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 Binary Adder Subtractor Circuit in ALU using Full Adder Subcircuit in eSim

---

Theory

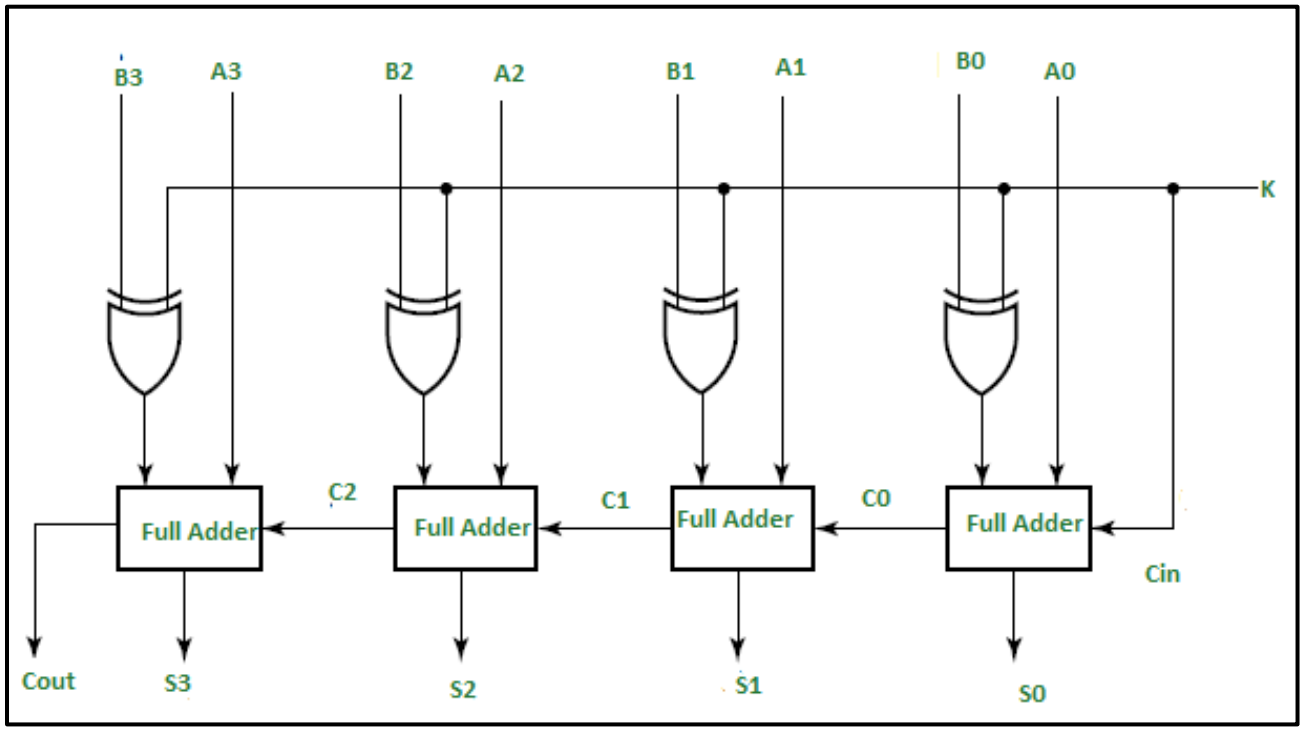
A Binary Adder-Subtractor is one which is capable of both addition and subtraction of binary numbers in one circuit itself. The operation being performed depends upon the binary value that the control signal holds {here the control signal is K}. It is one of the components of the ALU (Arithmetic Logic Unit).

(i) If K = 0 ➔ Output of XOR gate is the same as the bit of 'B' operand and Cin = 0
Operation performed = A + B ➔ Addition {works as Adder}

(ii) If K = 1 ➔ Output of XOR gate is the negation of the bit of 'B' operand and Cin = 1
Operation performed = A - B ➔ Subtraction {works as Subtractor}

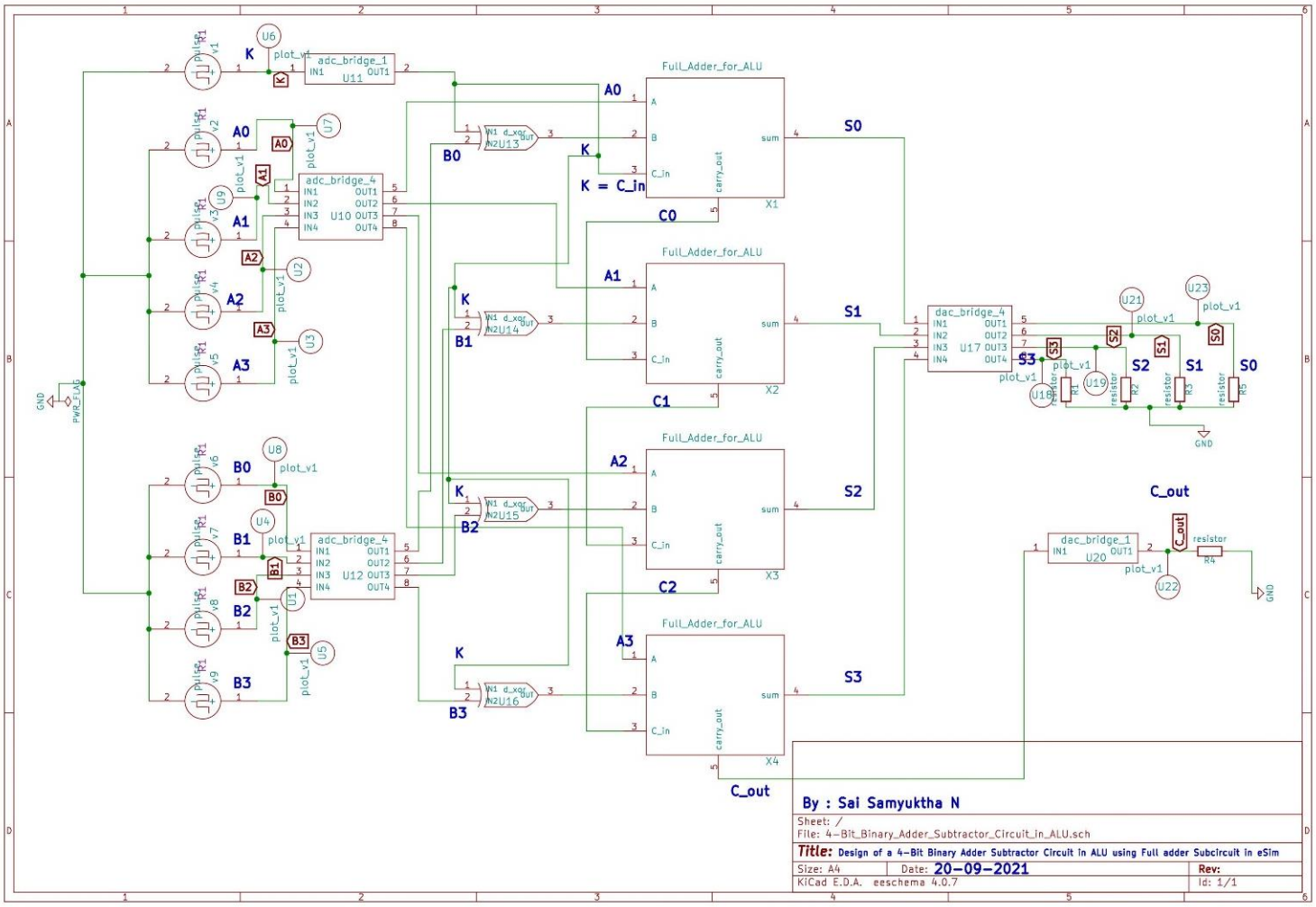
Circuit Diagram:

The circuit can be implemented using four x-or gates and four full adders

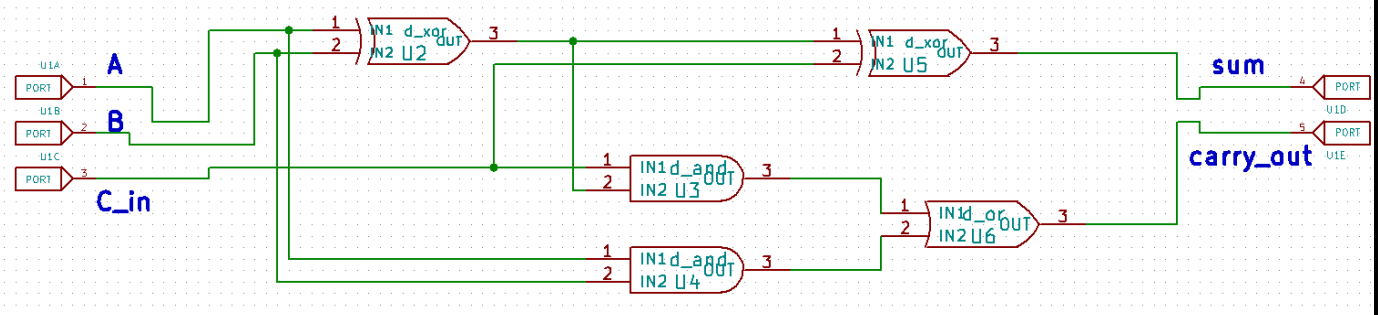


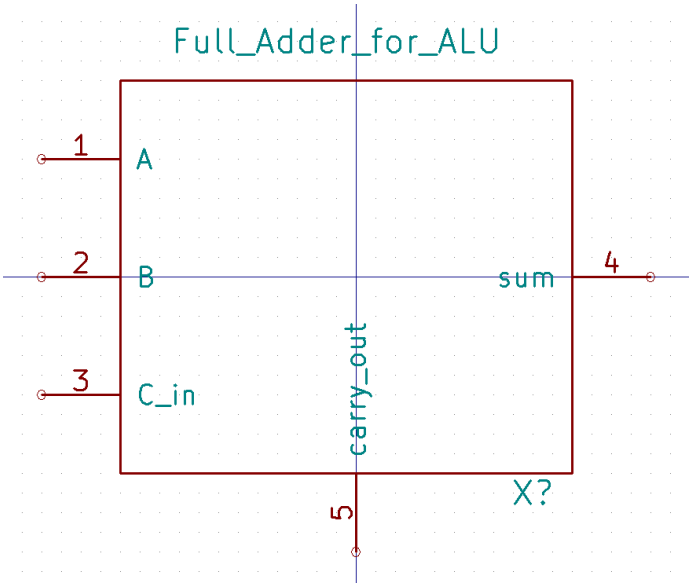
eSim Implementation

Main circuit Implementation



We make use of a Full Adder Sub circuit:





The main circuit has three parts:

1. **Input**

The input consists of control line ‘K’. Depending on the value of ‘K’, either Addition or Subtraction is performed by the circuit

The individual bits of the 4-bit Binary input is given:

For A : A3 A2 A1 A0

For B : B3 B2 B1 B0

Where A3 is the MSB of A and B3 is the MSB of B.

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 output consists of the 4-bit Binary Output and the Carry_out bit

Binary output (S) : S3 S2 S1 S0

Where S3 is the MSB bit of S.

Carry_out - C_out

Since the carry out bit can either be 0 or 1, we only one bit to represent the C_out

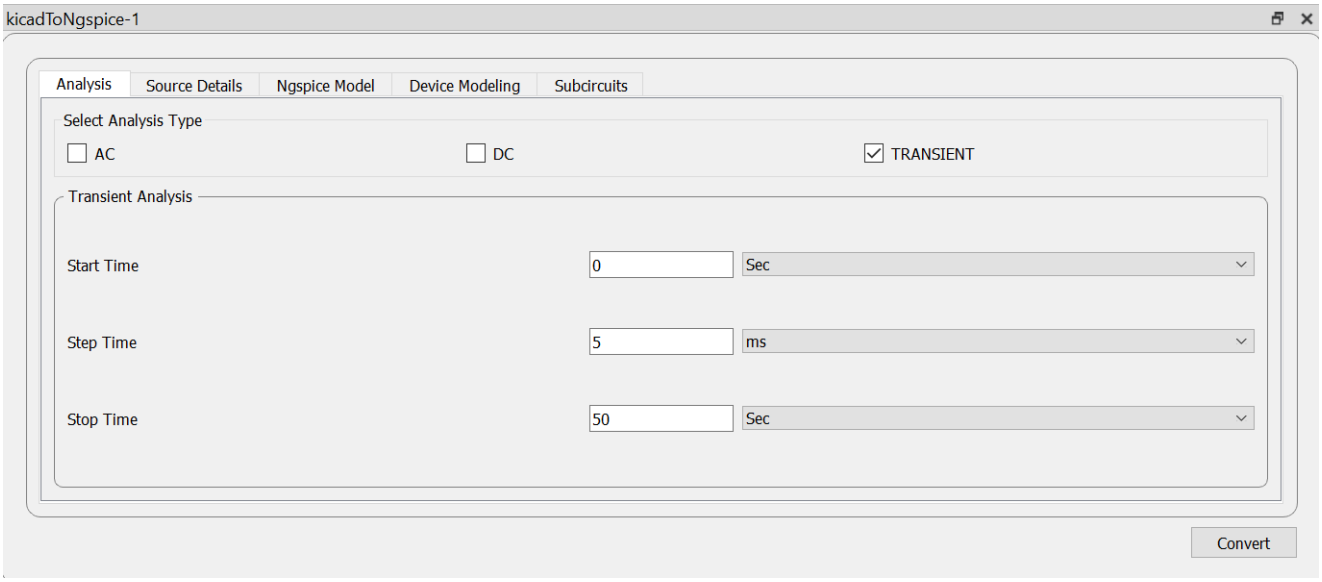
We make use of the digital to analog converter to convert the signals back into analog and compute the output

