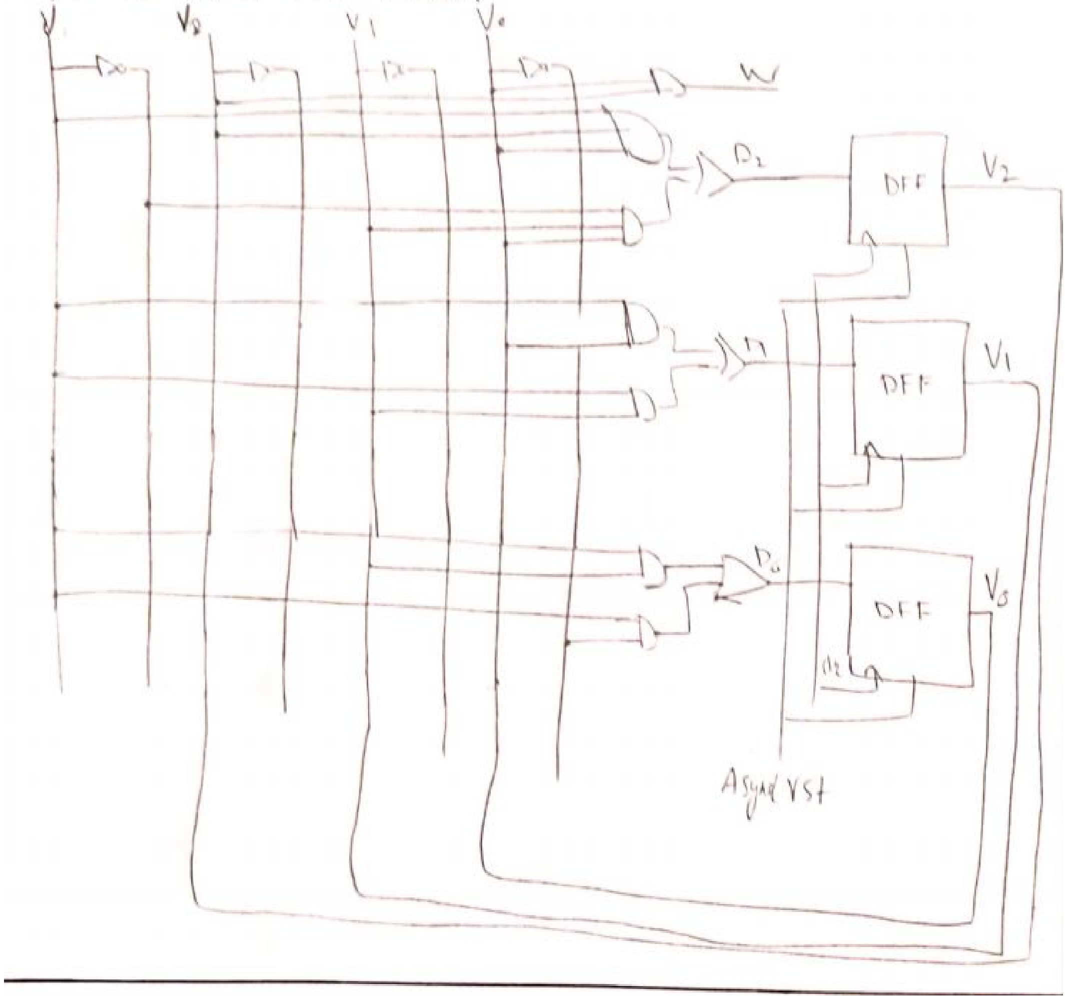
D_0

$D_0 = JV_1 + J\bar{V}_0$

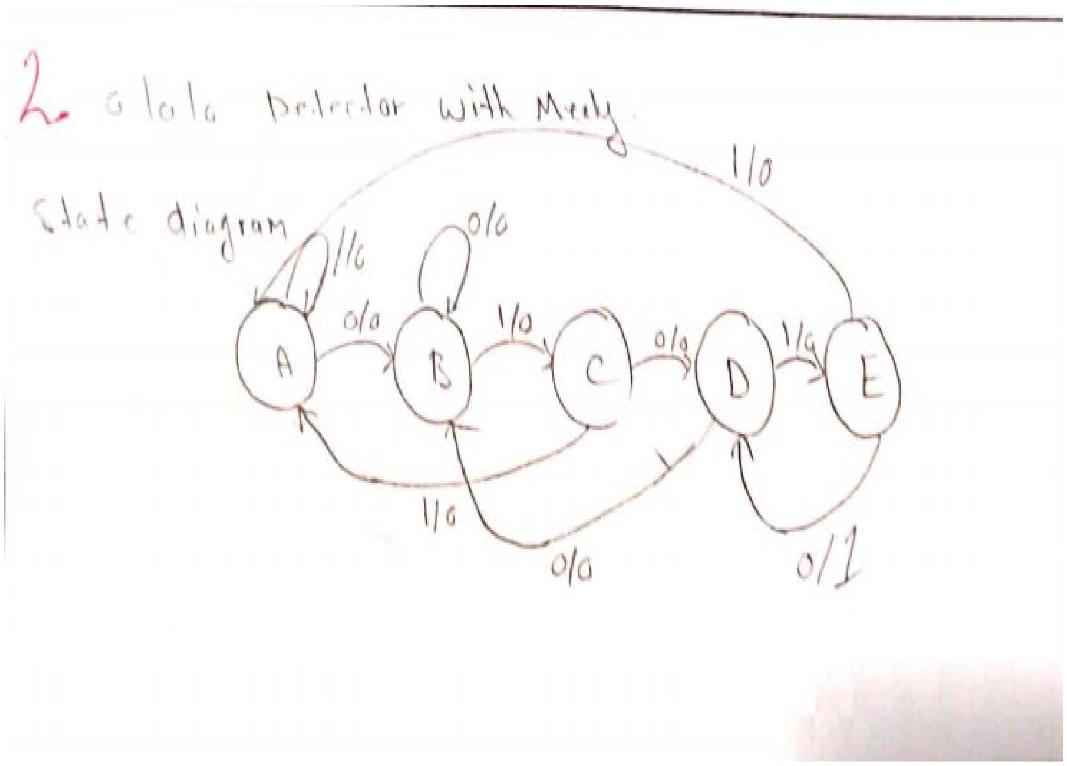
$V_0 \backslash V_2V_1$	00	01	11	10
0	0	0	–	0
1	0	0	–	1

$W = V_2 V_0$

Now we draw the circuit



Question 2



2. 01010 Detector with Mealy.

State diagram

	State	0	1	0	1
000	A	B	A	0	0
001	B	B	C	0	0
010	C	D	A	0	0
011	D	B	E	0	0
100	E	D	A	1	0
		State		W	

State table

$\xrightarrow{\text{Excitation}}$

	J			
$V_2V_1V_0$	0	1	0	1
000	001	000	0	0
