* Digital Electronics and Logic Design (DELD) - Practical Number - 12

Name :- Kaustubh Shrikant Kabra.
Class :- Second Year Engineering
Div :- A Roll Number :-
Batch :-
Department :- Computer Department
College :- AISSMS's IOIT.

Title :-
Sequence Generator.

Aim :-
Design and implement sequence generator using D flip-flop.

Objectives :-
Design sequence generator for :-
① Odd Sequence
② Even Sequence

Theory :-
For the design of sequence generator particular sequence can be determined as follows.
No. of flip-flop required to generate particular sequence can be determined as follows :-
① Find the number of 1's in the sequence
② Find the number of 0's in the sequence

③ Take the maximum out of two.

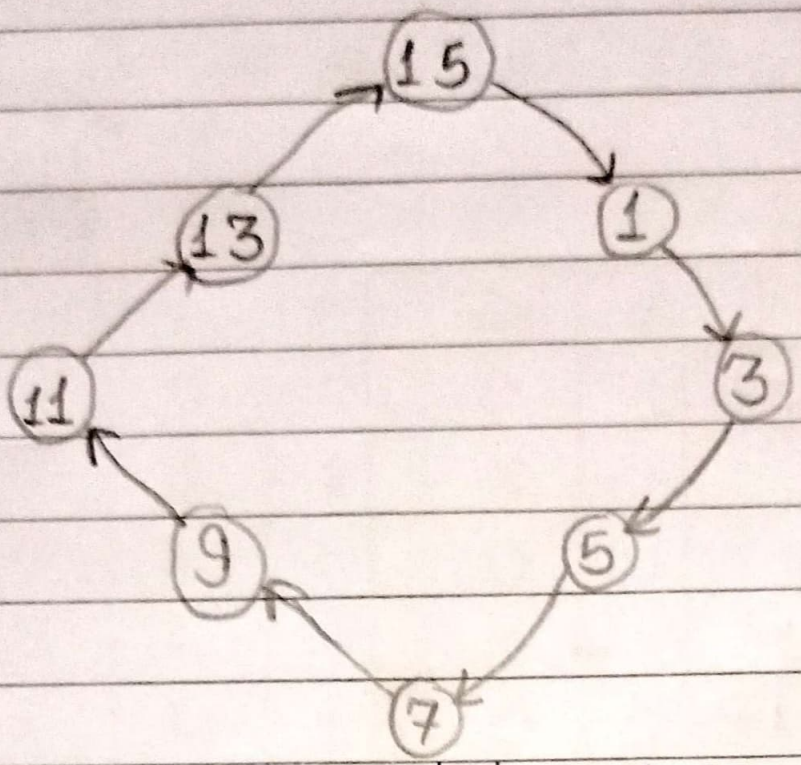
④ If N is the required no. of flip-flop, choose minimum value of 'n' to satisfy equation given below:

$$Max(0's, 1's) \leq 2^{n-1}$$

Truth Table :-

1) Sequence Generator for Odd Number (4-bit)

Present State				Next State			
Q_A	Q_B	Q_C	Q_D	Q_{A+1}	Q_{B+1}	Q_{C+1}	Q_{D+1}
0	0	0	0	X	X	X	X
0	0	0	1	0	0	1	1
0	0	1	0	X	X	X	0
0	0	1	1	0	1	0	1
0	1	0	0	X	X	X	X
0	1	0	1	0	1	1	1
0	1	1	0	X	X	X	X
0	1	1	1	1	0	0	1
1	0	0	0	X	X	X	X
1	0	0	1	1	0	1	1
1	0	1	0	X	X	X	X
1	0	1	1	1	1	0	1
1	1	0	0	X	X	X	X
1	1	0	1	1	1	1	1
1	1	1	0	X	X	X	X
1	1	1	1	0	0	0	1



