Circuit Simulation Project

https://esim.fossee.in/circuit-simulation-project

Name of Participant: S Gopi

Project Guide: Dr. Maheswari. R

Title of the Project: -

Design Of a 4-Bit Binary to BCD Code Converter Ciruit using Subcircuit Builder in eSIM

Theory/Description: -

- **BCD** - BCD or Binary Coded Decimal is a number system which assigns a four-bit binary code to each digit from 0 through 9 in a decimal (base-10) numeral. The decimal numbers 1- 9 are represented in their normal 4-bit binary form, but from the decimal number 10, we represent each number as its 4-bit binary equivalent.

  For example, the two-digit decimal number = 10 is divided into individual digits: 1 and 0. The binary representation of individual digits (4-bit)

  1 = 0001
  0 = 0000
  Hence, 10 = 0001 0000 = 1 0000
  Here, the leading zeros can be omitted as they have no significance.

Note: -

- *Similarly, when Binary representations of Decimal numbers 10 – 15 are converted to BCD, the leading zeros are omitted (as they have no significance) in the below designed 4-Bit Binary to BCD Code Converter.*

- *Though this is a 4-Bit Code Converter but when Binary representations of Decimal numbers 10 – 15 are converted to BCD, the output is a 5-Bit (ignoring leading zeros) BCD code and hence to incorporate and show the conversion of 16 combinations of Binary code to BCD code, I have designed a Code Converter circuit which takes 4-Bit binary code as input and produces 5-Bit BCD code as output. So, if you want the output also to be a 4-Bit BCD code, then we can show conversion of only 10 combinations of Binary code.*

- **Binary** - A binary code represents text, computer processor instructions, or any other data using a two-symbol system. The two-symbol system used is often "0" and "1" from the binary number system. The binary code assigns a pattern of binary digits, also known as bits, to each character, instruction, etc. For example, a binary string of eight bits can represent any of 256 possible values and can, therefore, represent a wide variety of different items.

Truth Table: -

The following truth table shows the conversion between the binary code input and the BCD code output. As you see from the table, the 4-bit binary number is converted into 5-bit BCD code. Decimal code is added in the table to understand the equivalence of Binary and BCD code.

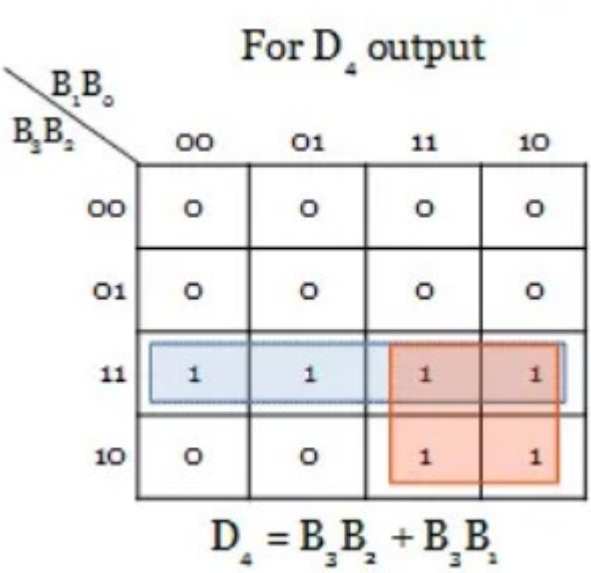
Decimal Number	Binary code (Input)				BCD code (Output)				
	B_3	B_2	B_1	B_0	D_4	D_3	D_2	D_1	D_0
0	0	0	0	0	0	0	0	0	0
1	0	0	0	1	0	0	0	0	1
2	0	0	1	0	0	0	0	1	0
3	0	0	1	1	0	0	0	1	1
4	0	1	0	0	0	0	1	0	0
5	0	1	0	1	0	0	1	0	1
6	0	1	1	0	0	0	1	1	0
7	0	1	1	1	0	0	1	1	1
8	1	0	0	0	0	1	0	0	0
9	1	0	0	1	0	1	0	0	1
10	1	0	1	0	1	0	0	0	0
11	1	0	1	1	1	0	0	0	1
12	1	1	0	0	1	0	0	1	0
13	1	1	0	1	1	0	0	1	1
14	1	1	1	0	1	0	1	0	0
15	1	1	1	1	1	0	1	0	1

Karnaugh Map (K-Map): -

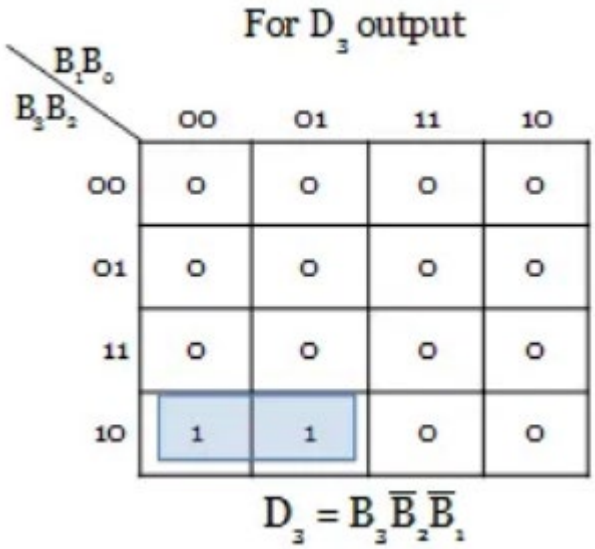
The converter has 5 outputs D0, D1, D2, D3 and D4. From the truth table, the minterms can be obtained for each output.

The minterms are plotted in the karnaugh map and the simplified boolean expressions are obtained.

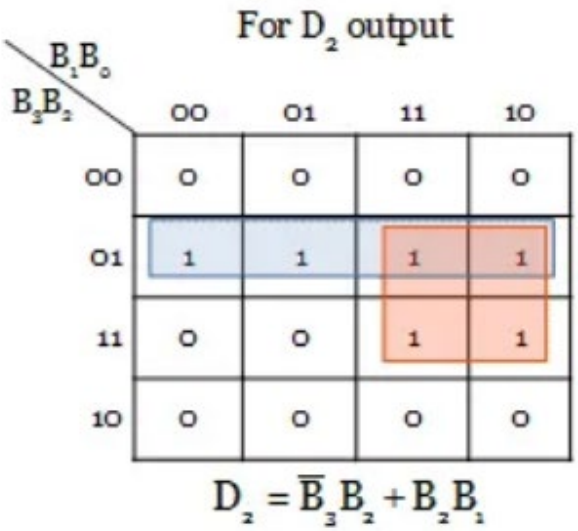
- D4 = ∑m(10, 11, 12, 13, 14, 15)



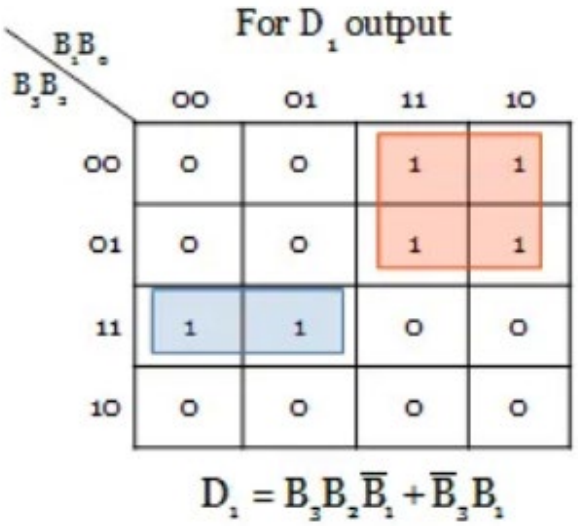
- D3 = ∑m(8, 9)



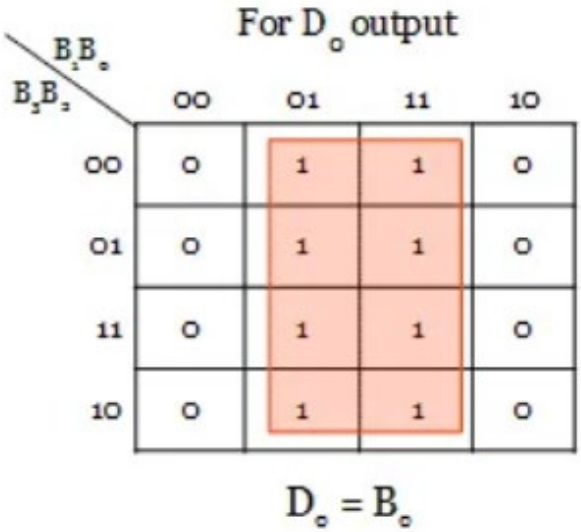
- D2 = ∑m(4, 5, 6, 7, 14, 15)



- D1 = ∑m(2, 3, 6, 7, 12, 13)

