Title of the Research Migration Project :
Design and Implementation of 2*4 decoder and 3*8 decoder using 2*4 decoders in eSim

Theory/Description :
A decoder is a combinational logic circuit that converts an N-bit binary input into one of its M distinct output channels. It is often used in memory addressing, data demultiplexing, and other digital applications. The primary function of a decoder is to ensure that only one output is active at any given time, corresponding to the binary value of the input. All decoders have an input called Enable (EN). If EN = 0, both outputs are zero. If EN = 1, the output corresponding to the input value is activated.
There are various types of decoders such as 1x2 decoder, 2x4 decoder, 3x18 decoder and so on.

This project aims to design and simulate a 1x2 decoder and a 2×4 Decoder using basic logic gates and then extend its application by constructing a 3×8 Decoder using multiple 2×4 Decoders and 1x2 decoder through an efficient modular design approach. By employing eSim, an open-source electronic design automation (EDA) tool, the design will be analyzed for its performance and correctness.

- **1x2 Decoder**: A 1x2 decoder has 1 input line(A), 1 enable input (EN), and 2 output lines (Y0, Y1). The enable input controls whether the decoder operates or not. The truth table for a 1x2 decoder is as follows:

EN	A	Y0	Y1
0	X(Don't care)	0	0
1	0	1	0
1	1	0	1

- **2x4 decoder** : The 2x4 decoder has two inputs (A1, A0), one enable input (EN), and four outputs (Y0,Y1,Y2,Y3). It can be designed using basic logic gates such as AND, NOT, and OR. The truth table for a 2x4 decoder is as follows:

EN	A1	A0	Y0	Y1	Y2	Y3
0	X (Don't care)	X (Don't care)	0	0	0	0
1	0	0	1	0	0	0
1	0	1	0	1	0	0
1	1	0	0	0	1	0
1	1	1	0	0	0	1

- **3x8 decoder** : A 3x8 decoder has 1 input enable(EN), 3 input lines(A,B,C) and 8 output lines(D0,D1,D2,D3,D4,D5,D6,D7), providing eight possible output states. The truth table for a 3x8 decoder is as follows:

EN	A	B	C	D0	D1	D2	D3	D4	D5	D6	D7
0	X (Don't care)	X (Don't care)	X (Don't care)	0	0	0	0	0	0	0	0
1	0	0	0	1	0	0	0	0	0	0	0
1	0	0	1	0	1	0	0	0	0	0	0
1	0	1	0	0	0	1	0	0	0	0	0
1	0	1	1	0	0	0	1	0	0	0	0
1	1	0	0	0	0	0	0	1	0	0	0
1	1	0	1	0	0	0	0	0	1	0	0
1	1	1	0	0	0	0	0	0	0	1	0
1	1	1	1	0	0	0	0	0	0	0	1

This project involves designing a 1x2 decoder and a 2x4 decoder using only AND and NOT gates. The 2x4 decoder uses 4 3-input AND gates as its subcircuit. A 3x8 decoder is then constructed using one 1x2 decoder and two 2x4 decoders, where the 1x2 decoder uses the most significant input bit to enable one of the 2x4 decoders. The use of hierarchical sheets inside the root schematic sheet organizes the design, simplifying the schematic and promoting modularity by encapsulating each decoder within a separate sheet for better clarity and management.

Circuit Diagrams :

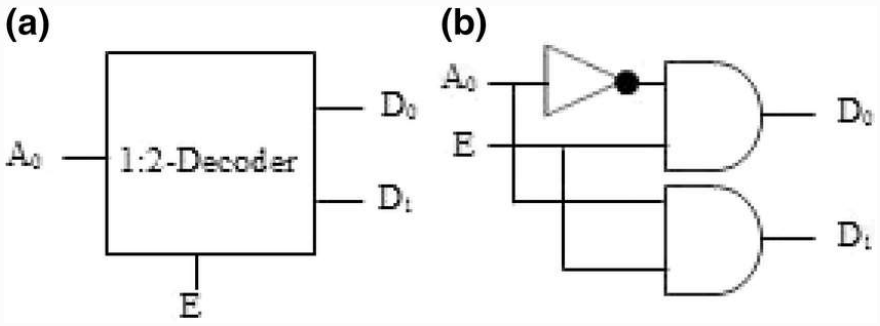


Fig1. (a)Symbol and (b)Circuit Diagram of 1x2 decoder using AND and NOT gates

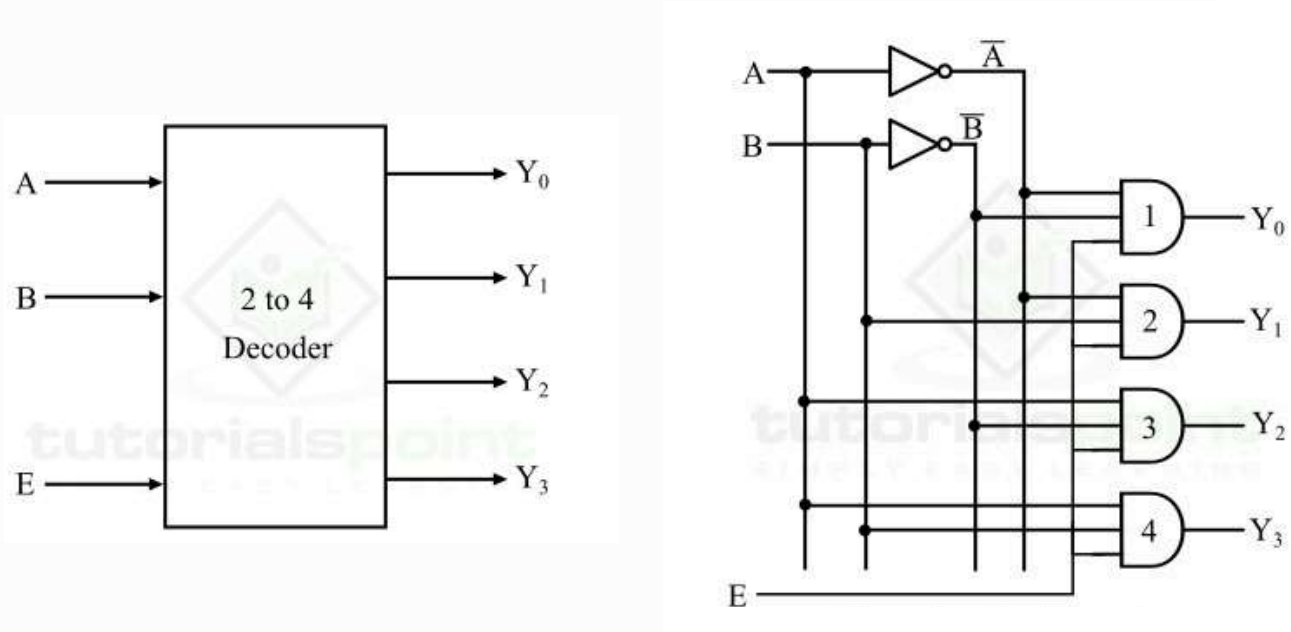


Fig2. (a)Symbol and (b)Circuit Diagram of 2x4 decoder using AND and NOT gates

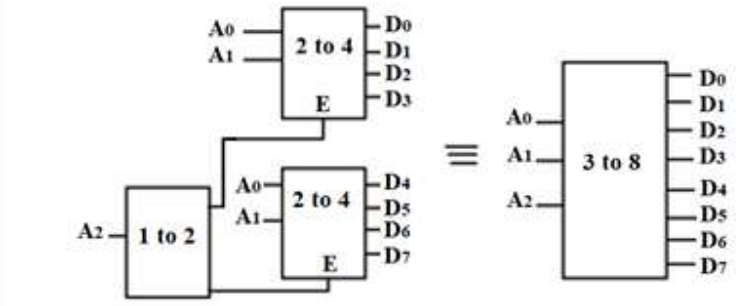


Fig3. Building a 3*8 decoder using a group of 2*4 decoders and a 1x2 decoder

Schematic diagram (in eSim):