3. Logic Circuit

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

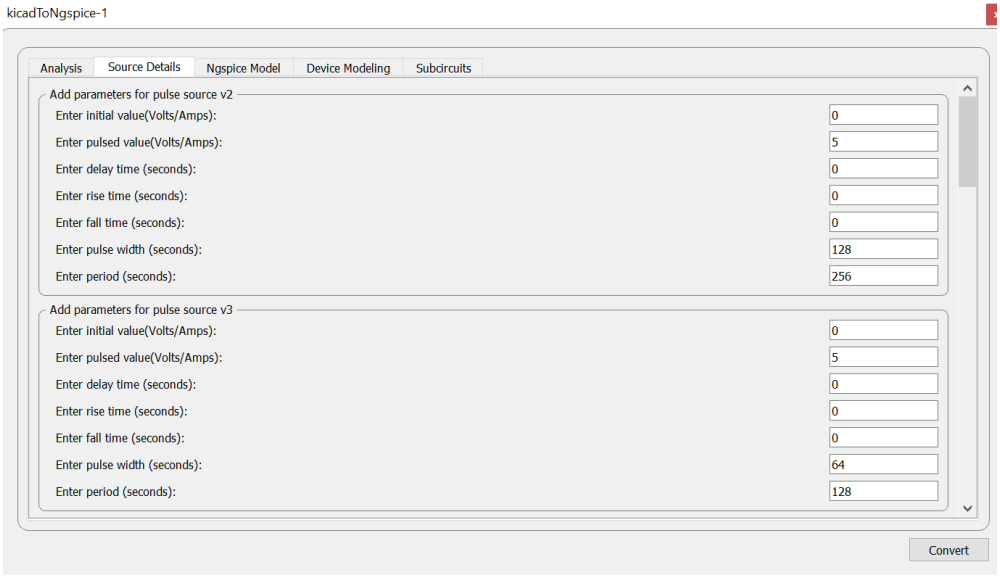
Kicad to Ngspice Conversion

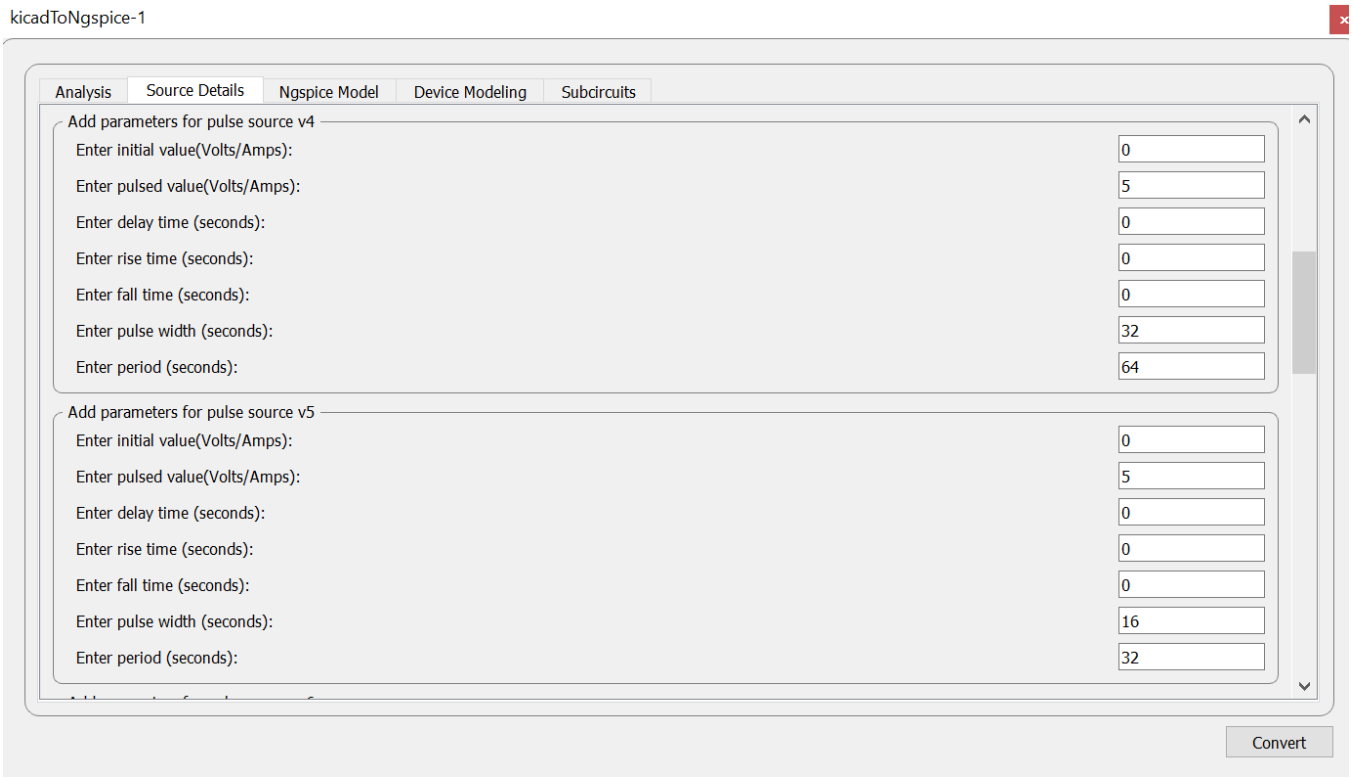
Here we make use of transient analysis:

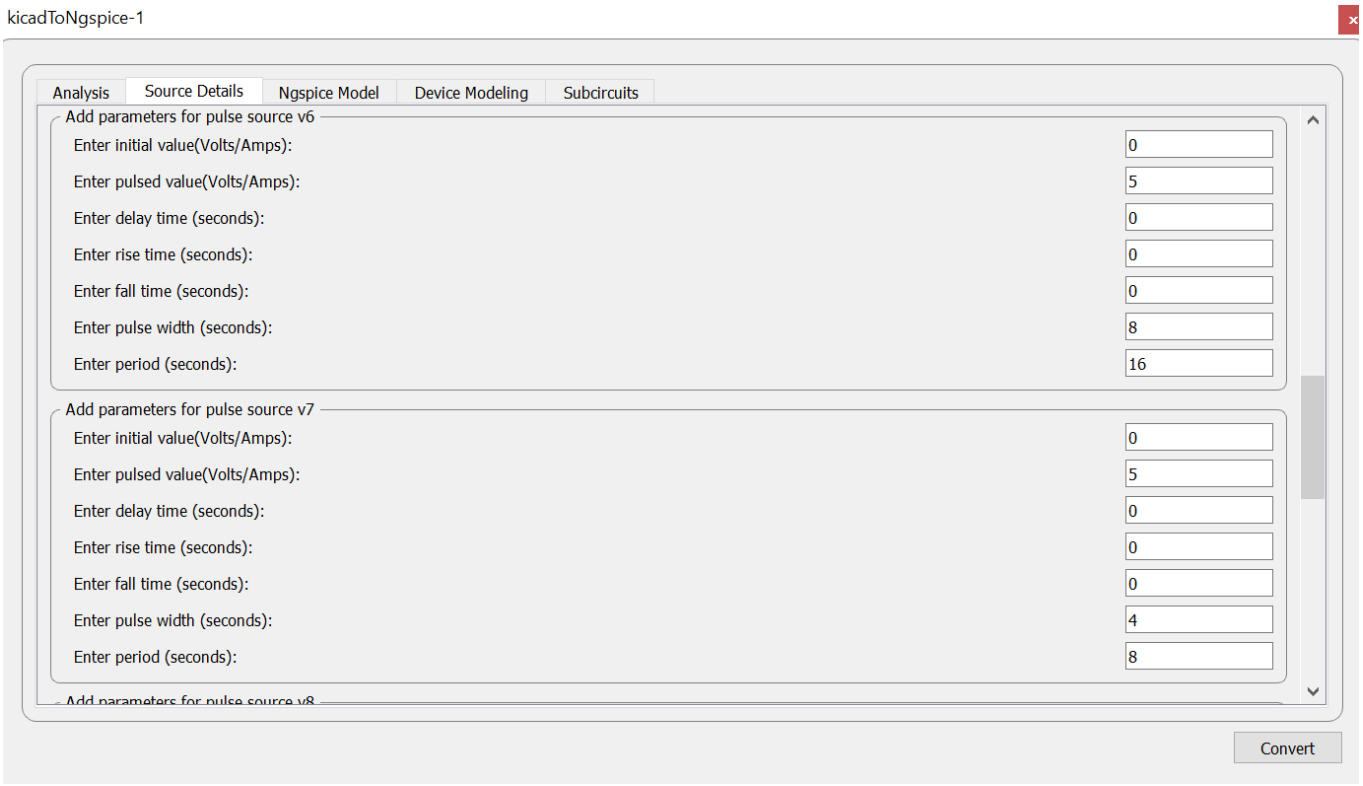


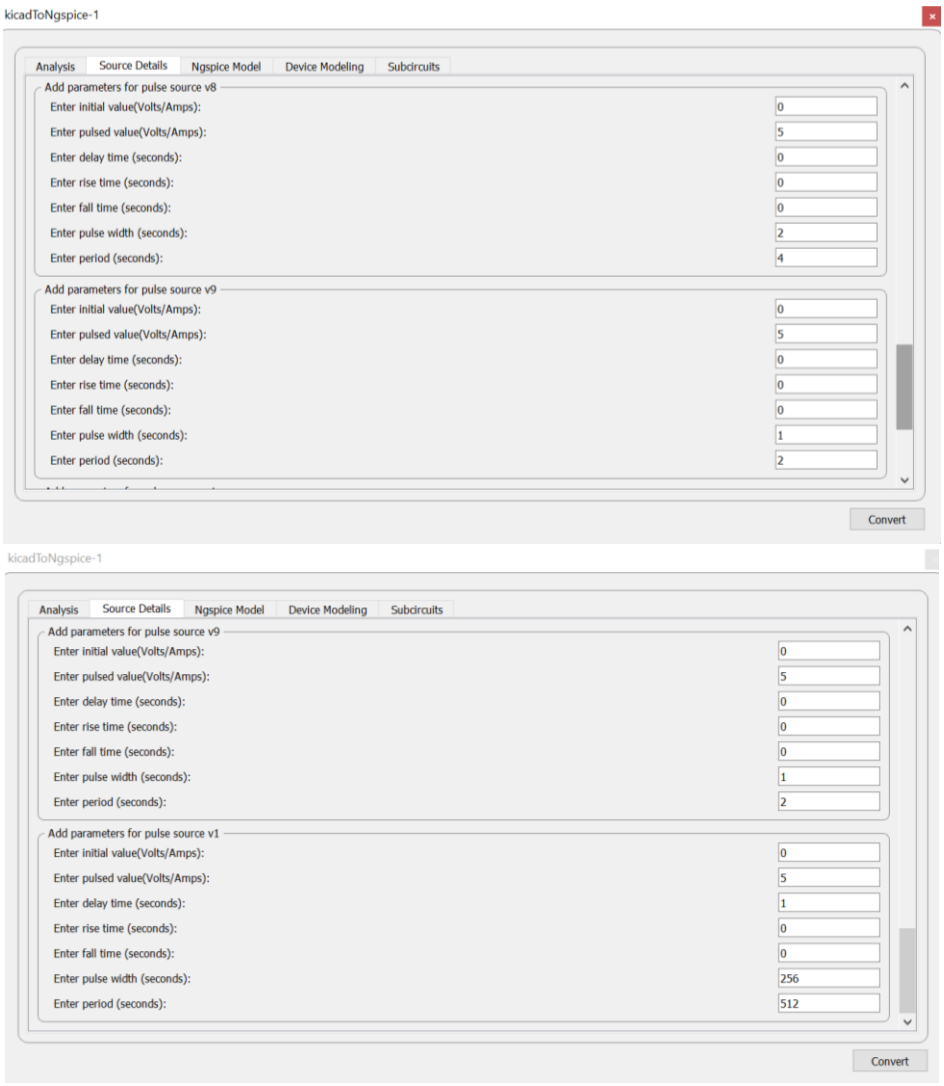
For better simulation output, we take the stop time as 50 sec. For analysis of other inputs, we can change the start time and stop time accordingly.