001	001	010	0	0
010	011	000	0	0
011	001	100	0	0
100	011	000	1	0
	$D_2D_1D_0$		W	

→ K-maps

$V_1V_0 \backslash JV_2$	00	01	11	10
00	0	0	0	0
01	0	-	-	0
11	0	-	-	1
10	0	-	-	0

D_2

$V_1V_0 \backslash JV_2$	00	01	11	10
00	0	1	0	0
01	0	-	-	1
11	0	-	-	0
10	1	-	-	0

D_1

$V_1V_0 \backslash JV_2$	00	01	11	10
00	1	1	0	0
01	1	-	-	0
11	1	-	-	0
10	1	-	-	0

D_0

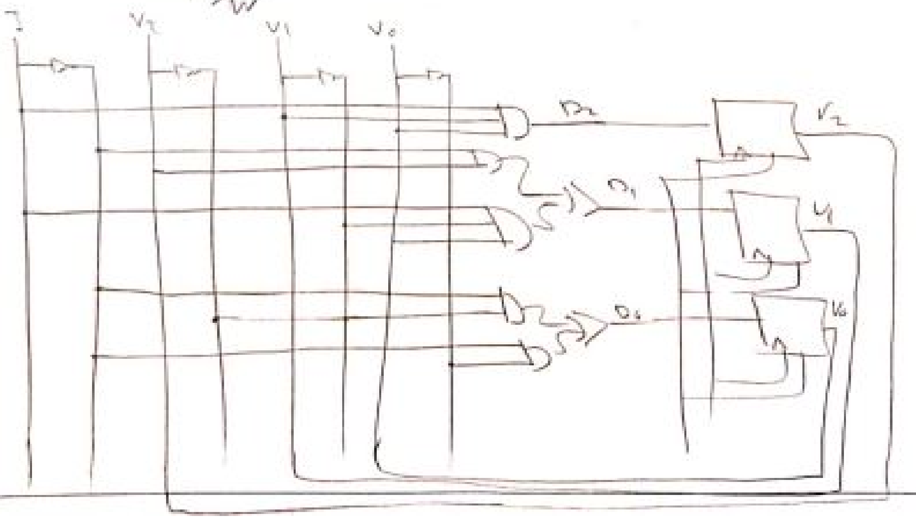
$V_1V_0 \backslash JV_2$	00	01	11	10
00	0	1	0	0
01	0	-	-	0
11	0	-	-	0
10	0	-	-	0