Simplification using truth table and kmaps :-

$$D_A = Q_A \overline{Q}_C + Q_A \overline{Q}_B + \overline{Q}_A Q_B Q_C$$

$$D_B = Q_B \overline{Q}_C + \overline{Q}_B Q_C$$

$$D_C = \overline{Q}_C$$

$$D_D = 1.$$

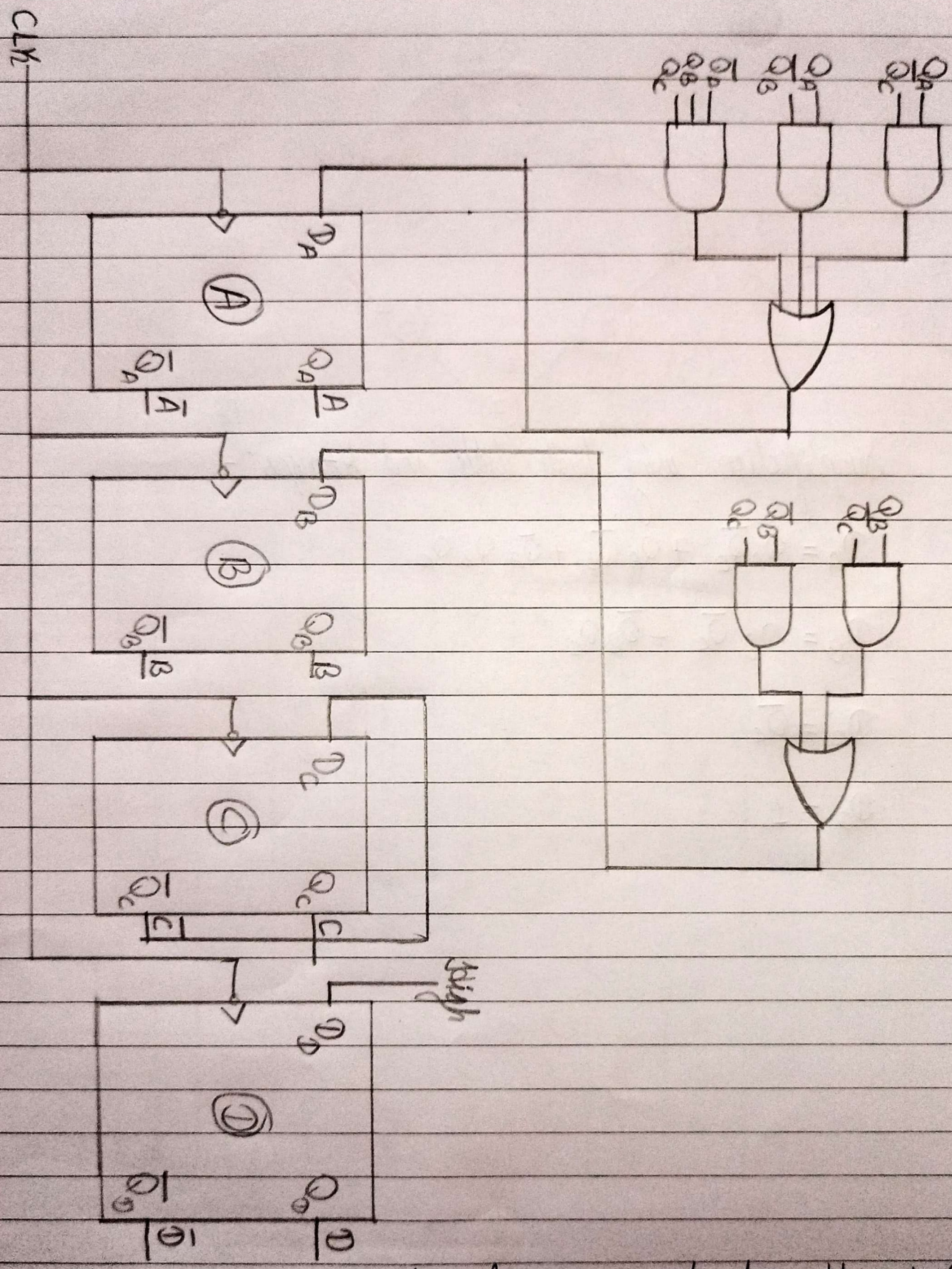
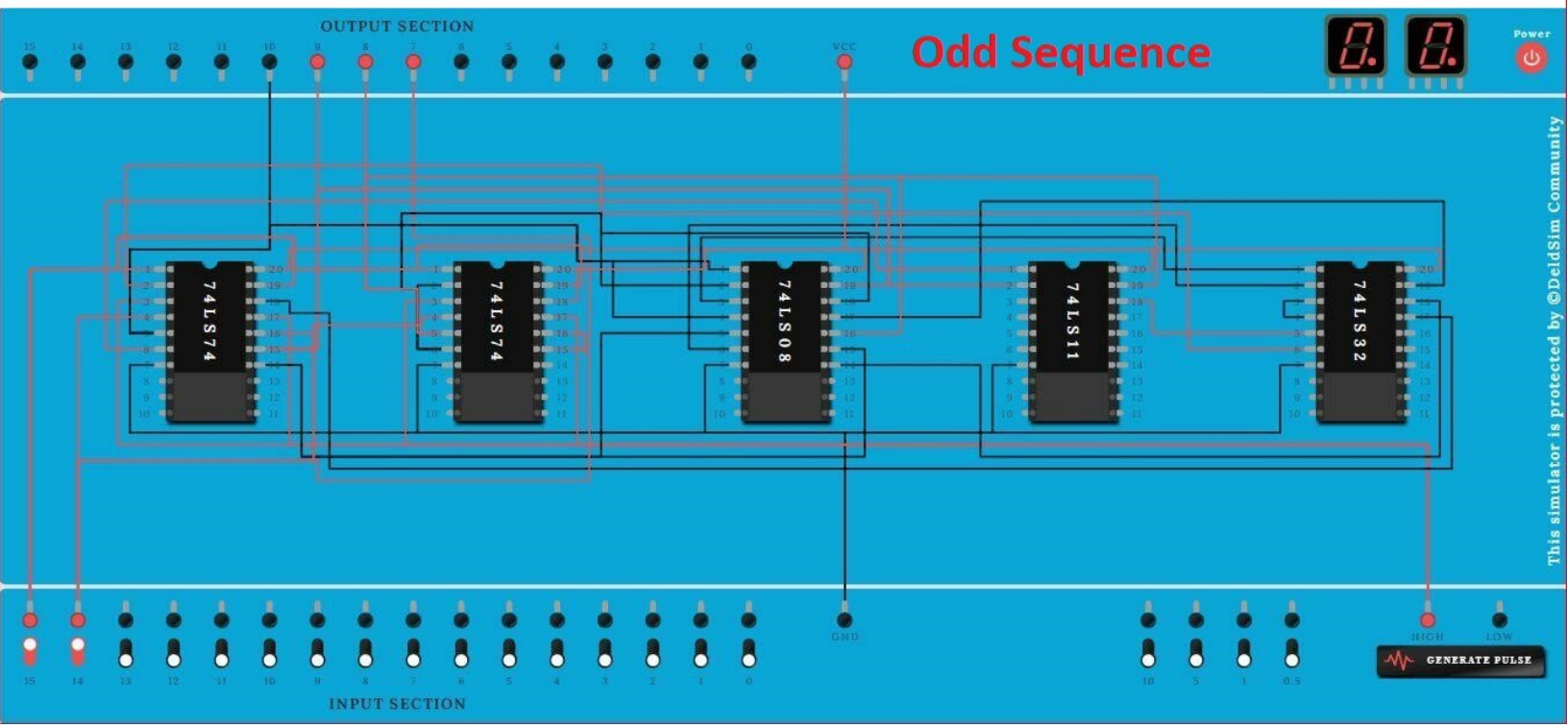
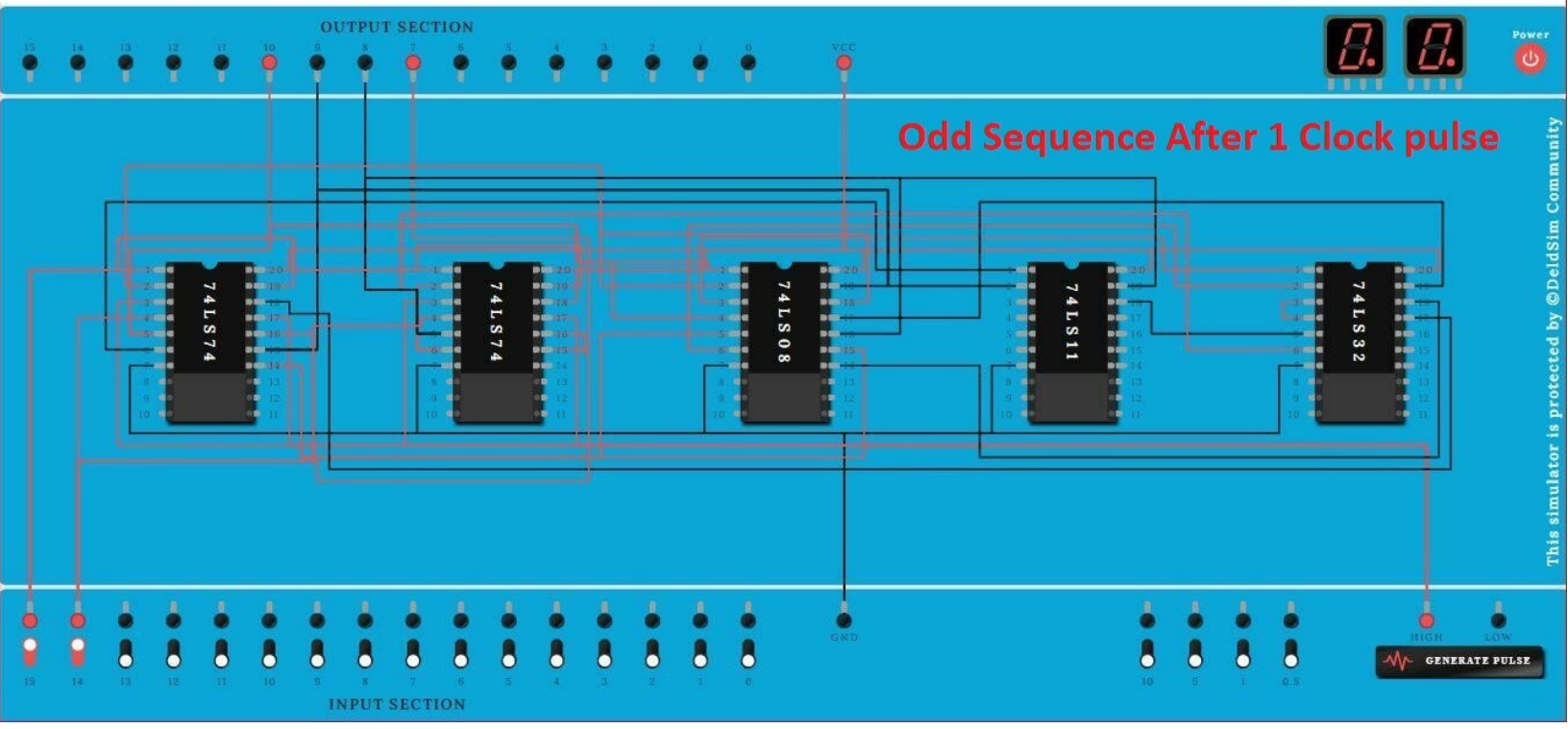


