Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

Circuit Simulation Project

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

Name of the Participant - Ms. Sai Samyuktha N

Project Guide - Dr. Maheswari.R

Title of the Project - Design of a 4-bit Gray to Binary code converter circuit with Main circuit and Subciruit implementation using eSim

Theory

- ***Binary*** - Binary code is based on a binary number system in which there are only two possible states, off and on, usually symbolized by 0 and 1.

- ***Gray code*** - Gray code is an ordering of the binary numeral system such that two successive values differ in only one bit.

- The following table is a comparison of Decimal, Gray code and Binary:

Decimal	Gray Code	Binary
0	0000	0000
1	0001	0001
2	0011	0010
3	0010	0011
4	0110	0100
5	0111	0101
6	0101	0110
7	0100	0111
8	1100	1000
9	1101	1001
10	1111	1010
11	1110	1011
12	1010	1100
13	1011	1101
14	1001	1110
15	1000	1111

Image source : https://www.dynapar.com/hs-fs/hubfs/uploadedImages/_Site_Root/Gray-Code-Encoder-Output.jpg?width=219&height=319&name=Gray-Code-Encoder-Output.jpg

- ***Gray to Binary code conversion:***

The truth table of Gray to Binary code conversion is:

Gray code number is the input and the corresponding Binary form is the Output. Decimal number is taken for reference (in the table)

Decimal Number rep.	**INPUT**				**OUTPUT**			
	A	**B**	**C**	**D**	**W**	**X**	**Y**	**Z**
0	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
1	**0**	**0**	**0**	**1**	**0**	**0**	**0**	**1**
2	**0**	**0**	**1**	**0**	**0**	**0**	**1**	**1**
3	**0**	**0**	**1**	**1**	**0**	**0**	**1**	**0**
4	**0**	**1**	**0**	**0**	**0**	**1**	**1**	**1**
5	**0**	**1**	**0**	**1**	**0**	**1**	**1**	**0**
6	**0**	**1**	**1**	**0**	**0**	**1**	**0**	**0**
7	**0**	**1**	**1**	**1**	**0**	**1**	**0**	**1**
8	**1**	**0**	**0**	**0**	**1**	**1**	**1**	**1**
9	**1**	**0**	**0**	**1**	**1**	**1**	**1**	**0**
10	**1**	**0**	**1**	**0**	**1**	**1**	**0**	**0**
11	**1**	**0**	**1**	**1**	**1**	**1**	**0**	**1**
12	**1**	**1**	**0**	**0**	**1**	**0**	**0**	**0**
13	**1**	**1**	**0**	**1**	**1**	**0**	**0**	**1**
14	**1**	**1**	**1**	**0**	**1**	**0**	**1**	**1**
15	**1**	**1**	**1**	**1**	**1**	**0**	**1**	**0**

Truth Table reduction using K-Map:

1) W

AB \ CD	00	01	11	10
00	0	0	0	0
01	0	0	0	0
11	**1**	**1**	**1**	**1**
10	**1**	**1**	**1**	**1**

Hence, W = A

2) X

AB \ CD	00	01	11	10
00	0	0	0	0
01	1	1	1	1
11	0	0	0	0
10	1	1	1	1

$$X = (A'.B) + (A.B')$$

Hence, $X = A \oplus B$

3) Y

CD / AB	00	01	11	10
00	0	0	1	1
01	1	1	0	0
11	0	0	1	1
10	1	1	0	0

Y = (A′.B′.C) + (A′.B.C′) + (A.B.C) + (A.B′.C′)

Hence, on simplification Y = X ⊕ C

4) Z

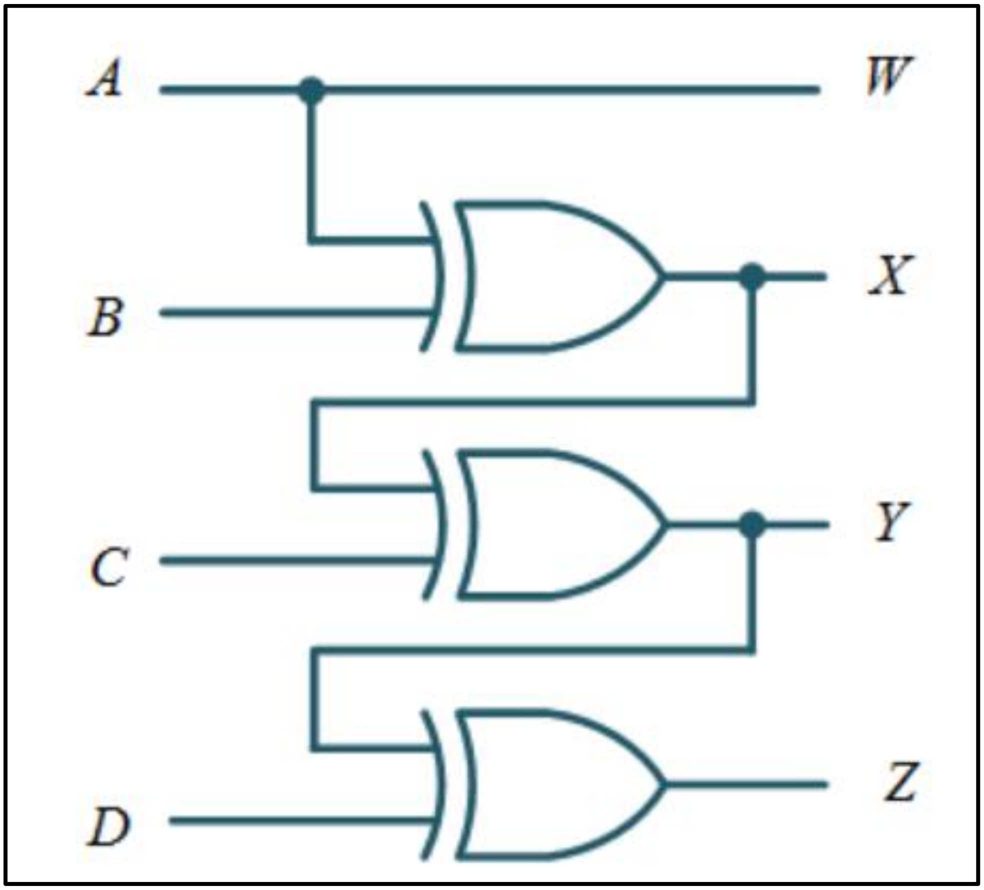
CD / AB	00	01	11	10
00	0	1	0	1
01	1	0	1	0
11	0	1	0	1
10	1	0	1	0

Z = (A′.B′.C′.D) + (A′.B′.C.D′) + (A′.B.C′.D′) + (A′.B.C.D) + (A.B.C′.D) + (A.B.C.D′) + (A.B′.C′.D′) + (A.B′.C.D)

Hence, on simplification, Z = Y ⊕ D

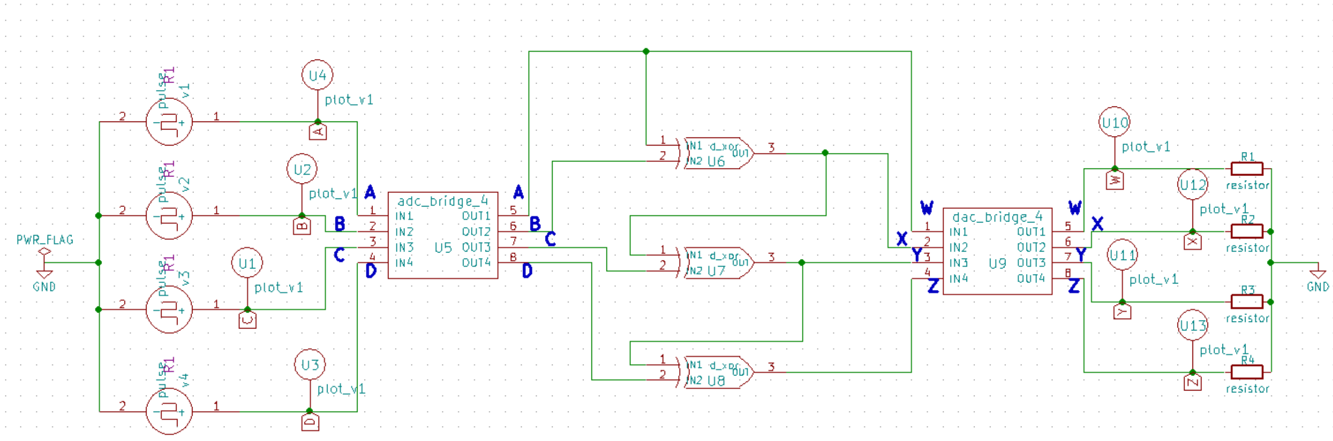
- ***Circuit Diagram:***

The circuit can be implemented using three x-or gates



eSim Implementation

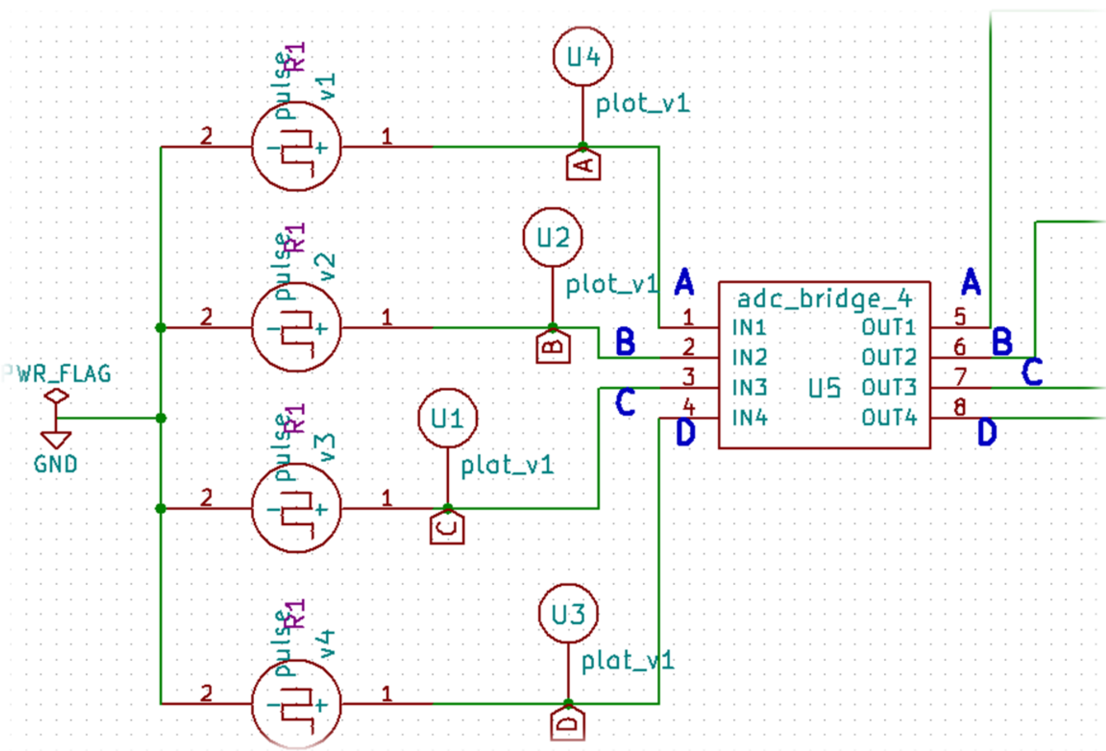
I. Main circuit Implementation



The main circuit has three parts:

1. **Input**

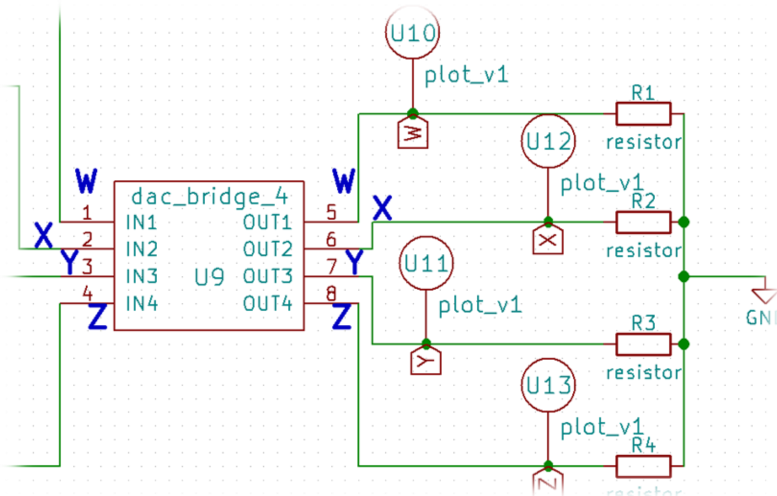
The 4-bit Gray code input is of the form :- **A B C D**



We make use of the analog to digital converter to convert the input analog pulses into digital as we make use of logic gates (that work only on digital signals)

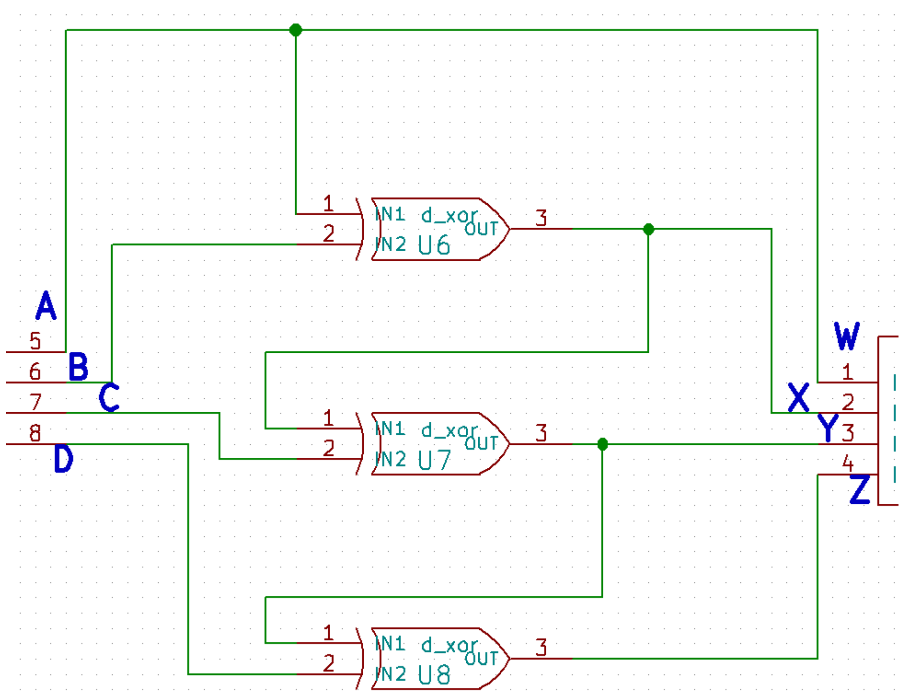
2. Output

The 4-bit Binary output is of the form :- **W X Y Z**



We make use of the digital to analog converter to convert the signals back into analog and compute the output