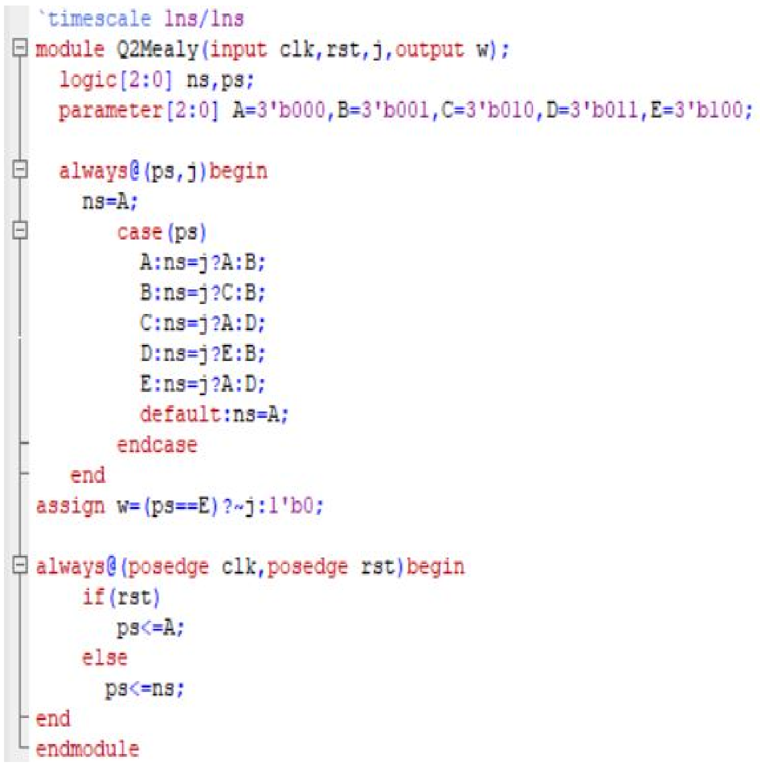
W

$D_2 = JV_1V_0$, $D_1 = \bar{J}V_2 + J\bar{V}_1V_0$

$D_0 = \bar{J}\bar{V}_2 + \bar{J}\bar{V}_1$

$W = \bar{J}V_2$



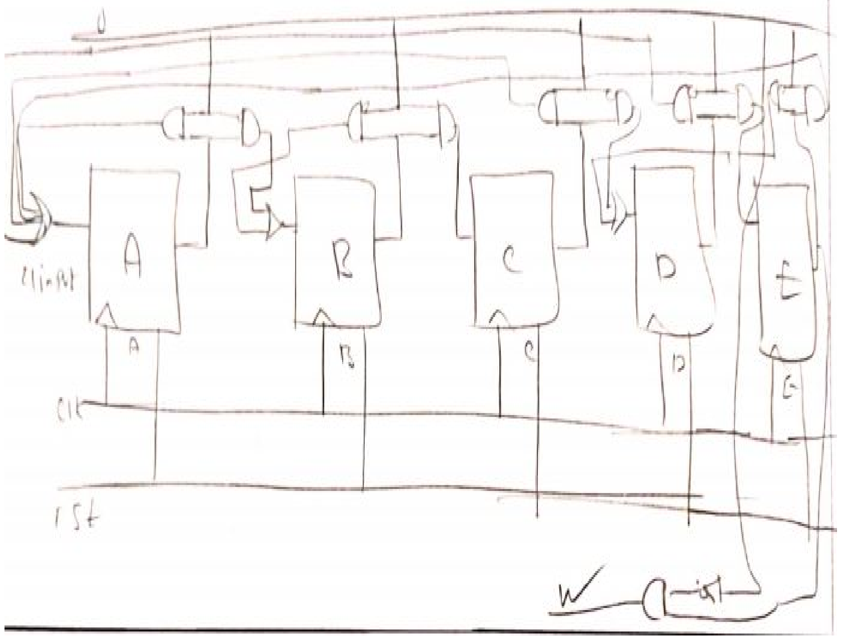


```
`timescale 1ns/1ns
module Q2Mealy(input clk,rst,j,output w);
  logic[2:0] ns,ps;
  parameter[2:0] A=3'b000,B=3'b001,C=3'b010,D=3'b011,E=3'b100;

  always@(ps,j)begin
    ns=A;
       case(ps)
         A:ns=j?A:B;
         B:ns=j?C:B;
         C:ns=j?A:D;
         D:ns=j?E:B;
         E:ns=j?A:D;
         default:ns=A;
       endcase
   end
assign w=(ps==E)?~j:1'b0;

always@(posedge clk,posedge rst)begin
    if(rst)
       ps<=A;
    else
      ps<=ns;
end
endmodule
```