Source Details:

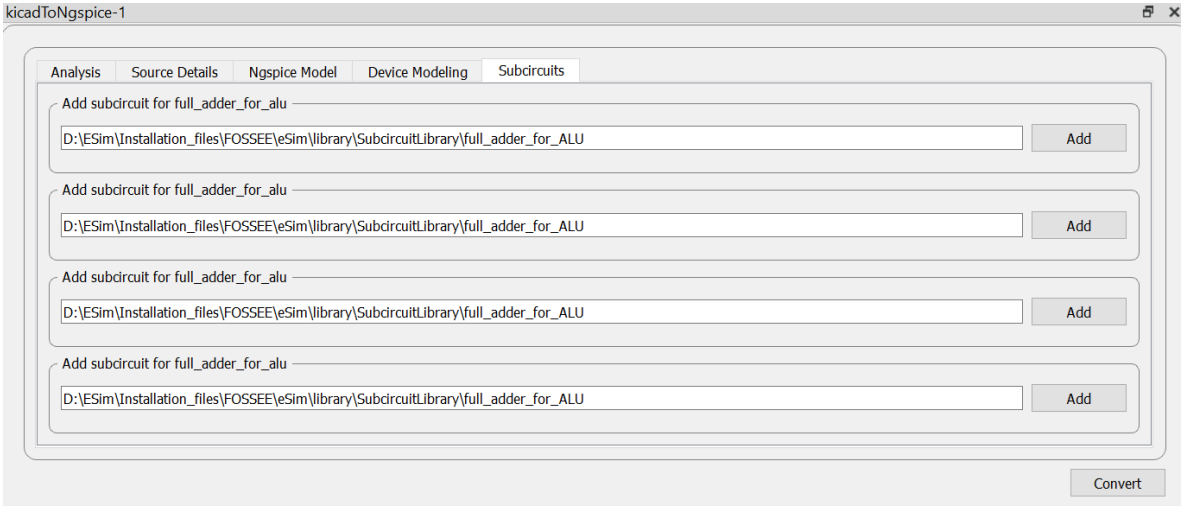



kicadToNgspice-1
Analysis
Source Details
Ngspice Model
Device Modeling
Subcircuits
Add parameters for pulse source v4
Enter initial value(Volts/Amps):
0
Enter pulsed value(Volts/Amps):
5
Enter delay time (seconds):
0
Enter rise time (seconds):
0
Enter fall time (seconds):
0
Enter pulse width (seconds):
32
Enter period (seconds):
64
Add parameters for pulse source v5
Enter initial value(Volts/Amps):
0
Enter pulsed value(Volts/Amps):
5
Enter delay time (seconds):
0
Enter rise time (seconds):
0
Enter fall time (seconds):
0
Enter pulse width (seconds):
16
Enter period (seconds):
32
Convert


kicadToNgspice-1
Analysis
Source Details
Ngspice Model
Device Modeling
Subcircuits
Add parameters for pulse source v6
Enter initial value(Volts/Amps):
0
Enter pulsed value(Volts/Amps):
5
Enter delay time (seconds):
0
Enter rise time (seconds):
0
Enter fall time (seconds):
0
Enter pulse width (seconds):
8
Enter period (seconds):
16
Add parameters for pulse source v7
Enter initial value(Volts/Amps):
0
Enter pulsed value(Volts/Amps):
5
Enter delay time (seconds):
0
Enter rise time (seconds):
0
Enter fall time (seconds):
0
Enter pulse width (seconds):
4
Enter period (seconds):
8
Convert



Subcircuits:

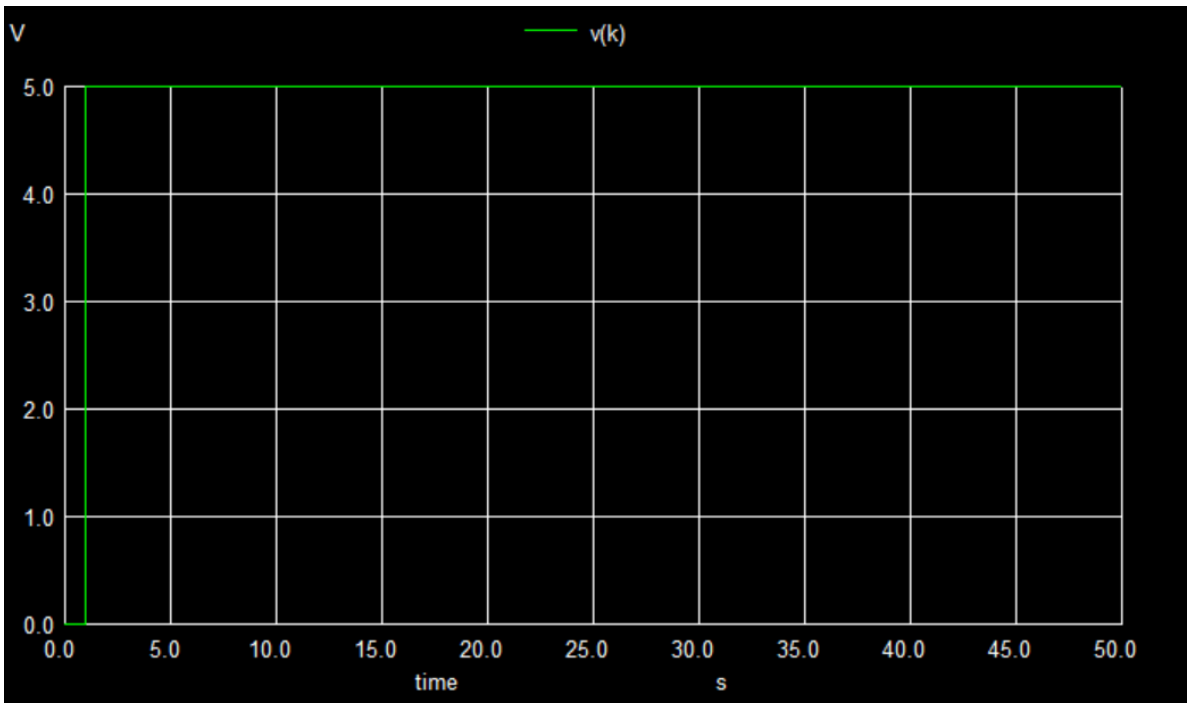


Other fields are left as default.

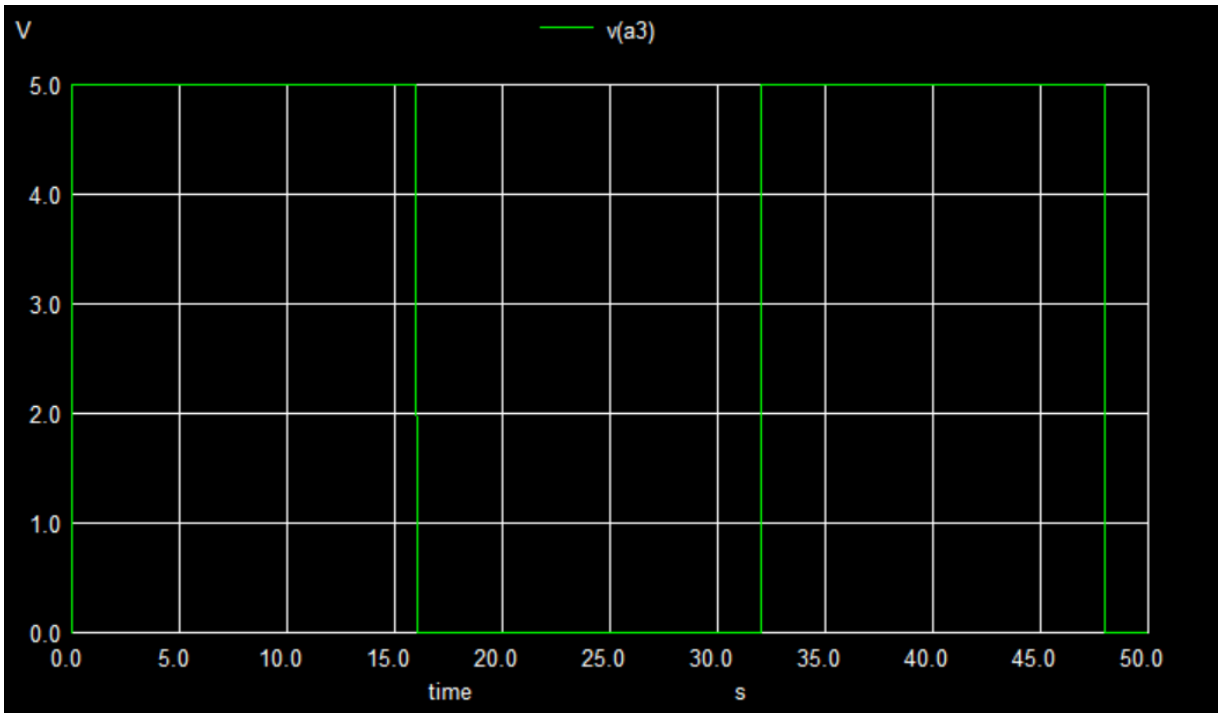
Circuit simulation Output

I. NGSPICE PLOTS:

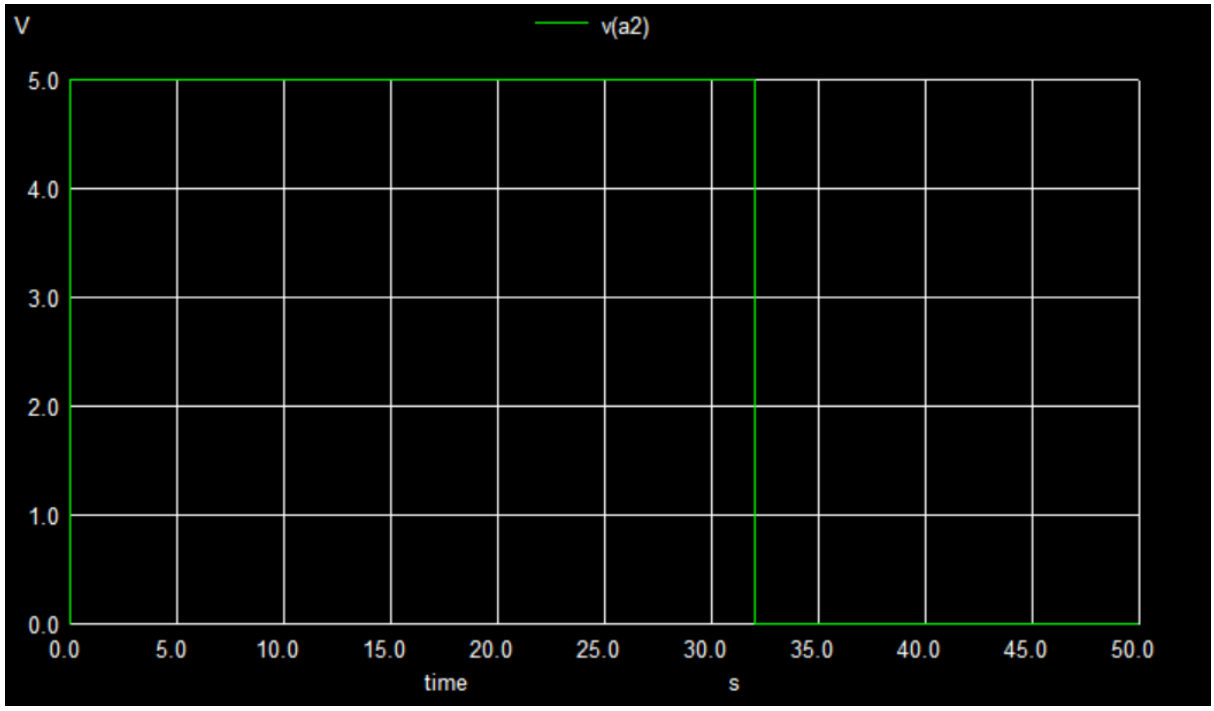
- Inputs:



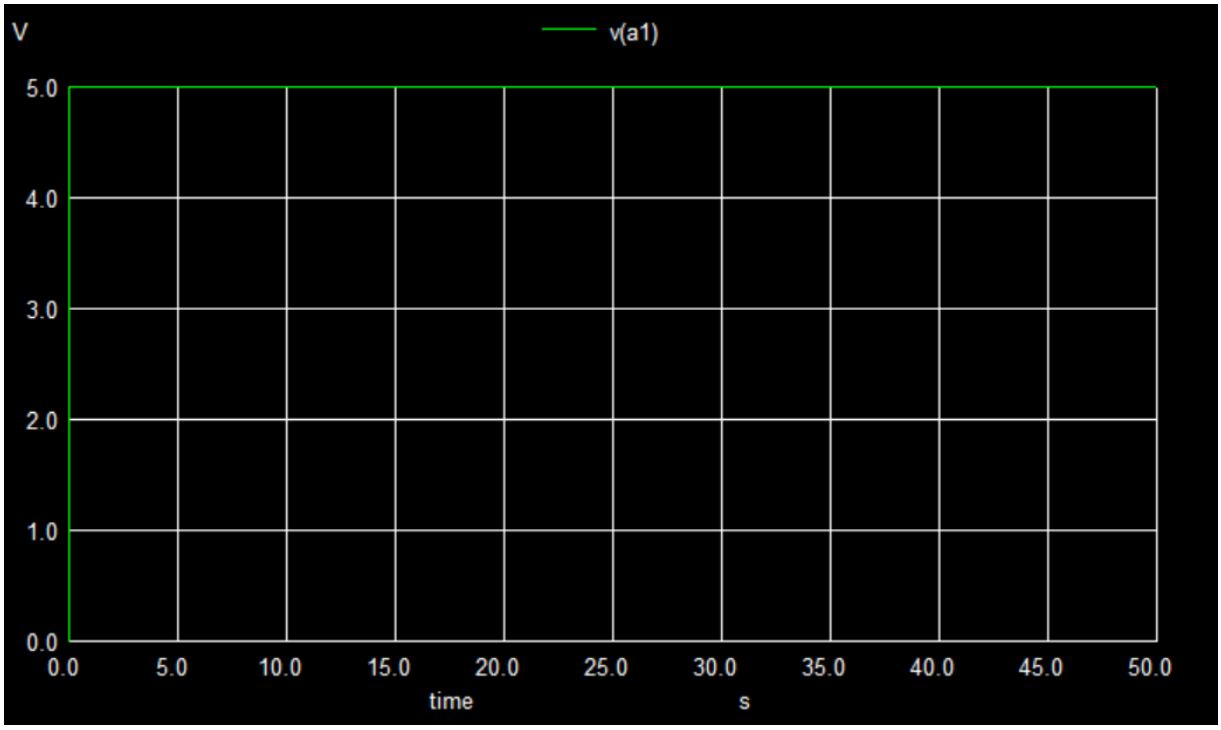
Ngspice plot of K



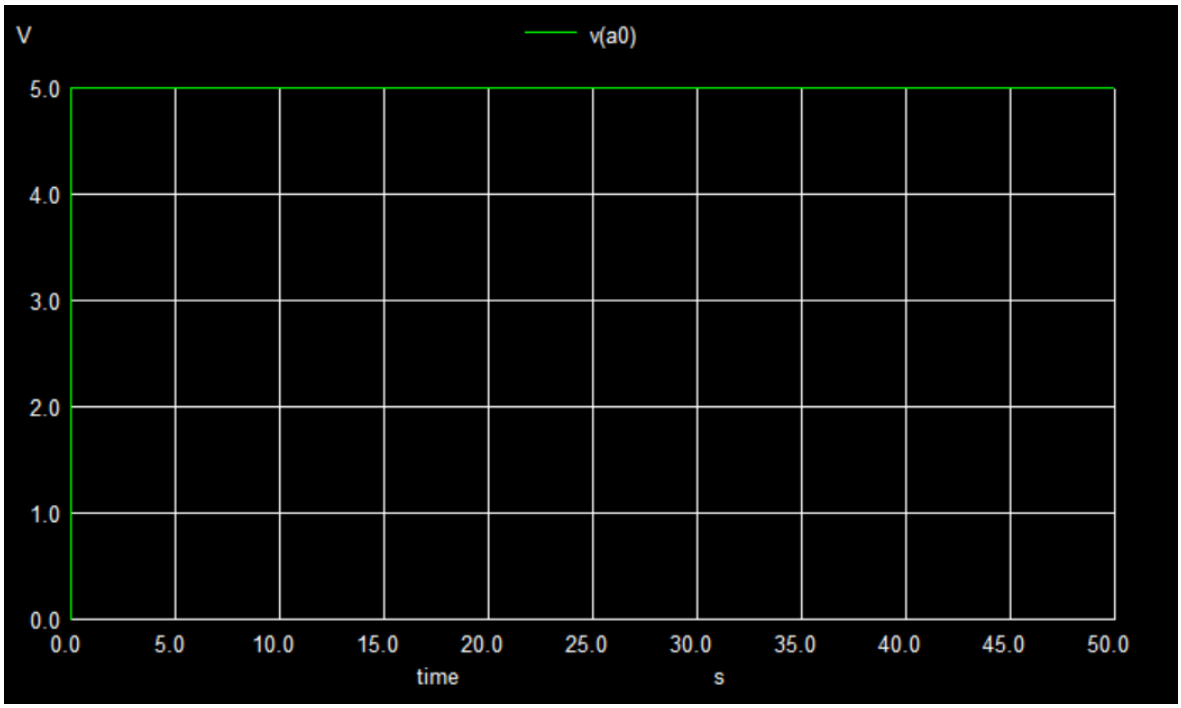
Ngspice plot of A3



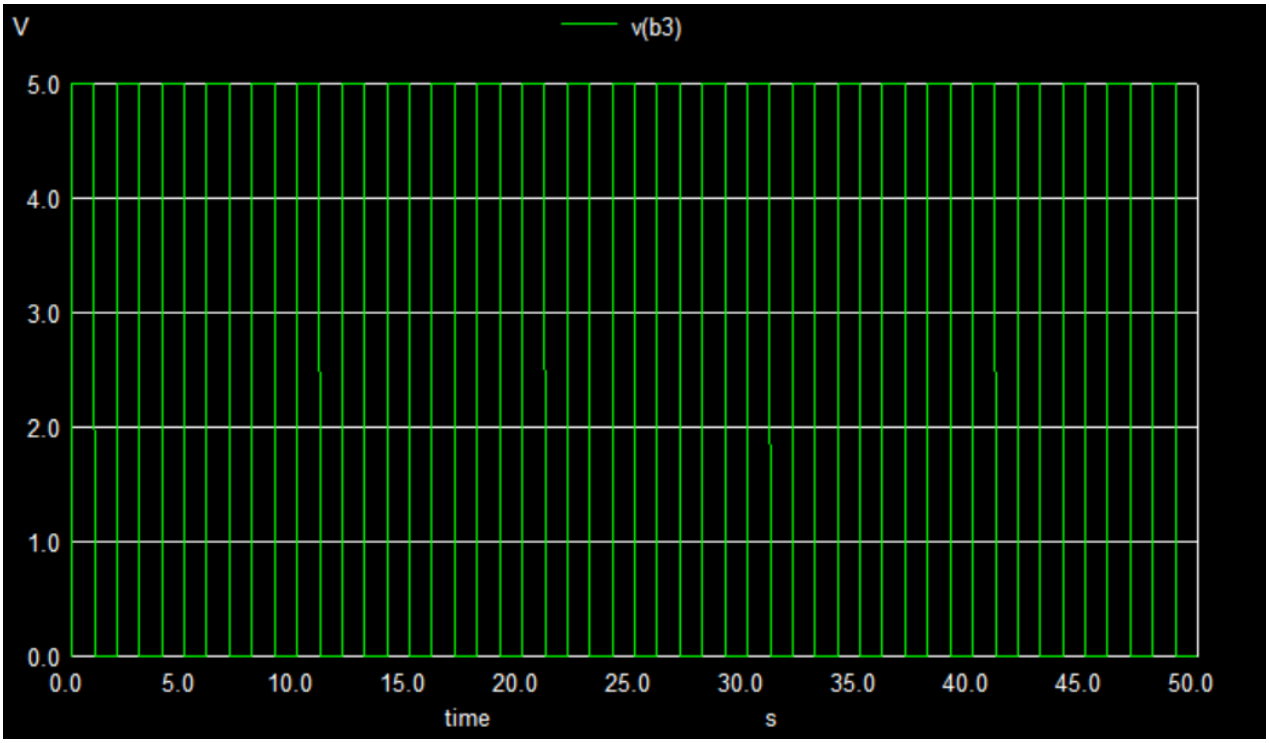
Ngspice plot of A2



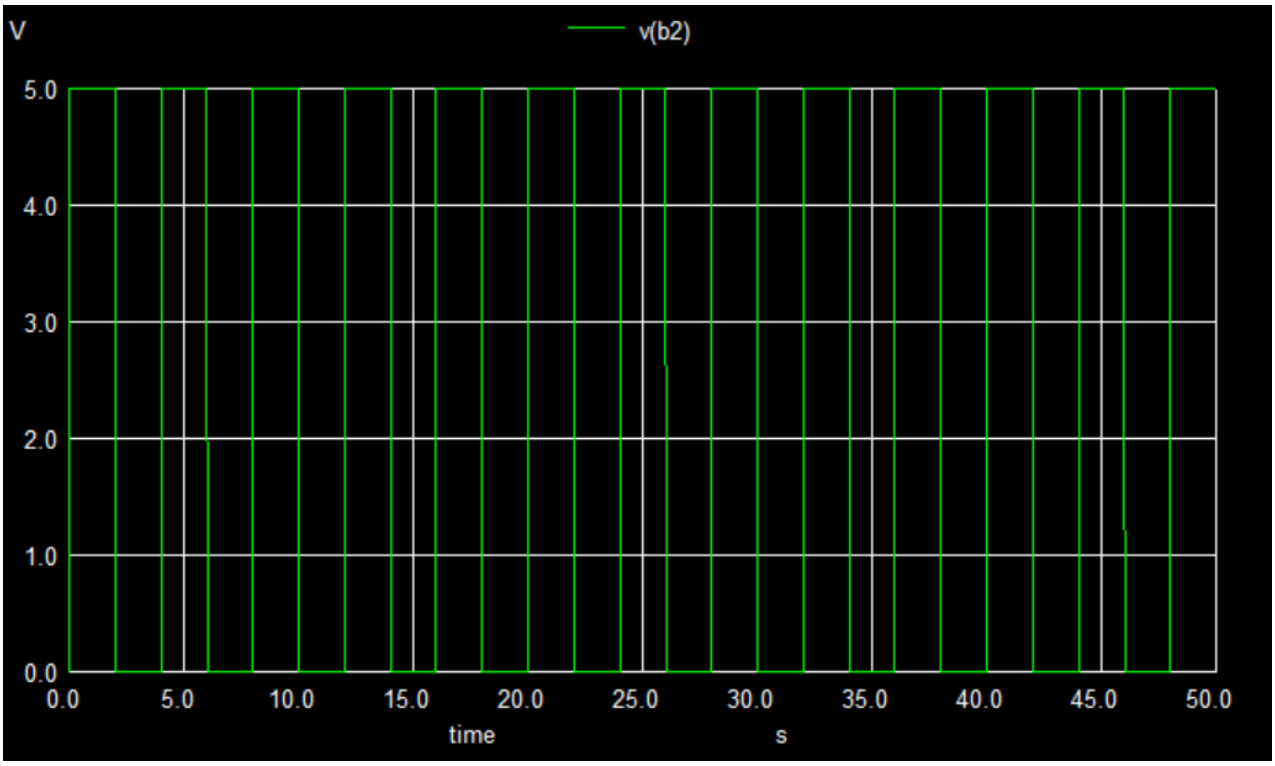
Ngspice plot of A1



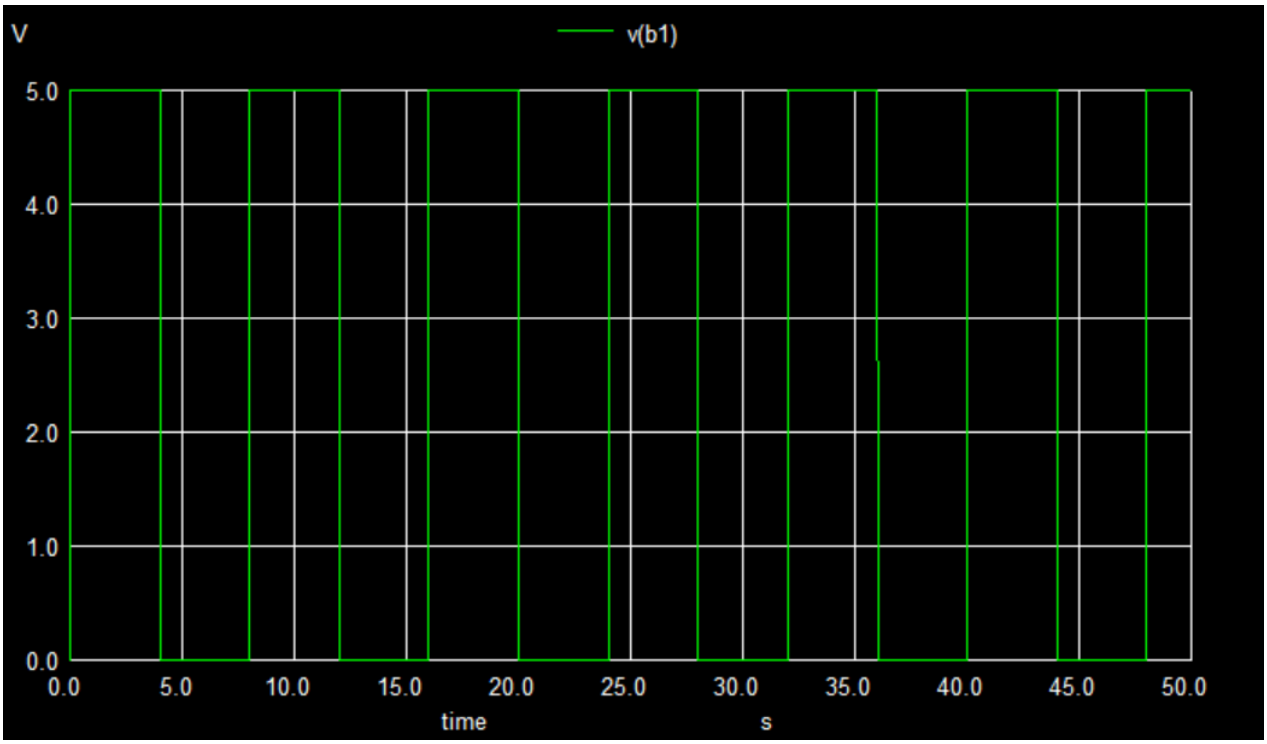