- Root Schematic diagram:

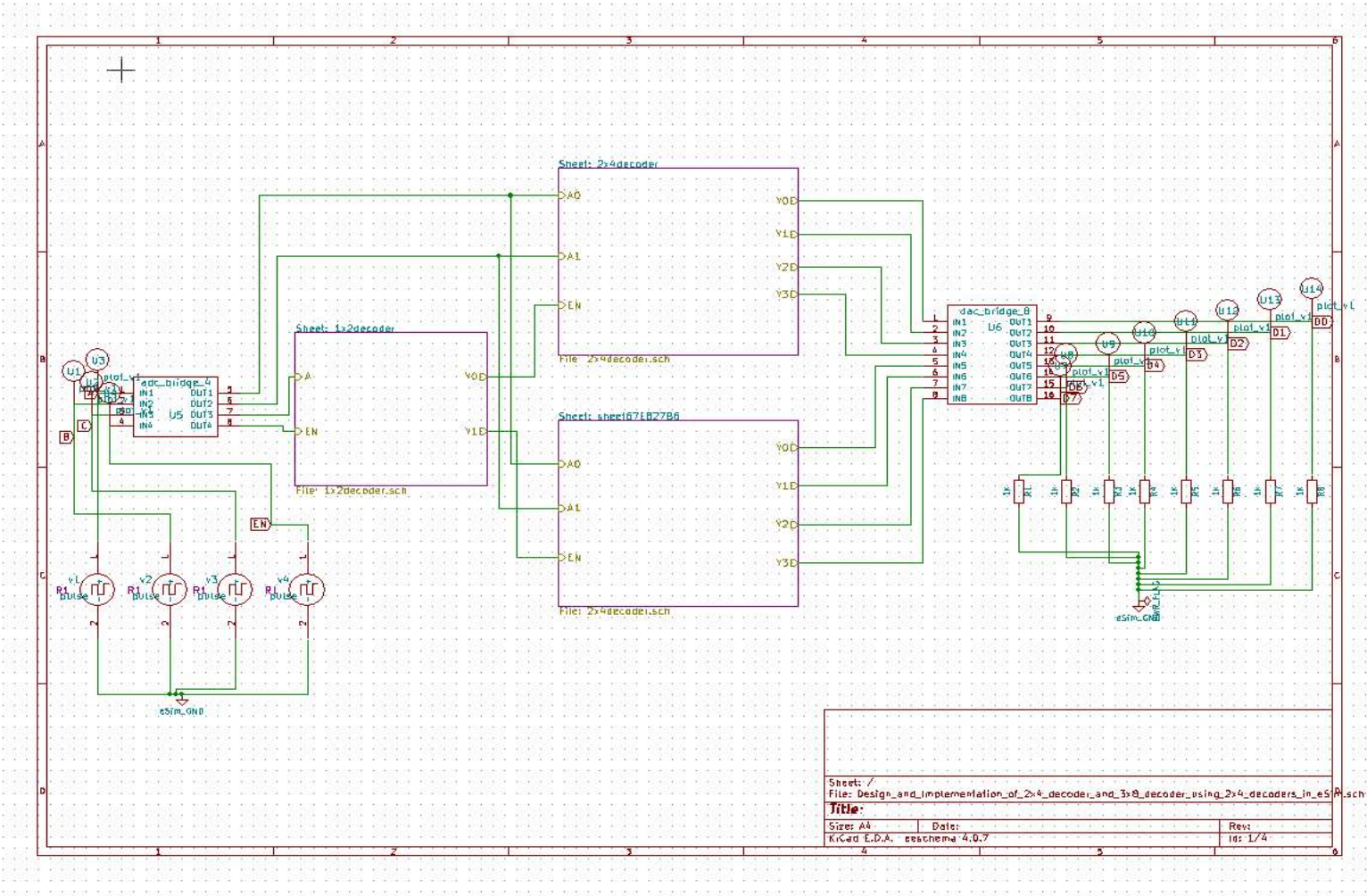
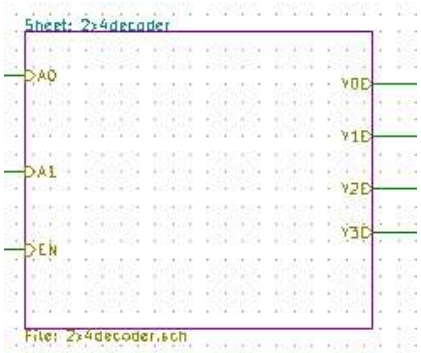
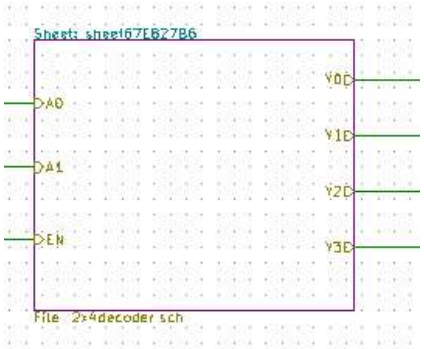


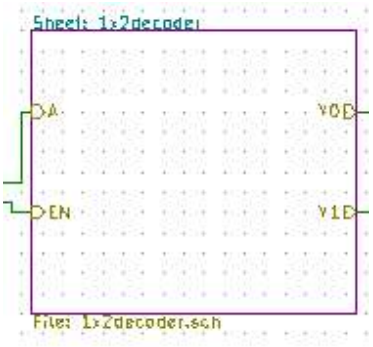
Fig4. Kicad Root Schematic diagram for 3x8 decoder made using one 1x2 decoder and two 2x4 decoders

- Hierarchical Sheet symbol for 2x4 decoder:

