CS226 - Lab 9

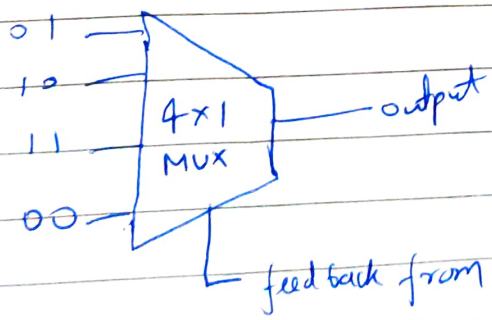
1)

I_0	I_1	Function
0	0	stop counting
0	1	count up by 1
1	0	count down by 1
1	1	count up by 2

Here the output is stored in a 2 bit register which is connected to the control switch of 4×1 MUX to get the next state.

i) for count up by 1 :

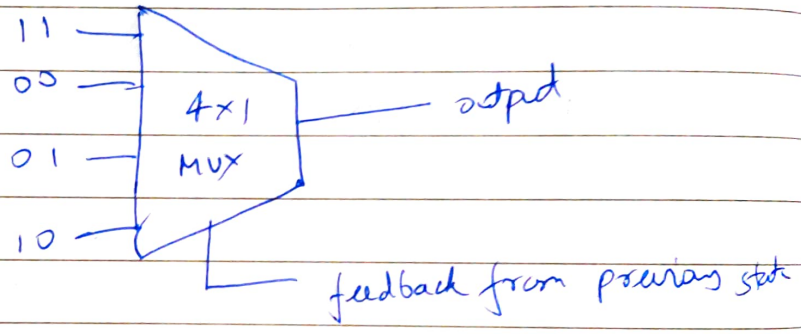
Present state		Next state	
Q_2	Q_1	Q_2^+	Q_1^+
0	0	0	1
0	1	1	0
1	0	1	1
1	1	0	0



* To stop counting ($I_0=0$, $I_1=0$) we connect feedback to the MUX which selects the operation. (first pin is connected)

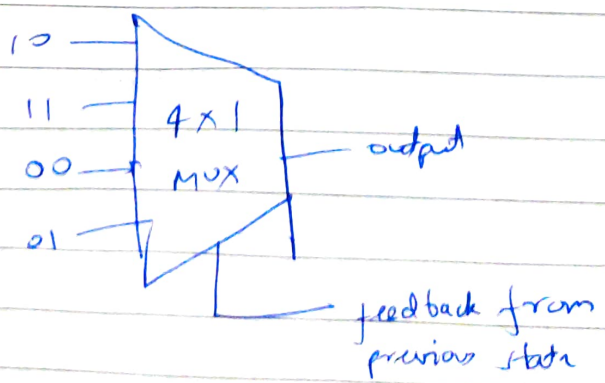
ii) For count down by 1:

Q_2	Q_1	Q_2^+	Q_1^+
0	0	1	1
0	1	0	0
1	0	0	1
1	1	1	0



iii) For count up by 2

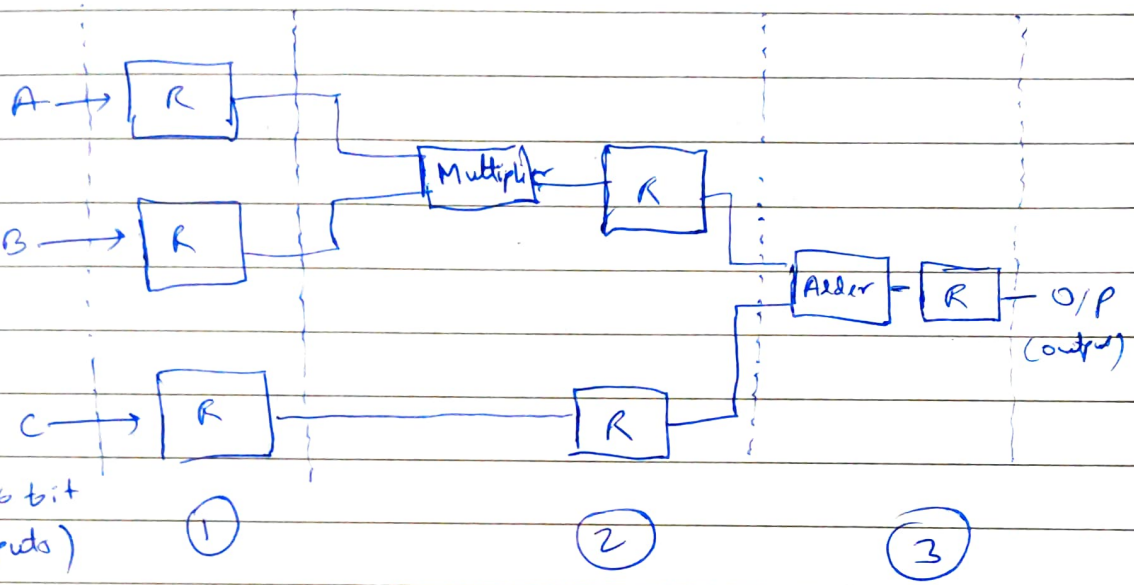
Q_2	Q_1	Q_2^+	Q_1^+
0	0	1	0
0	1	1	1
1	0	0	0
1	1	0	1



2) $Y = A * B + C$

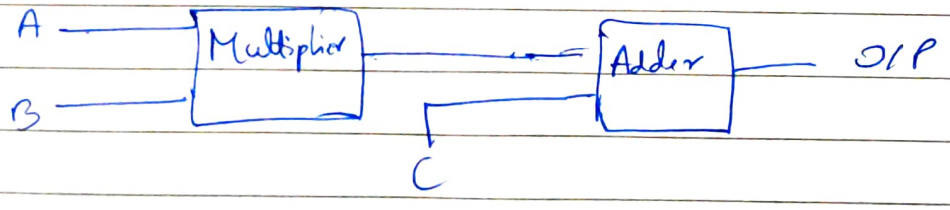
i) With pipeline register :

Block level simple design : R → register



(all registers are synchronized with a single clock)

ii) Without pipeline register :



	With pipeline Register	Without Register
i)	Requires 3 clock ticks to get the final output.	No clock used and outputs final answer at once.
ii)	stores output obtained at intermediate levels in registers.	No storage element present.
iii)	Slower as requires 3 clock cycles.	Faster as operations are performed immediately after input is given

3) Sequence detector (MUX based FSM design)

↳ detects 0100, 1001, 1011 sequences and produces output 1 (when last 4 inputs form any of the three sequence) otherwise outputs 0.

* Design: (MUX based FSM)

Last 4 inputs is stored in a shift register and then used as a control input to a 16x1 MUX with 1 at ports (pins) → 4 (0100), 9 (1001), and 11 (1011) and zero at all other ports.