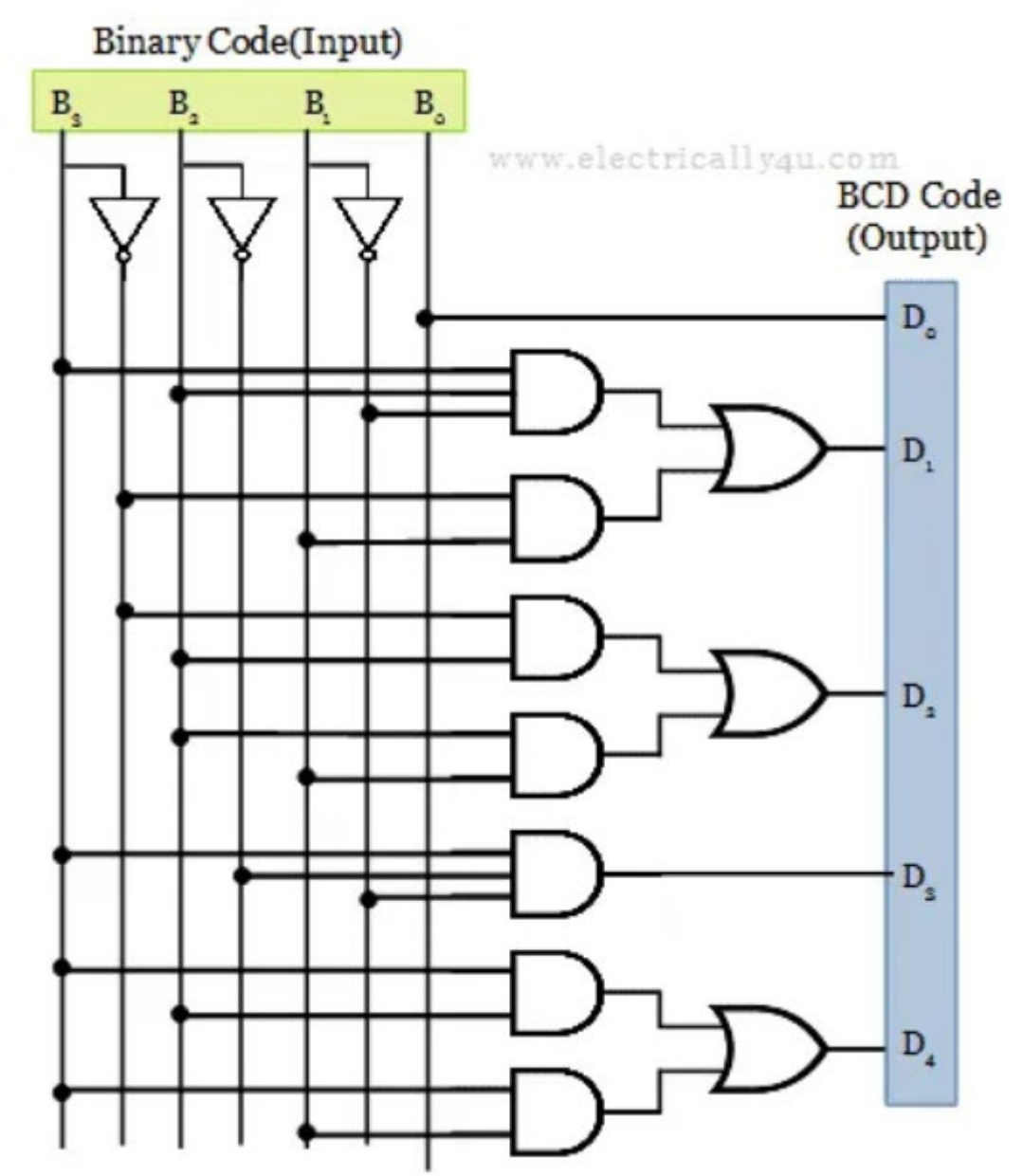


- D0 = ∑m(1, 3, 5, 7, 9, 11, 13, 15)



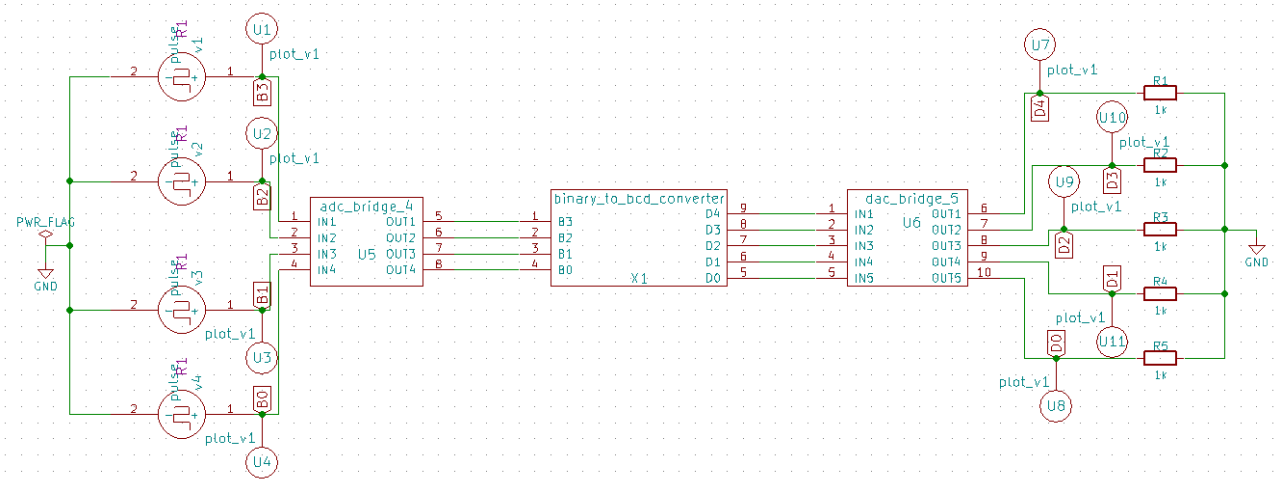
Digital Logic Circuit: -

The digital logic circuit for Binary to BCD code converter is designed from the simplified output expressions obtained from karnaugh map.



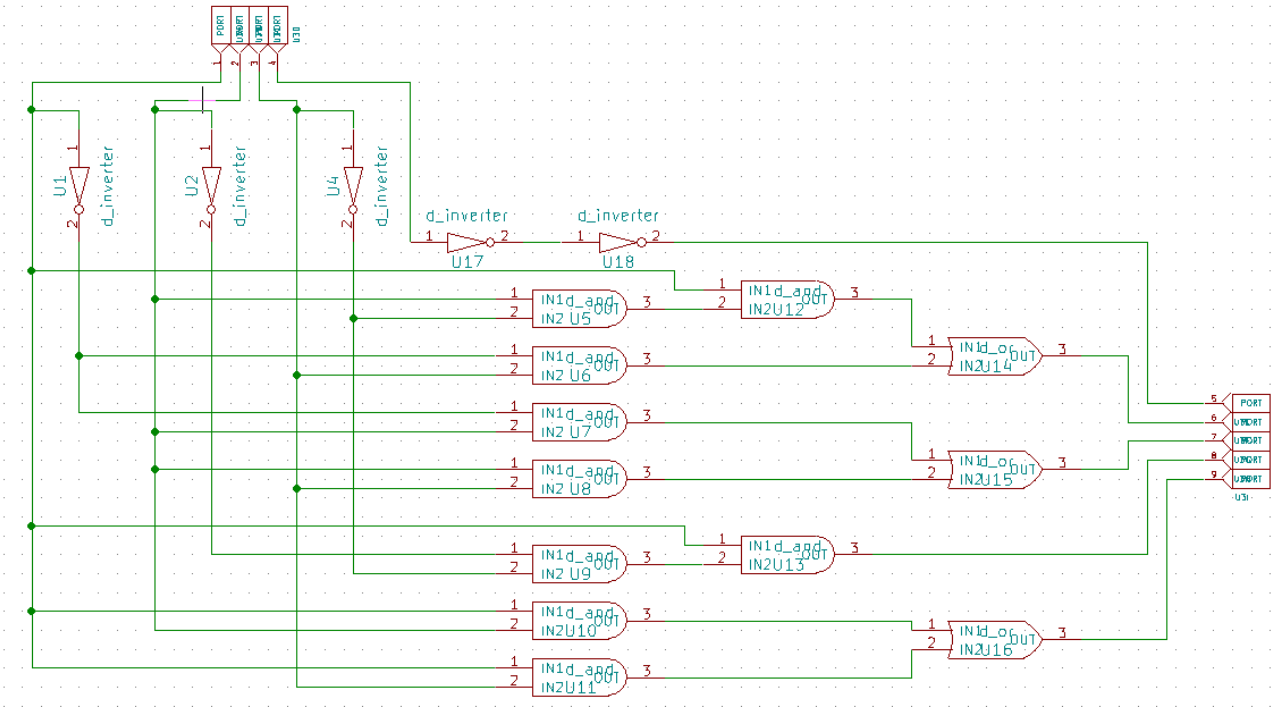
Circuit Diagrams: -

- This is the main functional circuit schematic of 4-Bit Binary to BCD Code Converter which uses a subcircuit:



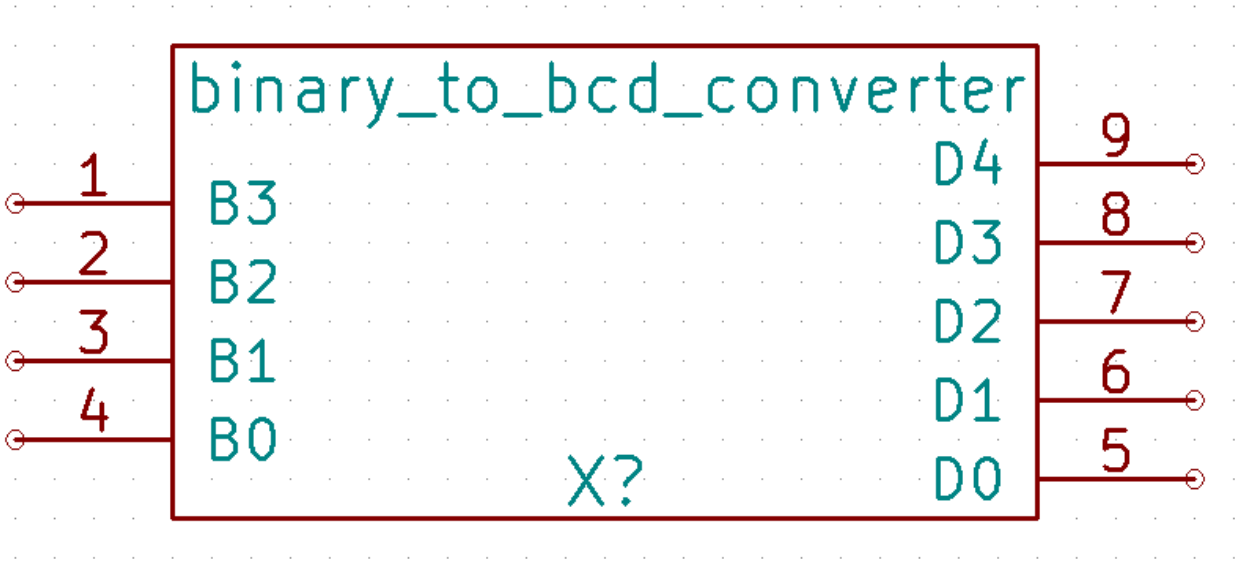
Main Circuit Schematic – 4-Bit Binary to BCD Code Converter using Subcircuit

- A subcircuit is used for converting Binary to BCD Code. The internal structure of the 4-Bit Binary to BCD Code Converter (subcircuit) is shown below:



Subcircuit Schematic for 4-Bit Binary to BCD Code Converter

- The symbol defined/designed to represent the subcircuit is shown below:

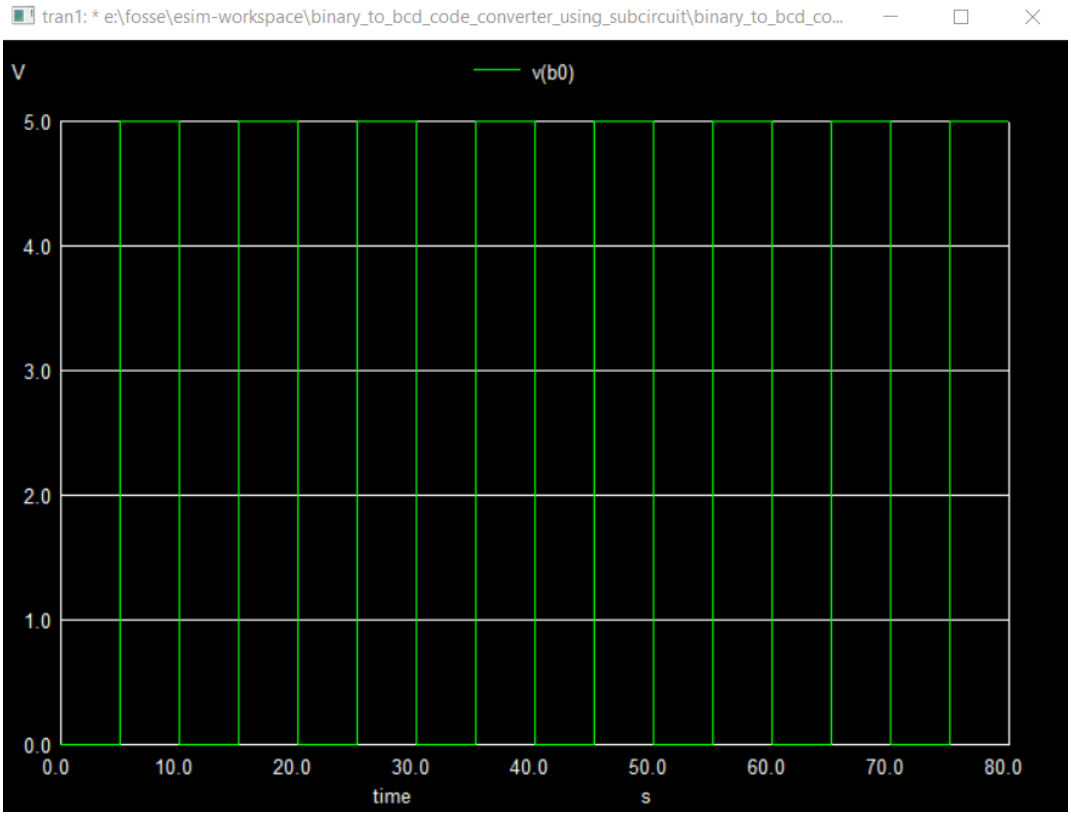


Subcircuit Symbol for 4-Bit Binary to BCD Code Converter

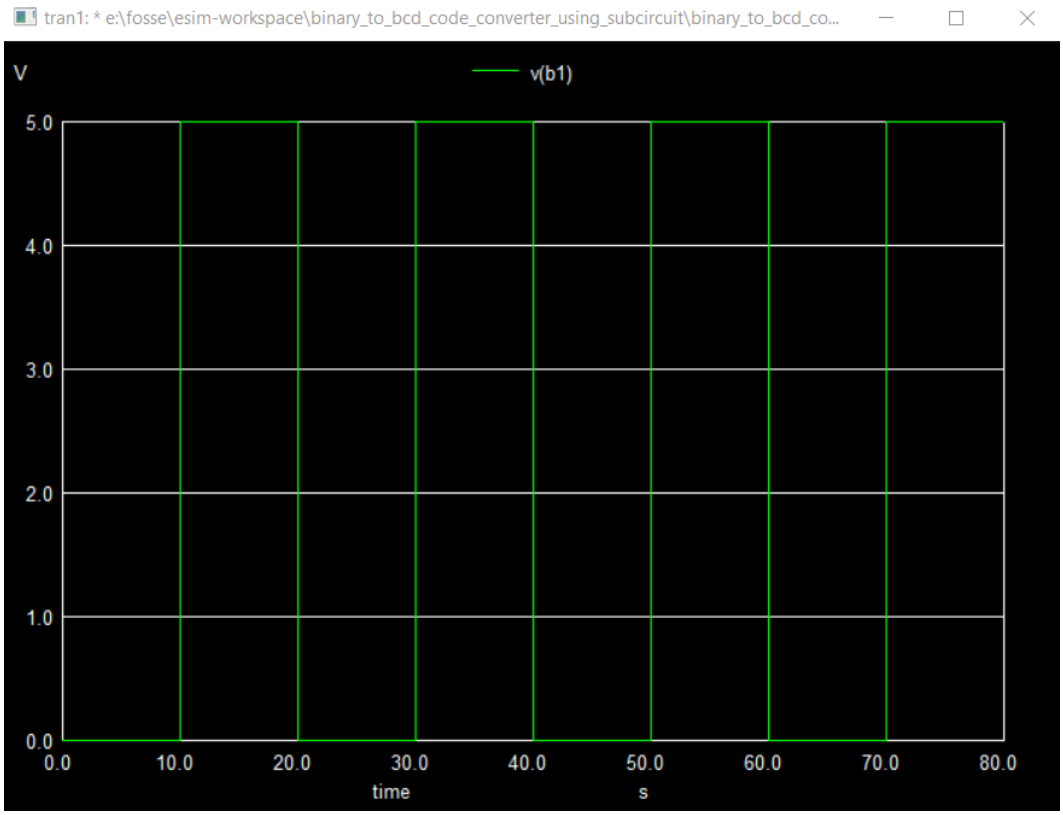
Result/Output: -

- **Ngspice Plots: -**
  - Inputs: -

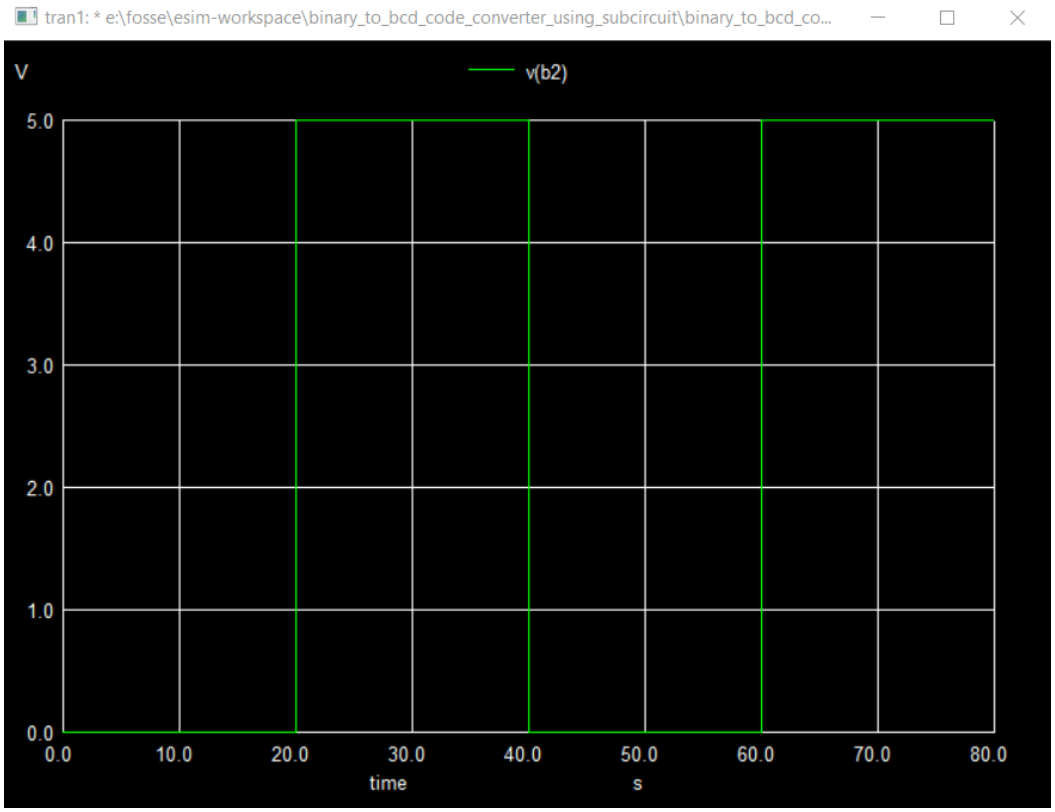
B0: -



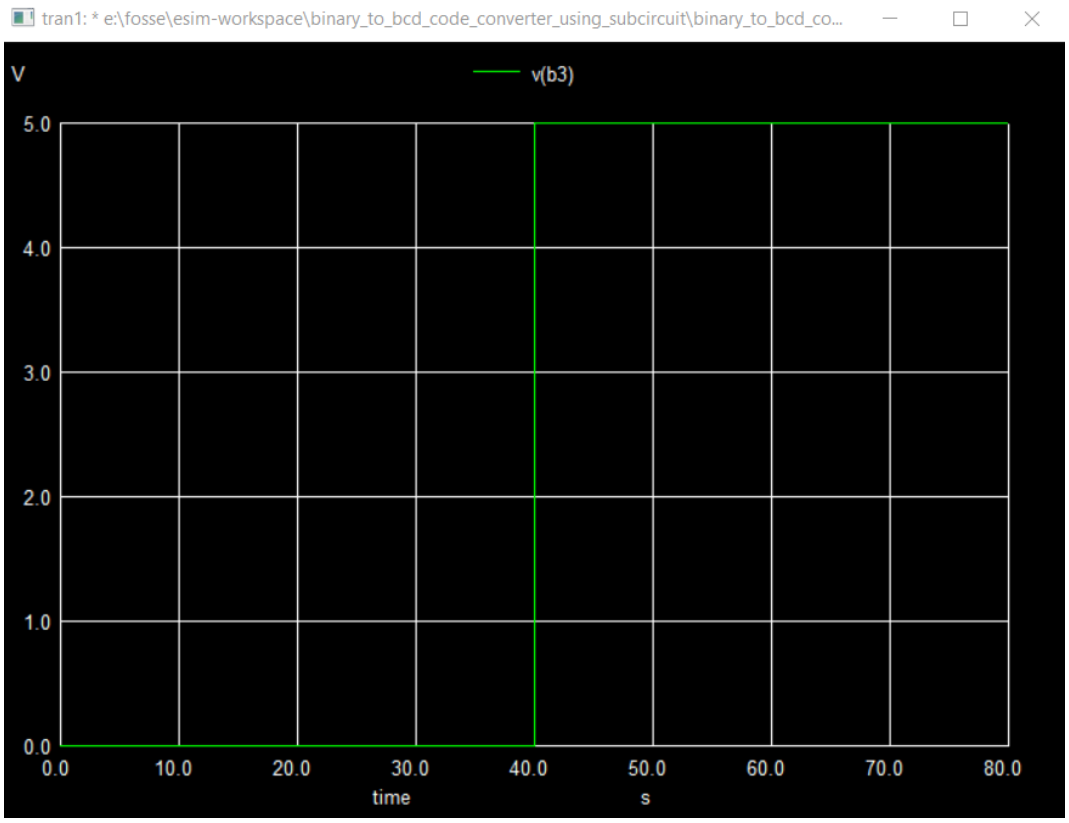
B1: -


tran1: * e:\fosse\esim-workspace\binary_to_bcd_code_converter_using_subcircuit\binary_to_bcd_co...
V
v(b1)
5.0
4.0
3.0
2.0
1.0
0.0
0.0
10.0
20.0
30.0
40.0
50.0
60.0
70.0
80.0
time
s