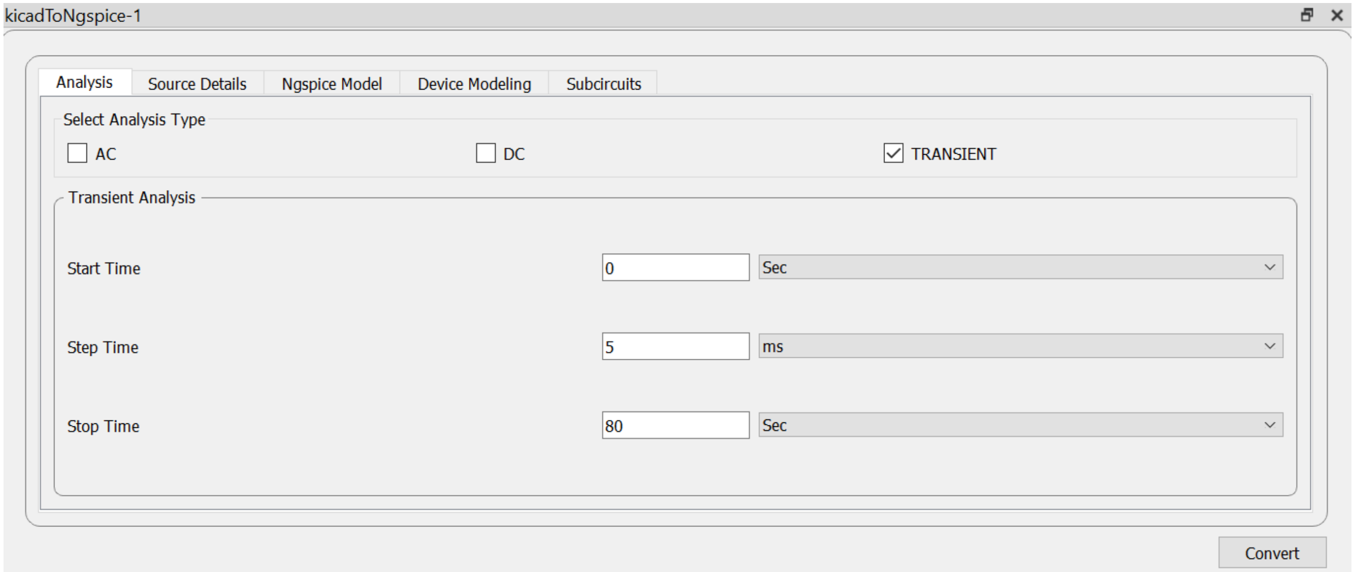
3. Logic Circuit



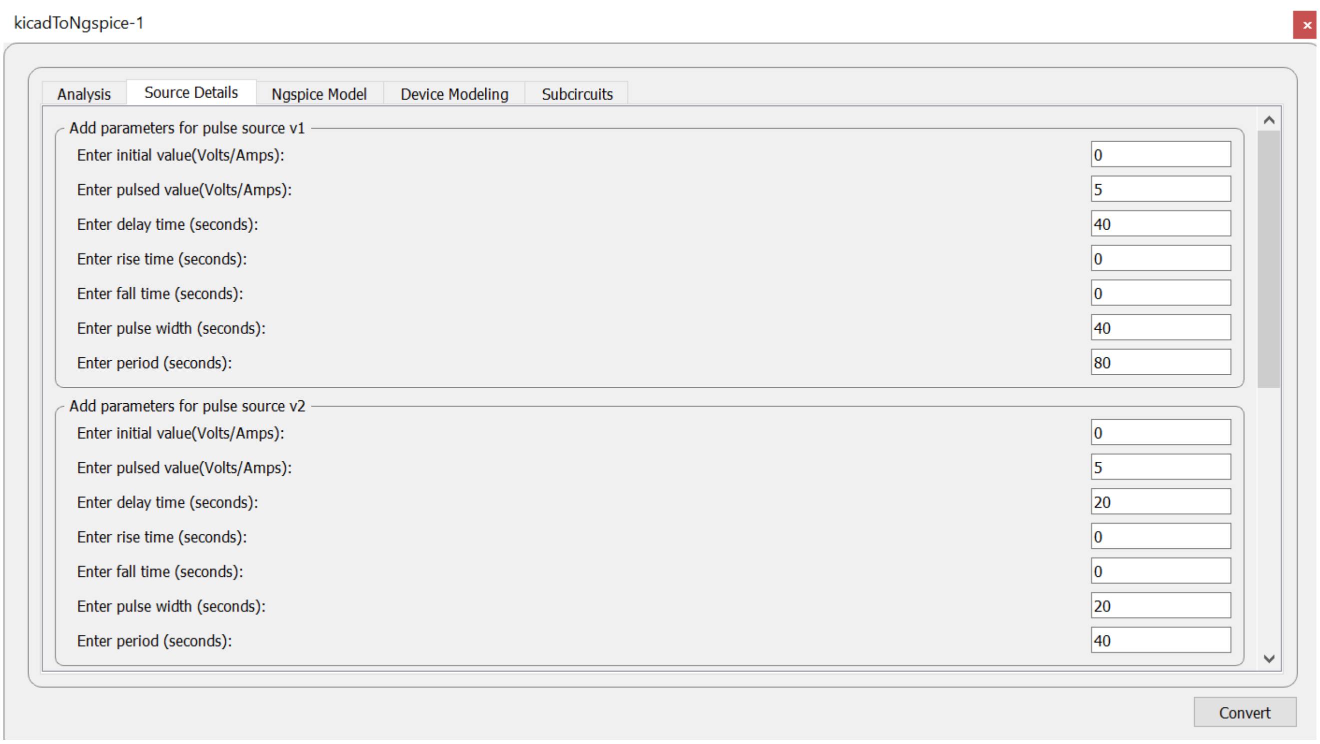
The circuit has been implemented from the previously derived logic circuit diagram.

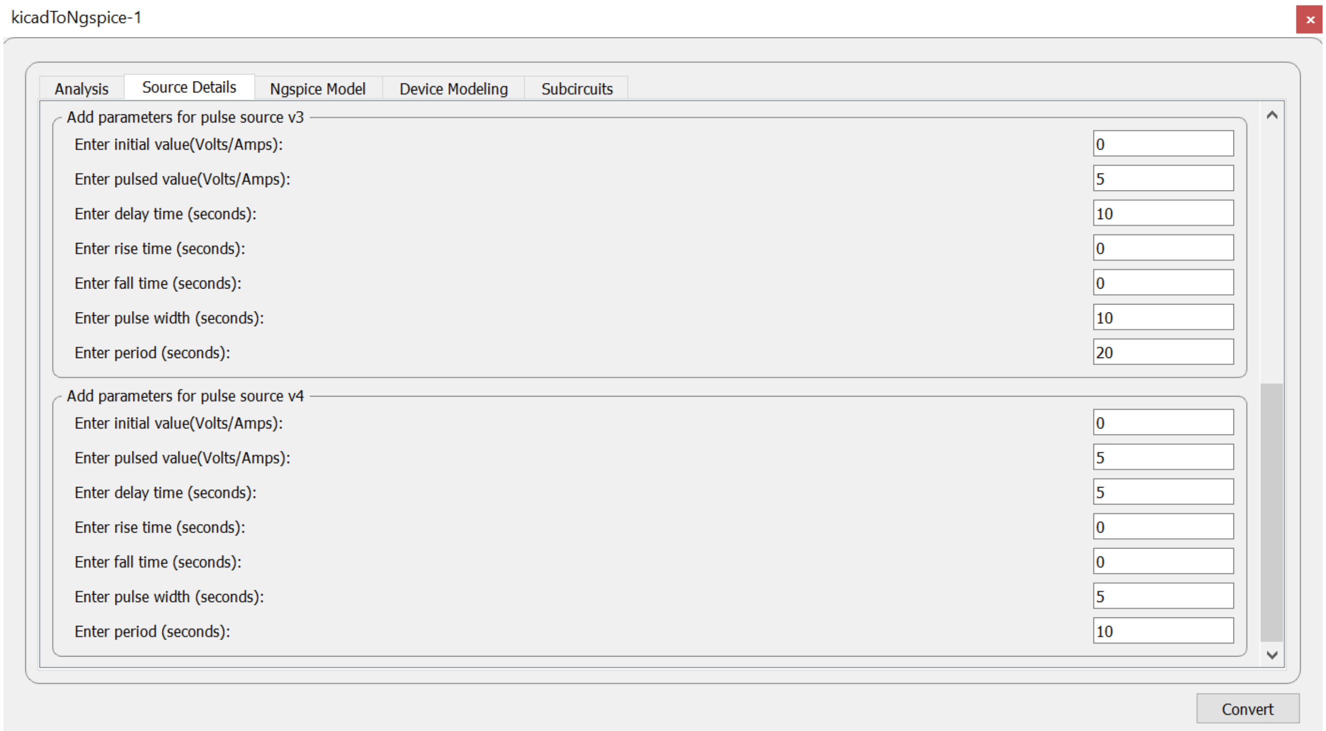
Kicad to Ngspice Conversion

Here we make use of transient analysis:



Source Details:



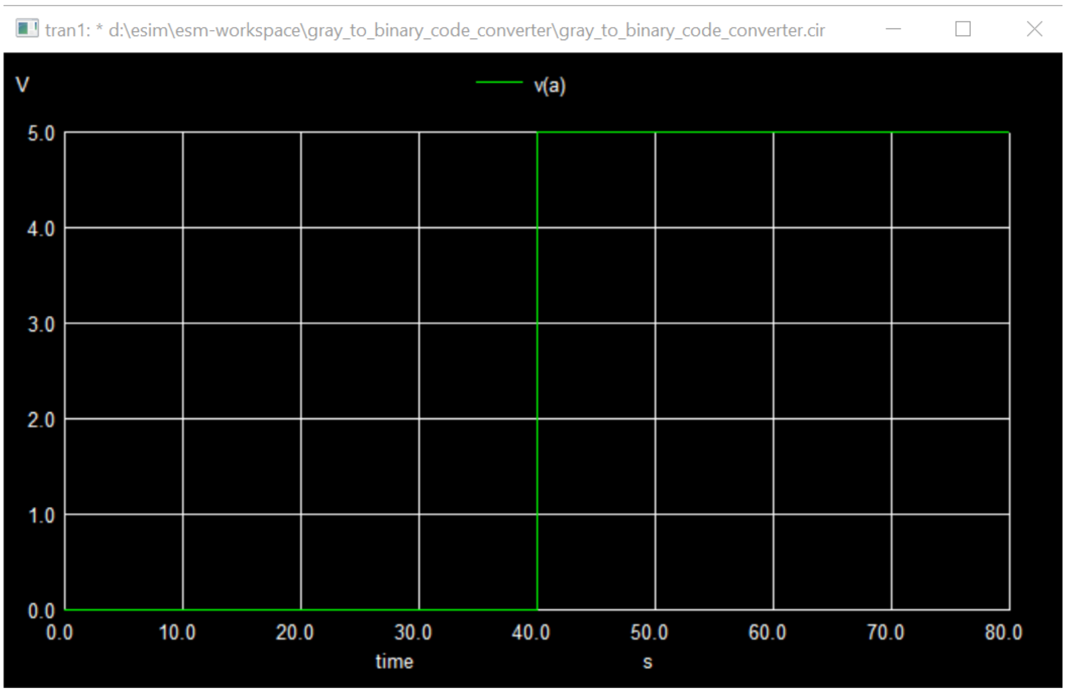


Other fields are left as default.