Figure 1: Mealy Machine Code

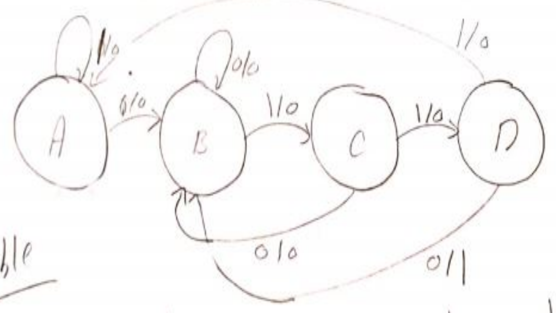
Question 3

3_ The state diagram is the same of the previous question, we need 5 flip flops



Question 4

4_ we start from Scratch   0 1 1 0



State table

	State	0	1	0	1
00	A	B	A	0	0
01	B	B	C	0	0
10	C	B	D	0	0
11	D	B	A	1	0
		State+		W	

transition table,

V_1V_0	0	1	0	1
00	01	00	0	0
01	01	10	0	0
10	01	11	0	0
11	01	00	1	0
	$V_1^+ V_0^+$		W	

Now we go to the exitation table

V_1V_0	J = 0	1	0	1
0 0	01	00	0	0
0 1	00	11	0	0
1 0	11	01	0	0
1 1	10	11	1	0
	T_1T_0		W	

K-maps

$V_0 \backslash JV_1$	00	01	11	10
0	0	1	0	0
1	0	1	1	1

T_1

$V_0 \backslash JV_1$	00	01	11	10
0	1	1	1	0
1	0	0	1	1

T_0

$V_0 \backslash JV_1$	00	01	11	10
0	0	0	0	0
1	0	1	0	0

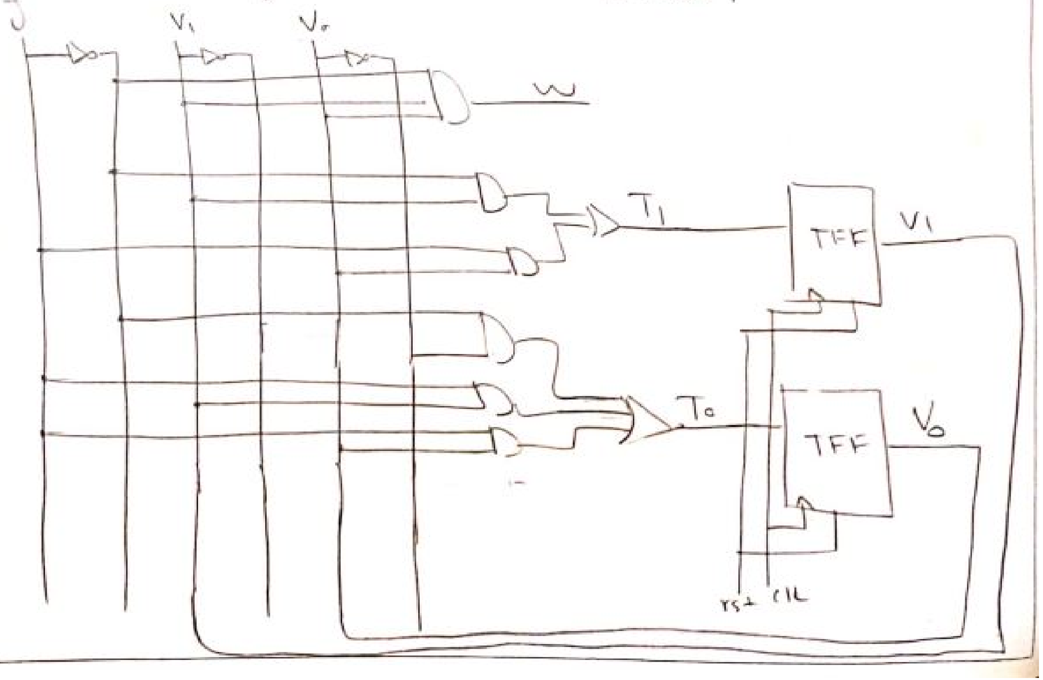
W

$T_1 = \bar{J}V_1 + JV_0$

$T_0 = \bar{J}\bar{V}_0 + JV_1 + JV_0$

$W = \bar{J}V_1V_0$

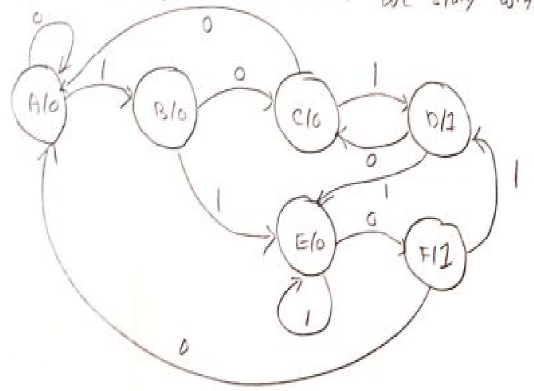
Hence the Diagram shall be as follows.