B2: -

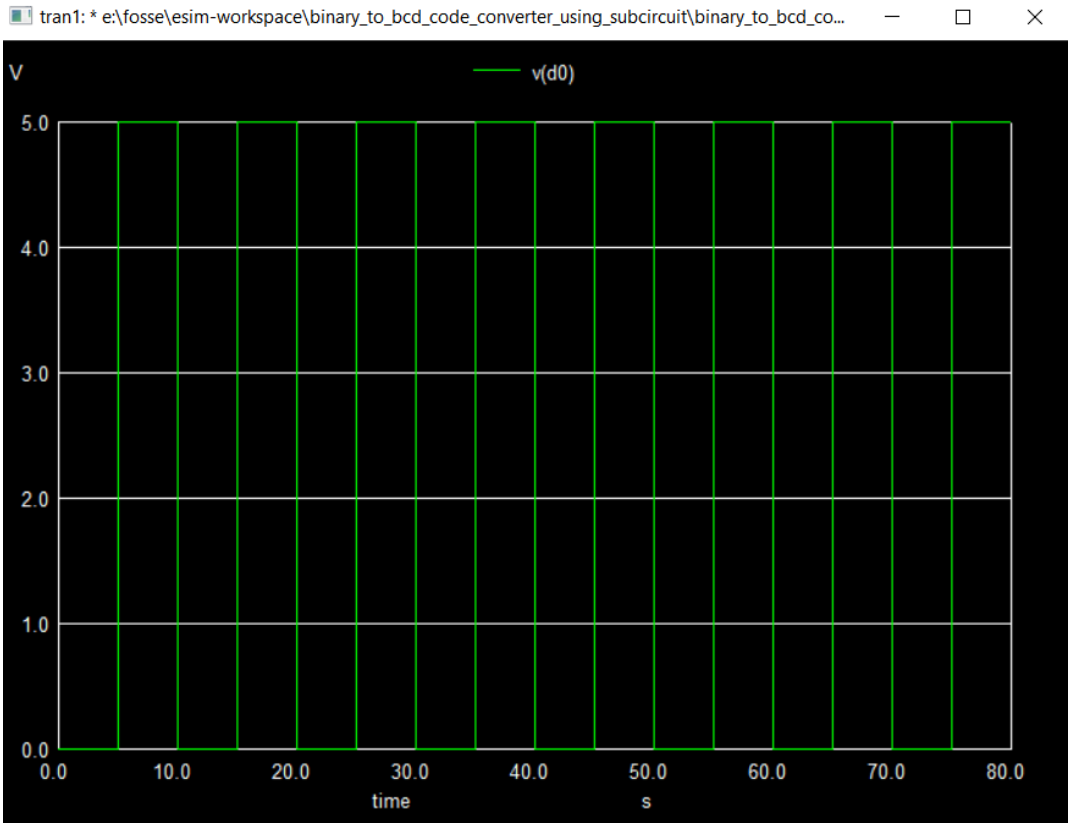

tran1: * e:\fosse\esim-workspace\binary_to_bcd_code_converter_using_subcircuit\binary_to_bcd_co...
V
v(b2)
5.0
4.0
3.0
2.0
1.0
0.0
0.0
10.0
20.0
30.0
40.0
50.0
60.0
70.0
80.0
time
s

