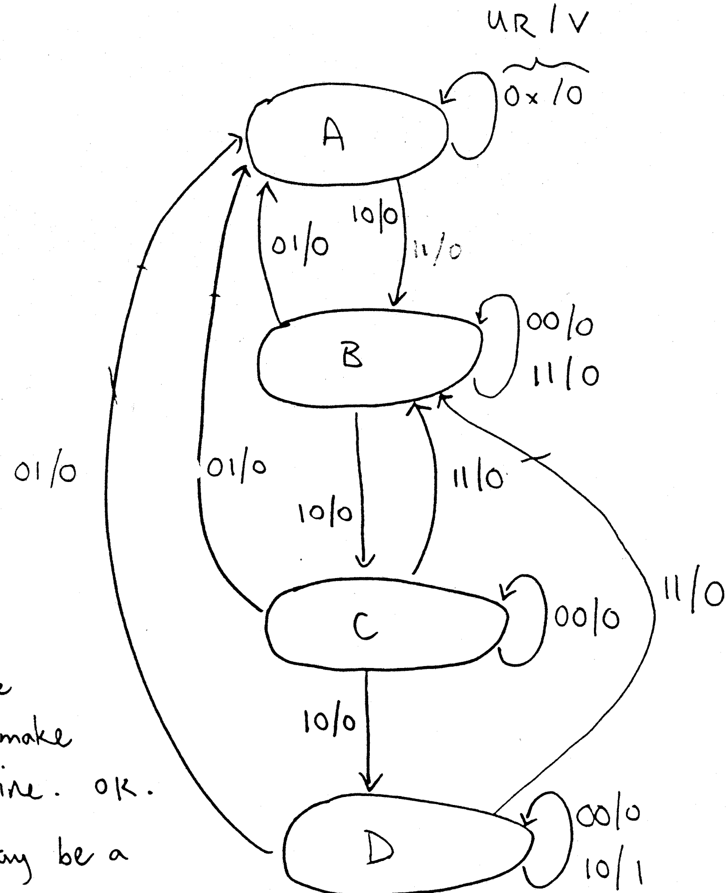
Question 1

Design a saturating 2-bit event counter which samples an input U (an event) *every* cycle and counts up if it is asserted. If an event arrives and the counter is already at maximum of 3, set an output V=1 to signify an overflow, but do not change the counter value (it is saturated). This means that when the counter is saturated, it will miss events until being reset.

When the counter value is read by setting input R=1, the counter should reset but it must not miss counting any new events.

a) Draw the FSM diagram for the counter. Label the states A, B, C, ...



NOTE: a fifth state may be used to make this a Moore machine. OK.

NOTE: the overflow may be a "sticky" state. OK.

b) Does this FSM correspond to a Mealy or a Moore model?

Mealy

c) Give the state table. Be sure to show the output V.

P.S.	N.S. UR= 00	01	11	10	V UR= 00	01	11	10
A	A	A	B	B	0	0	0	0
B	B	A	B	C	0	0	0	0
C	C	A	B	D	0	0	0	0
D	D	A	B	D	0	0	0	1

d) Give a state assignment.

A 000

B 001

C 010

D 011

E 111 ← only if needed

e) Give the state-assigned table.

P.S. $y_1 y_0$	N.S. UR=00 $Y_1 Y_0$	01 $Y_1 Y_0$	11 $Y_1 Y_0$	10 $Y_1 Y_0$	V UR=10	else
00	00	00	01	01	0	0
01	01	00	01	10	0	0
10	10	00	01	11	0	0
11	11	00	01	11	1	0

f) Implement the FSM in logic gates and flip-flops. Minimize your logic and draw the resulting circuit.

Y_1:

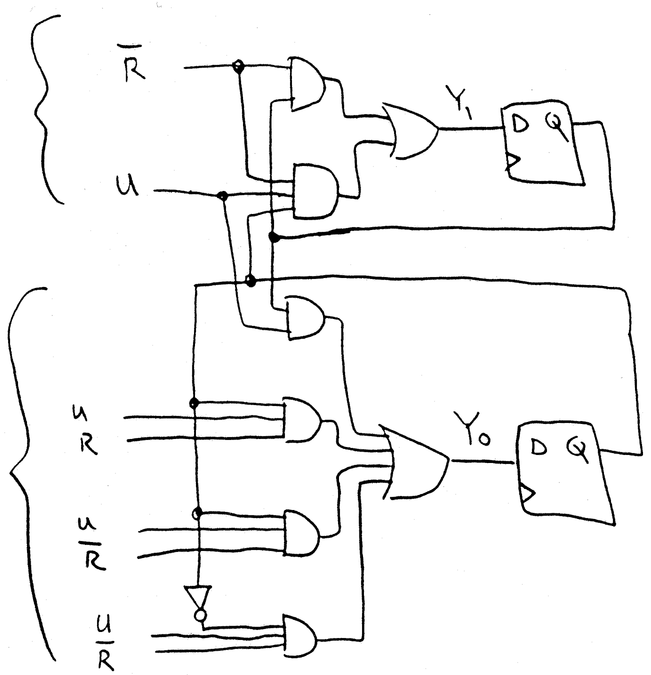
y_1y_0 \ UR=	00	01	11	10
00	0	0	0	0
01	0	0	0	1
11	1	0	0	1
10	1	0	0	1