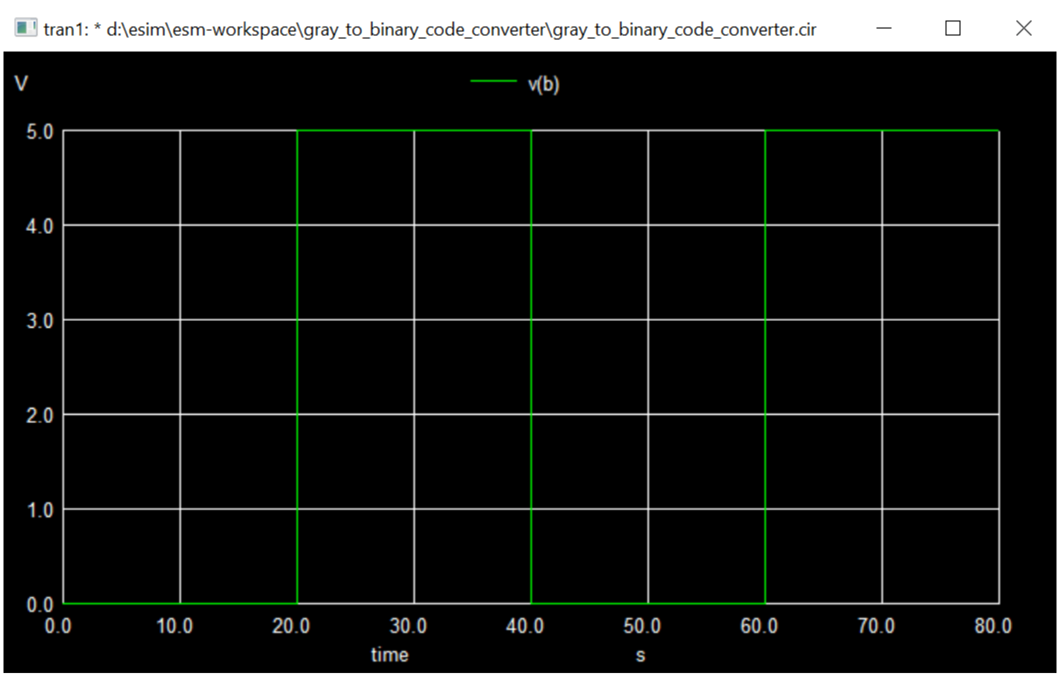
Circuit simulation Output

I. NGSPICE PLOTS:

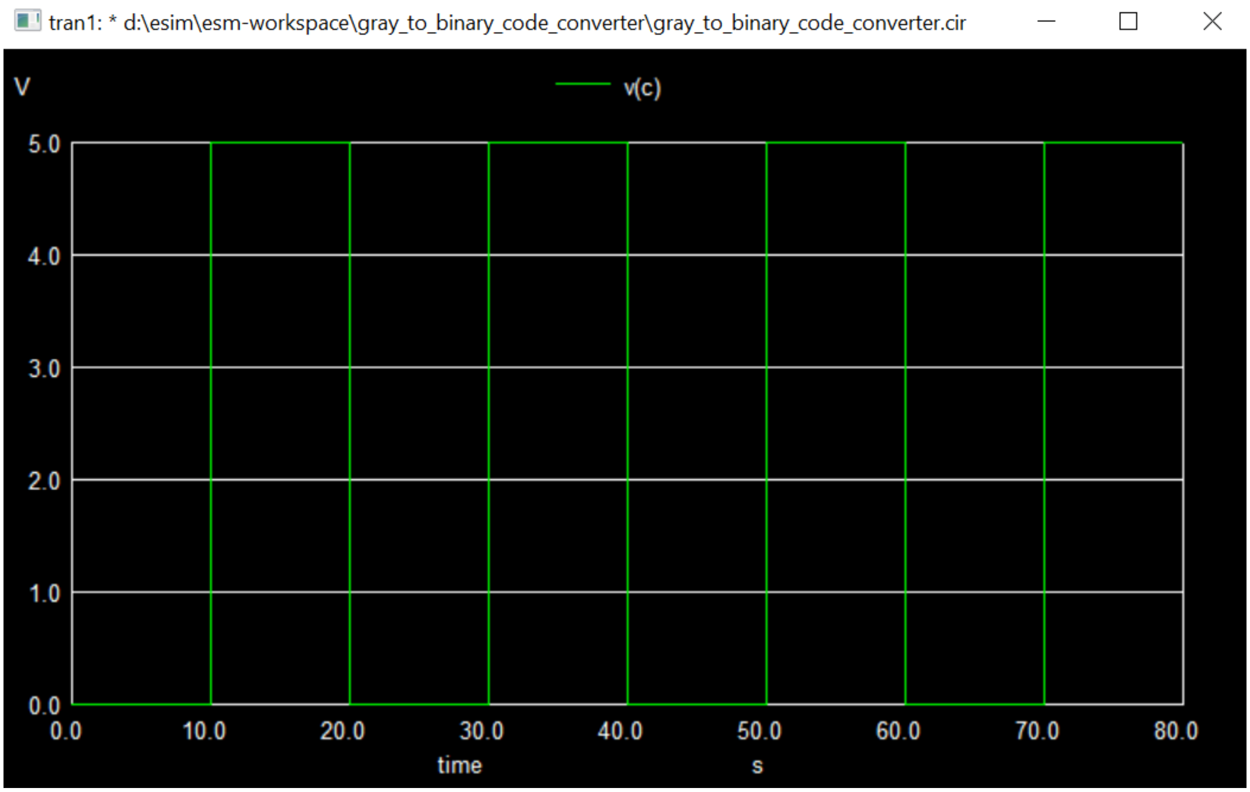
- Inputs:



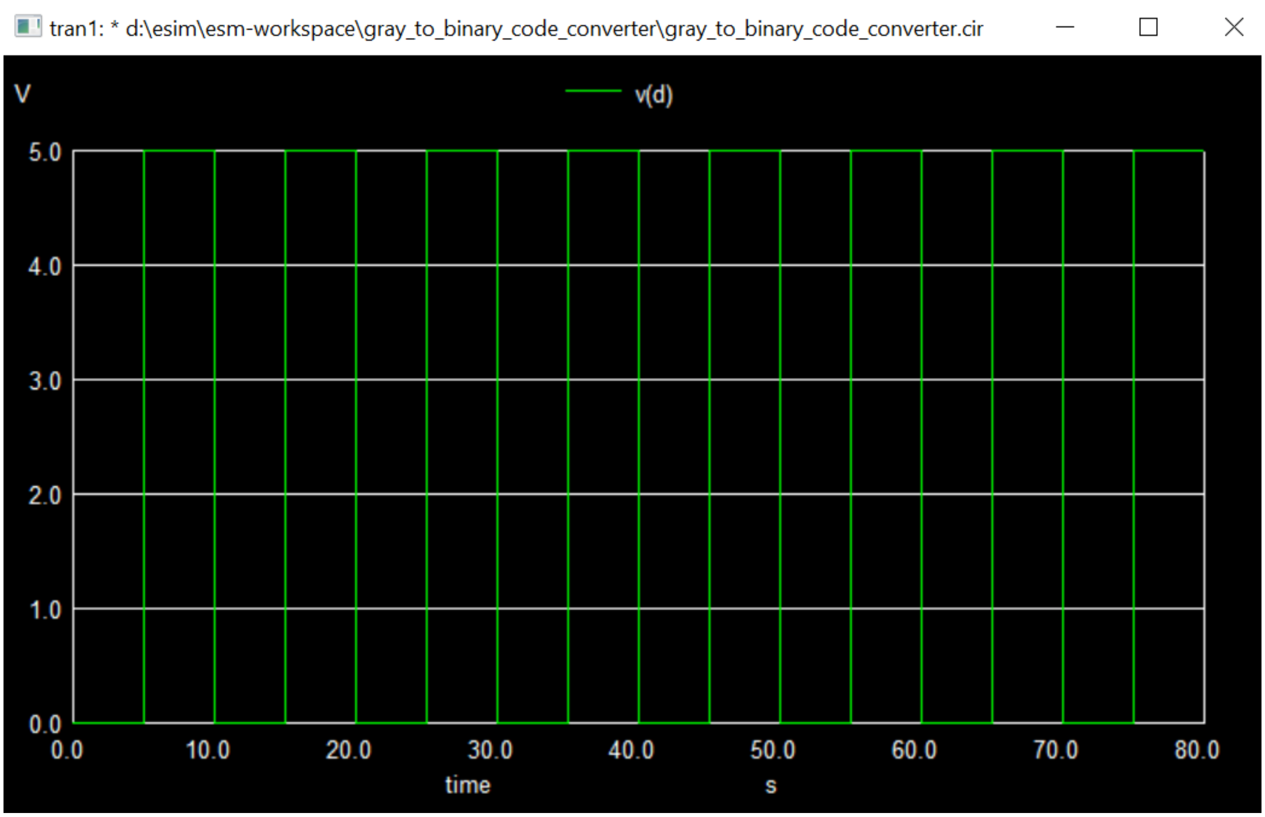
Ngspice plot of A



Ngspice plot of B

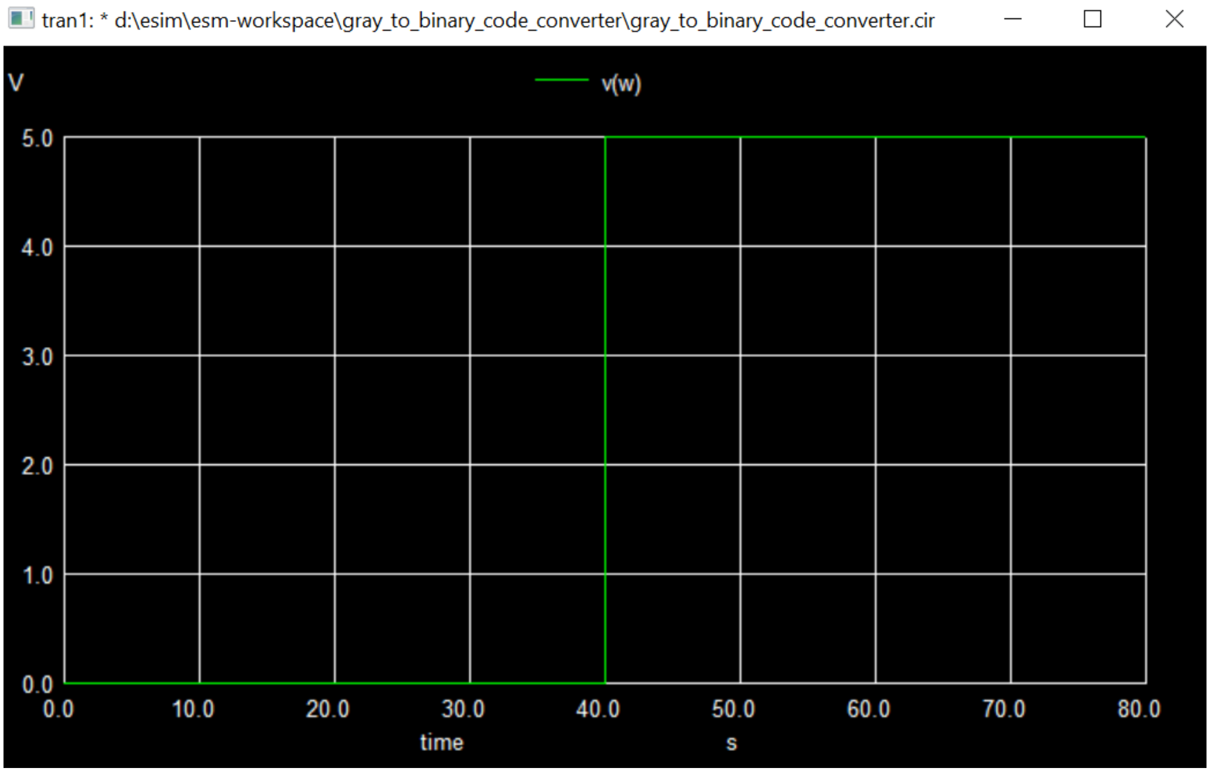


Ngspice plot of C

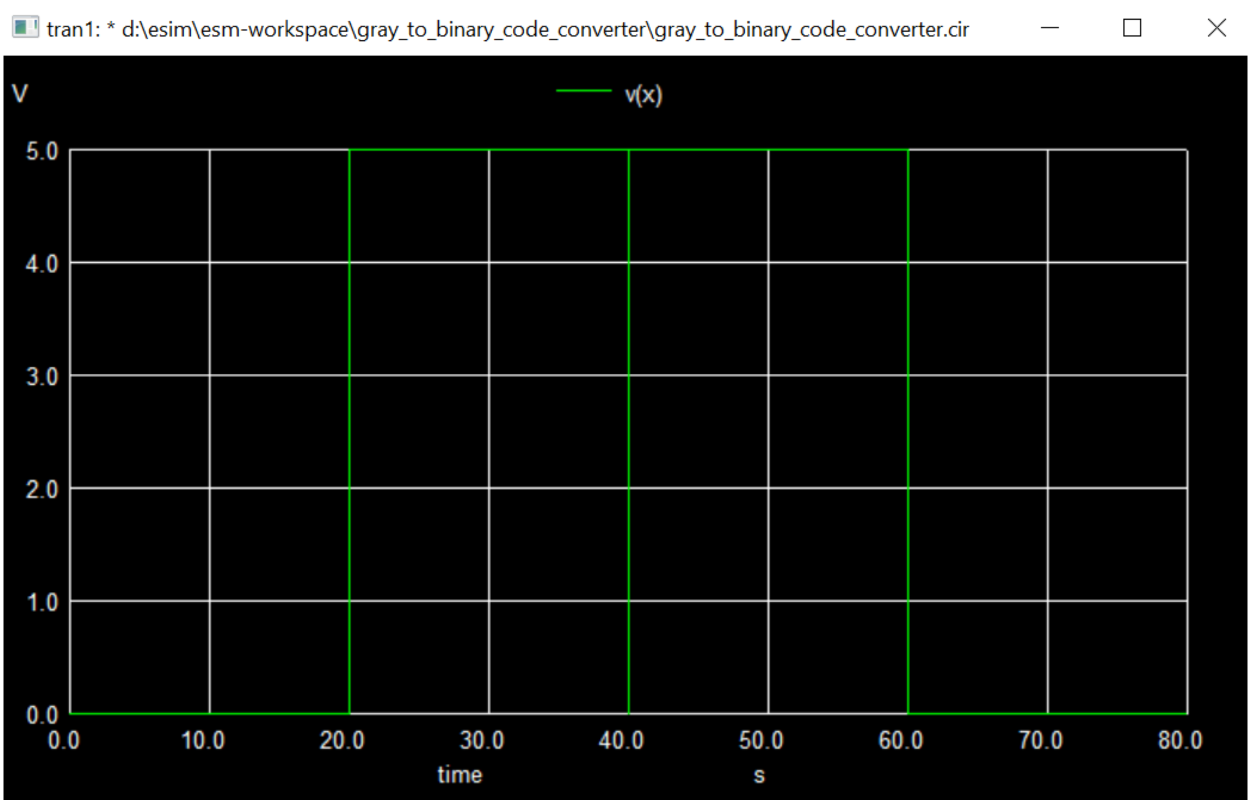


Ngspice plot of D

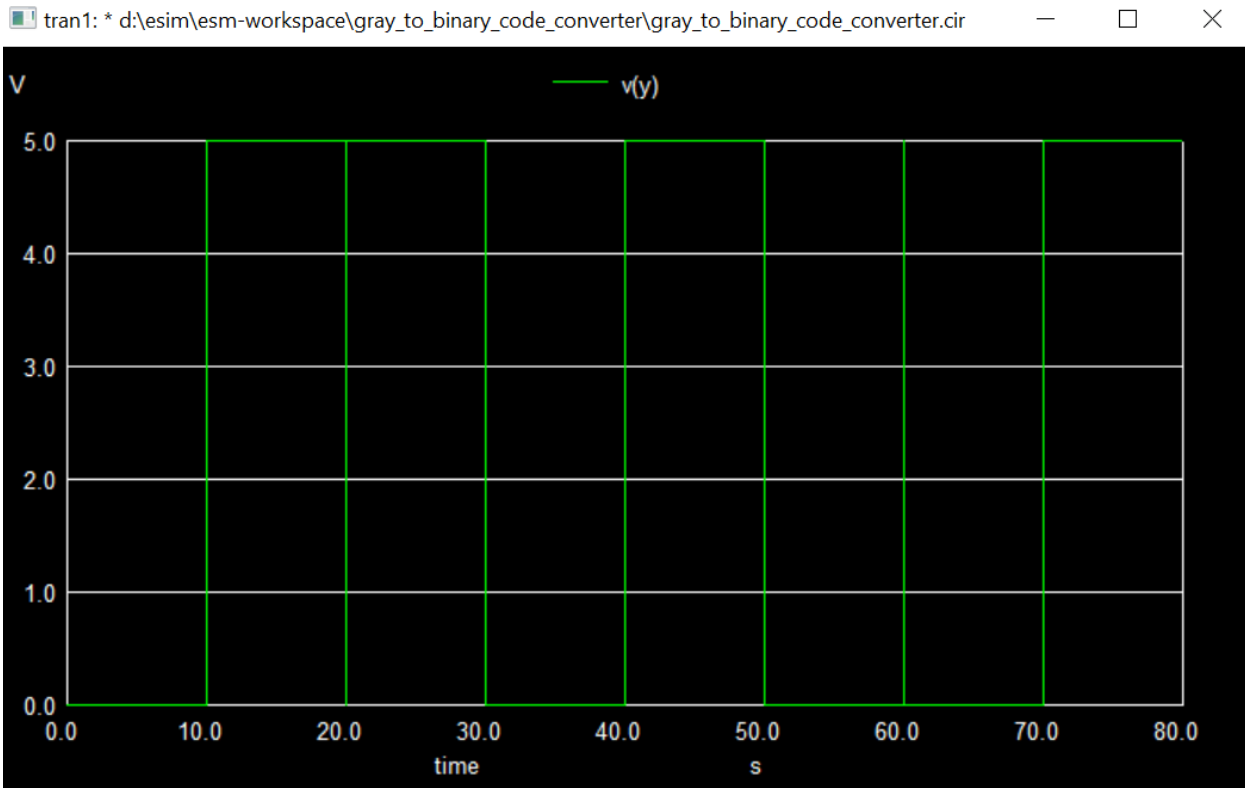
- Outputs:



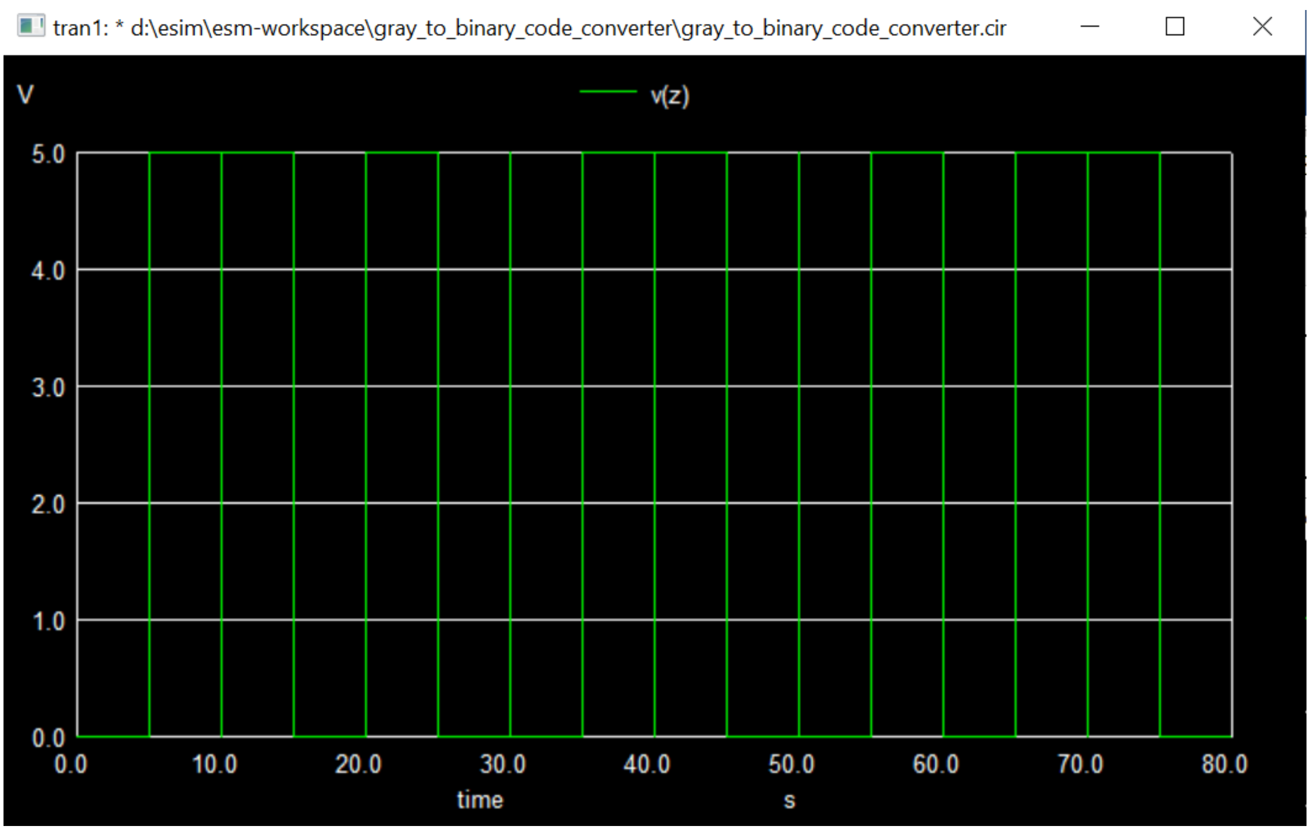
Ngspice plot of W



Ngspice plot of X

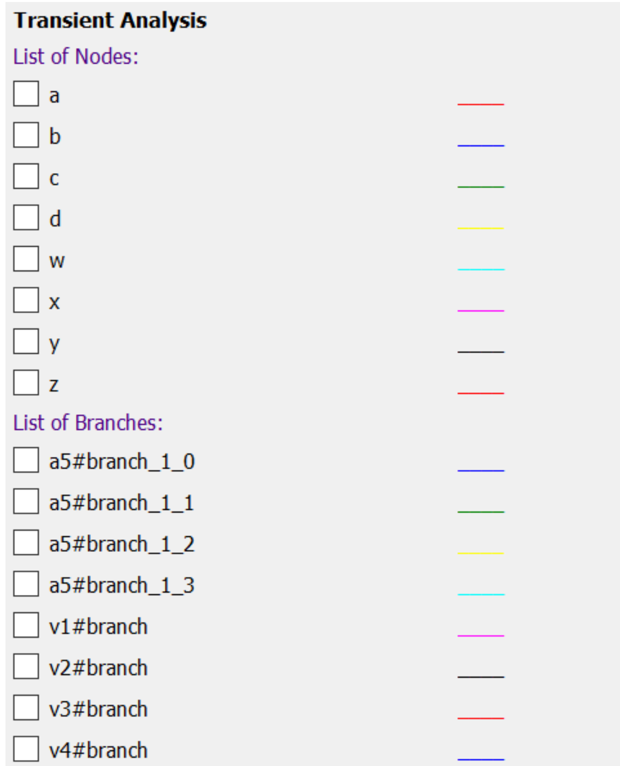


Ngspice plot of Y

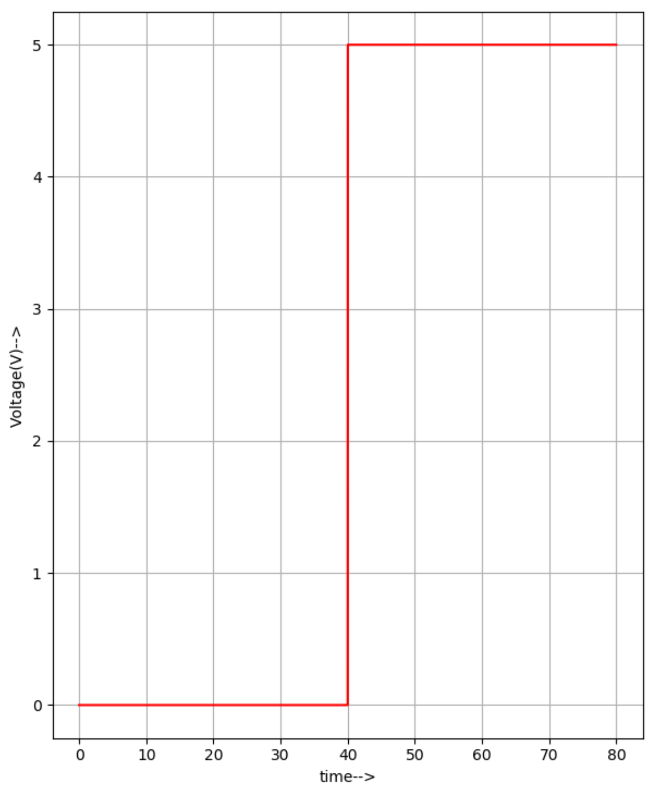


Ngspice plot of Z

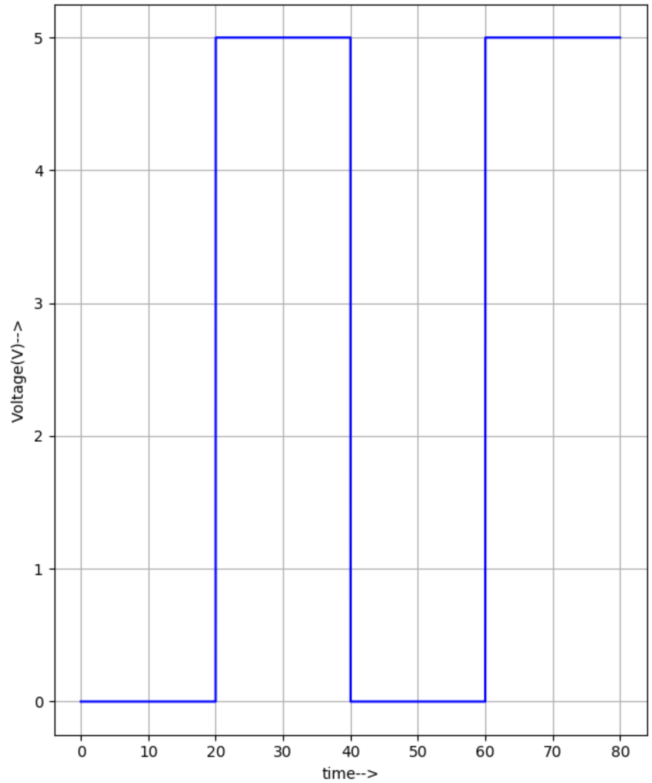
II. PYTHON PLOTS:



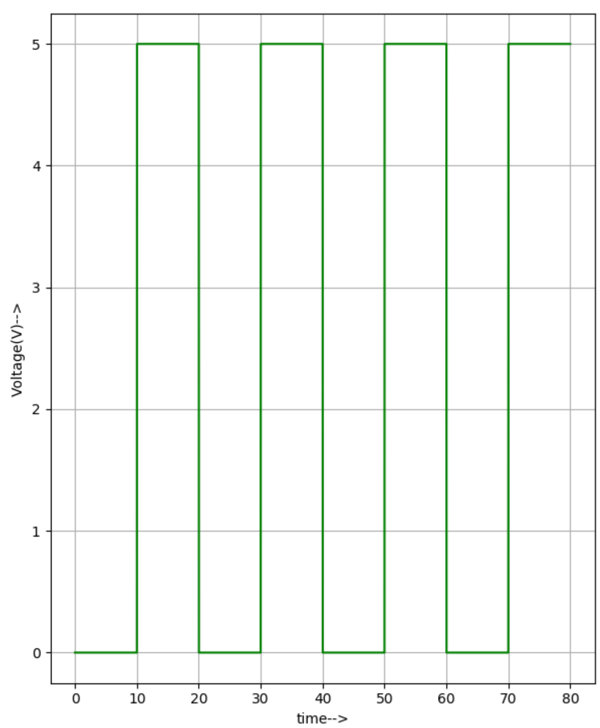
- Inputs:



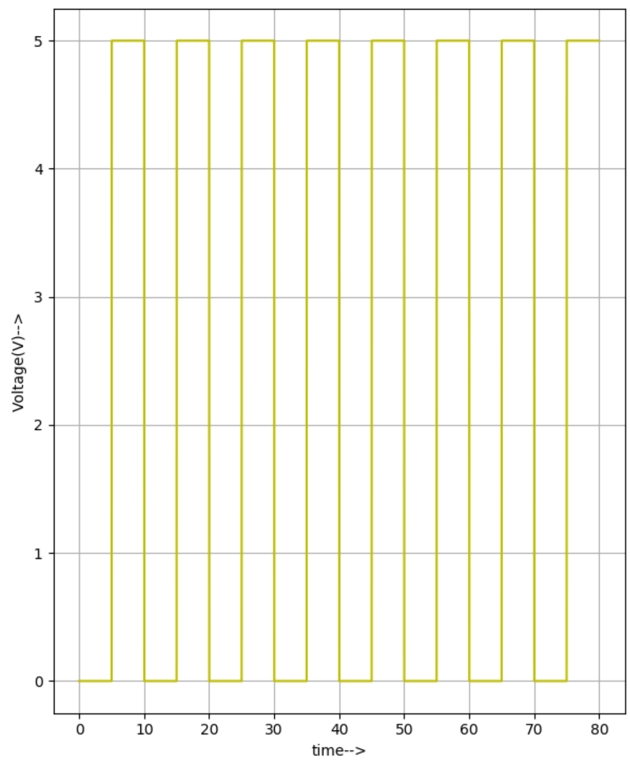
Python plot of A



Python plot of B

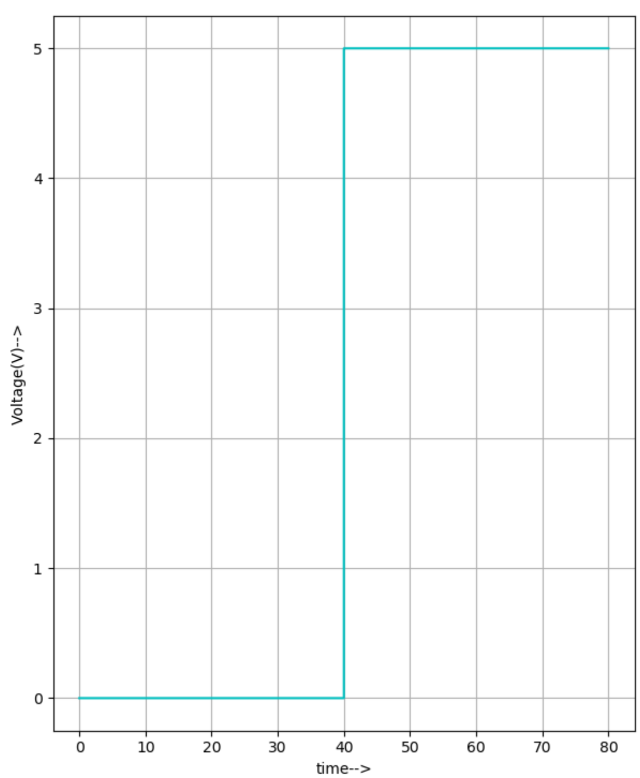


Python plot of C

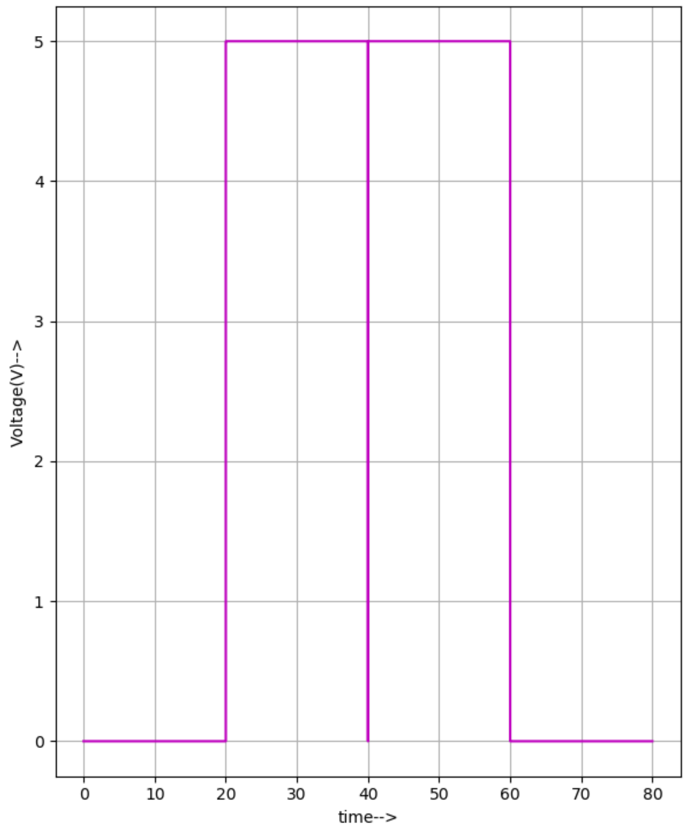


Python plot of D

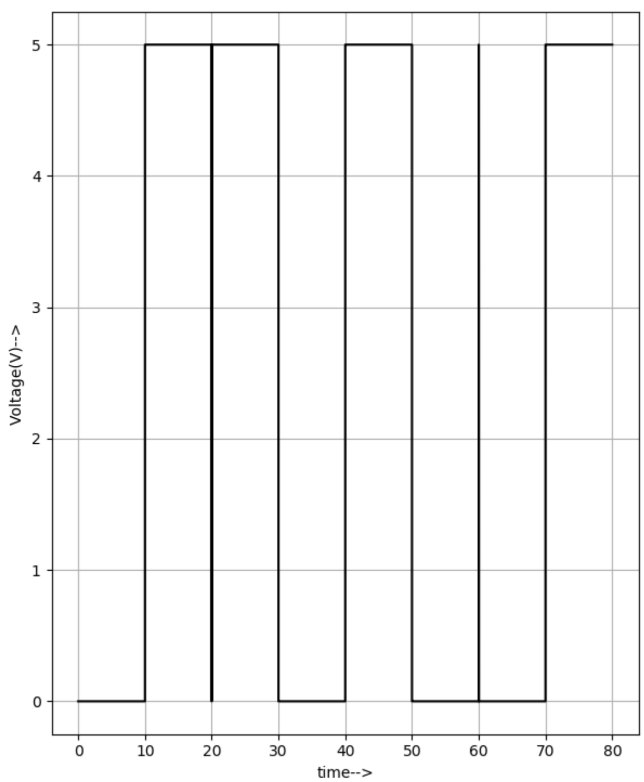
- Outputs:



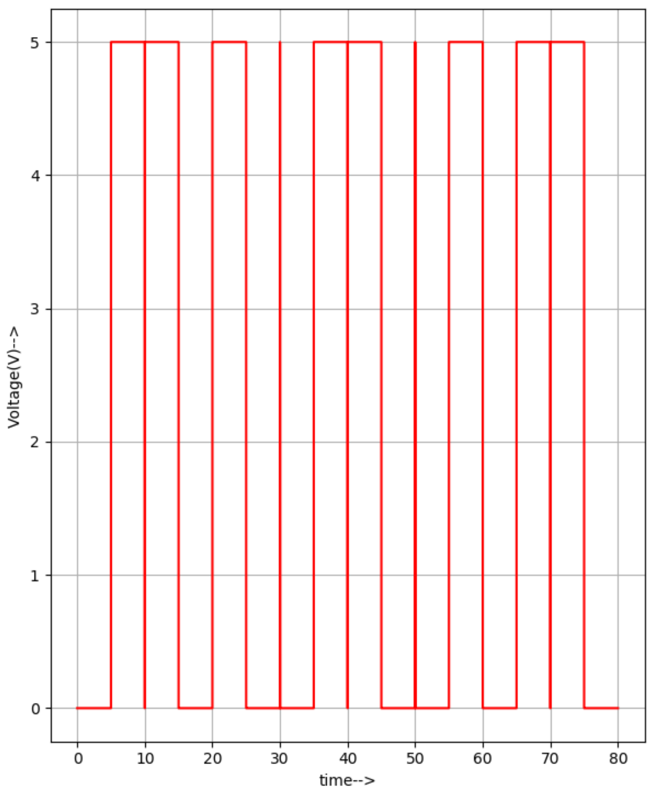
Python plot of W



Python plot of X



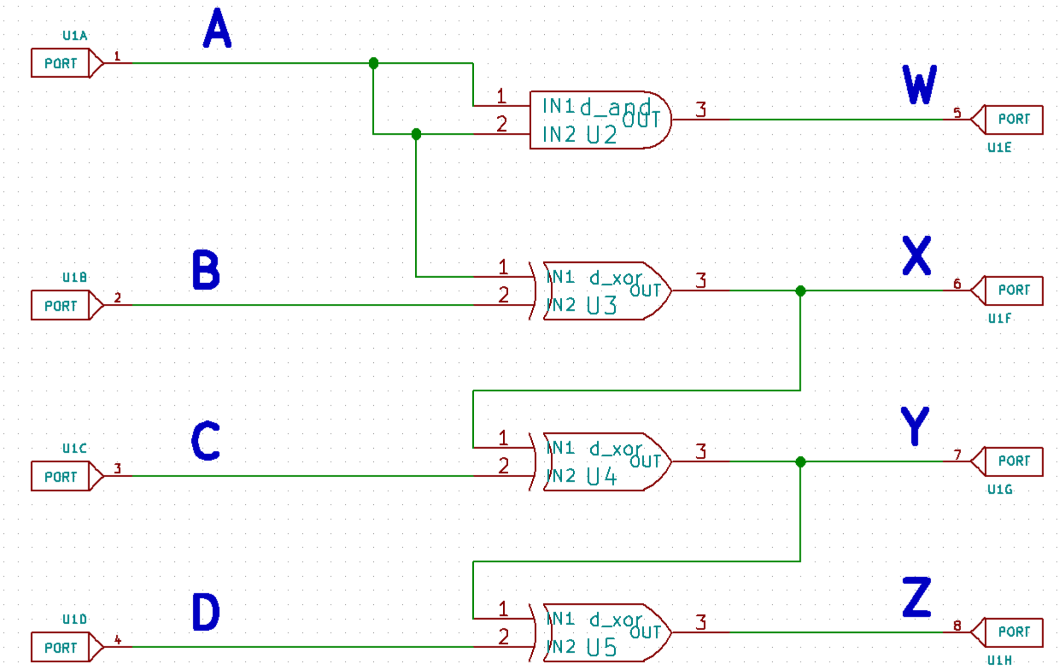
Python plot of Y



Python plot of Z

II. Sub circuit Implementation

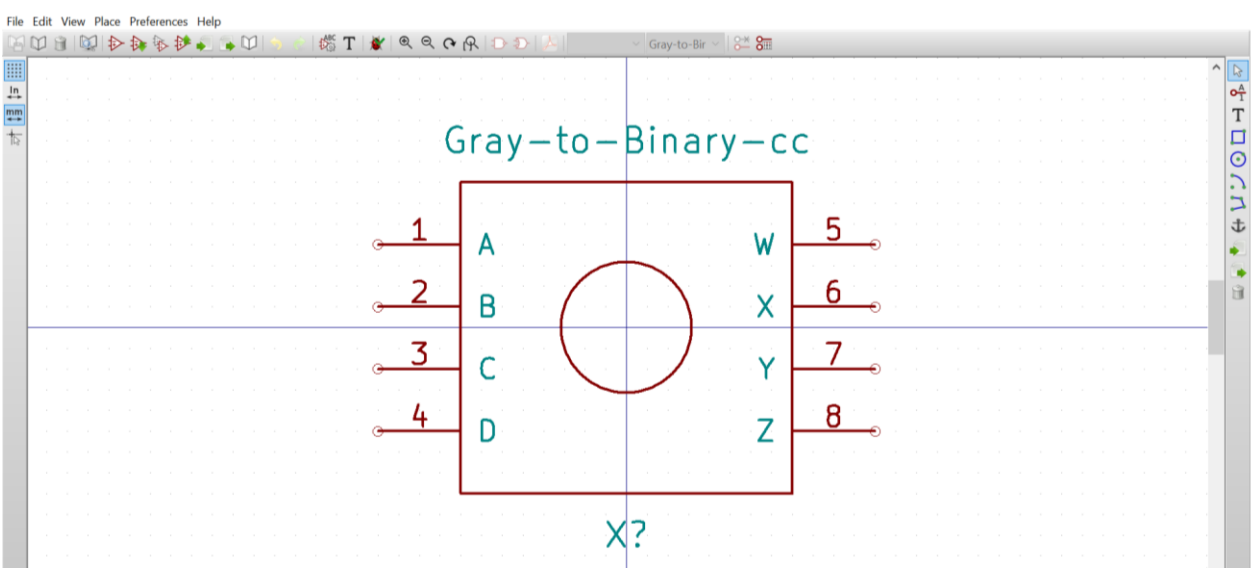
Creating the Subcircuit:



Here, we make use of an additional AND gate for sub-circuit implementation. When both the inputs of the AND gate is A, it gives the same (A) as the output. Hence, it does not affect the functionality of the circuit.

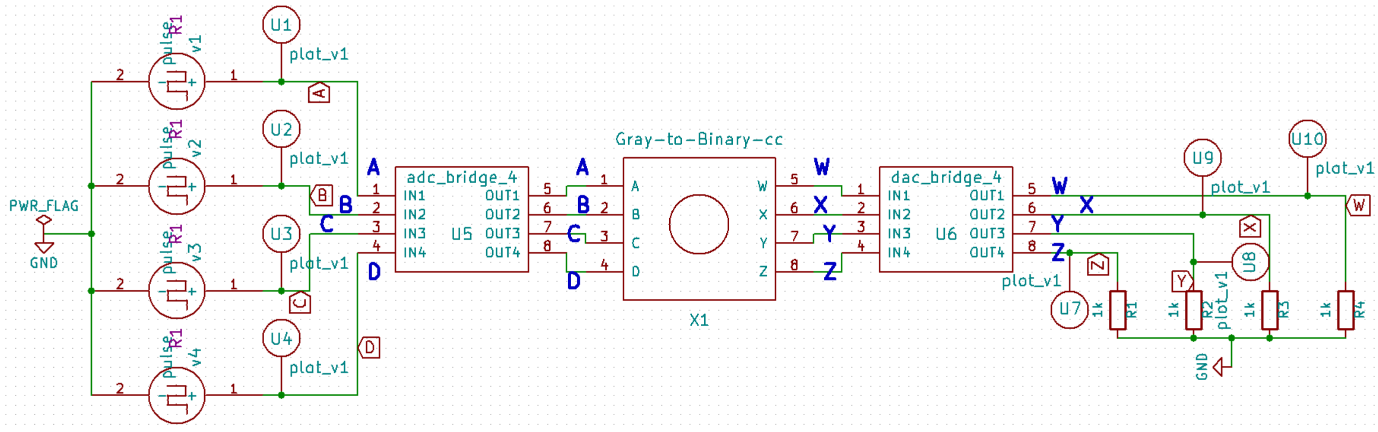
Creating the circuit symbol using Library editor :

- Create new component -> Enter component name and Default reference designator (X - since user defined)
- Draw the symbol, Generate netlist and save it (under eSim_Subckt library)