B3: -

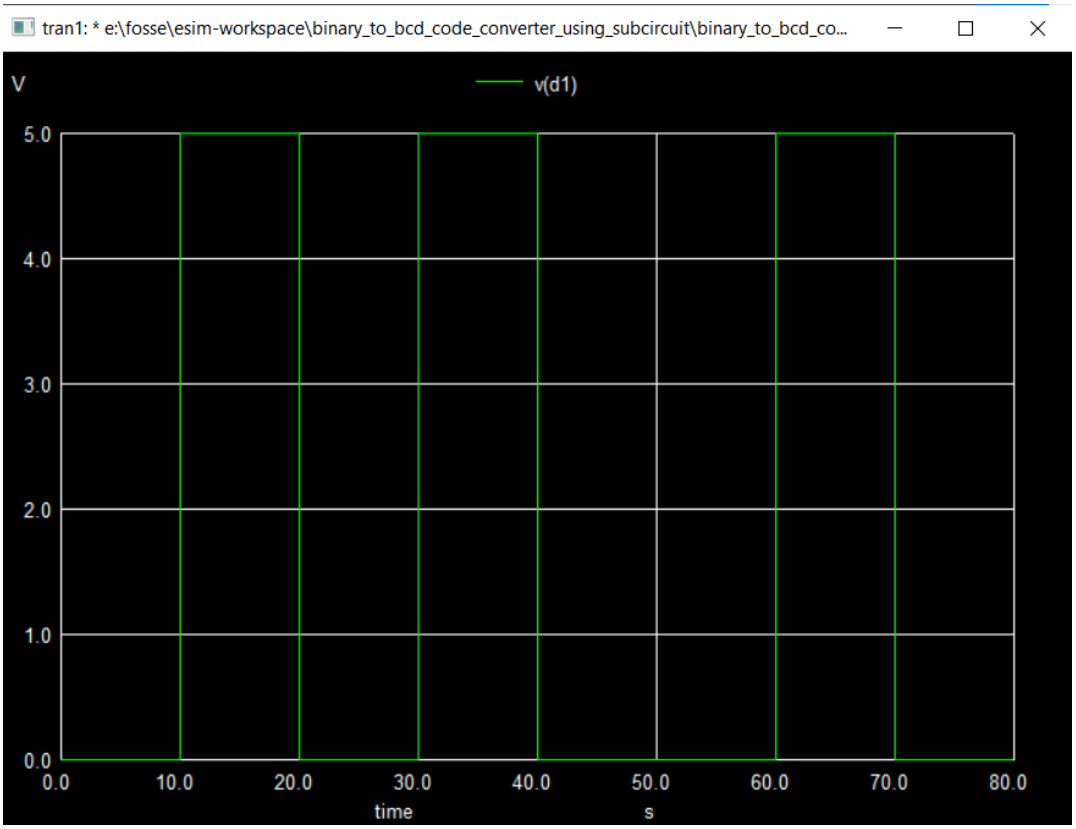


- Outputs: -

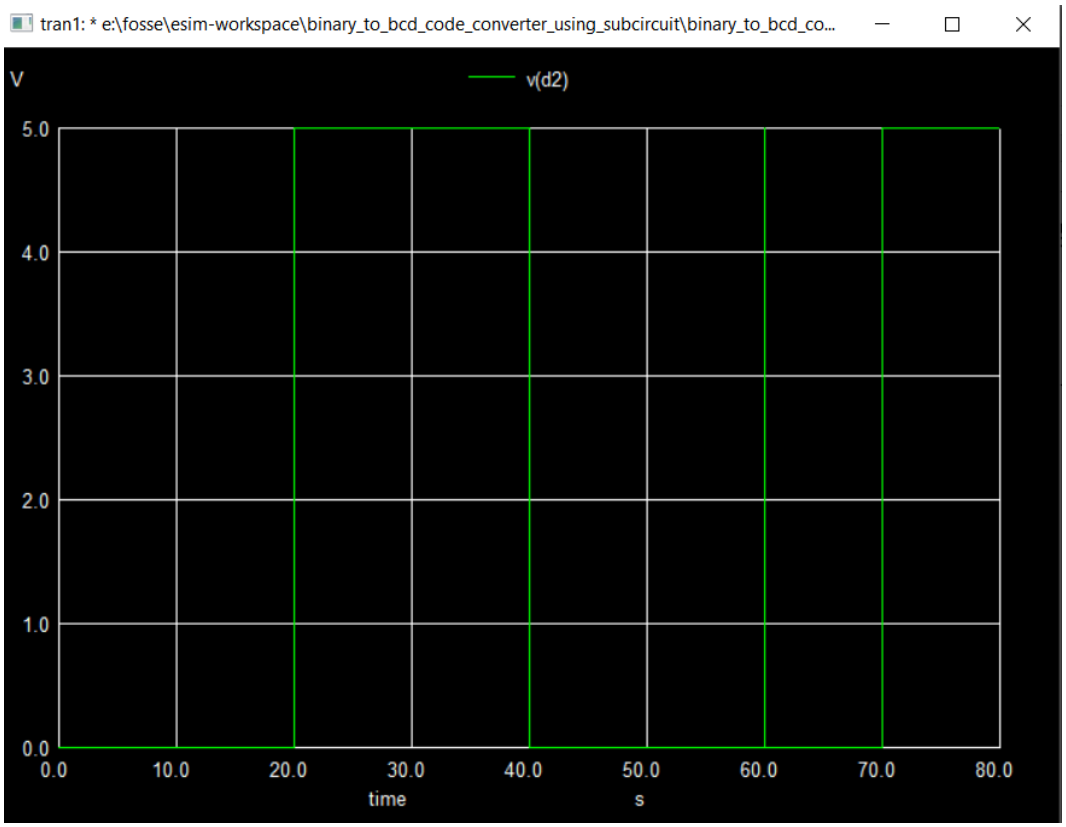
D0: -



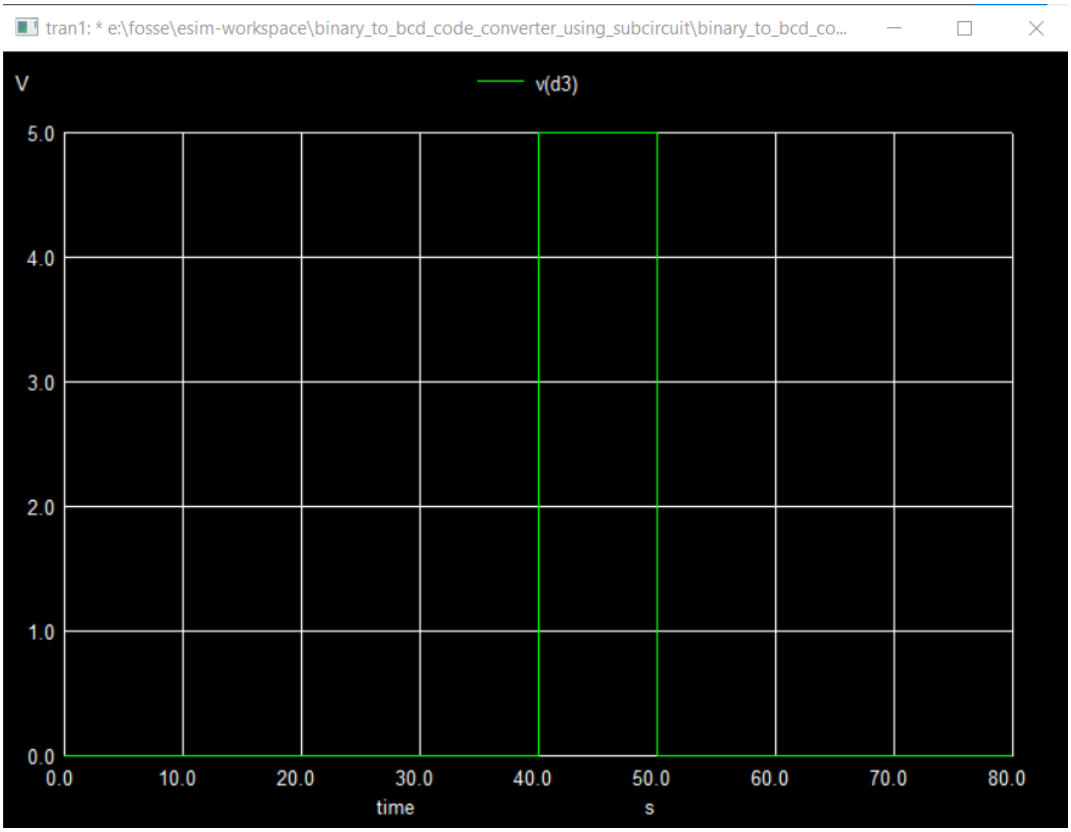
D1: -


tran1: * e:\fosse\esim-workspace\binary_to_bcd_code_converter_using_subcircuit\binary_to_bcd_co...
V
v(d1)
5.0
4.0
3.0
2.0
1.0
0.0
0.0
10.0
20.0
30.0
40.0
50.0
60.0
70.0
80.0
time
s

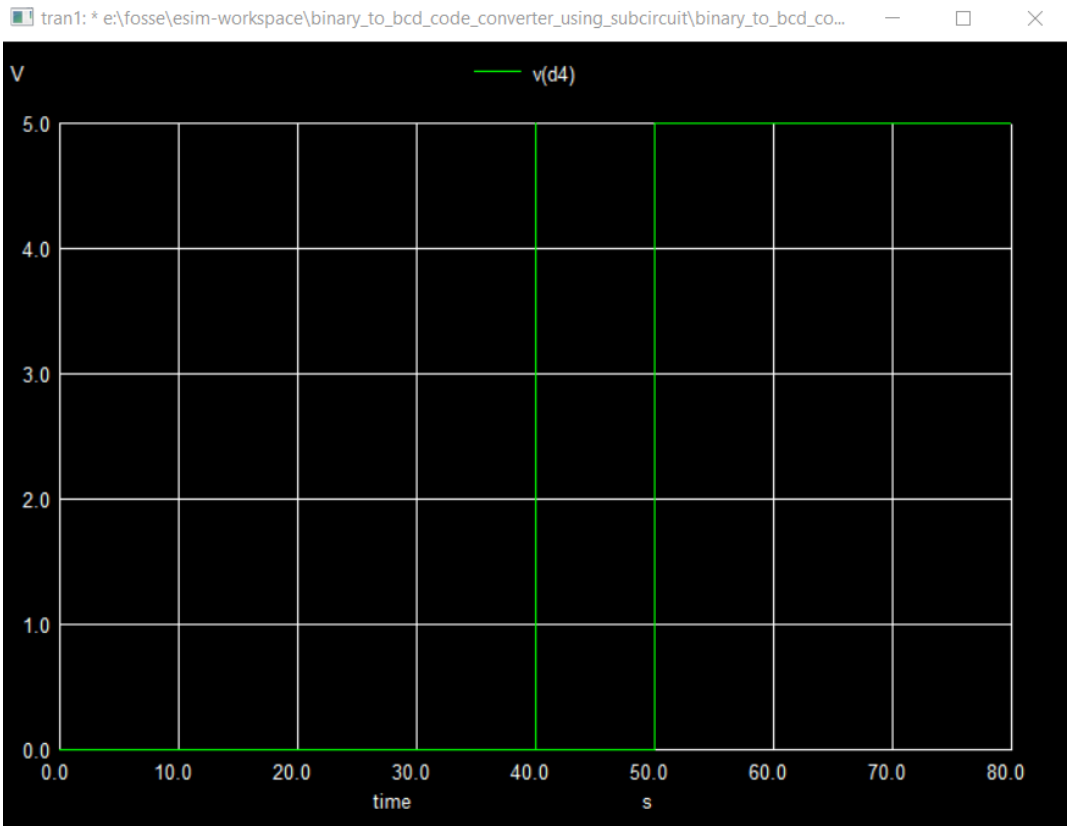
D2: -


tran1: * e:\fosse\esim-workspace\binary_to_bcd_code_converter_using_subcircuit\binary_to_bcd_co...
V
v(d2)
5.0
4.0
3.0
2.0
1.0
0.0
0.0
10.0
20.0
30.0
40.0
50.0
60.0
70.0
80.0
time
s

D3: -

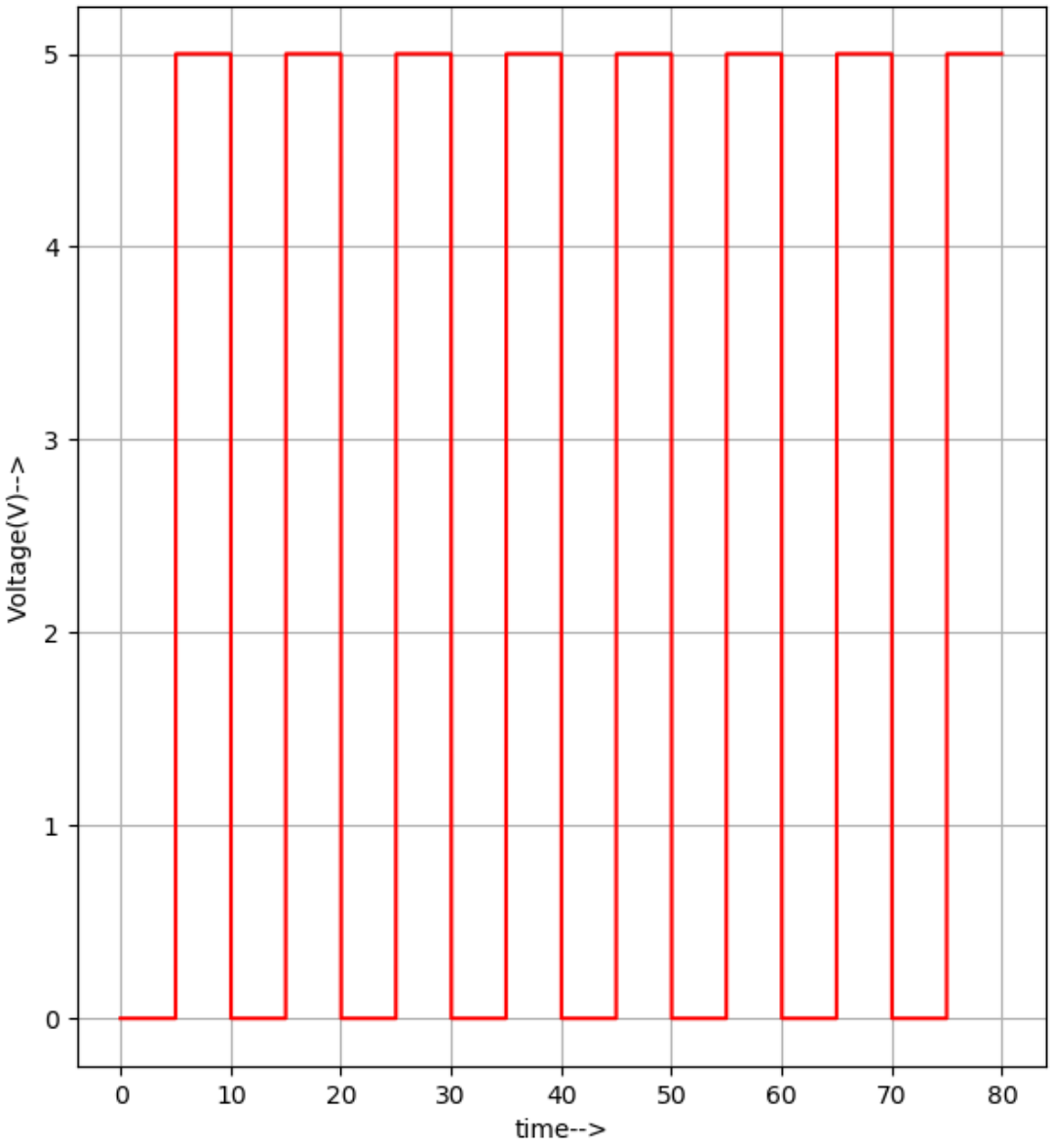

tran1: * e:\fosse\esim-workspace\binary_to_bcd_code_converter_using_subcircuit\binary_to_bcd_co...
V
v(d3)
5.0
4.0
3.0
2.0
1.0
0.0
0.0
10.0
20.0
30.0
40.0
50.0
60.0
70.0
80.0
time
s