Question 5

5. To implement this machine we start with the state diagram

101 & 110



We jump straight to the excitation table

Excitation table:

		J		
	$V_2V_1V_0$	0	1	W
A	000	000	001	0
B	001	010	100	0
C	010	000	011	0
D	011	010	100	0
E	100	101	100	1
F	101	000	011	1
		$D_2\ D_1\ D_0$		

$V_1V_0 \backslash JV_2$	00	01	11	10
00	0	1	1	0
01	0	0	0	1
11	0	–	–	1
10	0	–	–	0

D_2

$D_2 = J\bar{V_2}V_0 + V_2\bar{V_1}\bar{V_0}$

$V_1V_0 \backslash JV_2$	00	01	11	10
00	0	0	0	0
01	1	0	1	0
11	1	–	–	0
10	0	–	–	1

D_1

$D_1 = \bar{J}\bar{V_2}V_0 + JV_2V_0 + JV_1\bar{V_0}$

$V_1V_0 \backslash JV_2$	00	01	11	10
00	0	1	0	1
01	0	0	1	0
11	0	–	–	0
10	0	–	–	1

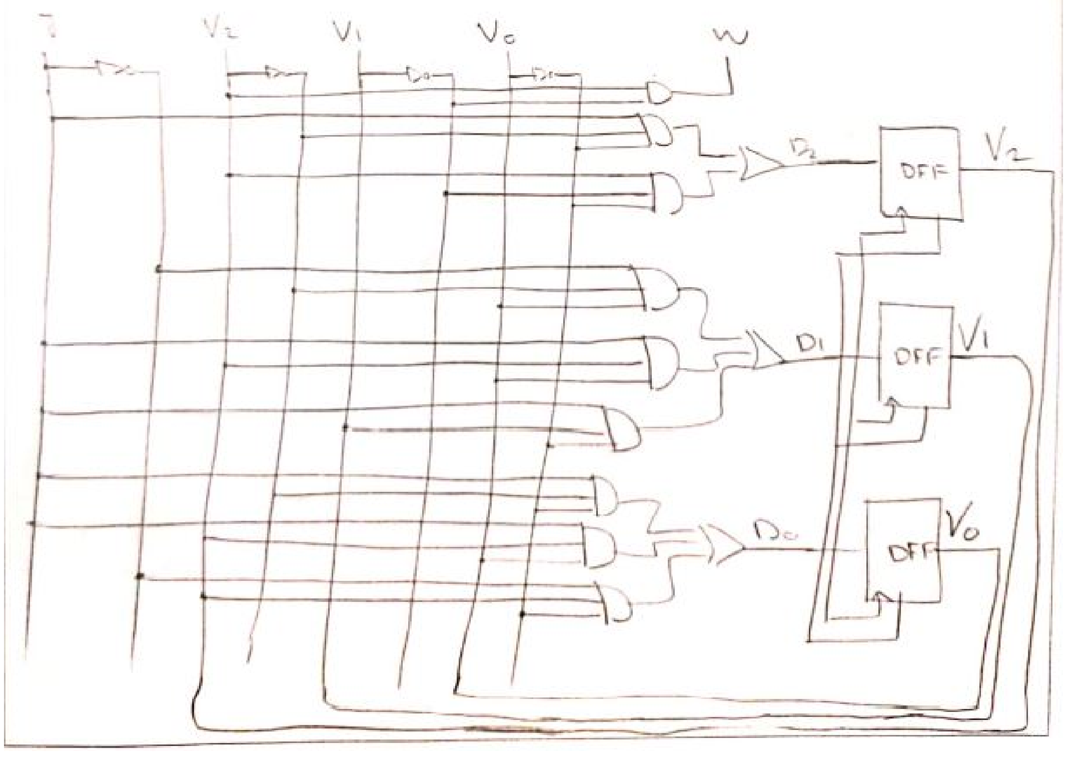
D_0

$D_0 = J\bar{V_2}\bar{V_0} + JV_2V_0 + \bar{J}V_2\bar{V_0}$

$V_0 \backslash V_2V_1$	00	01	11	10
0	·	·	0	1
1	0	0	0	1

W

$W = V_2\bar{V_1}$

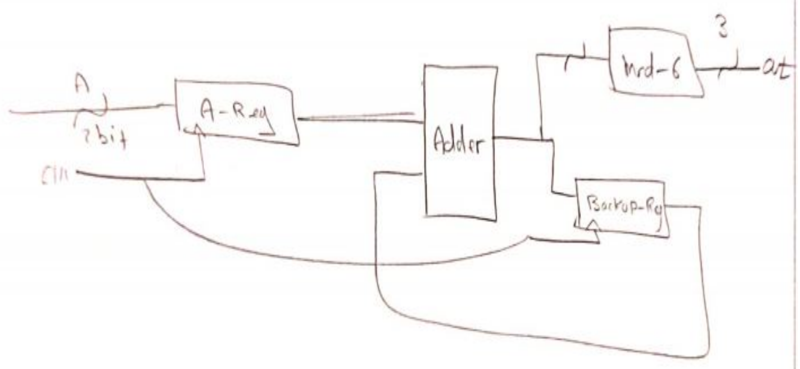


Question 6

6- This is a RTL Methodology problem.

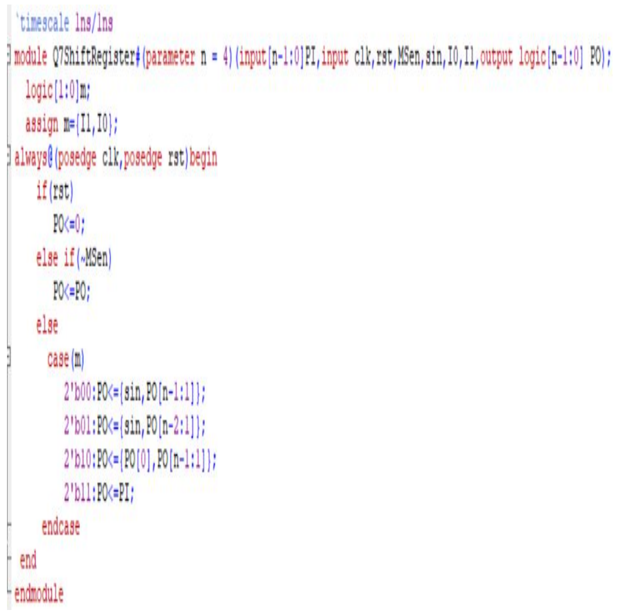
What we need: A Register for A, A Register to keep the Sum of each level init. An adder, A modulo-6 element