Y_0:

y_1y_0 \ UR=	00	01	11	10
00	0	0	0	1
01	1	0	1	0
11	1	0	1	1
10	0	0	1	1

$$Y_1 = \bar{R}y_1 + U\bar{R}y_0$$

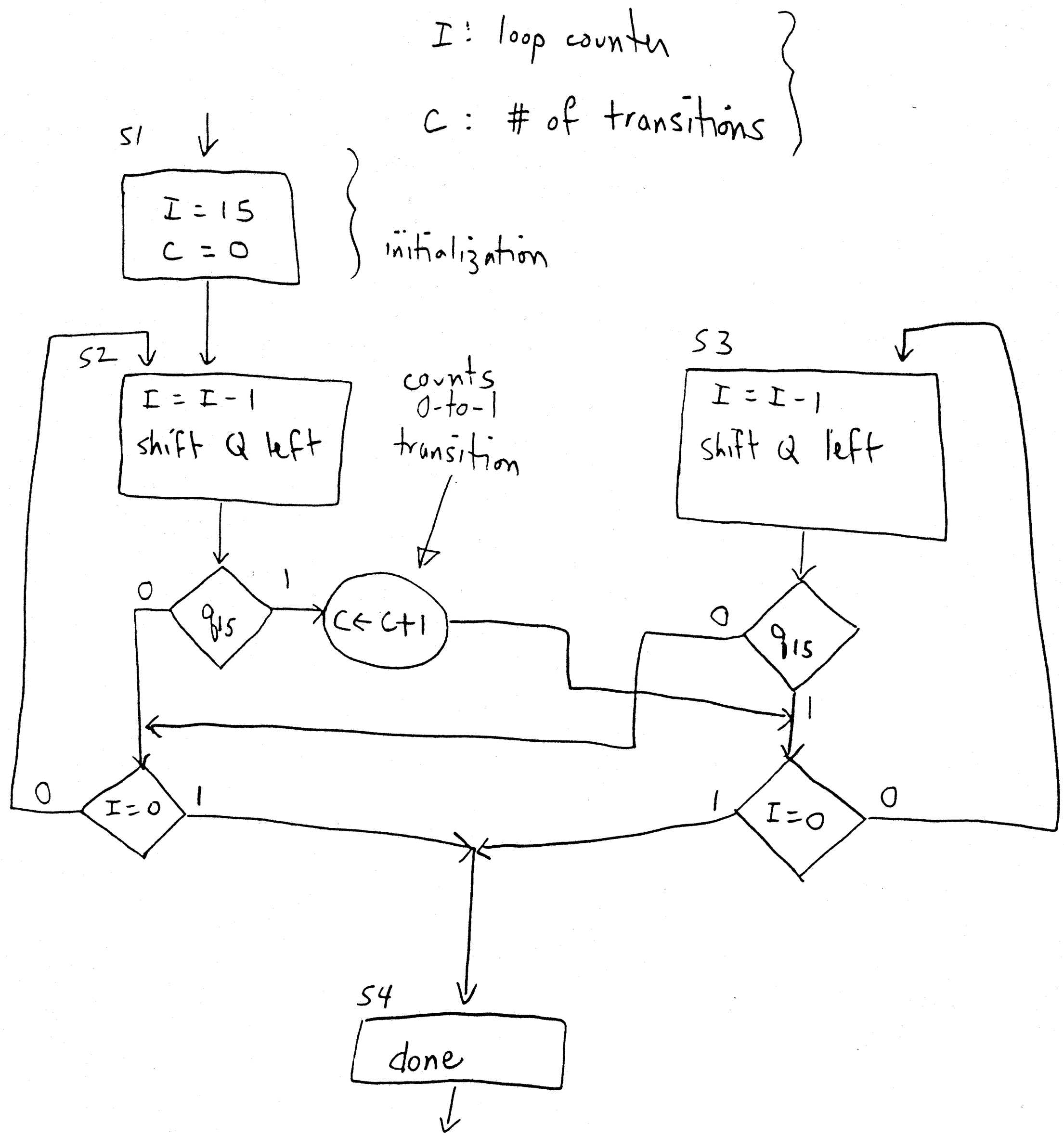
$$Y_0 = \bar{U}\bar{R}y_0 + URy_0 + U\bar{R}\bar{y}_0 + Uy_1$$



Question 2

You are given a shift register Q which holds a 16-bit value.

a) Draw an ASM diagram to count the number of 0-to-1 transitions in Q when reading from left to right, or MSB to LSB. You should assume there is an implied leading '0'.