D4: -

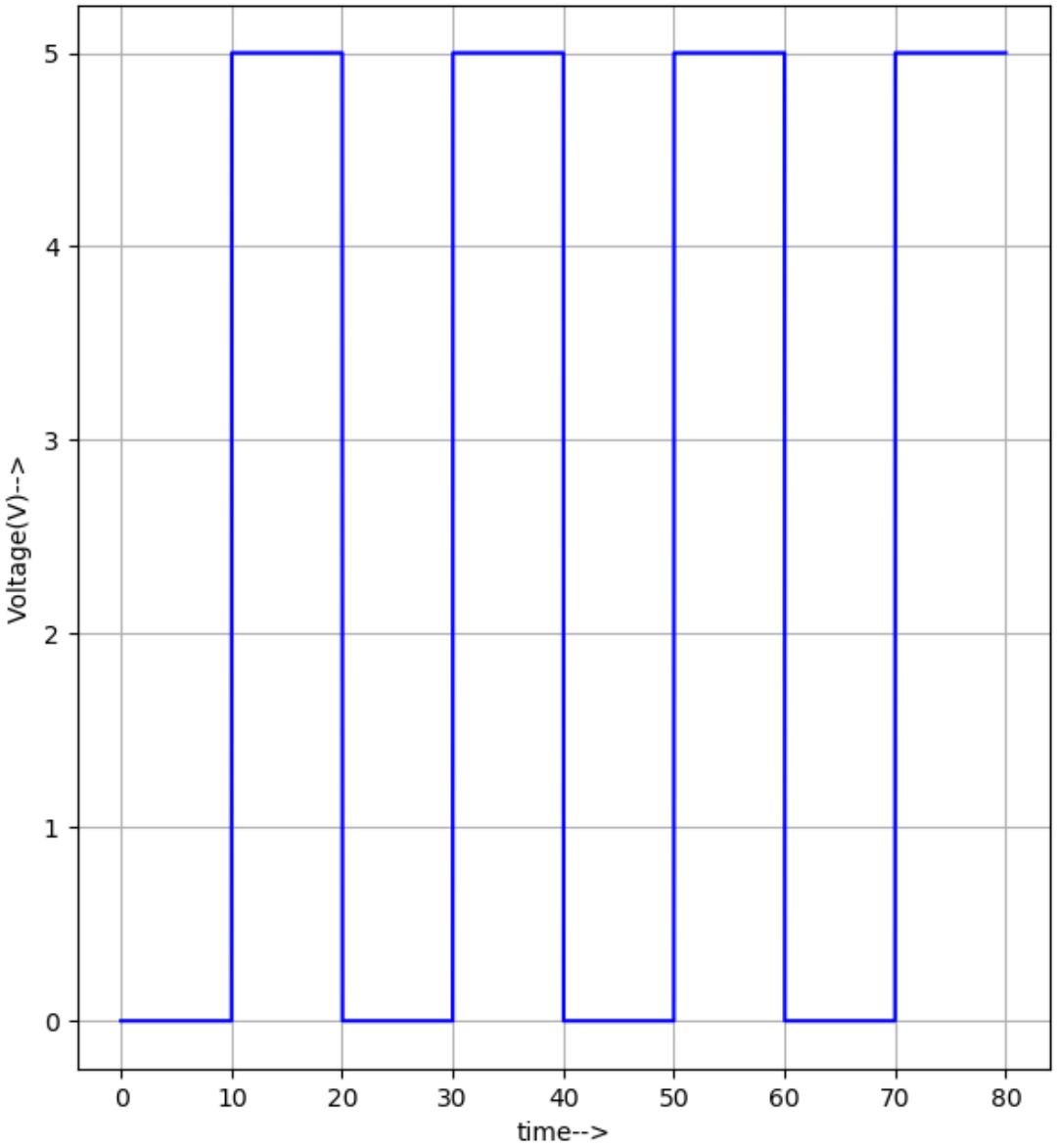

tran1: * e:\fosse\esim-workspace\binary_to_bcd_code_converter_using_subcircuit\binary_to_bcd_co...
V
v(d4)
5.0
4.0
3.0
2.0
1.0
0.0
0.0
10.0
20.0
30.0
40.0
50.0
60.0
70.0
80.0
time
s