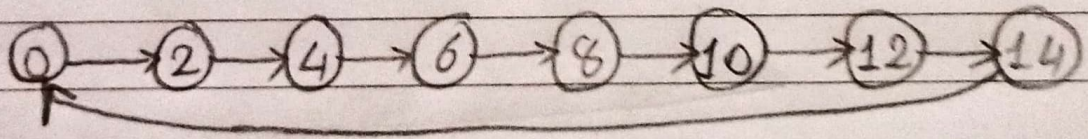
fig:- Sequence generator for odd number (4-bit).


OUTPUT SECTION
Odd Sequence
Power
VCC
74LS74
74LS74
74LS08
74LS11
74LS32
GND
HIGH
LOW
GENERATE PULSE
INPUT SECTION
This simulator is protected by ©DeldSim Community


OUTPUT SECTION
Odd Sequence After 1 Clock pulse
Power
VCC
74LS74
74LS74
74LS08
74LS11
74LS32
GND
HIGH
LOW
GENERATE PULSE
INPUT SECTION
This simulator is protected by ©DeldSim Community

2) Sequence Generator for even number (4-bit)-

Present State				Next State			
Q_A	Q_B	Q_C	Q_D	Q_{A+1}	Q_{B+1}	Q_{C+1}	Q_{D+1}
0	0	0	0	0	0	1	0
0	0	0	1	X	X	X	X
0	0	1	0	0	1	0	0
0	0	1	1	X	X	X	X
0	1	0	0	0	1	1	0
0	1	0	1	X	X	X	X
0	1	1	0	1	0	0	0
0	1	1	1	X	X	X	X
1	0	0	0	1	0	1	0
1	0	0	1	X	X	X	X
1	0	1	0	1	1	0	0
1	0	1	1	X	X	X	X
1	1	0	0	1	1	1	0
1	1	0	1	X	X	X	X
1	1	1	0	0	0	0	0
1	1	1	1	X	X	X	X