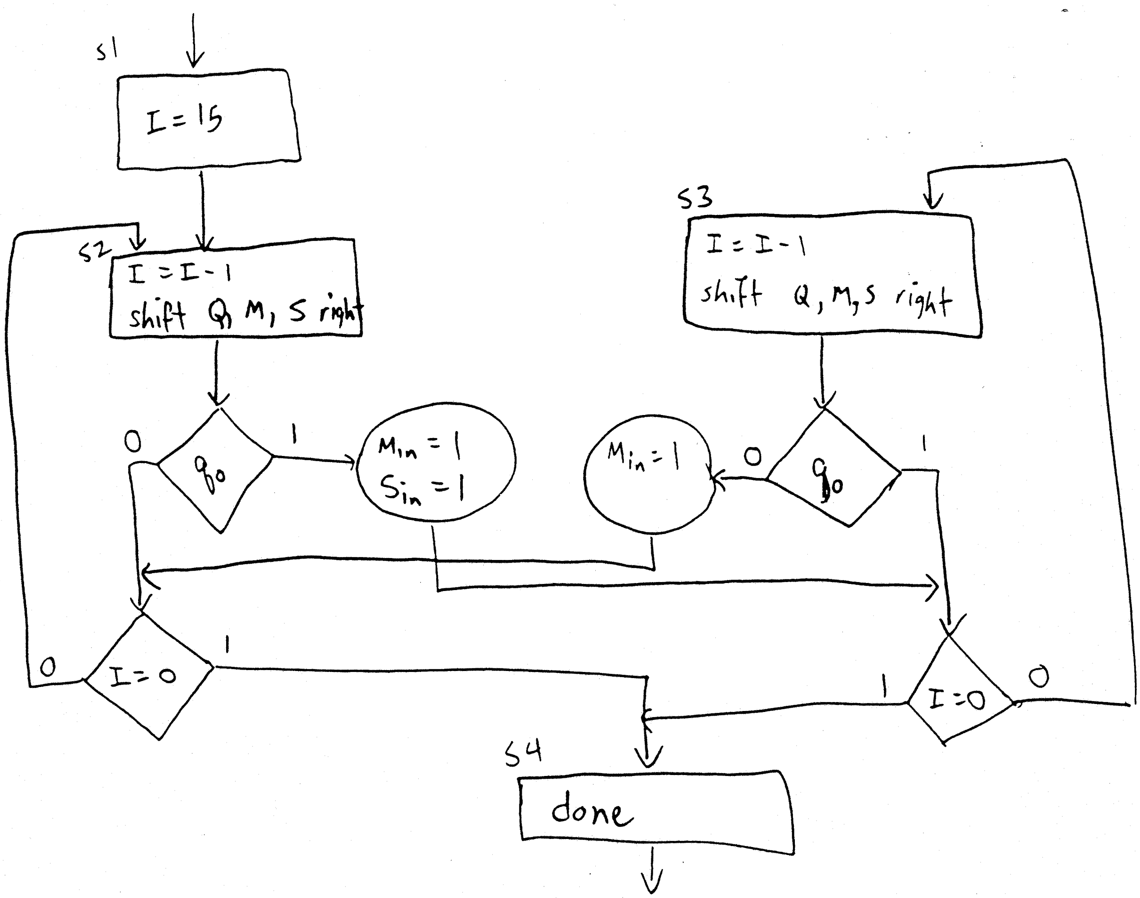
b) Draw an ASM diagram to compute the radix-2 Booth encoding of the number serially. Store the answer in two 16-bit shift registers: a magnitude register M and a sign register S. The register M has a '1' for every '1' or '-1' digit of the Booth-encoded value. The register S has a '1' for every '-1' digit in the Booth-encoded value.

Booth encoding replaces bit Q_i with Q'_i as follows:

Q_i	Q_{i-1}	Q'_i
0	0	0
0	1	1
1	0	-1
1	1	0

You should assume that bit $Q_{-1} = 0$.

For example, the input Q = 11010 (-6) has a Booth-encoded value of Q'={ 0 -1 1 -1 0 }. Your system should compute this as M = 01110 and S = 01010.



c) Draw the datapath circuit implied by b).