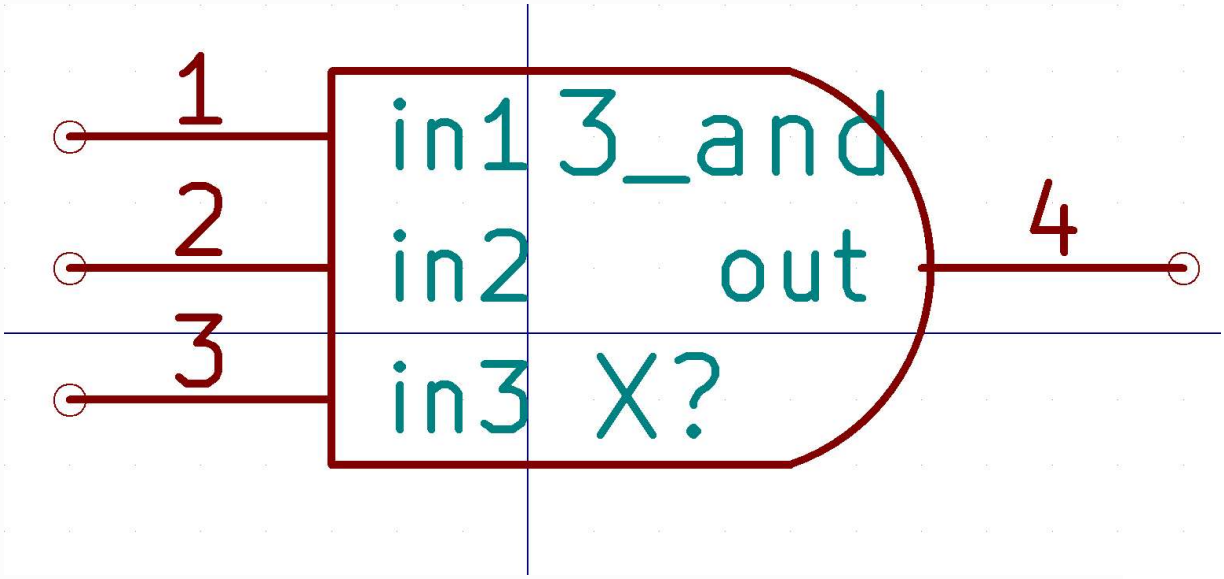




- Hierarchical Sheet symbol for 1x2 decoder:



- Symbol for 3 input AND gate subcircuit used in 2x4 decoder(Already exists in eSim_Subckt):



- 2x4 Decoder (made using hierarchical sheets)

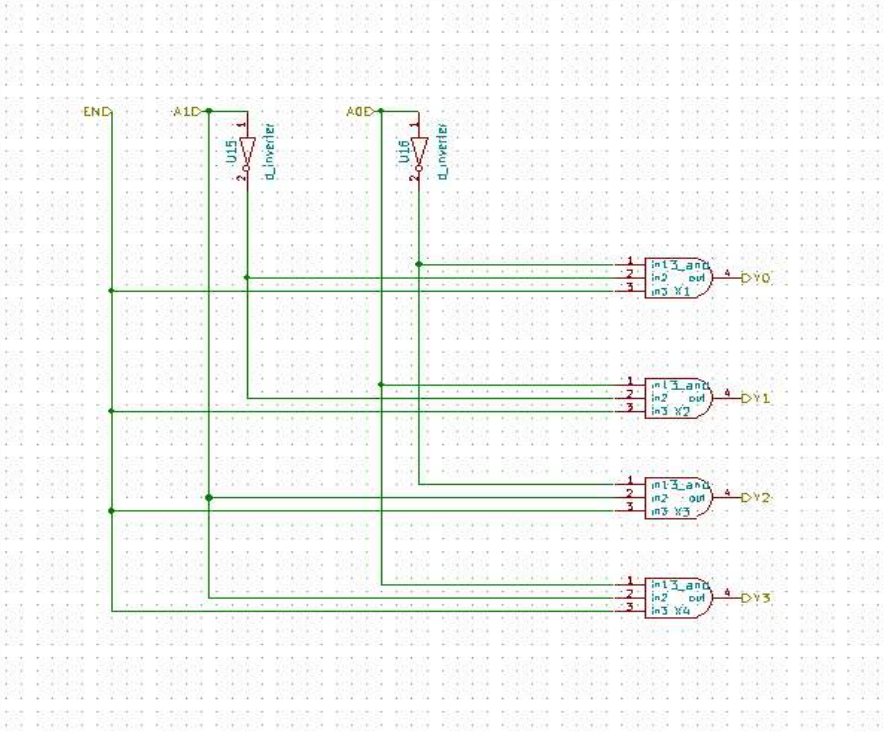


Fig5. 2x4 Decoder schematic diagram

- 1x2 Decoder(made using hierarchical sheets)

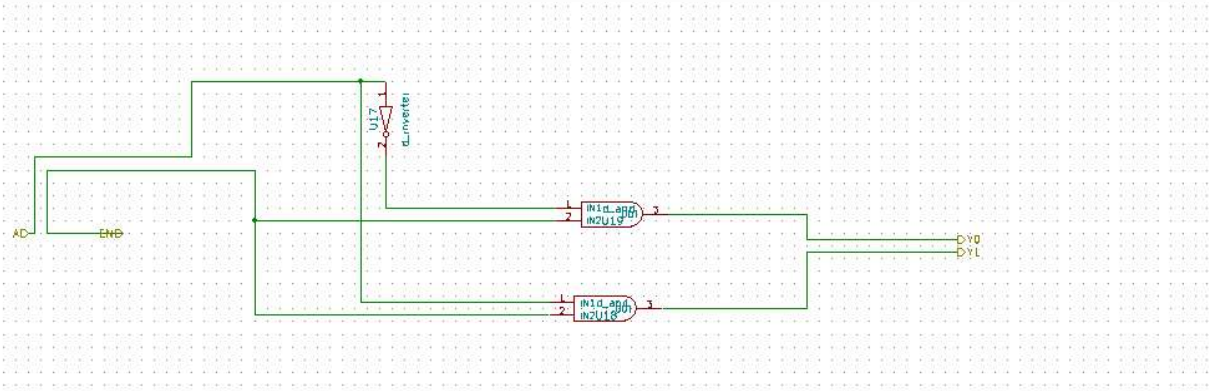
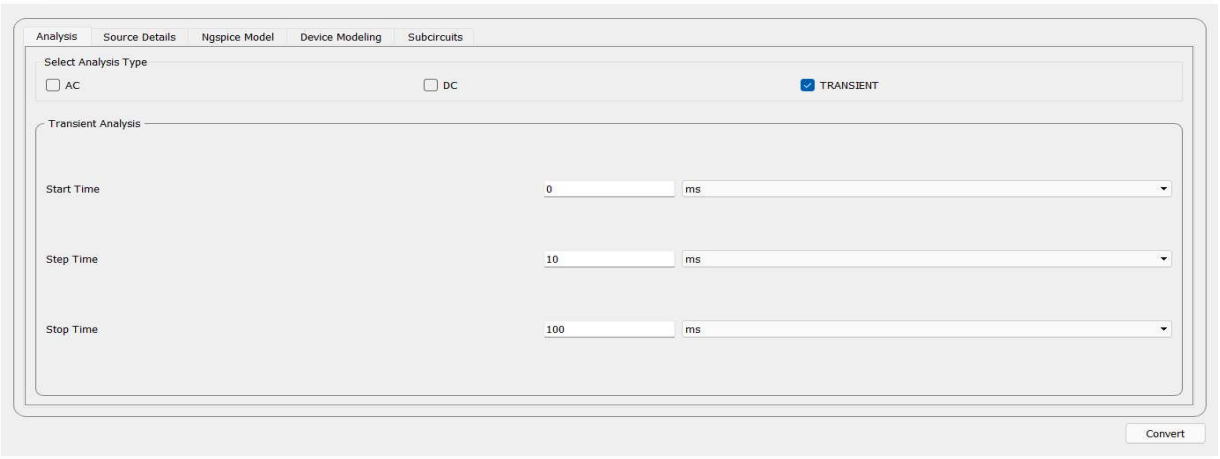