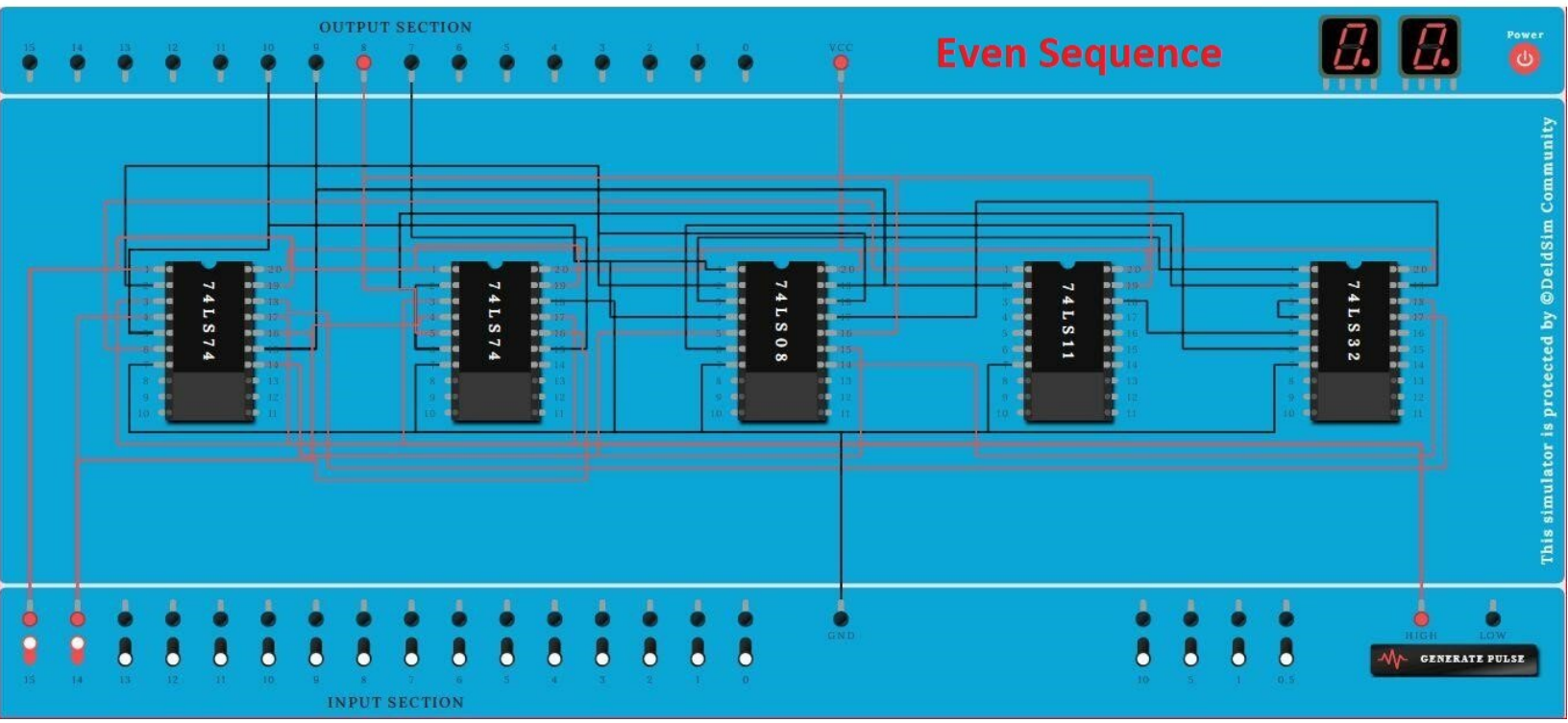
Using truth table and kmap for simplification.