Ngspice plot of A0



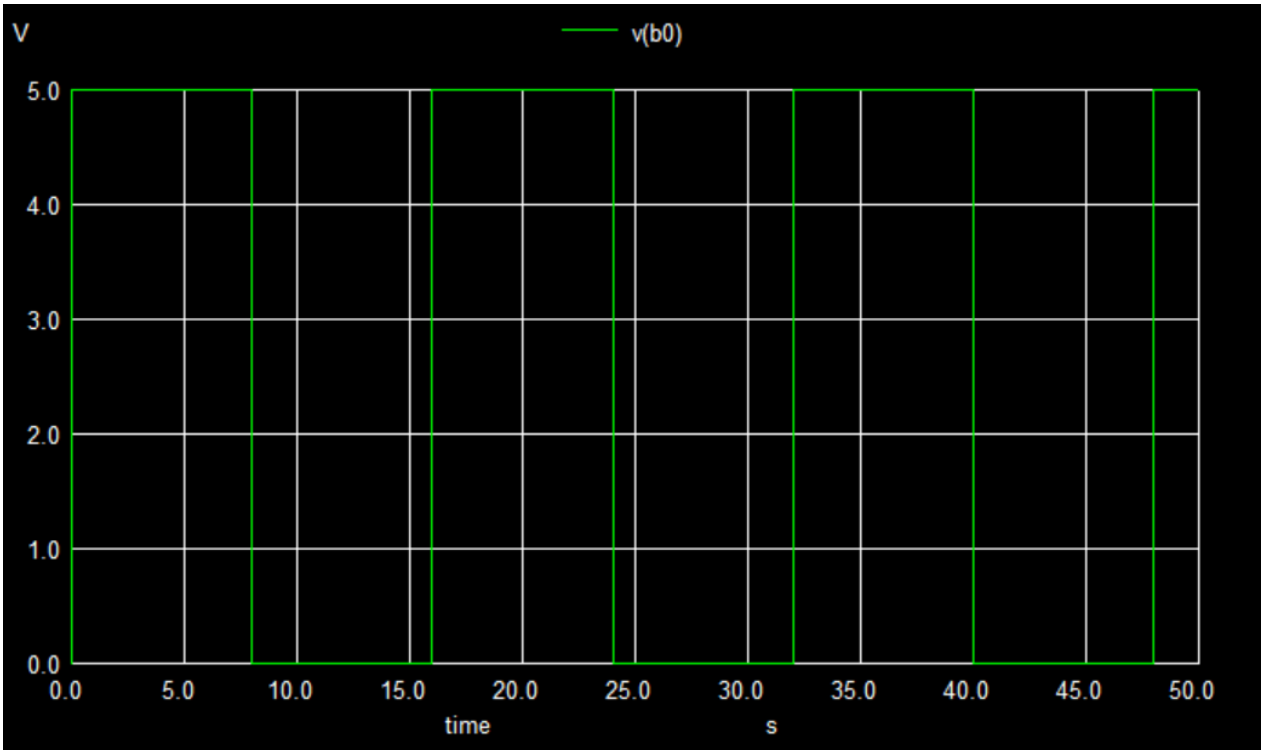
Ngspice plot of B3



Ngspice plot of B2

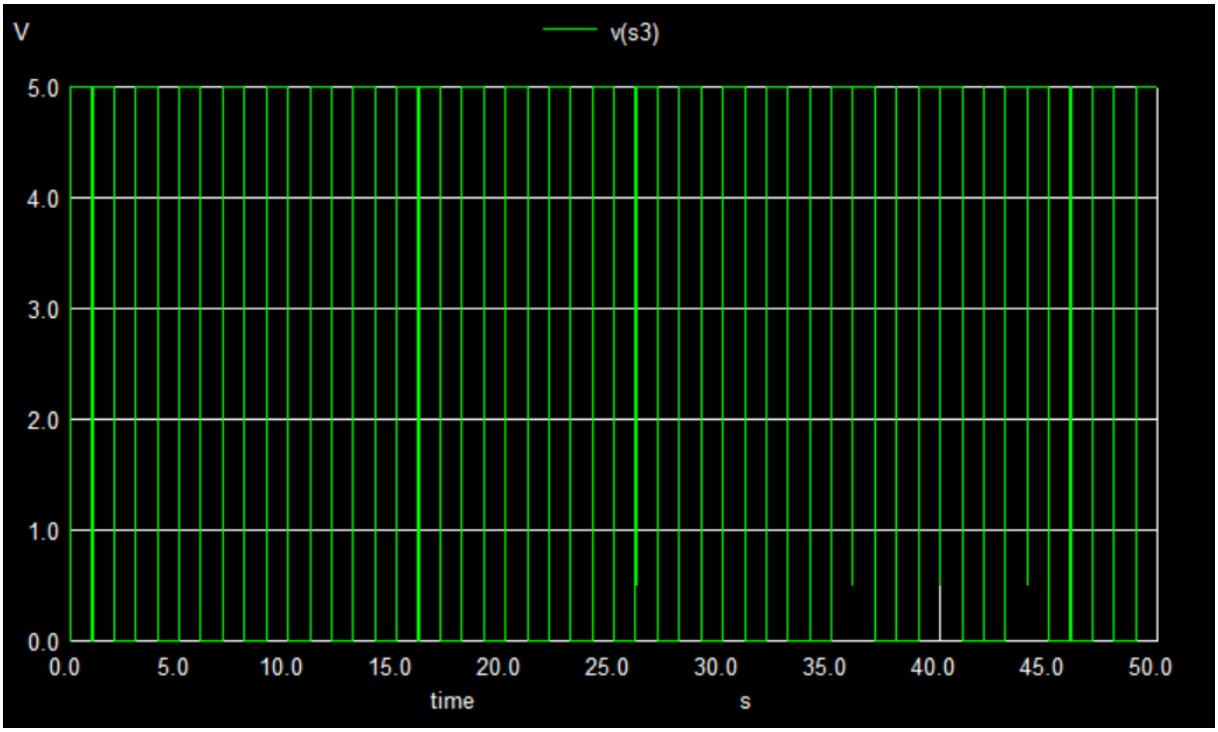


Ngspice plot of B1

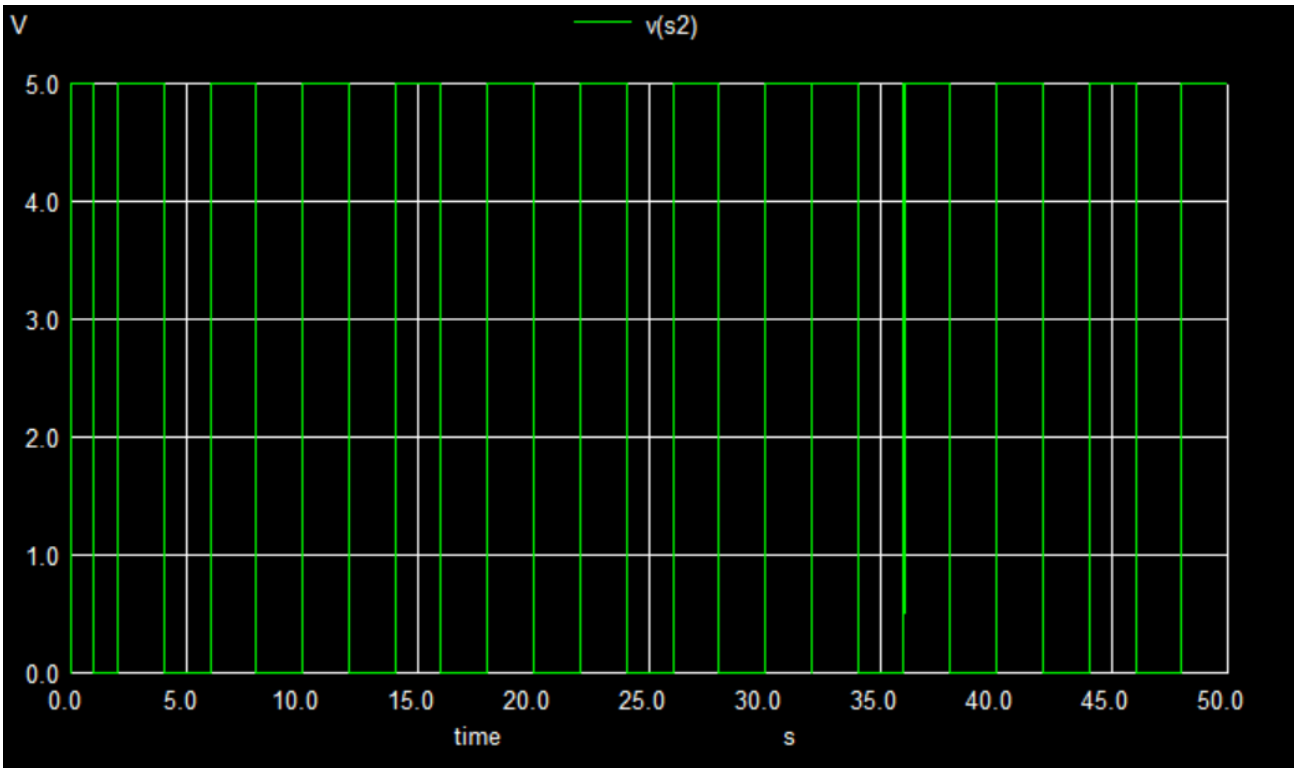


Ngspice plot of B0

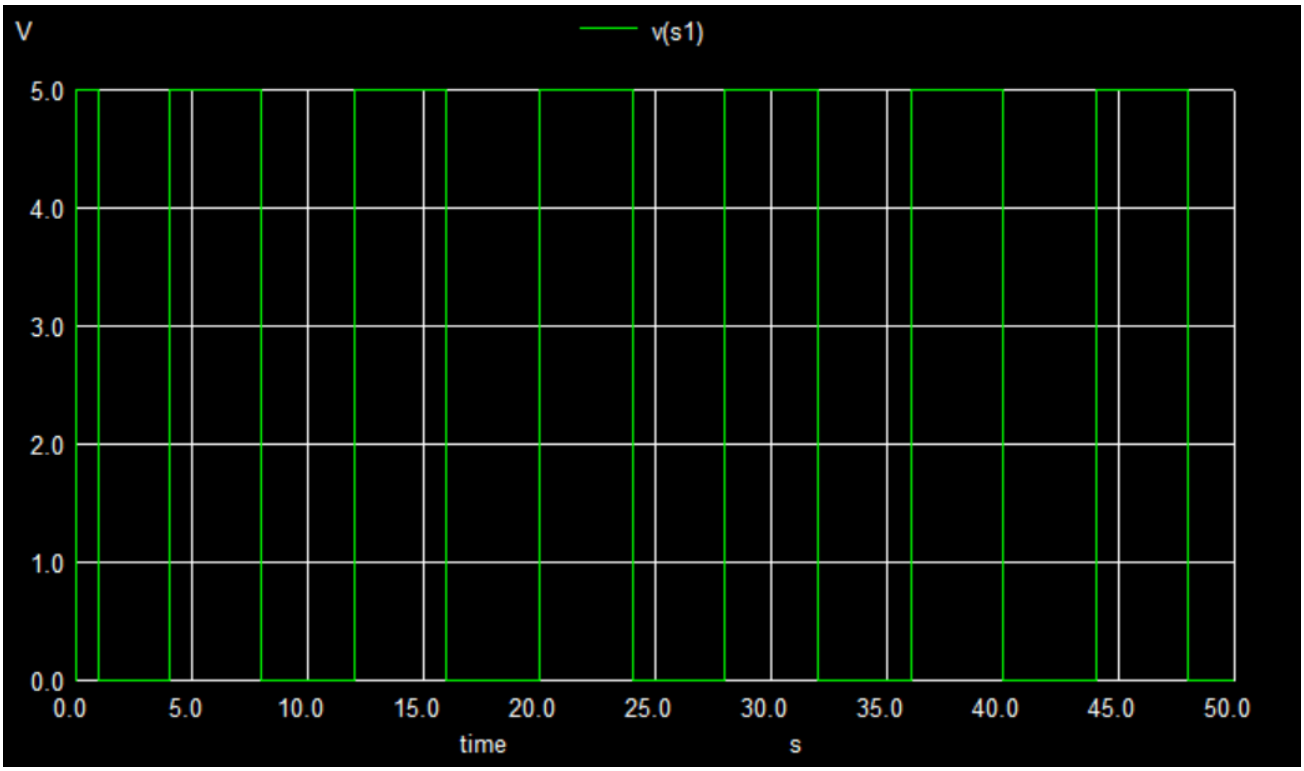
- Outputs:



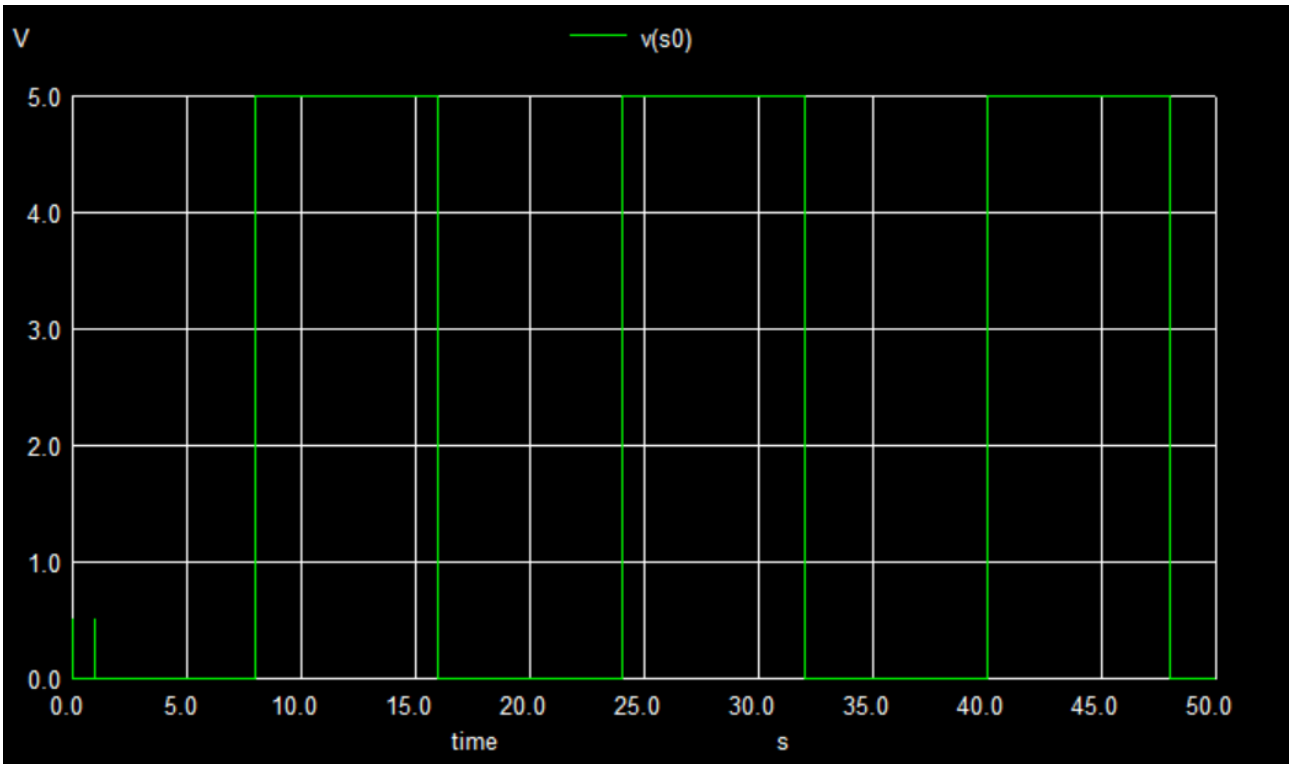
Ngspice plot of S3



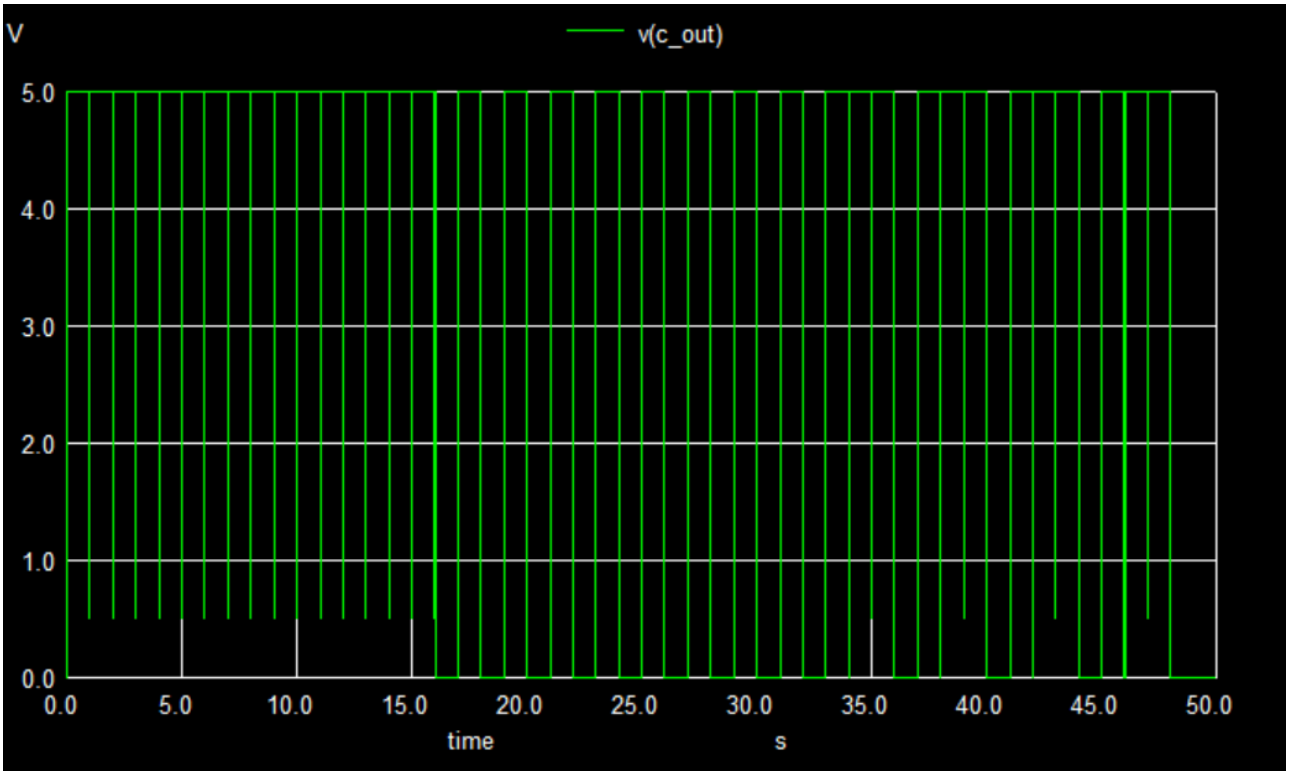
Ngspice plot of S2



Ngspice plot of S1

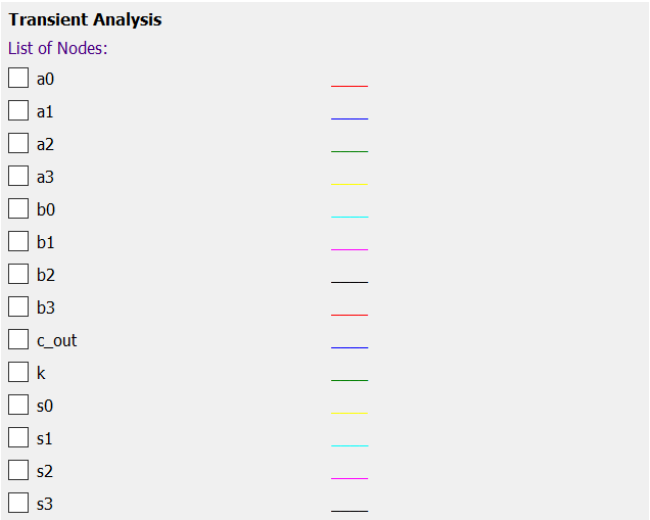


Ngspice plot of S0

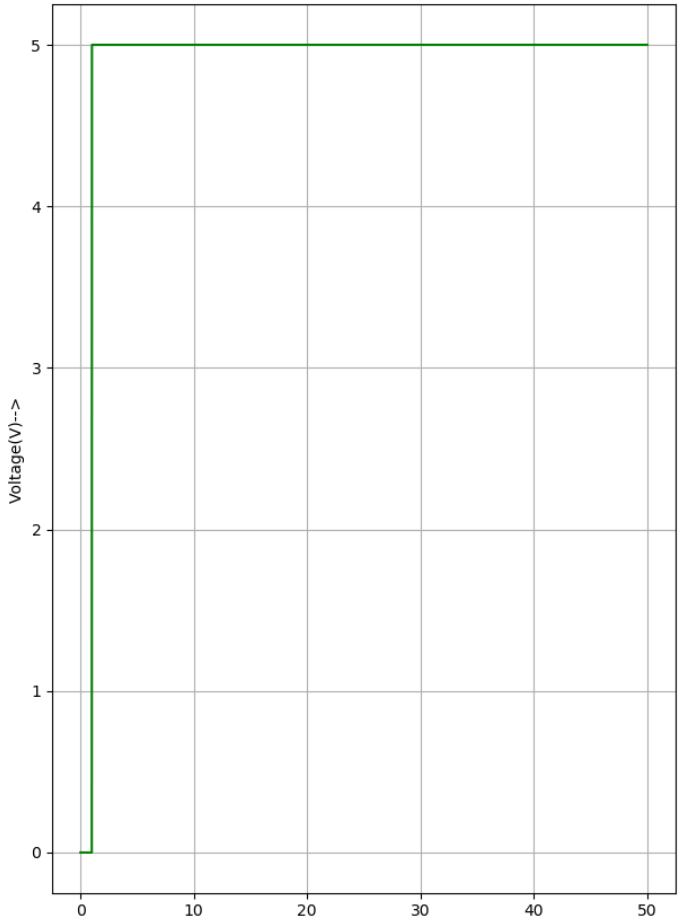


Ngspice plot of c_out

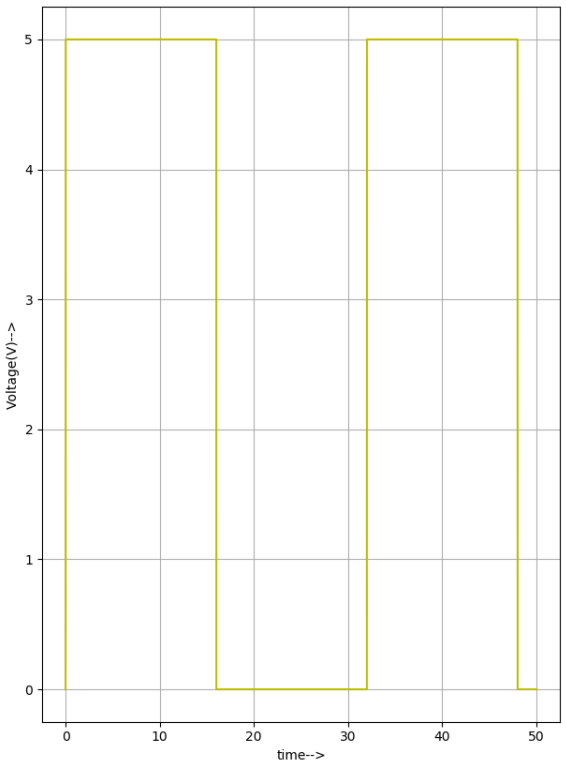
II. PYTHON PLOTS:



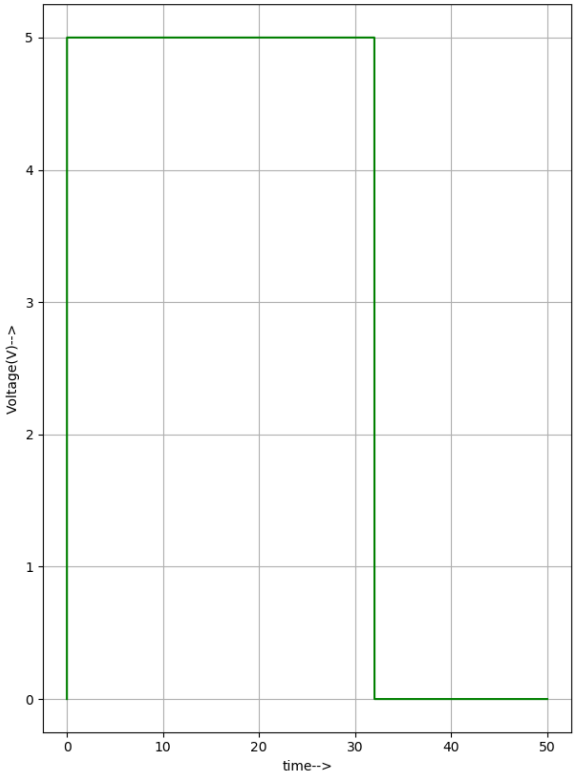
- Inputs:



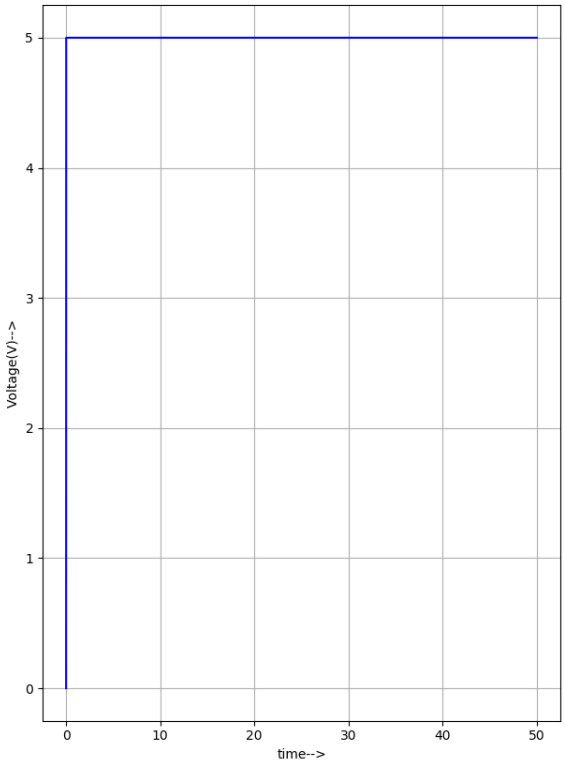
Python plot of K



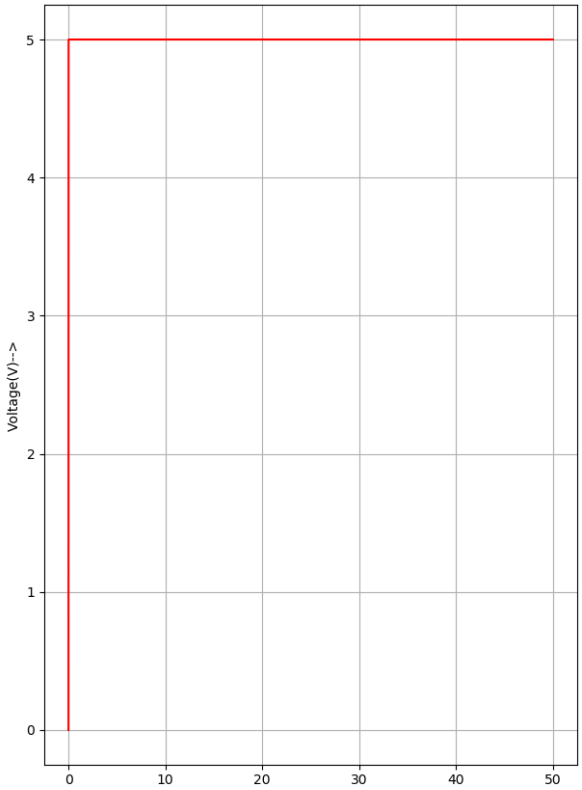
Python plot of A3



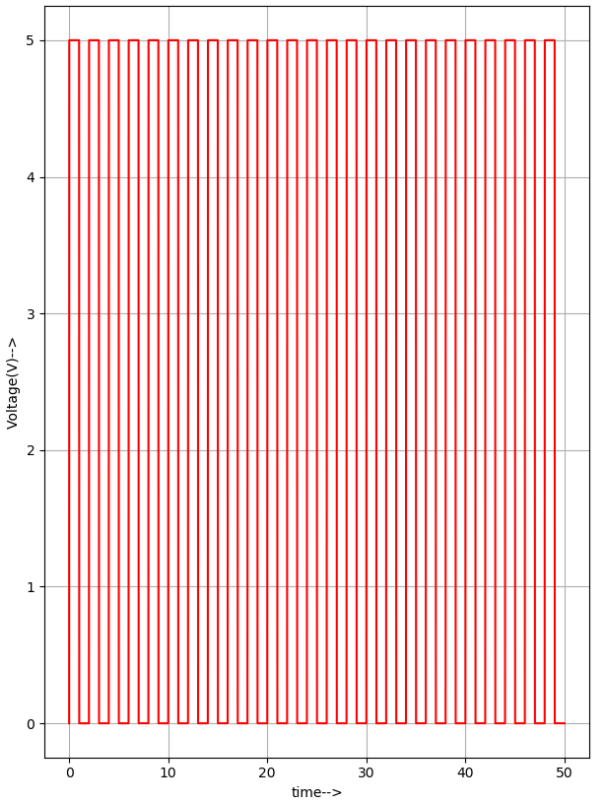
Python plot of A2



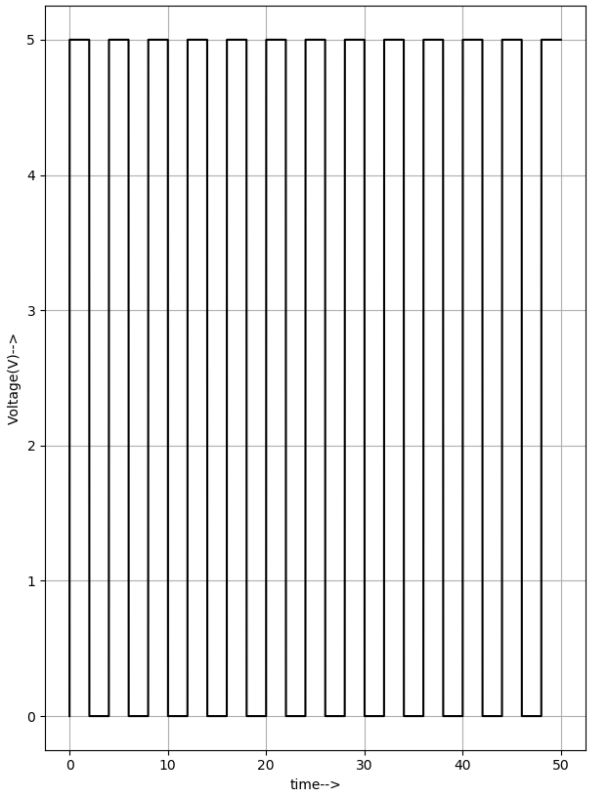
Python plot of A1



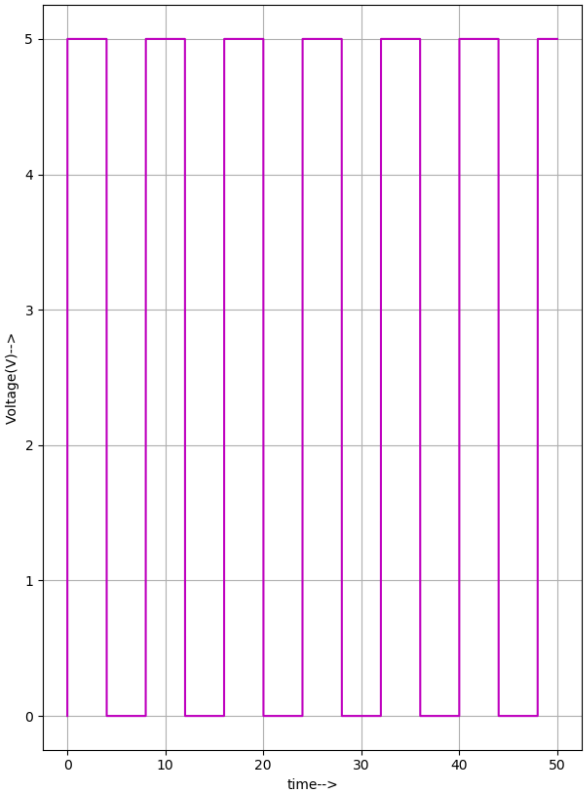
Python plot of A0



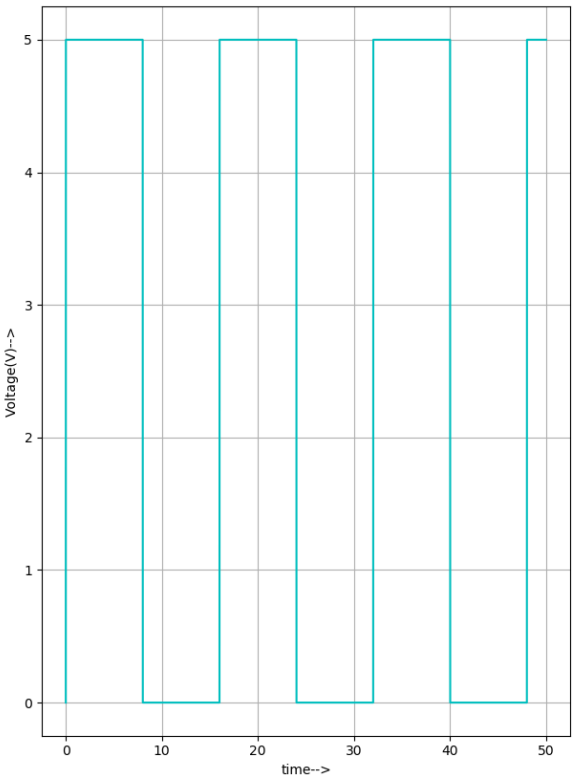
Python plot of B3



Python plot of B2

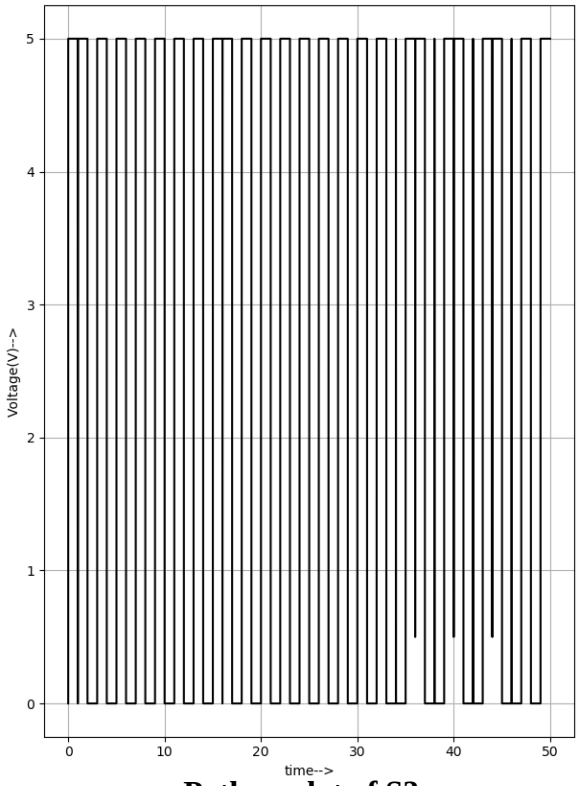


Python plot of B1

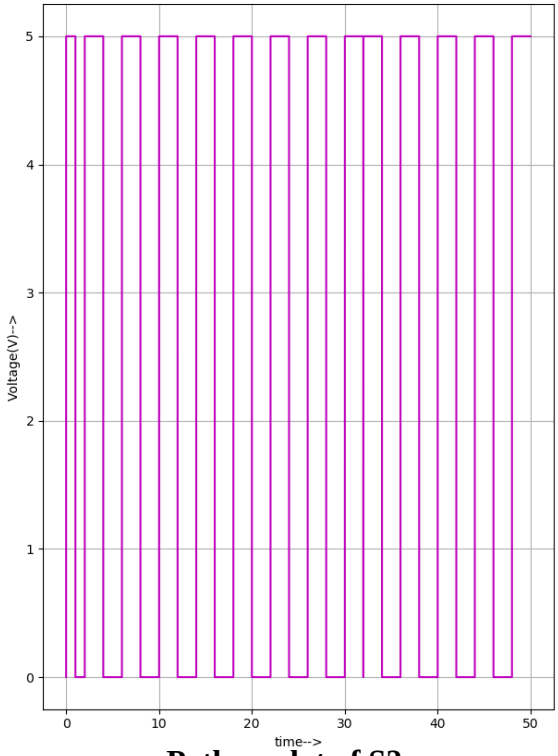


Python plot of B0

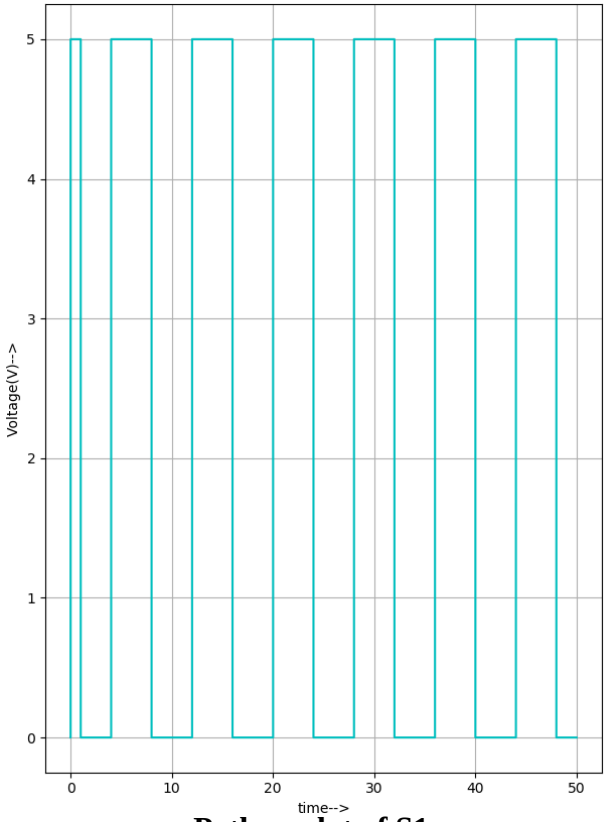
- Outputs:



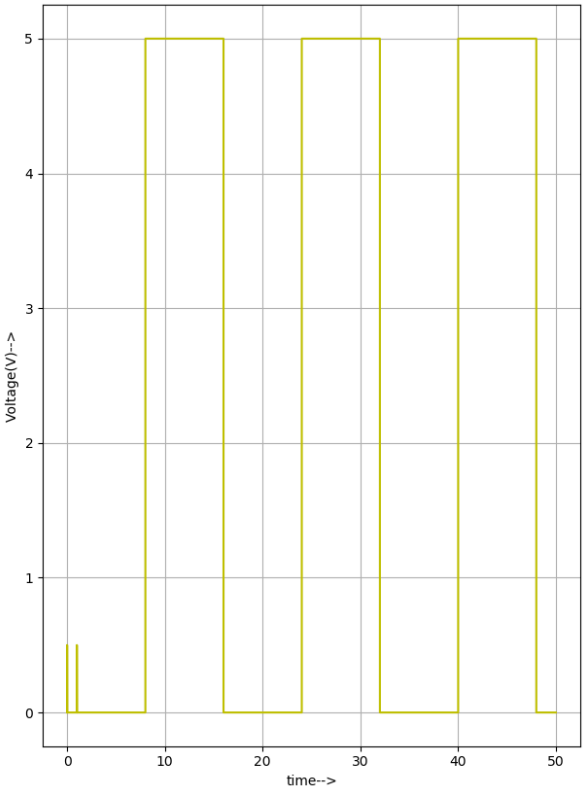
Python plot of S3



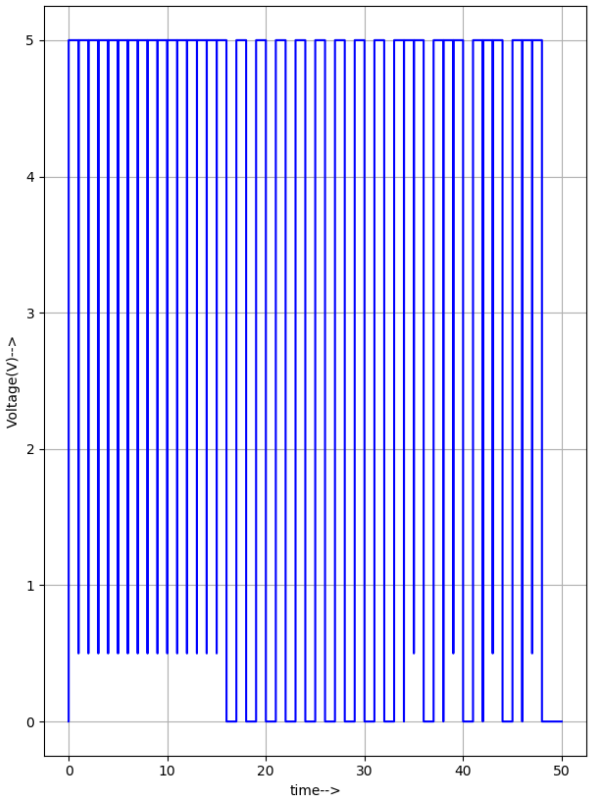
Python plot of S2



Python plot of S1



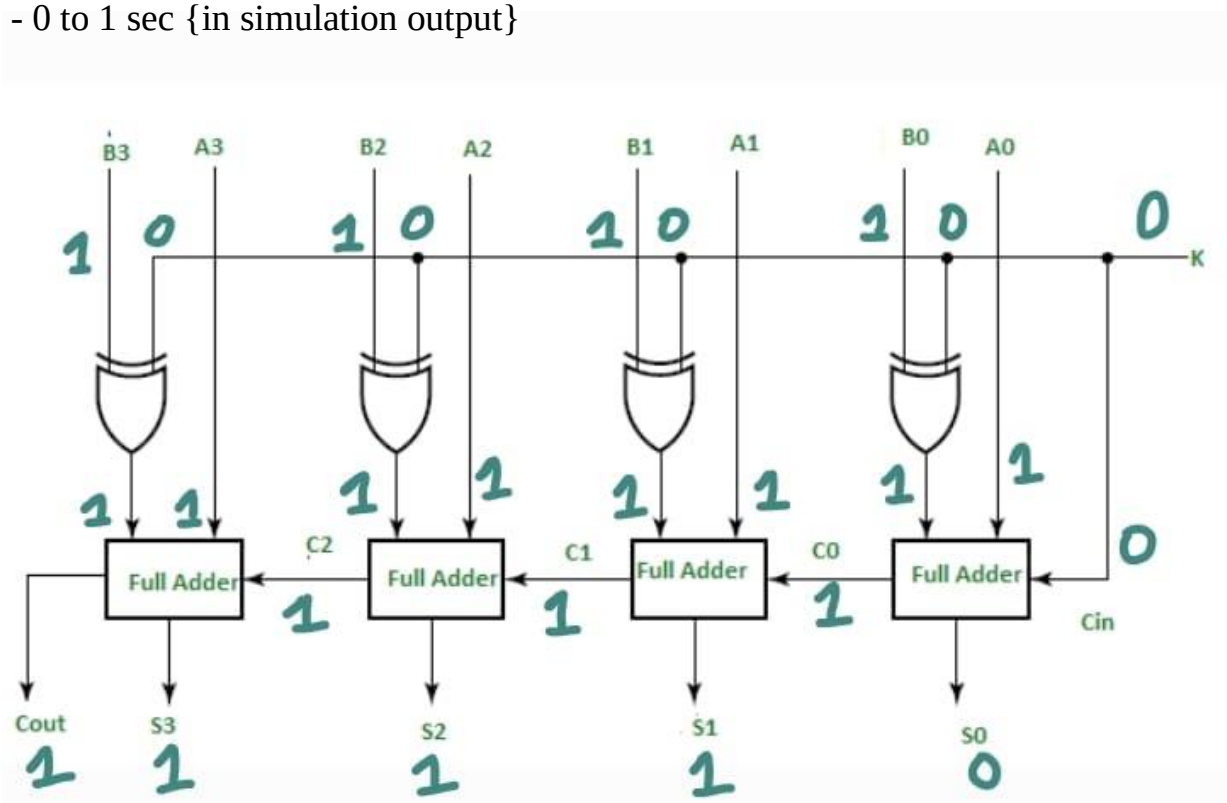
Python plot of S0



Python plot of c_out

Verification of Outputs:

time - 0 to 1 sec {in simulation output}



Logical verification:

A3	A2	A1	A0
1	1	1	1

B3	B2	B1	B0
1	1	1	1

K = 0 ⇒ Addition

1 1 1 → Cin

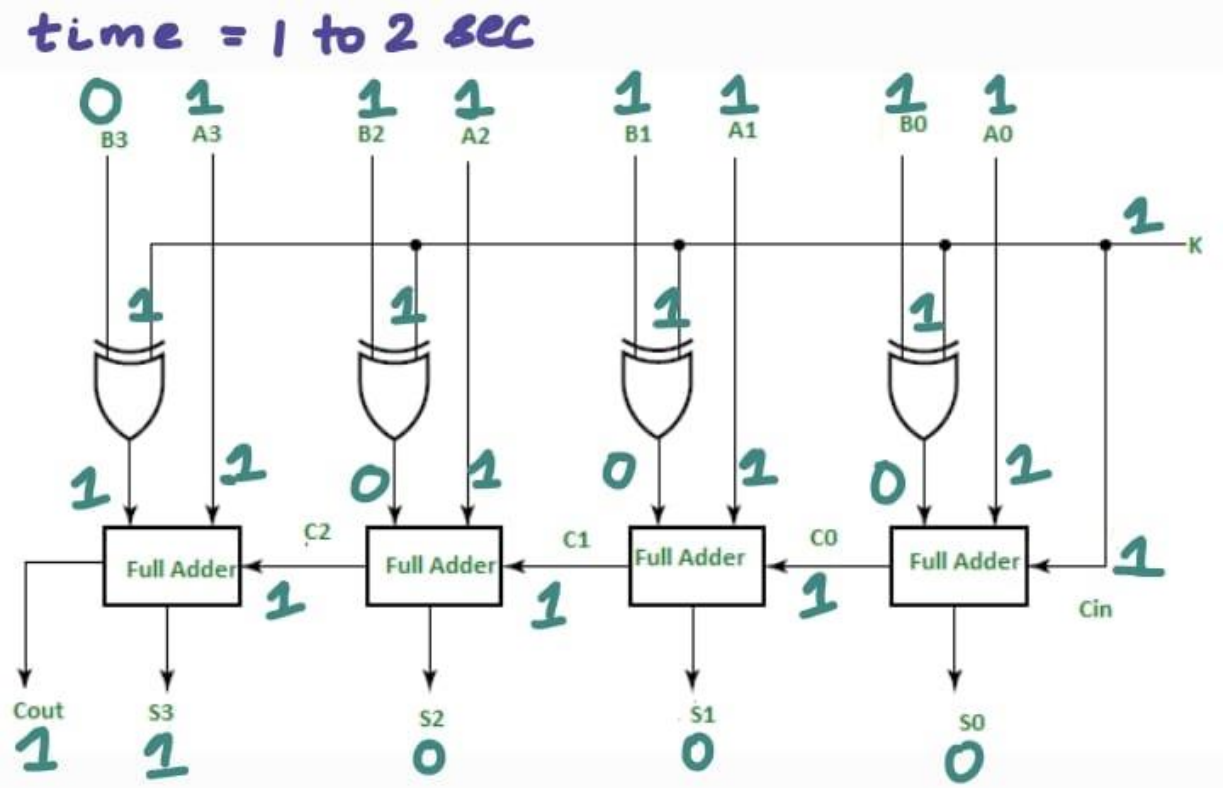
A = 1 1 1 1

B = 1 1 1 1

1 1 1 1 0 ⇒ sum (S3 S2 S1 S0)

1 → carry out

time - 1 to 2 sec {in simulation output}



Logical verification:

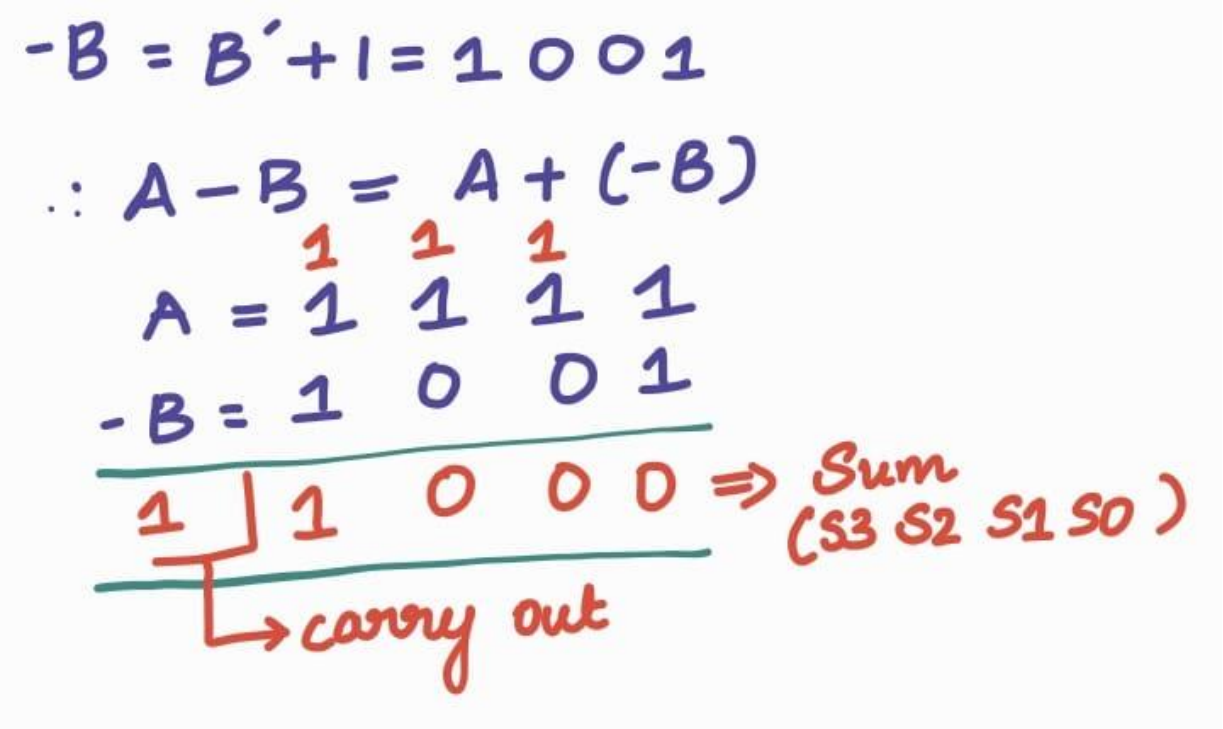
A3	A2	A1	A0	B3	B2	B1	B0
1	1	1	1	0	1	1	1

K=1 ⇒ Subtraction

A-B = A + 2's complement (B)

B = 0 1 1 1

B' = 1 0 0 0



Result:

Thus, a 4-Bit Binary Adder Subtractor Circuit in has been designed using Full Adder Subcircuit in eSim. The outputs have also been verified.

References:

https://www.geeksforgeeks.org/4-bit-binary-adder-subtractor/

https://youtu.be/4q68_3Bs5O4

https://youtu.be/tWPWBvExh1Q