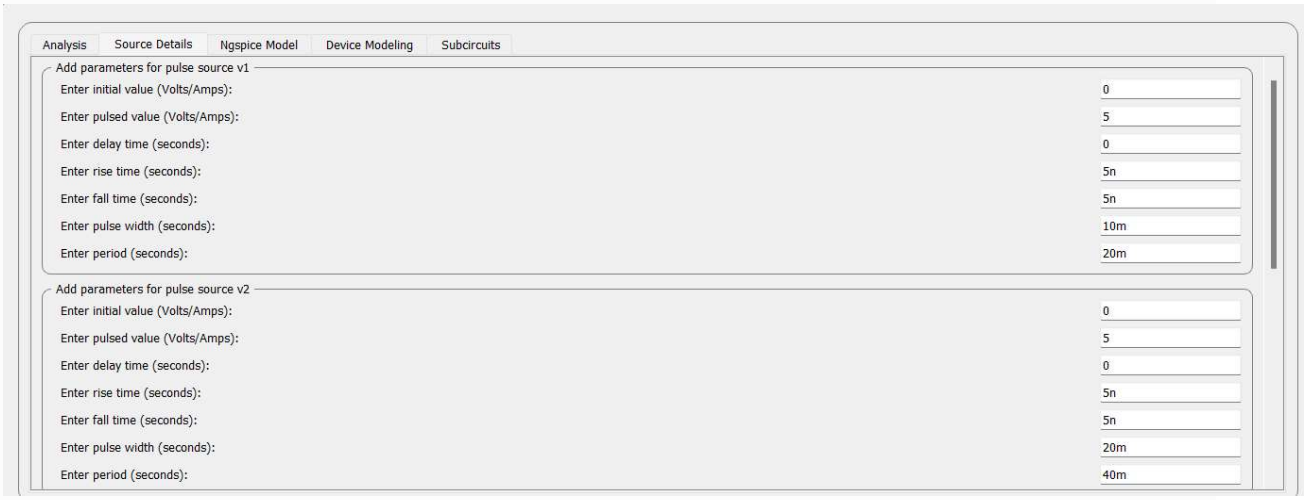
Fig6. 1x2 Decoder schematic diagram

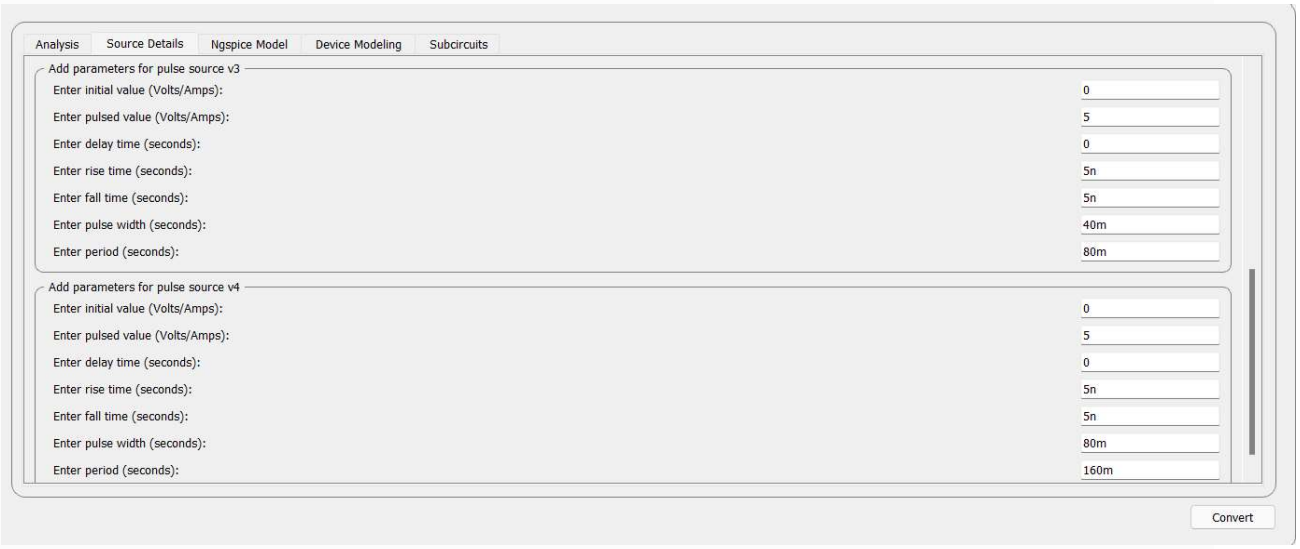
Kicad to NgSpice Conversion details:

- Analysis:



- Source Details:



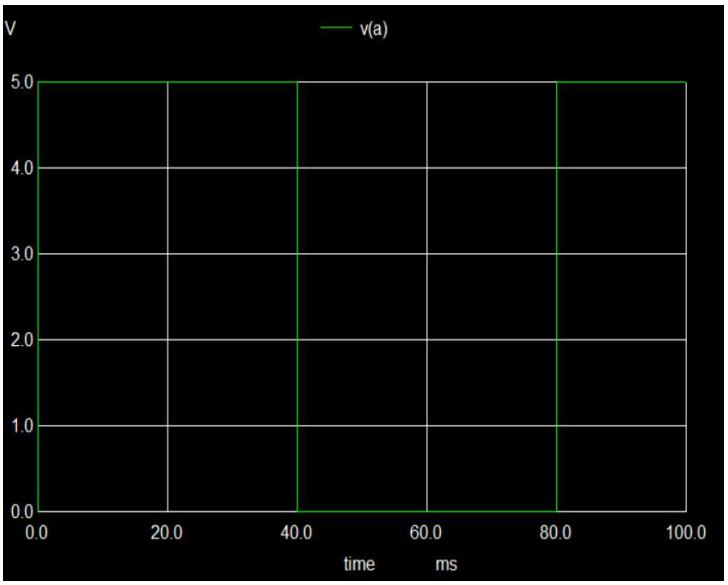


Results/Output:

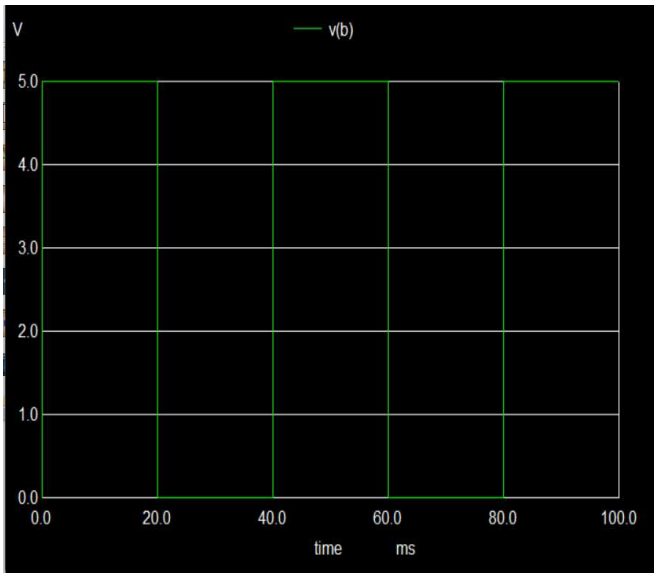
A. Ngspice Plots

- Input Waveforms:

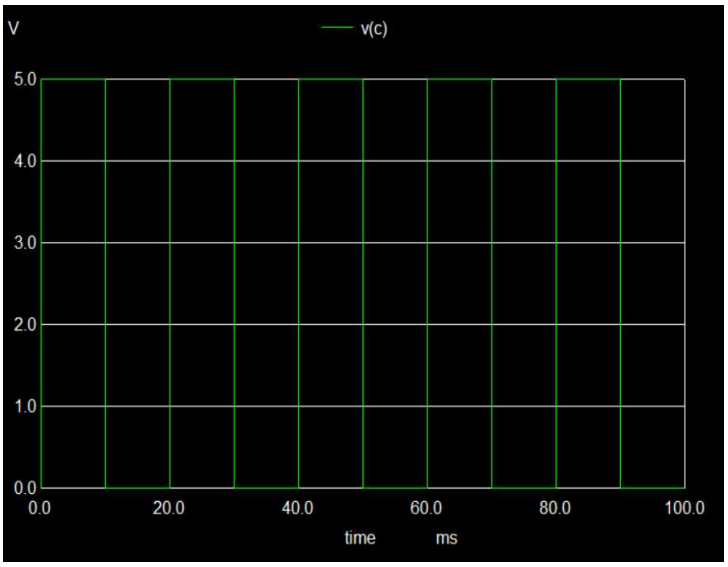
1. A:



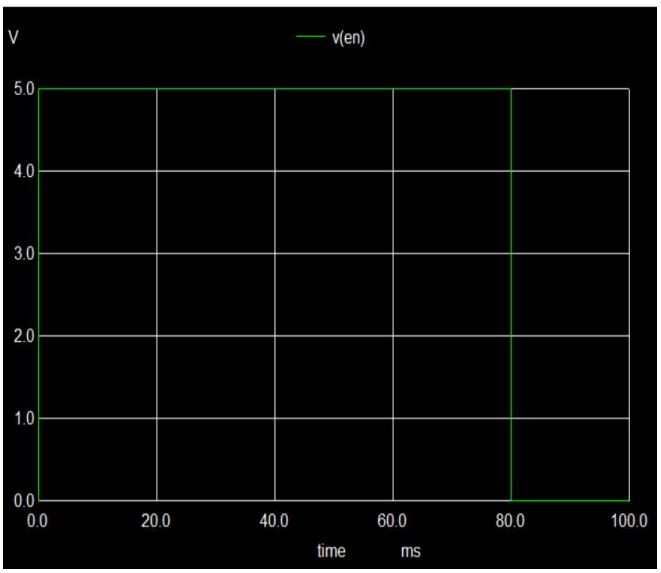
2. B:



3. C:

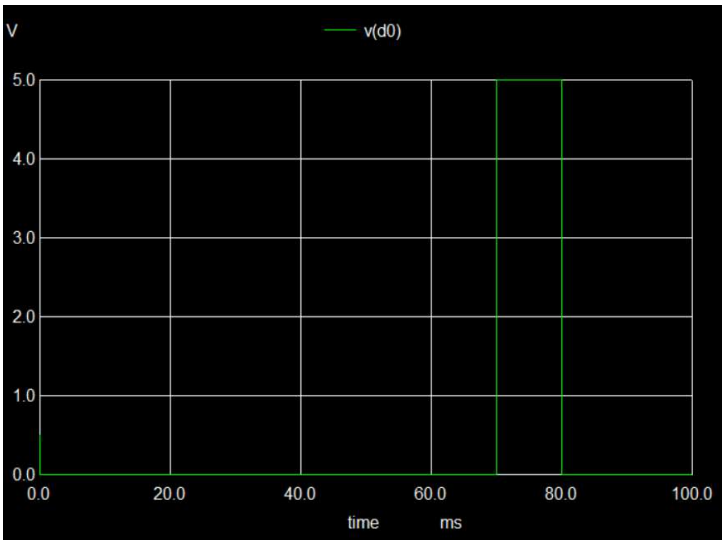


4. EN:

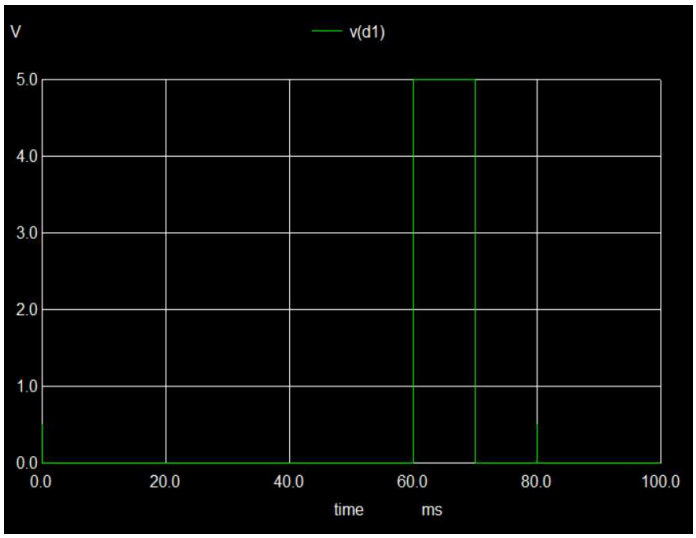


- Output Waveforms:

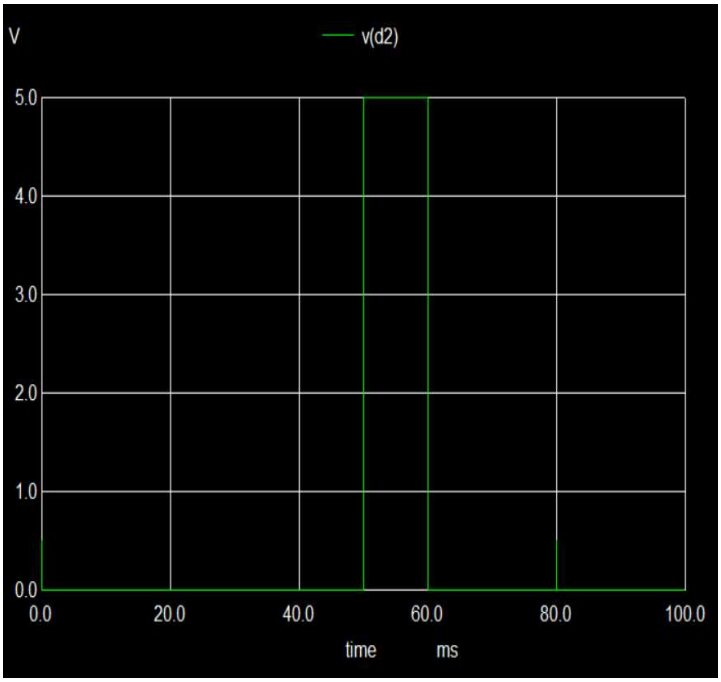
1. D0:



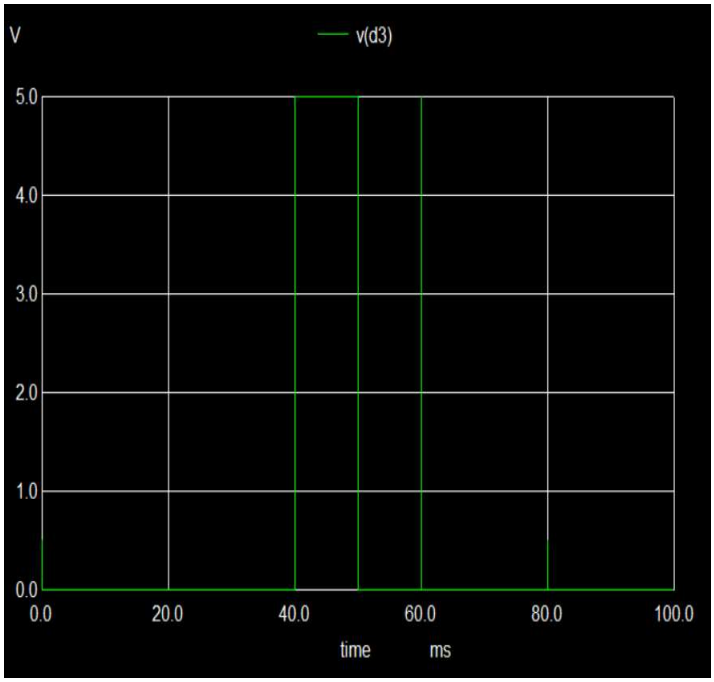
2. D1:



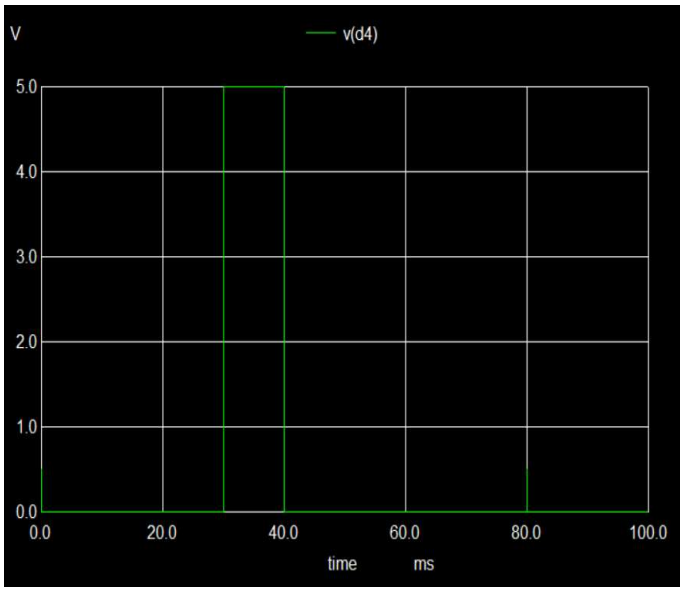
3. D2:



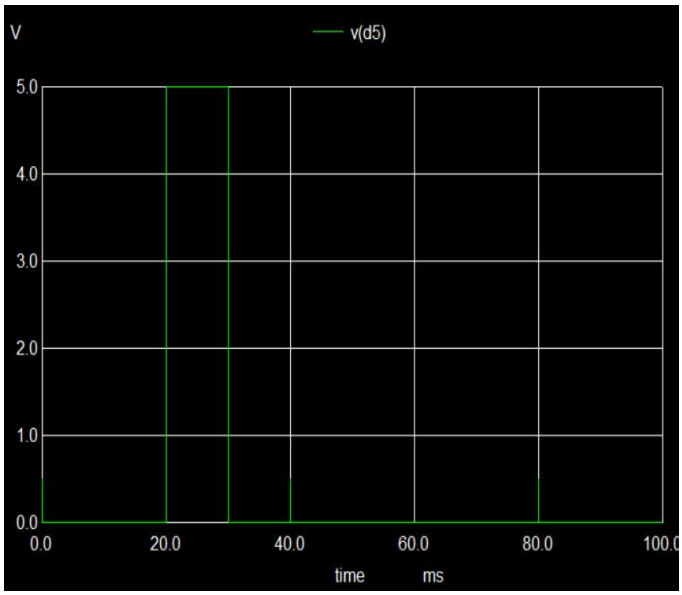
4. D3:



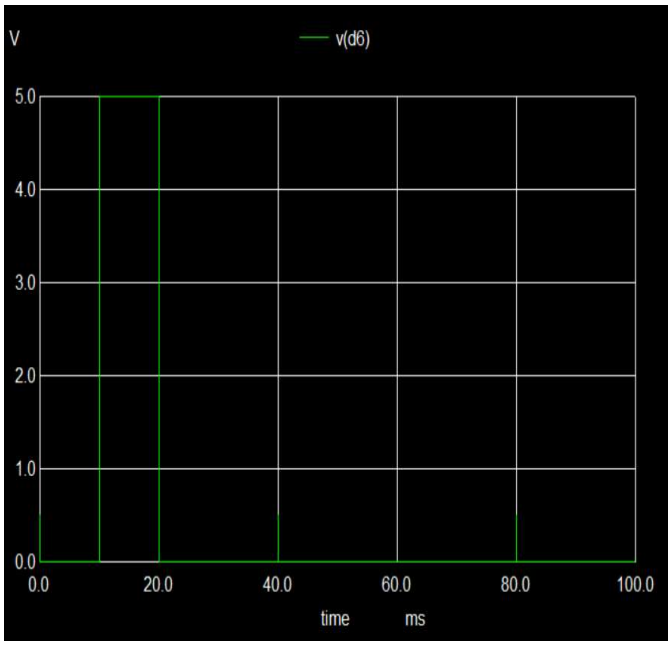
5. D4:



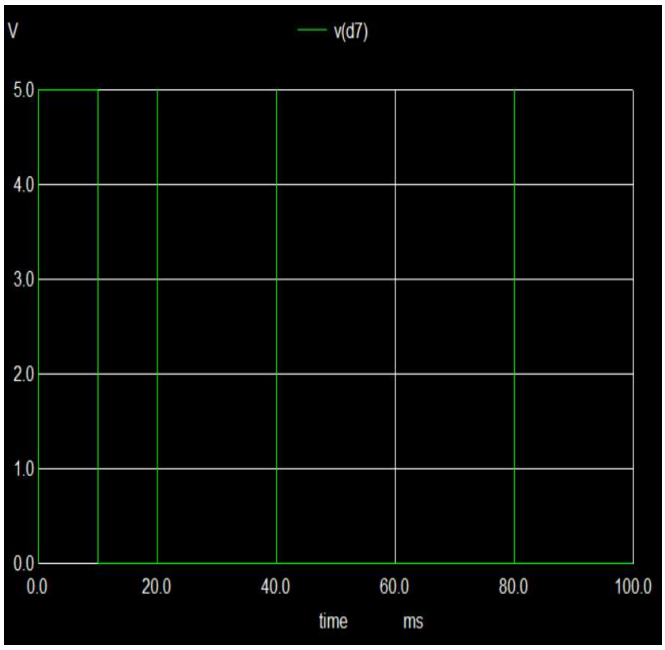
6. D5:



7. D6:



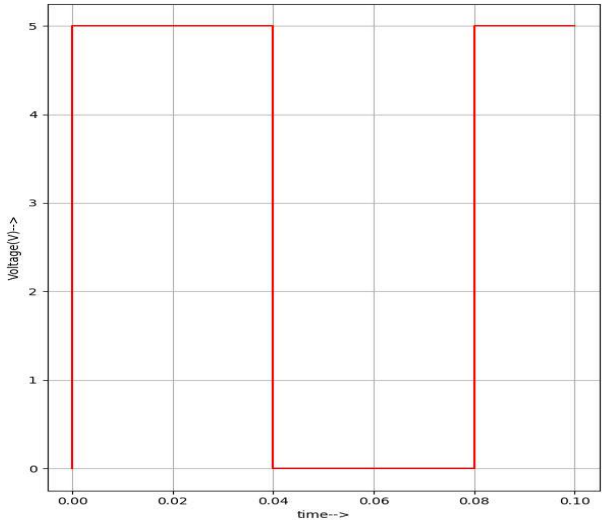
8. D7:



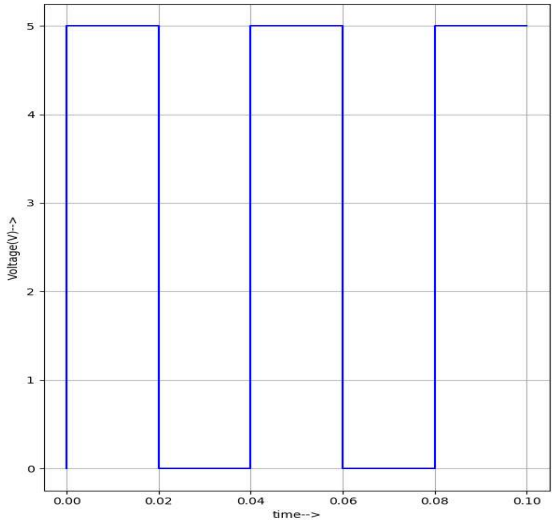
B. Python Plots

- Input Waveforms:

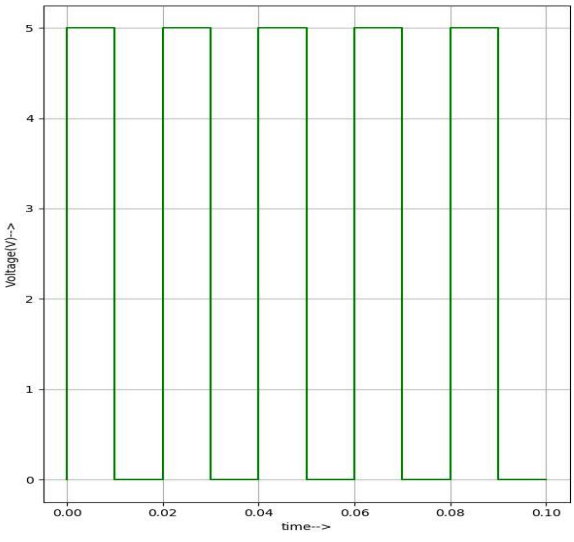
1. A:



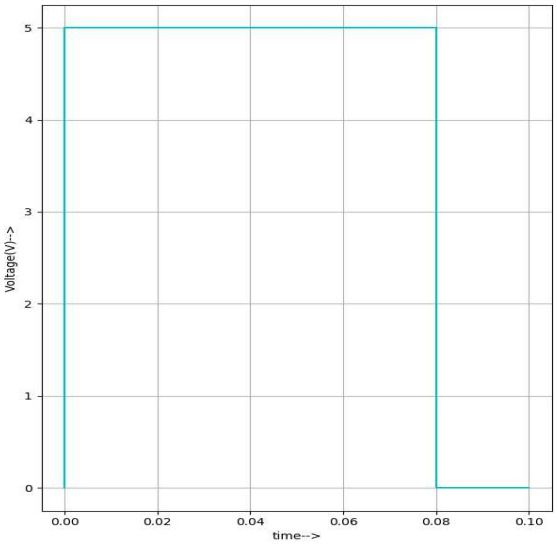
2. B:



3. C:

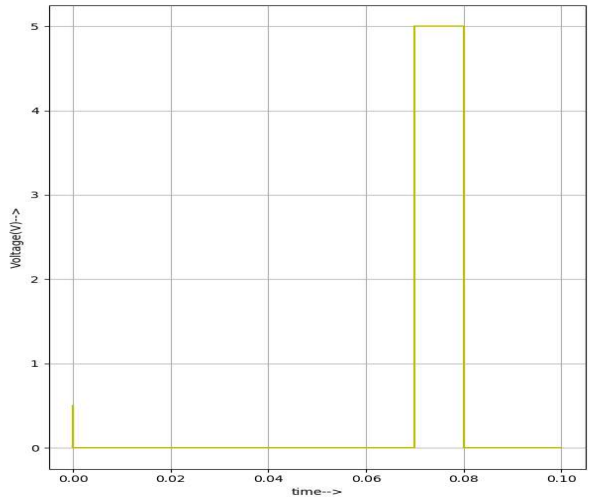


4. EN:

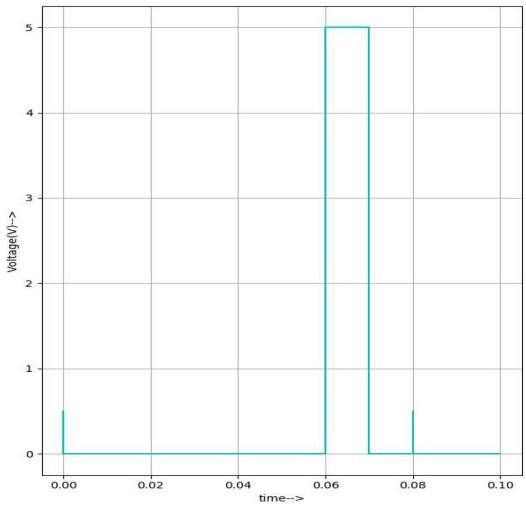


- Output Waveforms:

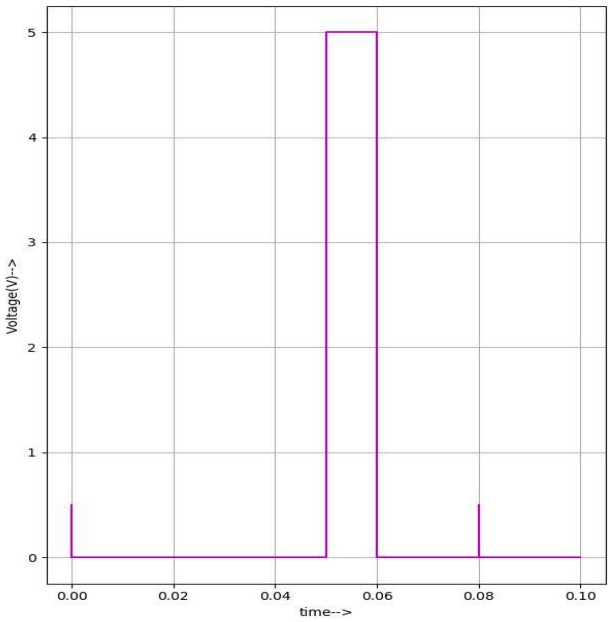
1. D0



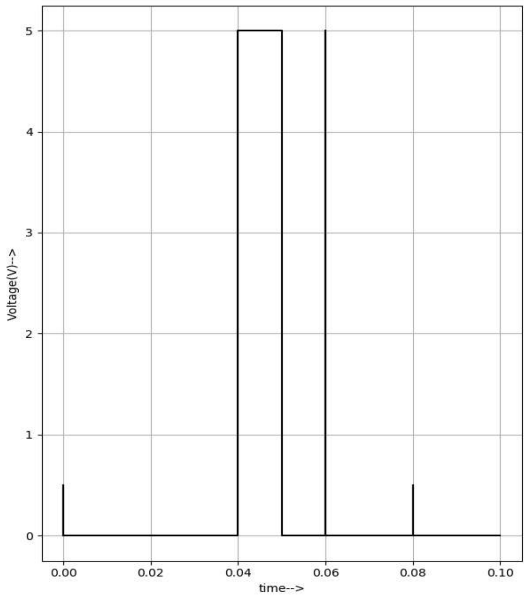
2. D1



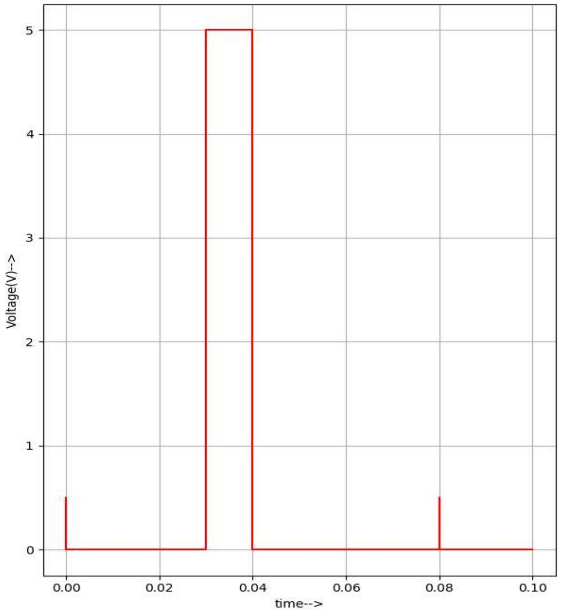
3. D2



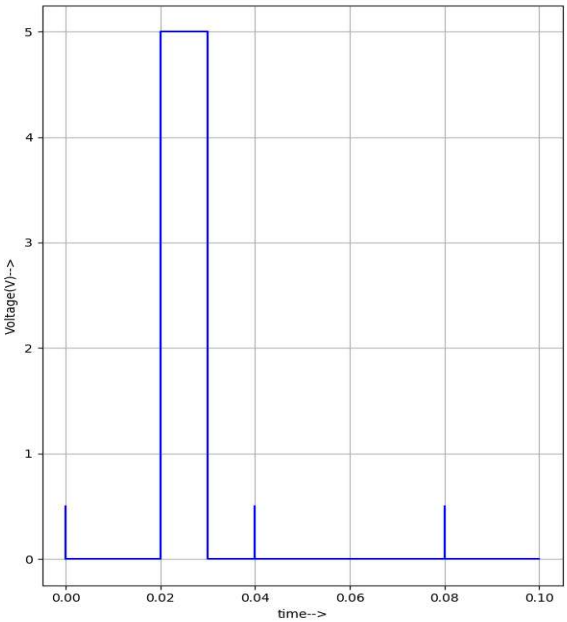
4. D3



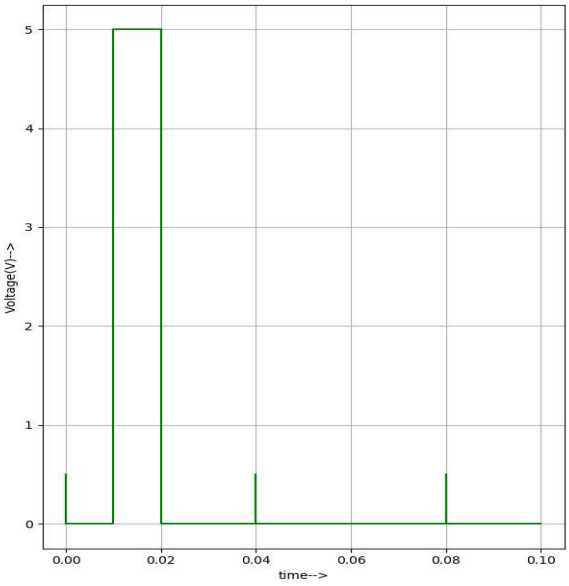
5. D4:



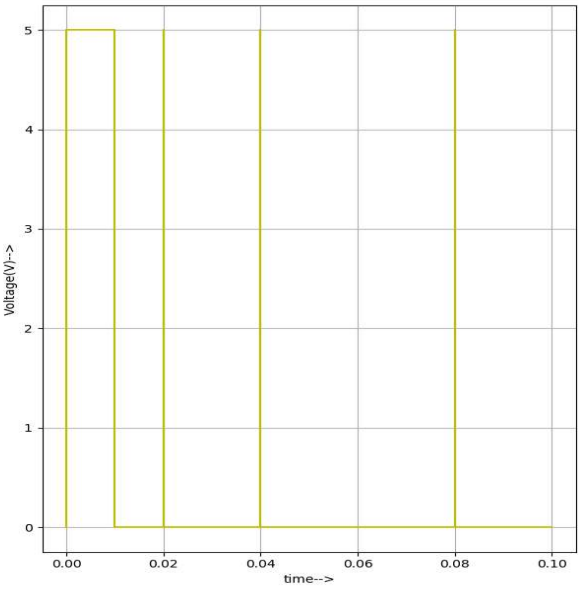
6. D5:



7. D6:



8. D7:



References:

- https://www.tutorialspoint.com/digital-electronics/digital-electronics-decoders.htm
- https://www.ijcaonline.org/archives/volume186/number38/baqar-2024-ijca-923962.pdf
- https://images.app.goo.gl/HsosoKtj7XpNtpjL9