- **Python Plots: -**
  - Inputs: -

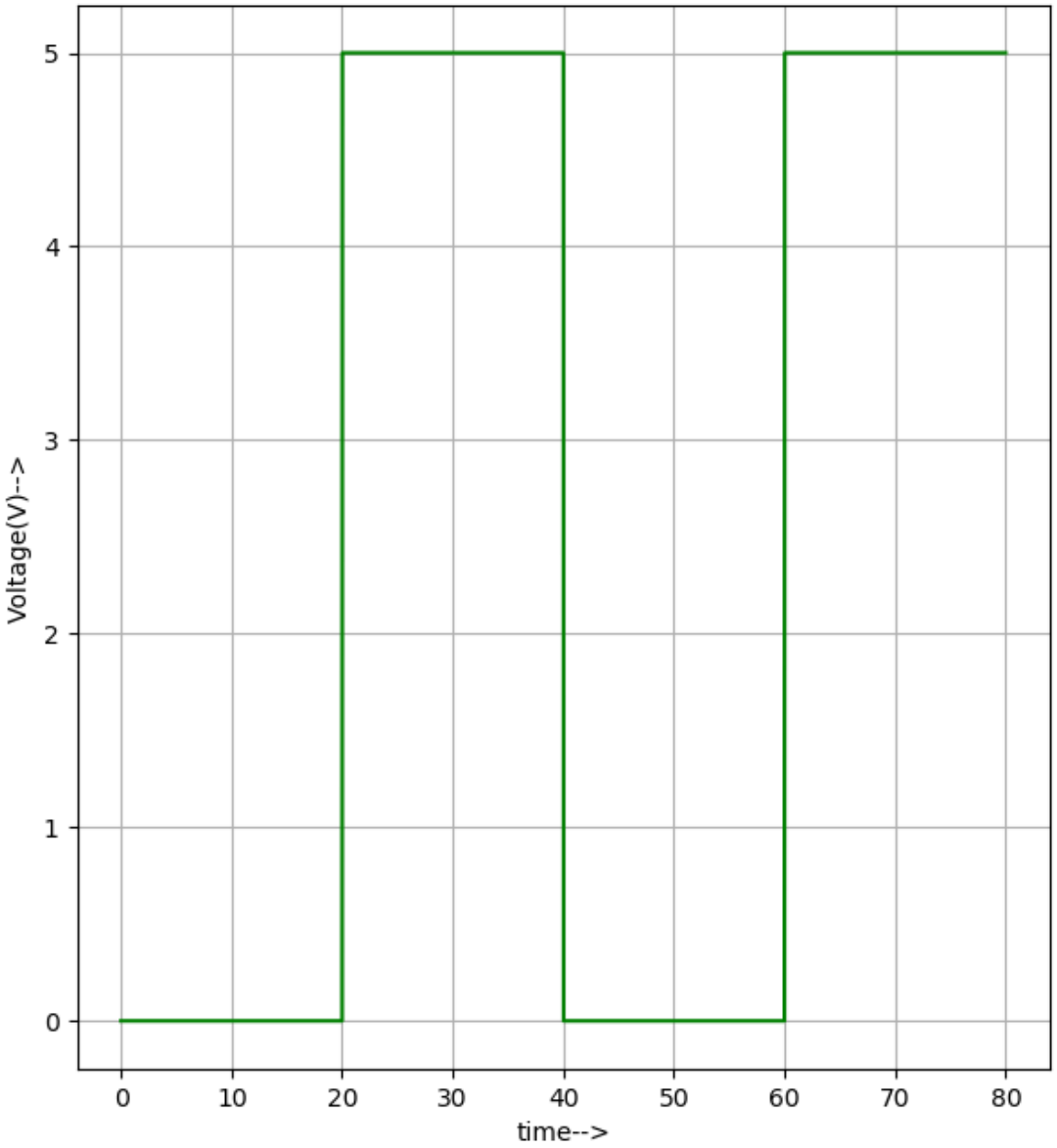
B0: -



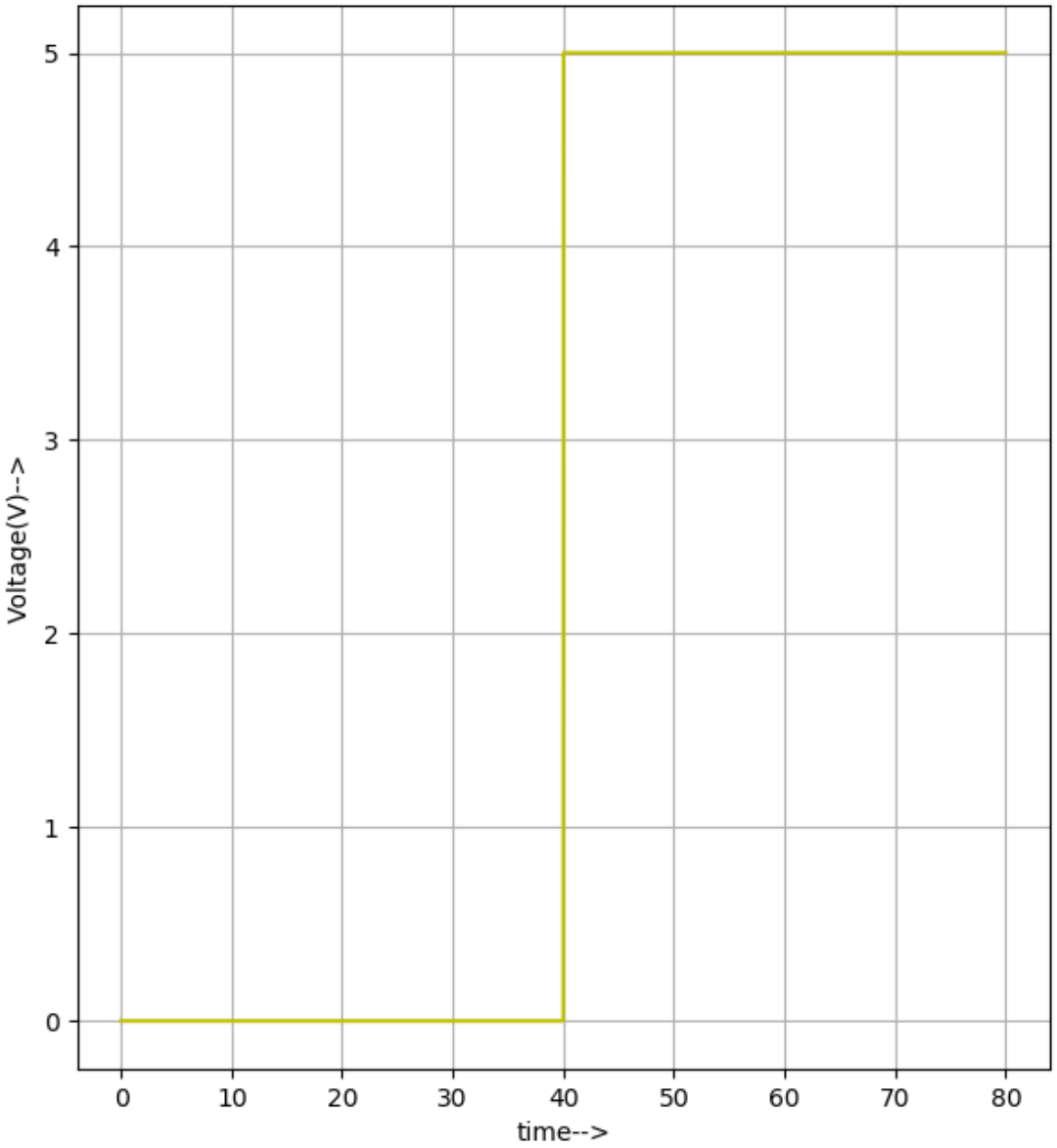
B1: -



B2: -

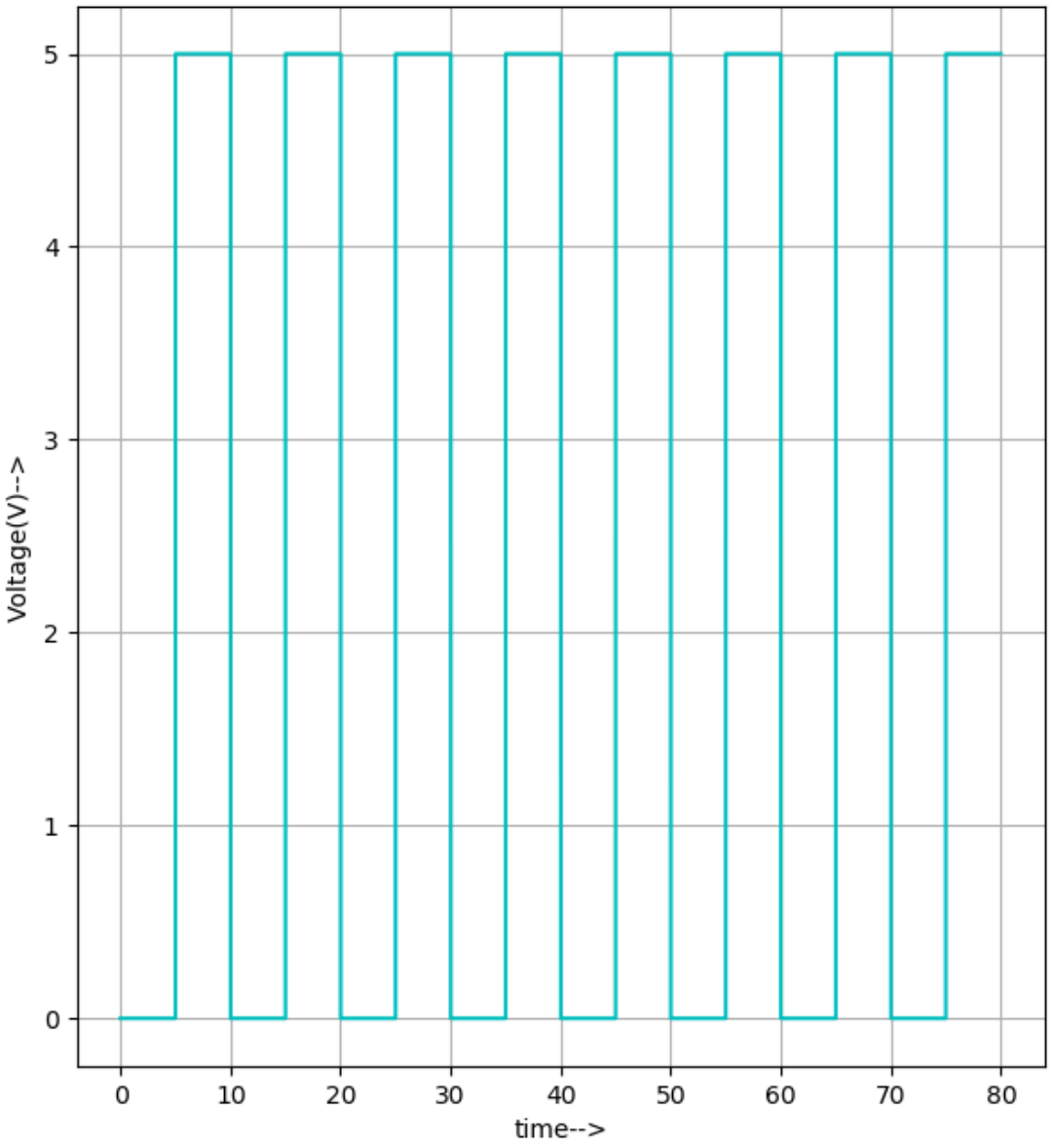


B3: -

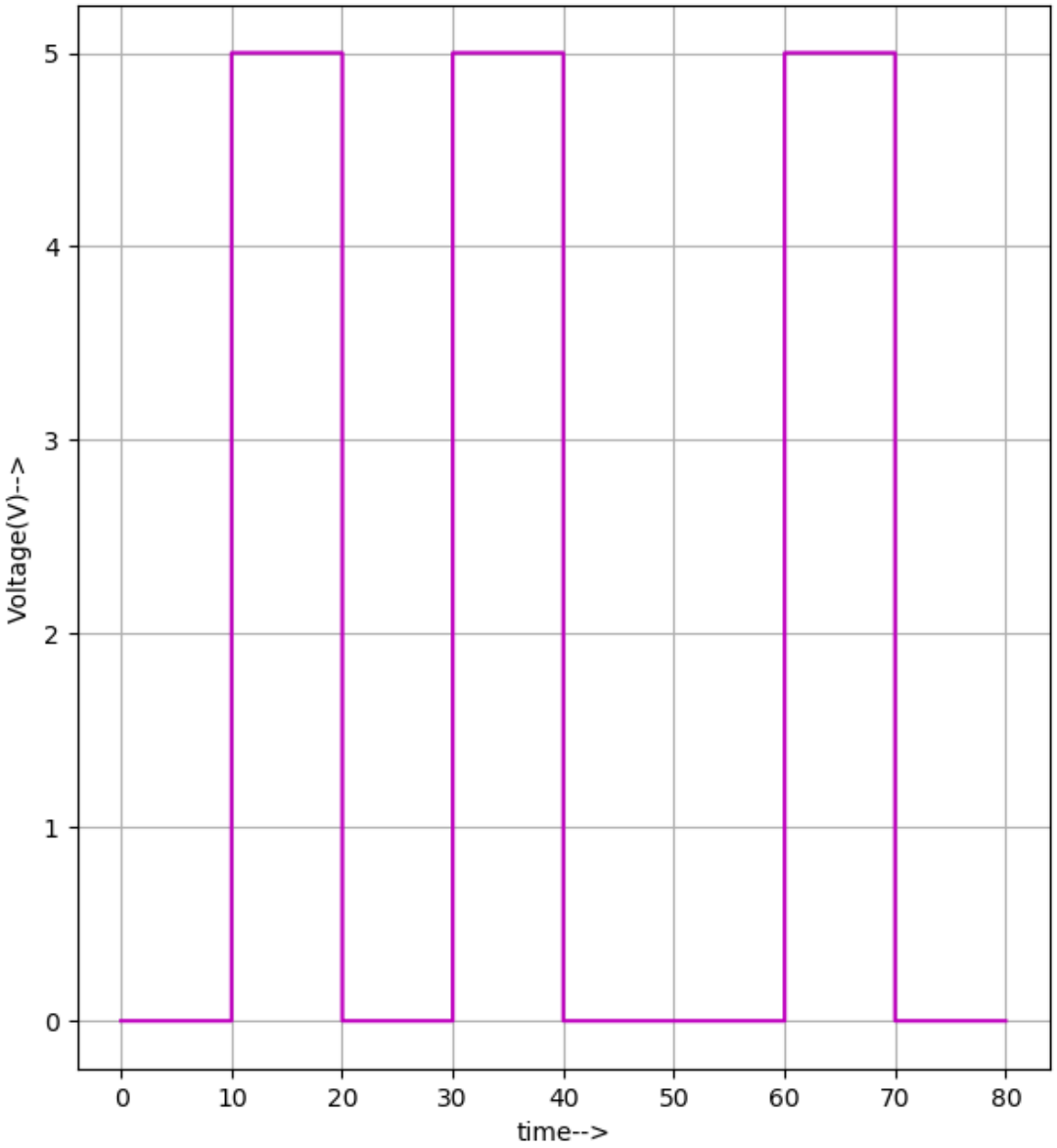


- Outputs: -