Hence we can create the following design:



Question 7

The system verilog code is depicted below:

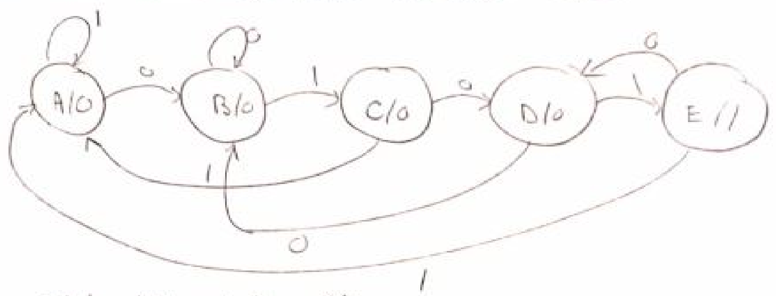


```
`timescale 1ns/1ns
module Q7ShiftRegister#(parameter n = 4)(input[n-1:0]PI,input clk,rst,MSen,sin,I0,I1,output logic[n-1:0] PO);
  logic[1:0]m;
  assign m={I1,I0};
always@(posedge clk,posedge rst)begin
    if(rst)
      PO<=0;
    else if(~MSen)
      PO<=PO;
    else
     case(m)
       2'b00:PO<={sin,PO[n-1:1]};
       2'b01:PO<={sin,PO[n-2:1]};
       2'b10:PO<={PO[0],PO[n-1:1]};
       2'b11:PO<=PI;
    endcase
 end
endmodule
```

Figure 2: Mealy Machine

Question 8 Part A

8- we start with the Moore machine: 0101



The state and transition table shall suffice

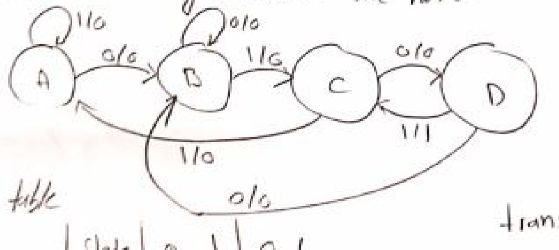
State table

	State	0	1	W
000	A	B	A	0
001	B	B	C	0
010	C	D	A	0
011	D	B	E	0
100	E	D	A	1

—

$V_2V_1V_0$	0	1	W
000	001	000	0
001	001	010	0
010	011	000	0
011	001	100	0
100	011	000	1
	$V_2^+ V_1^+ V_0^+$		

Now for the Mealy Machine we have.