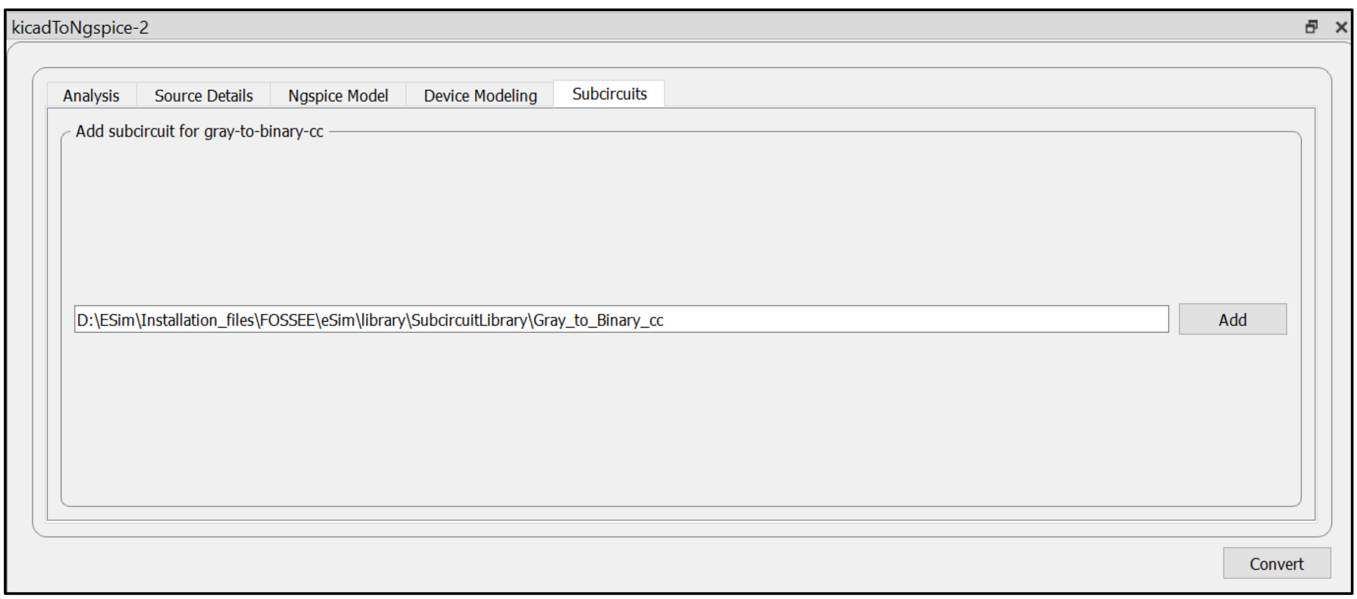
Create new project - new schematic:

Schematic design using subcircuit



Kicad to Ngspice Conversion:

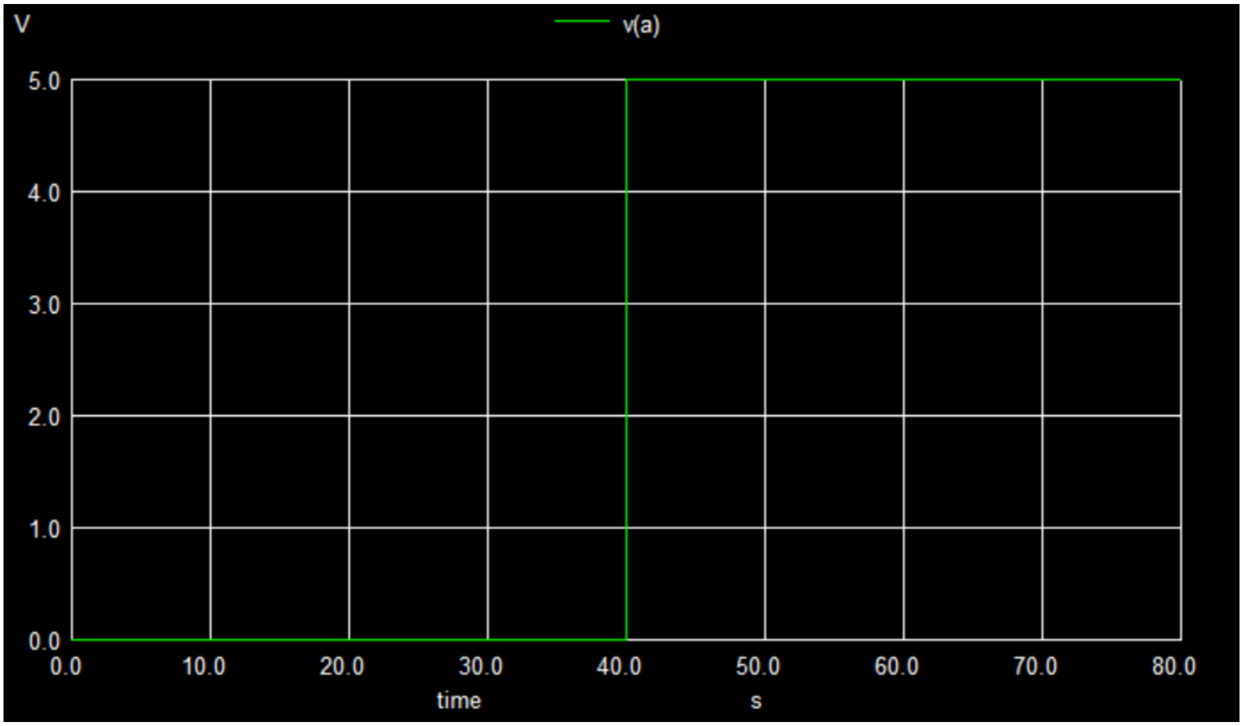
We use the same transient analysis parameters as the main circuit, but in addition, we mention the path of the sub circuit used:



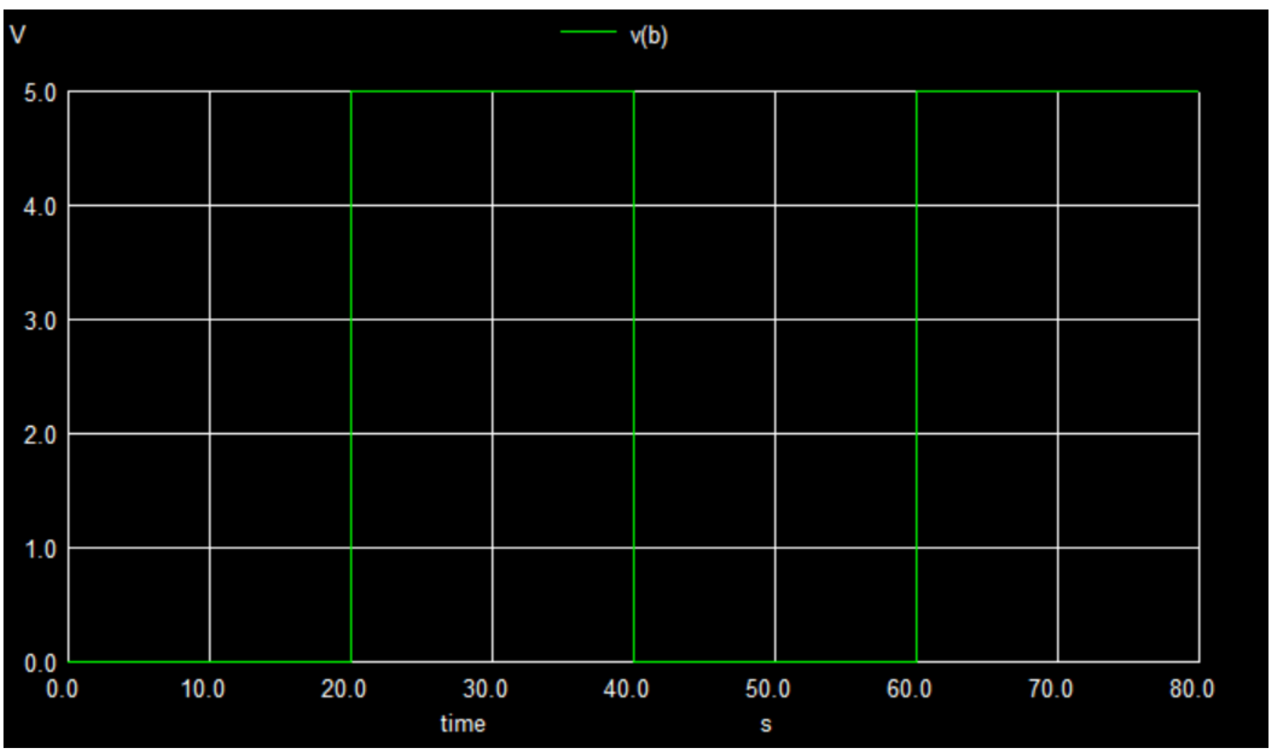
Circuit simulation Output

I. Ngspice Plots

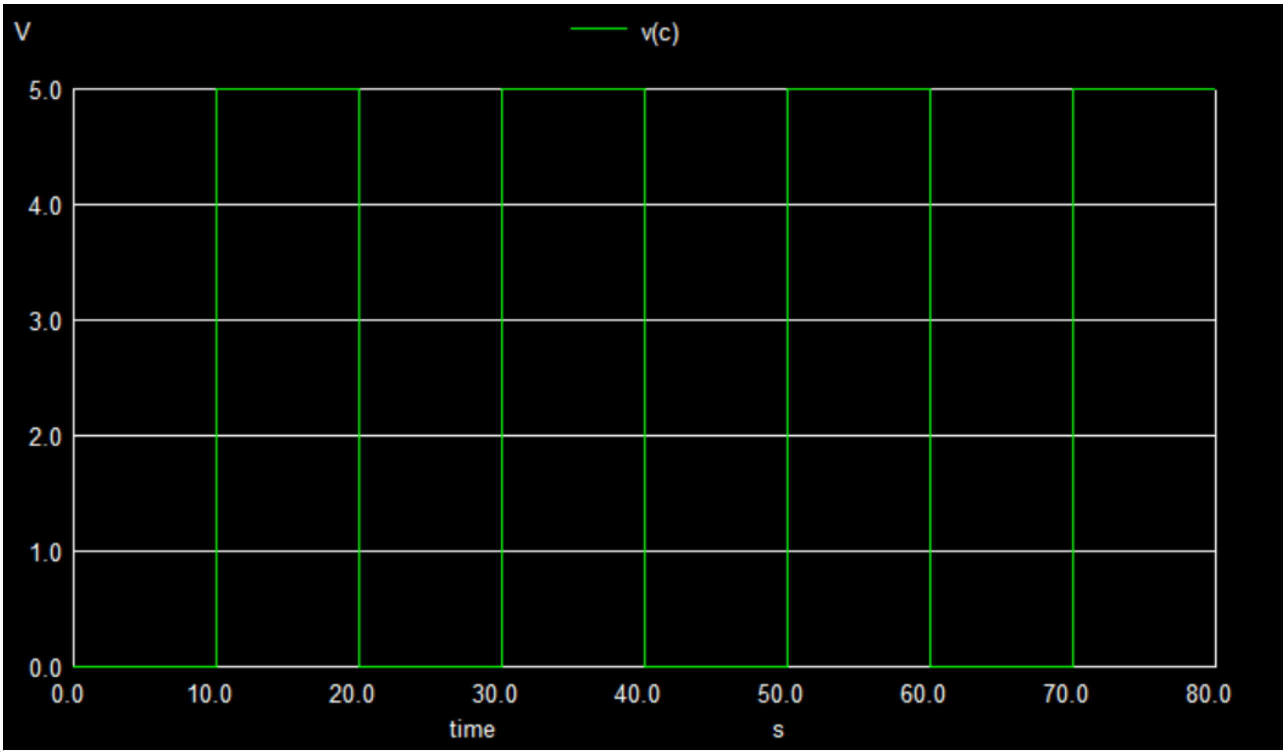
Inputs:



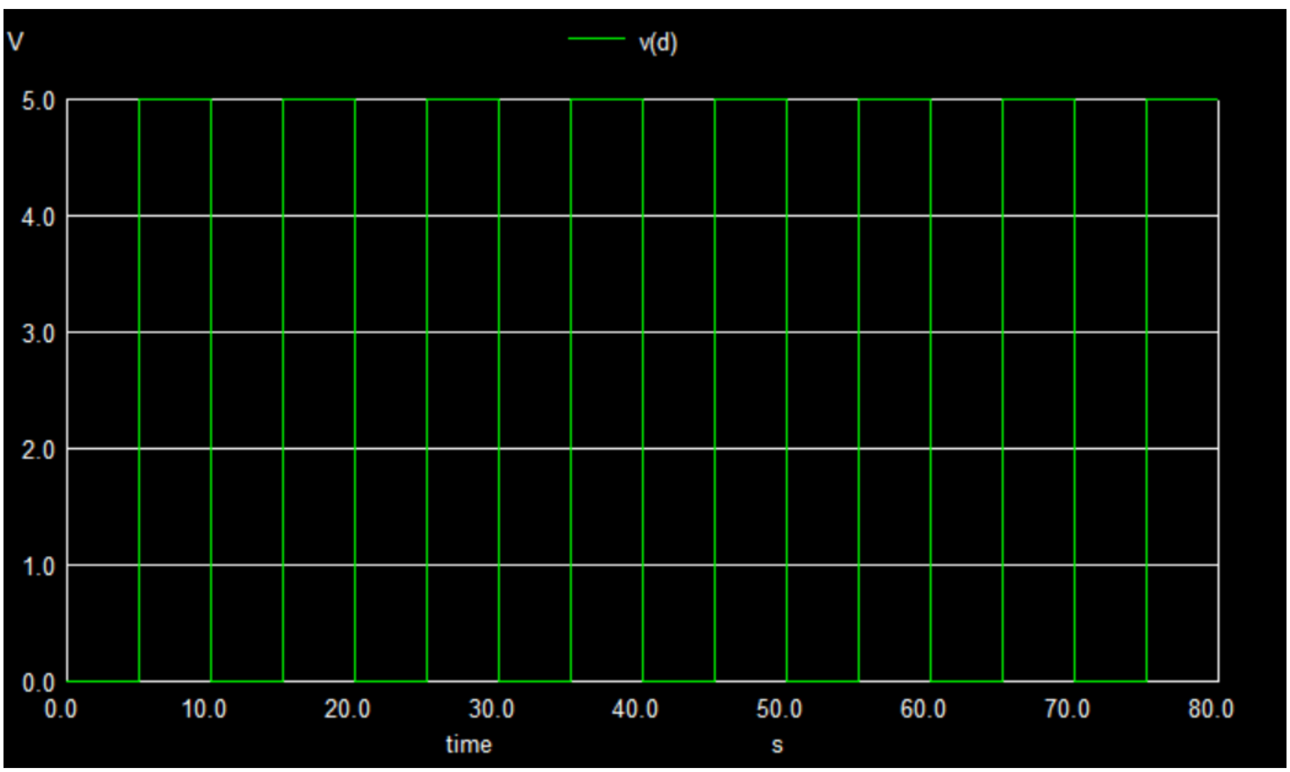
Ngspice plot of A



Ngspice plot of B

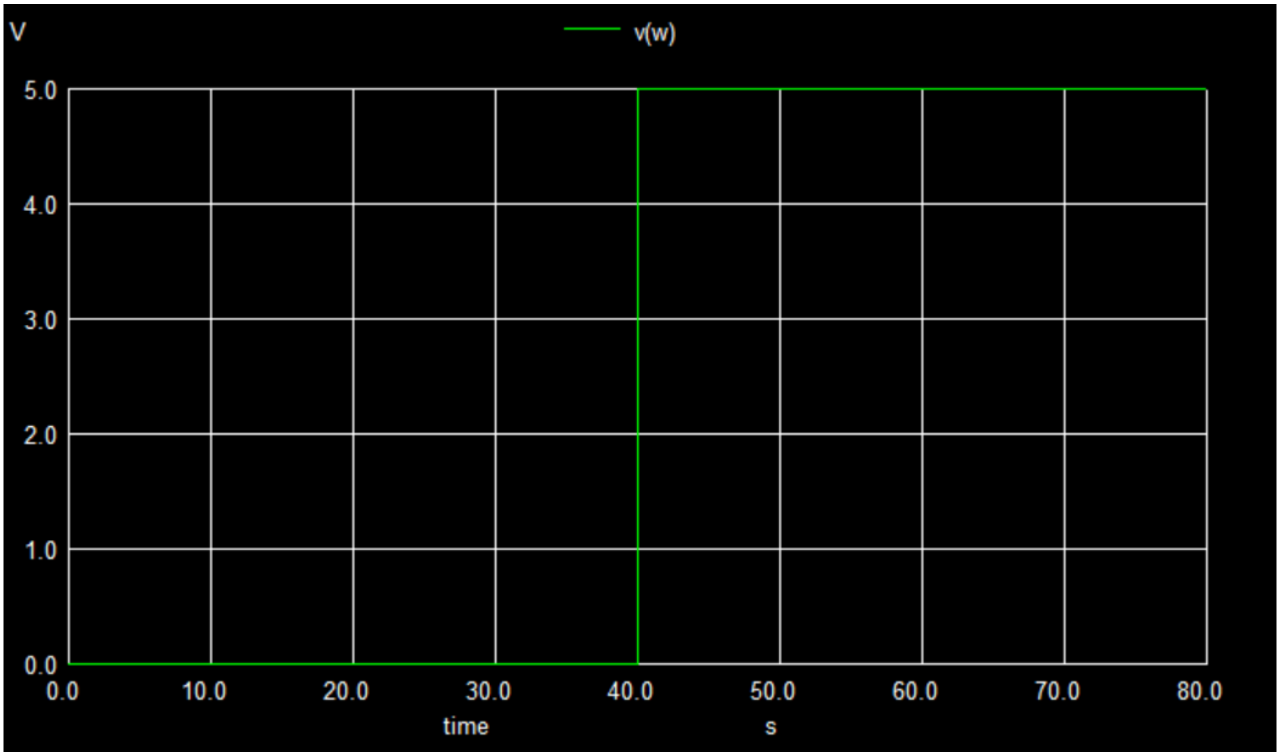


Ngspice plot of C

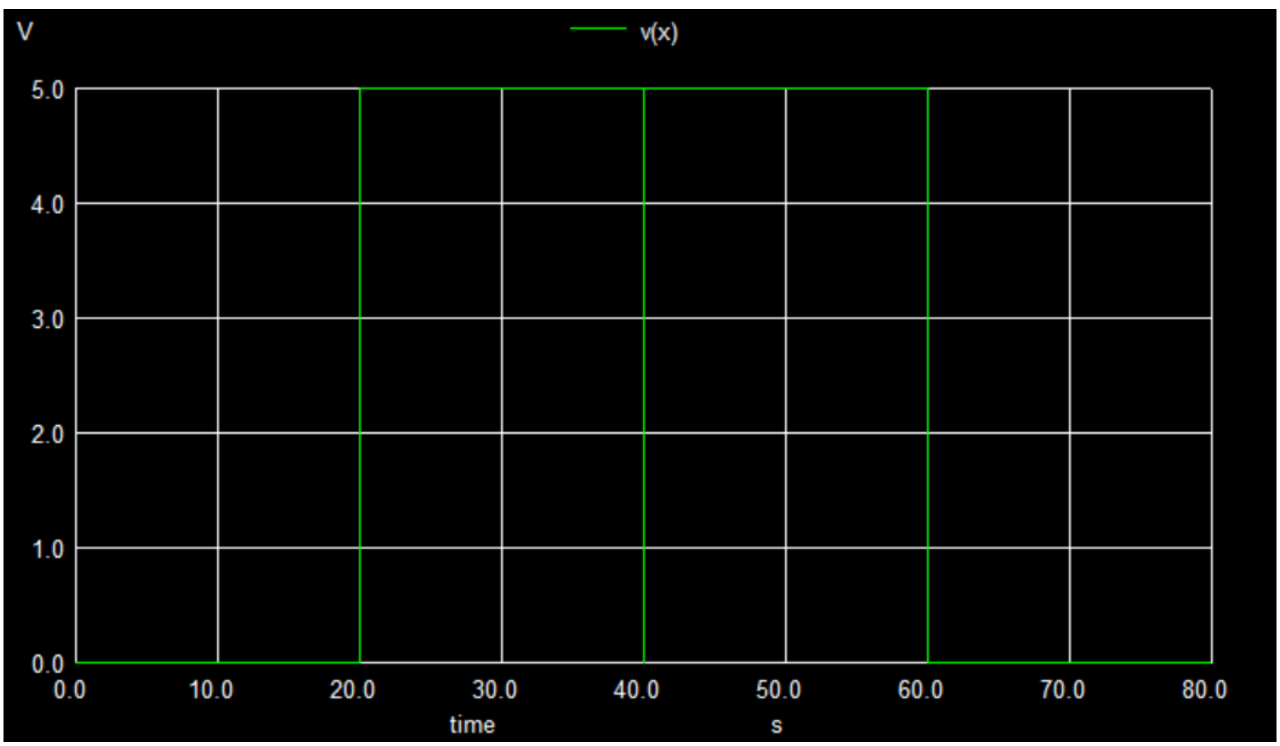


Ngspice plot of D

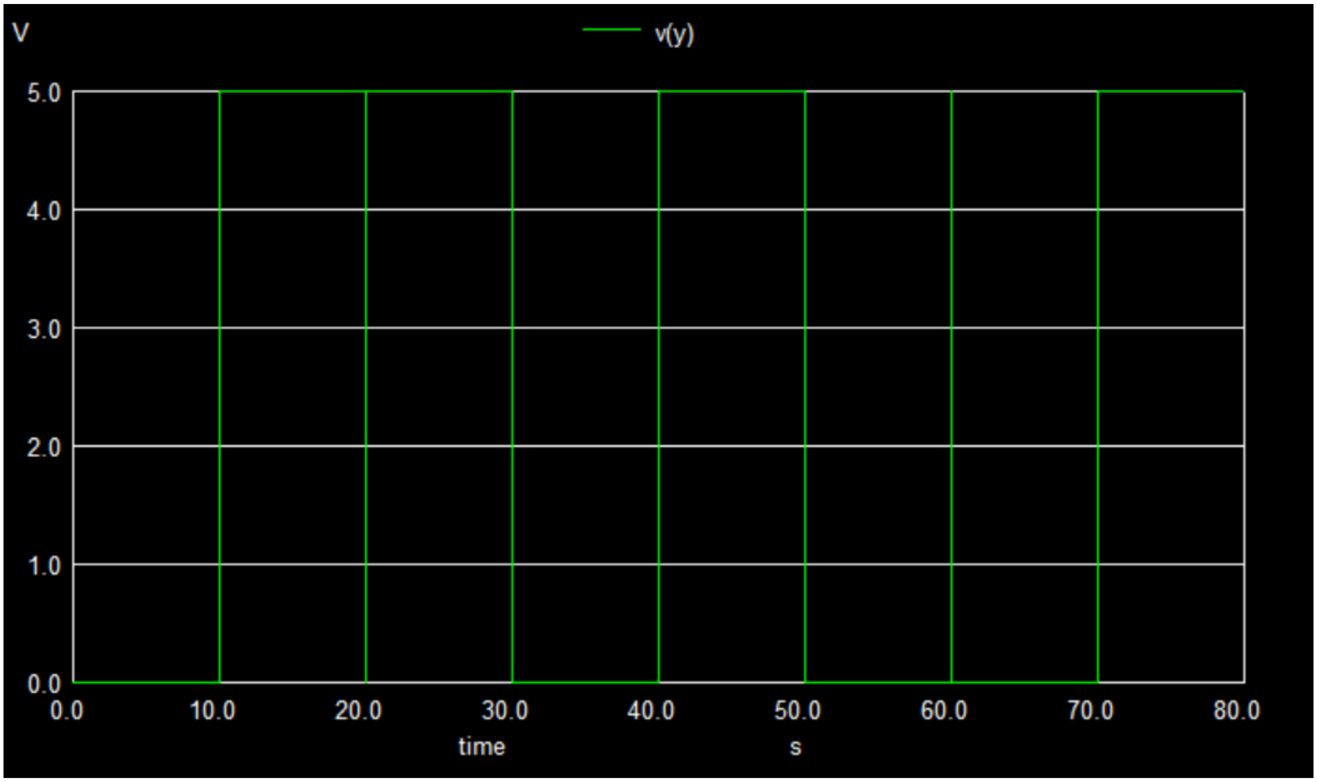
Outputs:



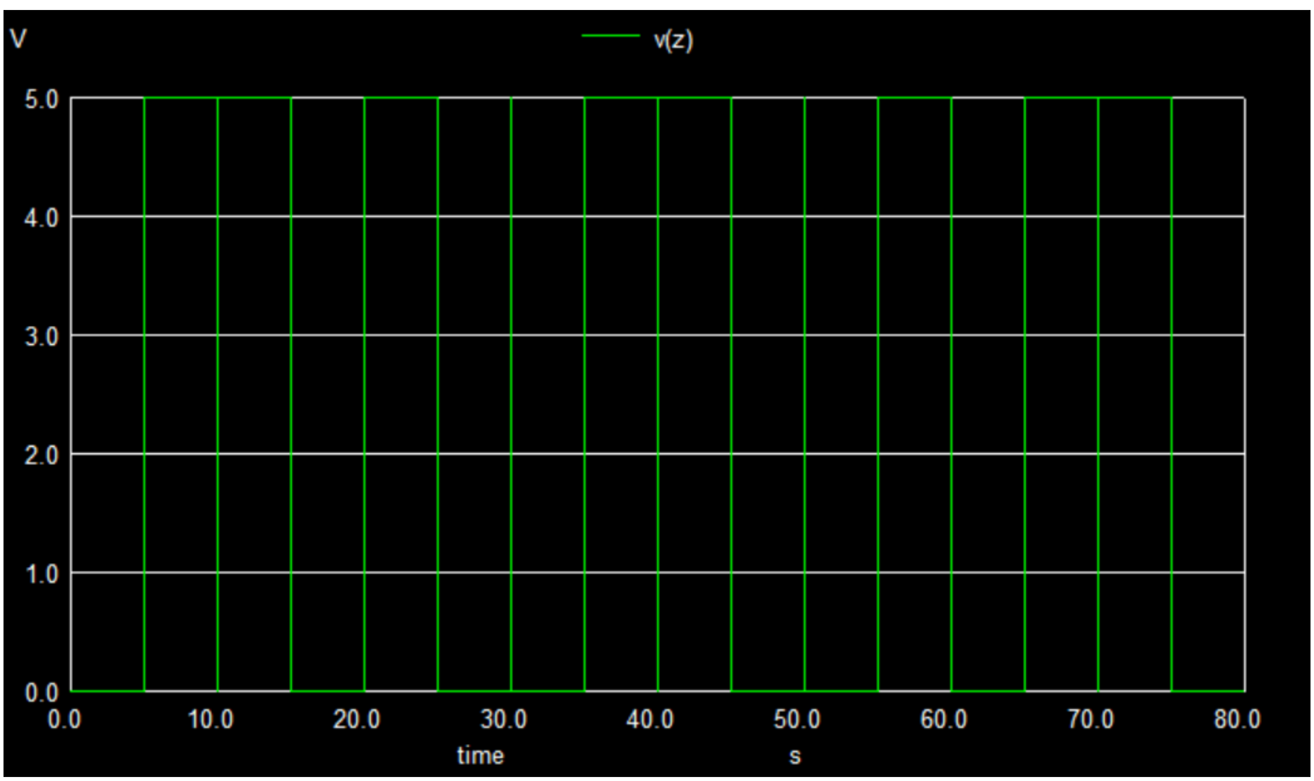
Ngspice plot of W



Ngspice plot of X

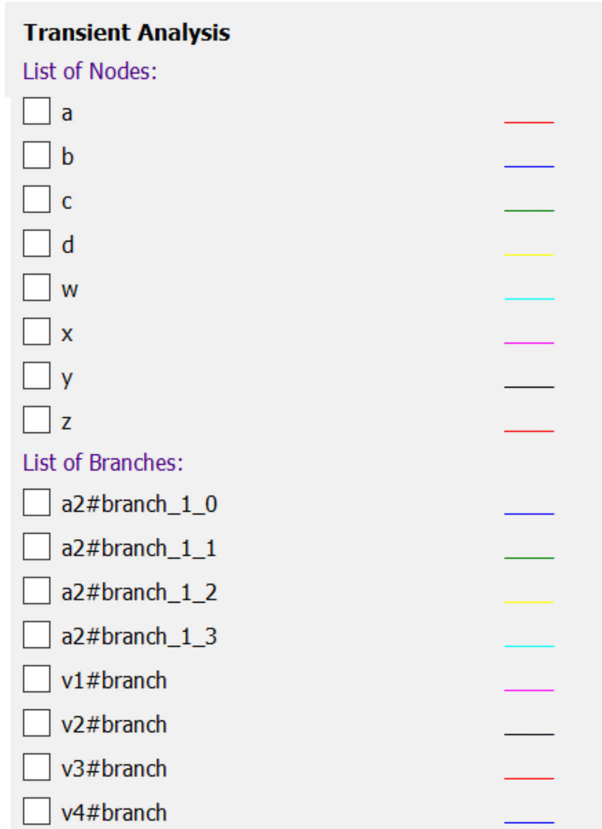


Ngspice plot of Y

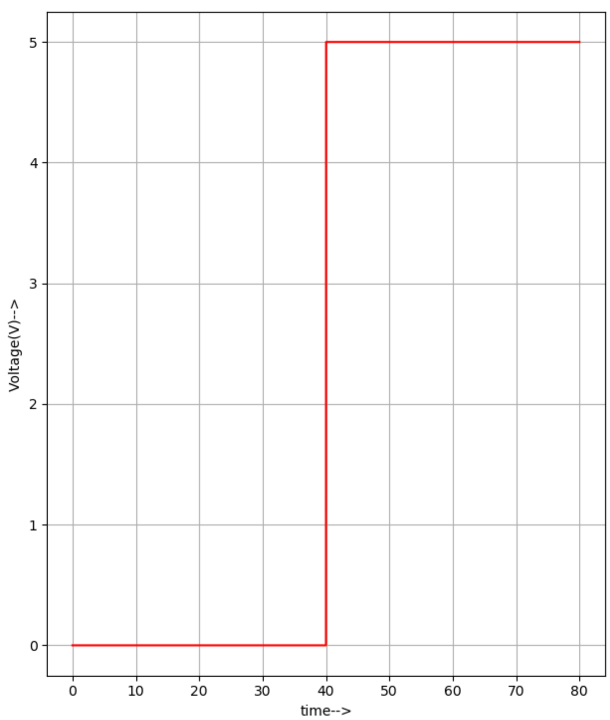


Ngspice plot of Z

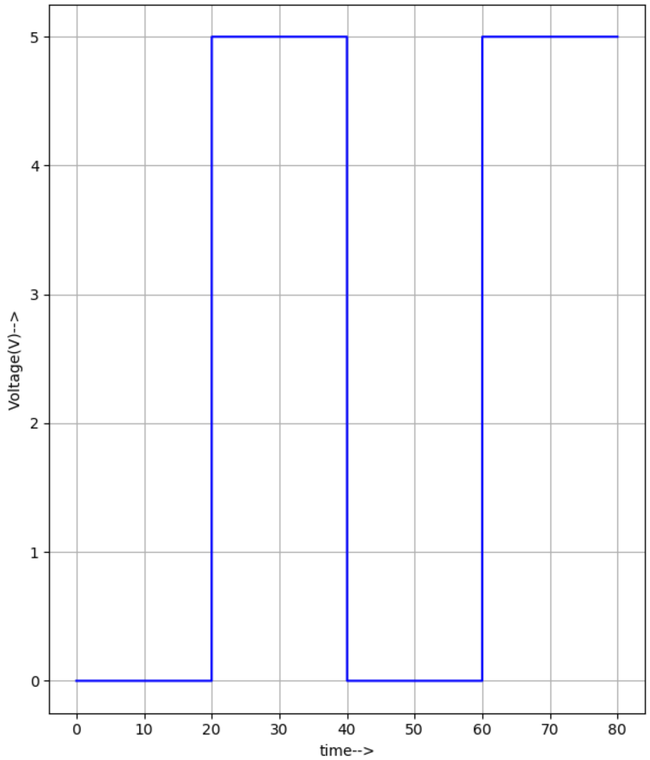
II. Python Plots



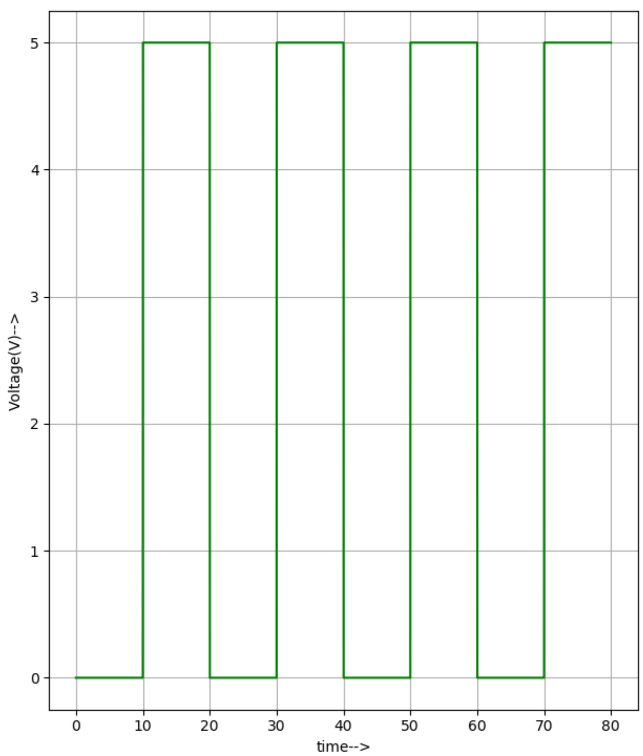
Inputs:



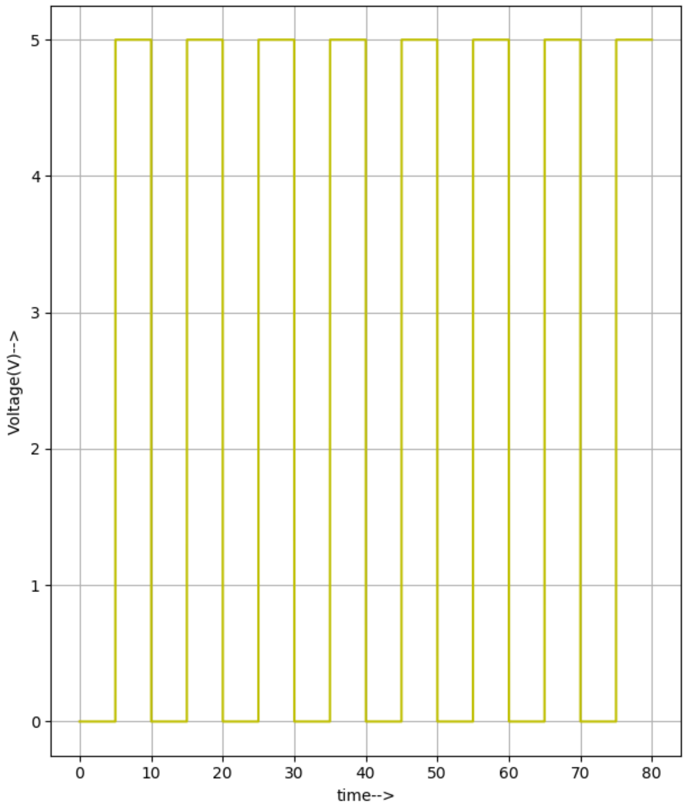
Python plot of A



Python plot of B

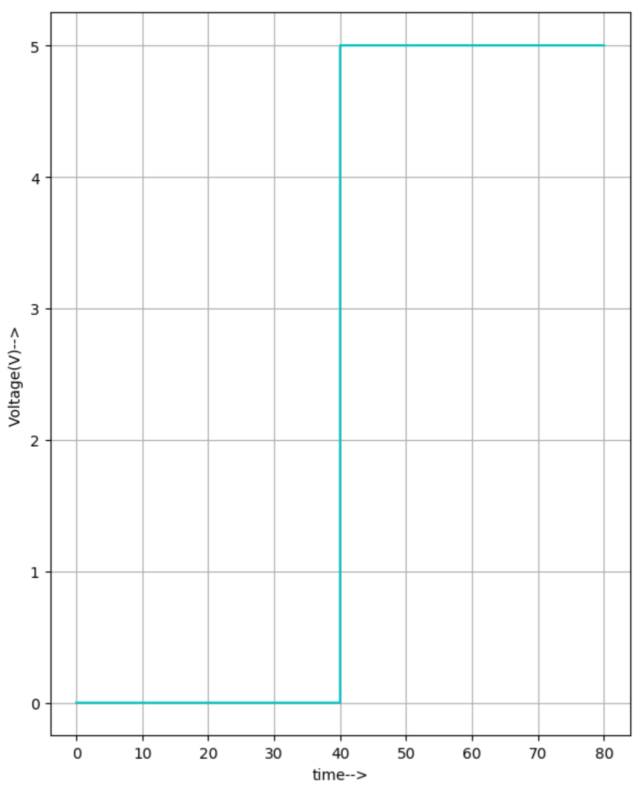


Python plot of C

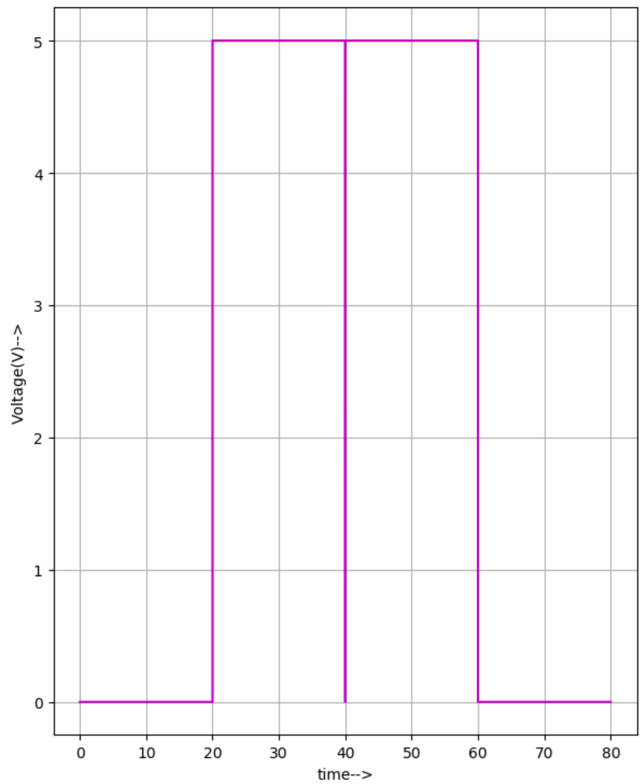


Python plot of D

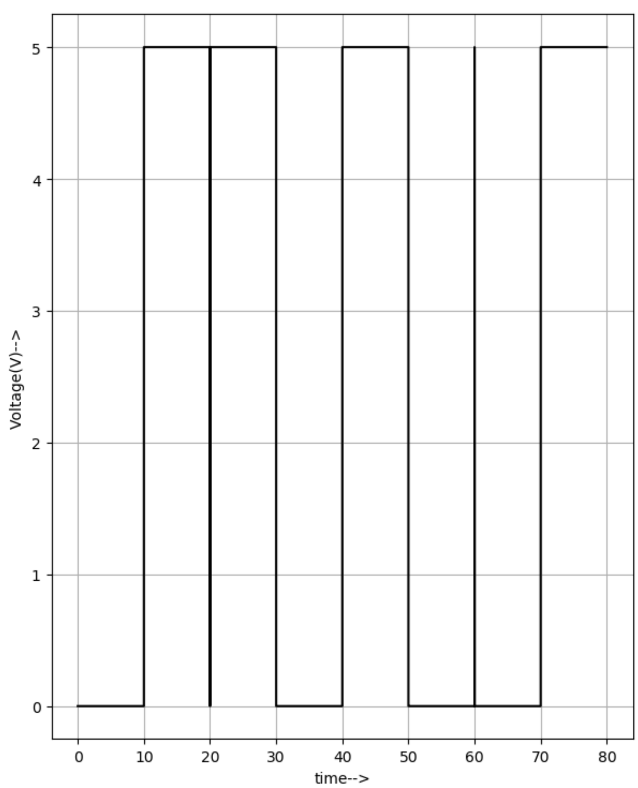
Outputs:



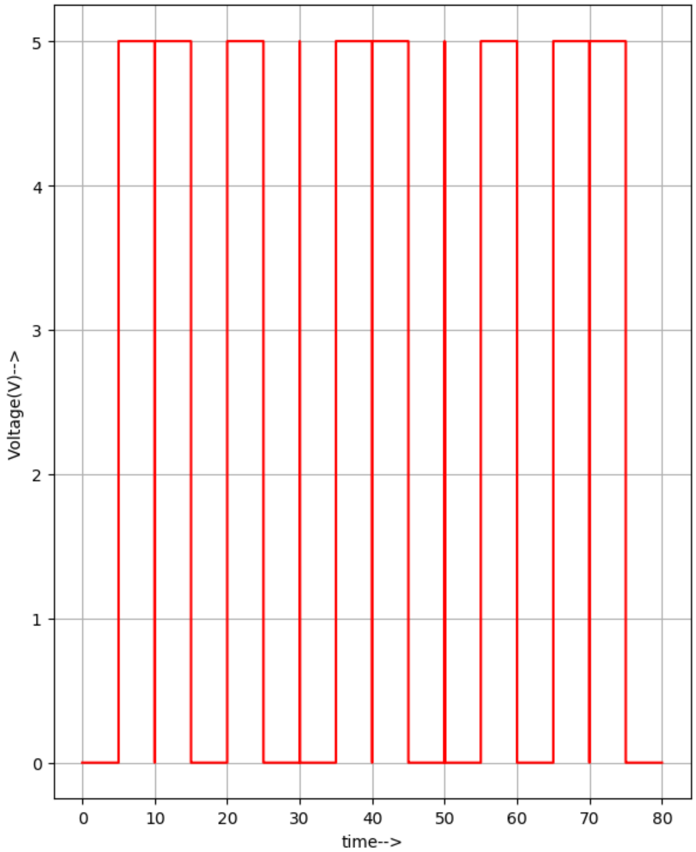
Python plot of W



Python plot of X



Python plot of Y



Python plot of Z

Result:

Both the circuits give the same output. Thus, a Gray to Binary code converter has been created along with Main circuit and Subcircuit implementation. The outputs have also been verified.

References:

https://electricalworkbook.com/design-of-binary-to-gray-code-converter-circuit/