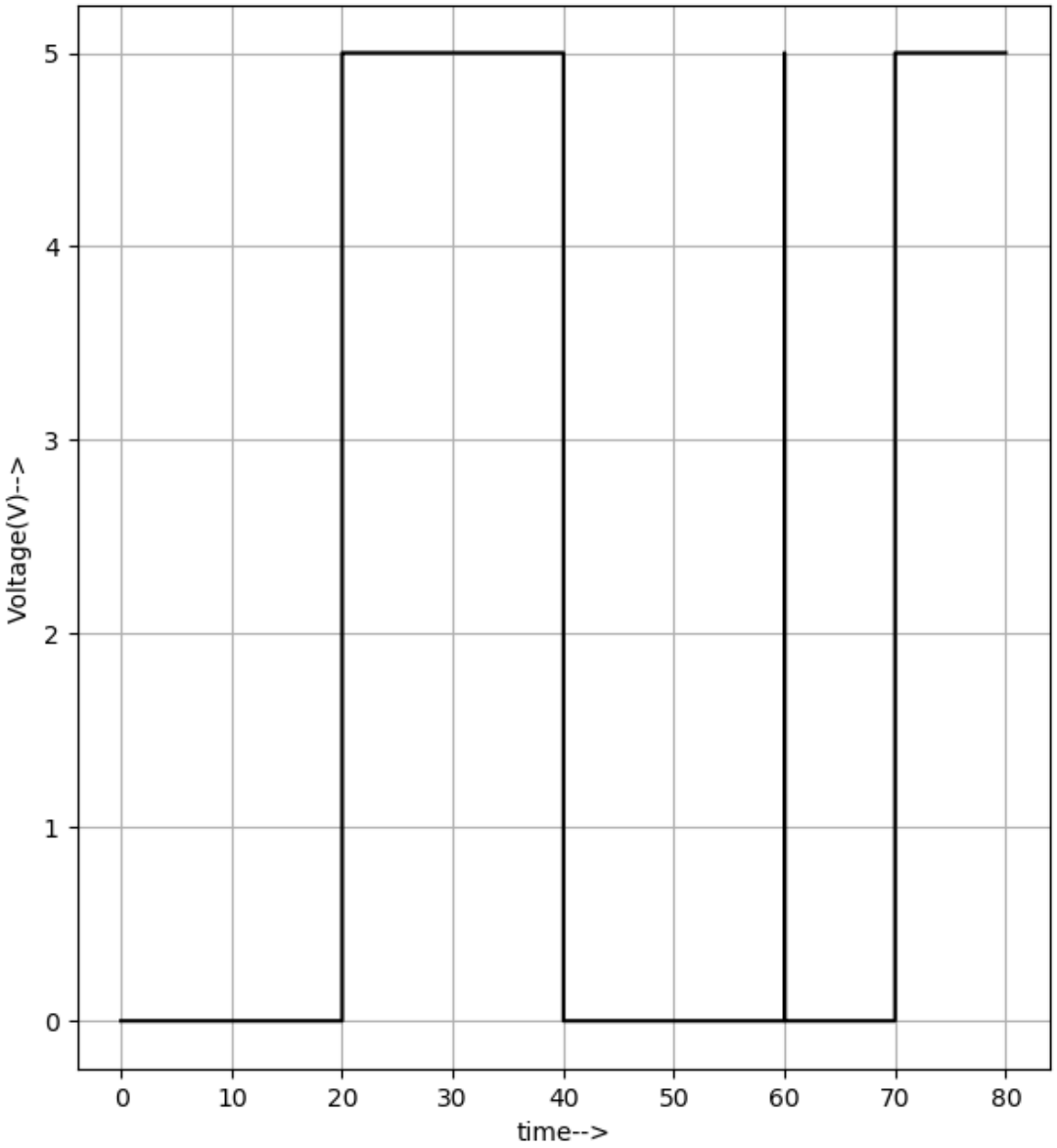
D0: -



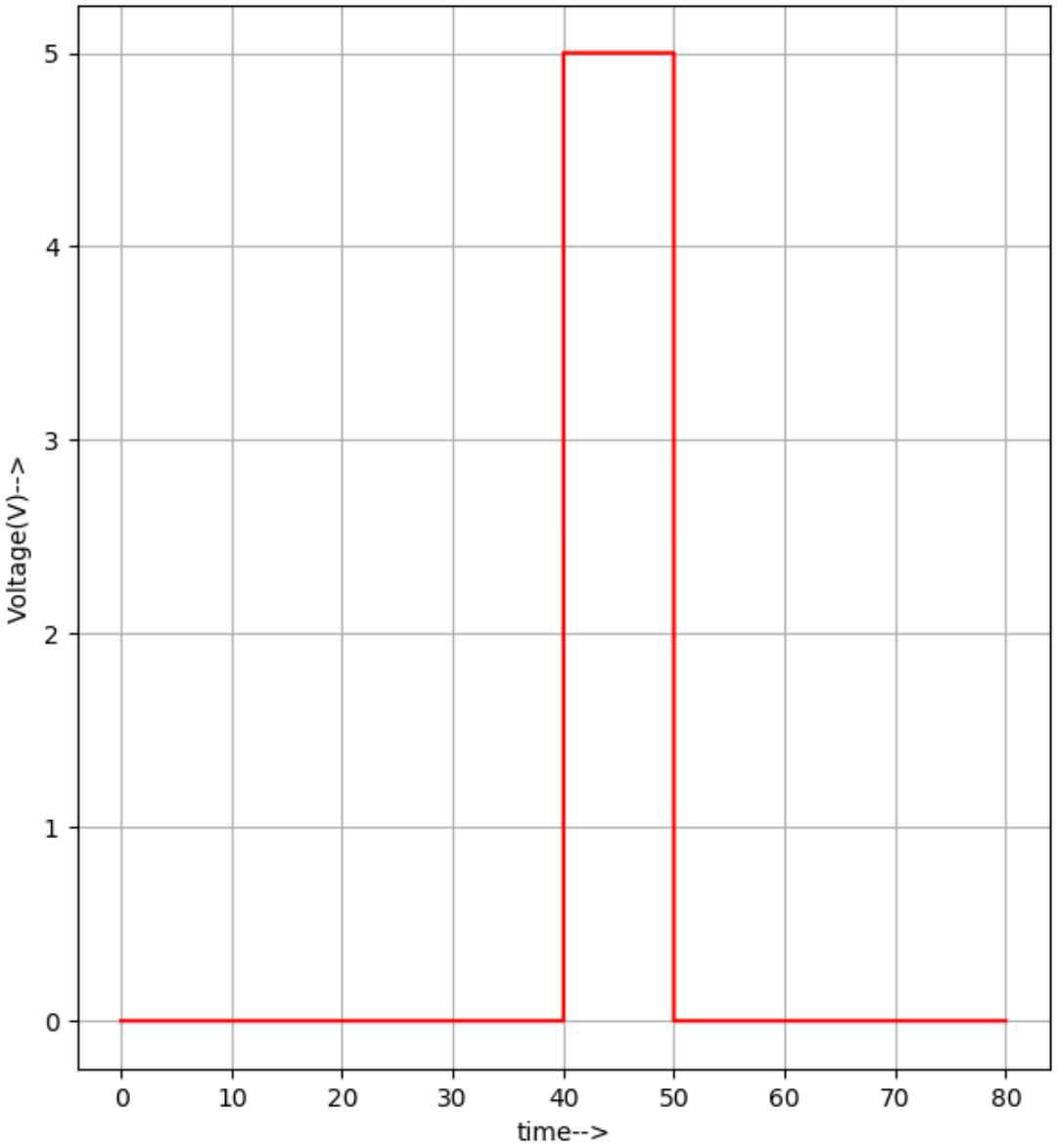
D1: -



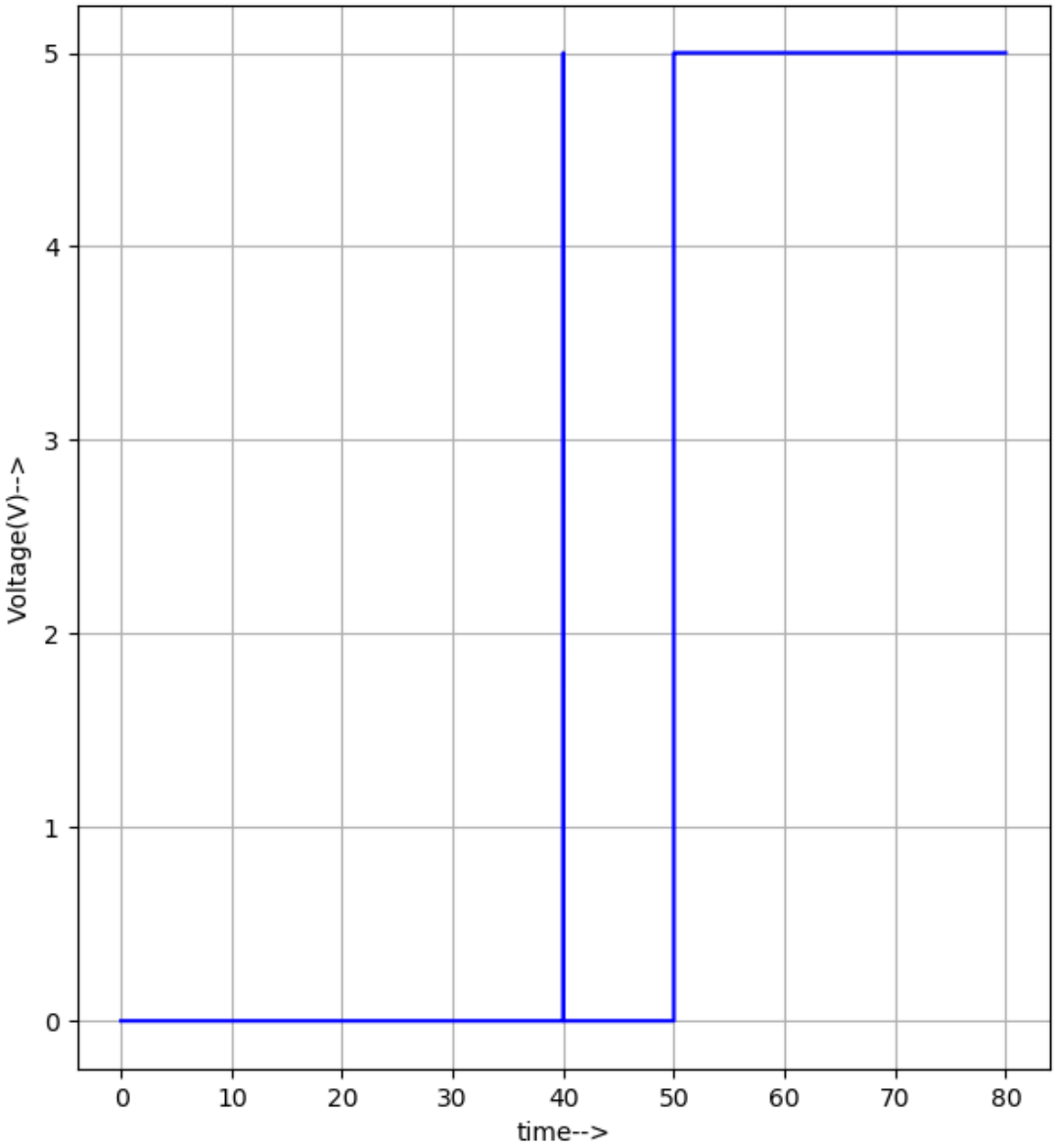
D2: -



D3: -



D4: -



References: -

- https://www.electrically4u.com/code-converter-types-truth-table-and-logic-circuits/