State table

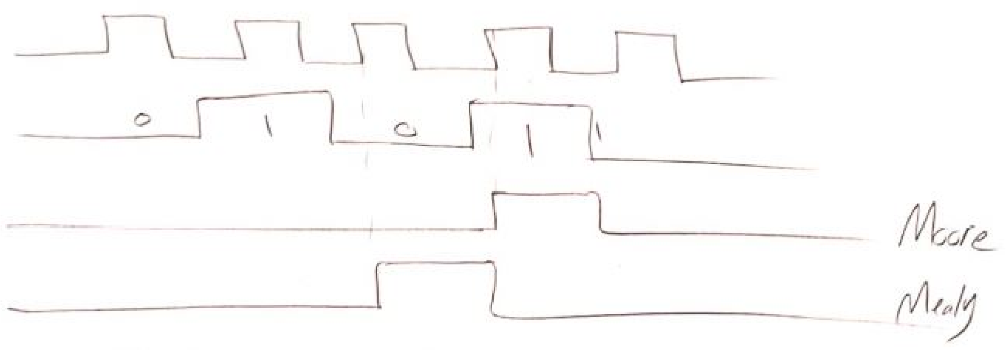
	State	0	1	0	1
00	A	B	A	0	0
01	B	B	C	0	0
10	C	D	A	0	0
11	D	B	C	0	1
		State+		W	

—

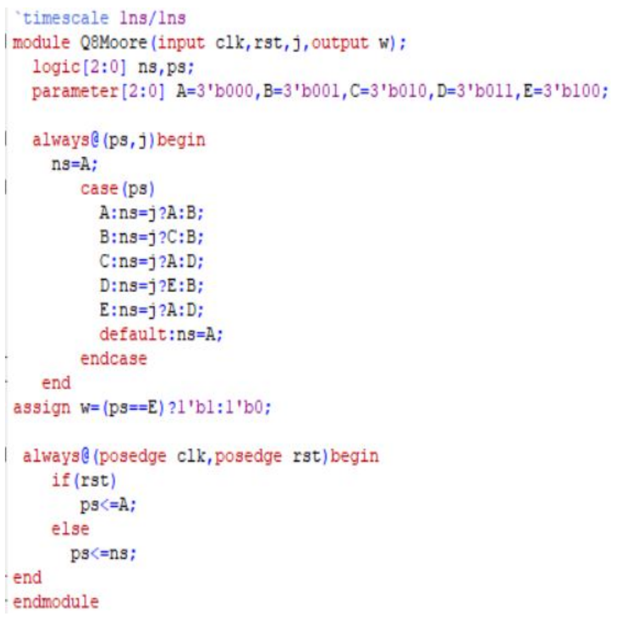
transition table

	0	1	0	1
00	01	00	0	0
01	01	10	0	0
10	11	00	0	0
11	01	10	0	1
	$V_1^+V_0^+$		W	

a) As we know the timing of the Moore and Mealy Machine differ from each other hence we have different outputs for the following example