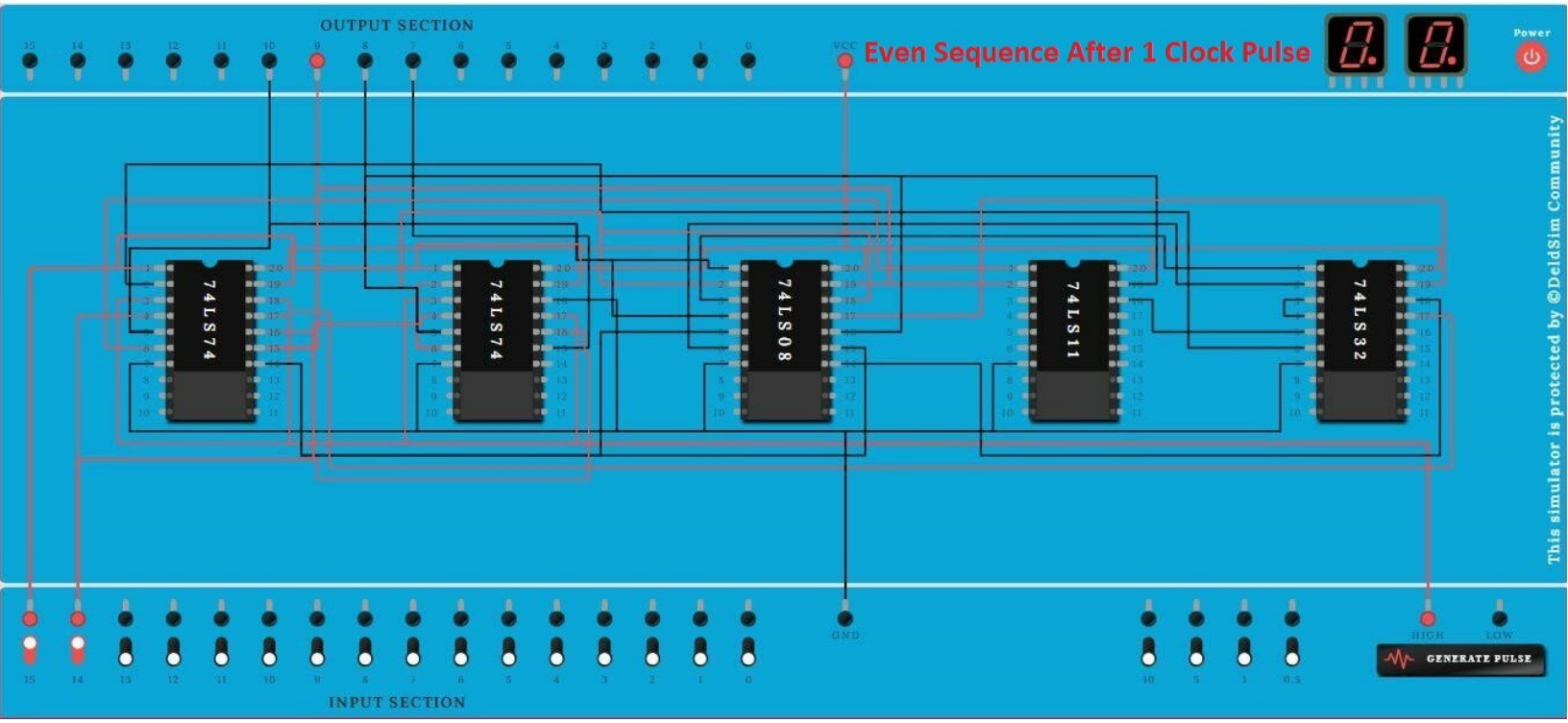
$D_A = Q_A\bar{Q}_C + Q_A\bar{Q}_B + \bar{Q}_A Q_B Q_C$

$D_B = Q_B\bar{Q}_C + \bar{Q}_B Q_C$

$D_C = \bar{Q}_C$

$D_D = 0$


OUTPUT SECTION
Even Sequence
Power
VCC
74LS74
74LS74
74LS08
74LS11
74LS32
This simulator is protected by ©DeldSim Community
GND
HIGH
LOW
GENERATE PULSE
INPUT SECTION


OUTPUT SECTION
Even Sequence After 1 Clock Pulse
Power
VCC
74LS74
74LS74
74LS08
74LS11
74LS32
This simulator is protected by ©DeldSim Community
GND
HIGH
LOW
GENERATE PULSE
INPUT SECTION

Conclusion:-

Hence, we have design and implimented the sequence generator using D flip flop.