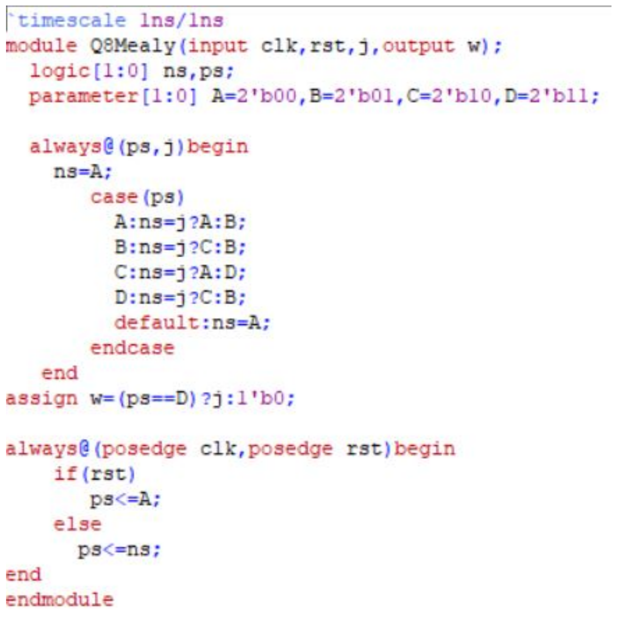


Part B



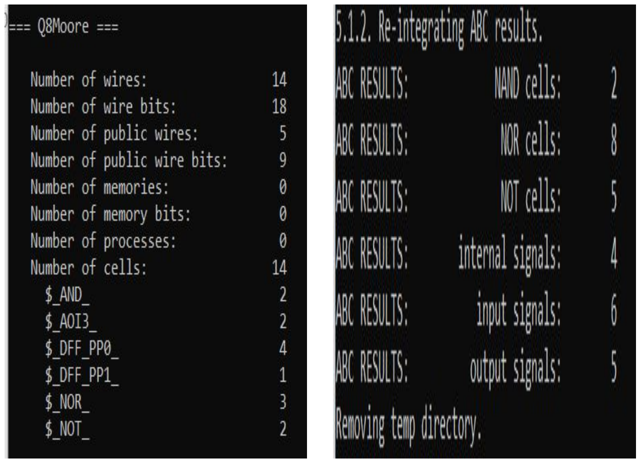
```
`timescale 1ns/1ns
module Q8Moore(input clk,rst,j,output w);
  logic[2:0] ns,ps;
  parameter[2:0] A=3'b000,B=3'b001,C=3'b010,D=3'b011,E=3'b100;

  always@(ps,j)begin
    ns=A;
       case(ps)
         A:ns=j?A:B;
         B:ns=j?C:B;
         C:ns=j?A:D;
         D:ns=j?E:B;
         E:ns=j?A:D;
         default:ns=A;
       endcase
    end
assign w=(ps==E)?1'b1:1'b0;

 always@(posedge clk,posedge rst)begin
    if(rst)
       ps<=A;
    else
      ps<=ns;
end
endmodule
```

Figure 3: Moore Machine



```
`timescale 1ns/1ns
module Q8Mealy(input clk,rst,j,output w);
  logic[1:0] ns,ps;
  parameter[1:0] A=2'b00,B=2'b01,C=2'b10,D=2'b11;

  always@(ps,j)begin
    ns=A;
       case(ps)
         A:ns=j?A:B;
         B:ns=j?C:B;
         C:ns=j?A:D;
         D:ns=j?C:B;
         default:ns=A;
       endcase
   end
assign w=(ps==D)?j:1'b0;

always@(posedge clk,posedge rst)begin
    if(rst)
       ps<=A;
    else
      ps<=ns;
end
endmodule
```

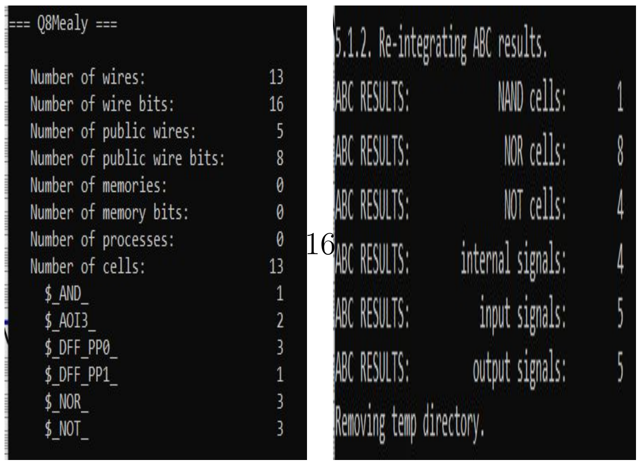
Figure 4: Mealy Machine



```
=== Q8Moore ===

   Number of wires:                14
   Number of wire bits:            18
   Number of public wires:          5
   Number of public wire bits:      9
   Number of memories:              0
   Number of memory bits:           0
   Number of processes:             0
   Number of cells:                14
     $_AND_                         2
     $_AOI3_                        2
     $_DFF_PP0_                     4
     $_DFF_PP1_                     1
     $_NOR_                         3
     $_NOT_                         2
```

```
5.1.2. Re-integrating ABC results.
ABC RESULTS:           NAND cells:     2
ABC RESULTS:            NOR cells:     8
ABC RESULTS:            NOT cells:     5
ABC RESULTS:     internal signals:     4
ABC RESULTS:        input signals:     6
ABC RESULTS:       output signals:     5
Removing temp directory.
```

Figure 5: Moore Synthesis



```
=== Q8Mealy ===

   Number of wires:                13
   Number of wire bits:            16
   Number of public wires:          5
   Number of public wire bits:      8
   Number of memories:              0
   Number of memory bits:           0
   Number of processes:             0
   Number of cells:                13
     $_AND_                         1
     $_AOI3_                        2
     $_DFF_PP0_                     3
     $_DFF_PP1_                     1
     $_NOR_                         3
     $_NOT_                         3
```

```
5.1.2. Re-integrating ABC results.
ABC RESULTS:           NAND cells:     1
ABC RESULTS:            NOR cells:     8
ABC RESULTS:            NOT cells:     4
ABC RESULTS:     internal signals:     4
ABC RESULTS:        input signals:     5
ABC RESULTS:       output signals:     5
Removing temp directory.
```

In the following pictures we see that in the result of the synthesis of the mealy machine fewer elements are needed.