SARDAR VALLABHBHAI NATIONAL INSTITUTE OF TECHNOLOGY, SURAT
ELECTRONICS ENGINEERING DEPARTMENT

Expt. No:	12	Multiplexers and Code Converters
Date:	11-11-2020	

AIM: To study, design and implement:

1. Binary to Gray and Gray to Binary Code Converter (2-Bit, 3-Bit and 4-Bit)
2. Multiplexer using basic Gates (2x1 and 4x1)
3. Realise all the basic gates using 2x1 Multiplexer
4. Function Implementation using Multiplexers

SOFTWARE TOOLS / OTHER REQUIREMENTS:

1. Multisim Simulator/Circuit Simulator

THEORY:

Codes and code converters:

Coding is the process of translating the input information which can be understandable by the machine or a particular device.

Coding can be used for security purpose to protect the information from steeling or interrupting.

Actually this is not the latest trend, in previous days also the king of the kingdom used to send the information to other kingdom which some code words.

The code converters are used to convert the information in to the code which we want. These are basically encoders and decoders which converts the data in to an encoded form. The below explains some digital codes used in digital electronics.

Excess-3 code:

It is also known as self-complementary code as the complement of any number (0-9) will be available within these 10 numbers. As the name implies it is excess of 3 for the regular BCD code i.e. if u add 3(0011) to the BCD Addition u can get Excess-3 code.

Gray Code:

Gray code – also known as **Cyclic Code**, **Reflected Binary Code** (RBC), **Reflected Binary** (RB) or **Grey code** – is defined as an ordering of the binary number system such that each incremental value can only differ by one bit. In gray code, while traversing from one step to another step only one bit in the code group changes. That is to say that two adjacent code numbers differ from each other by only one bit.

Binary to Gray Code Converter:

The logical circuit which converts the binary code to equivalent gray code is known as **binary to gray code converter**. An n-bit gray code can be obtained by reflecting an n-1 bit code about an axis after 2^{n-1} rows and putting the MSB (Most Significant Bit) of 0 above the axis and the MSB of 1 below the axis. Reflection of Gray codes is shown below.

The 4 bit binary to gray code conversion table is given below:

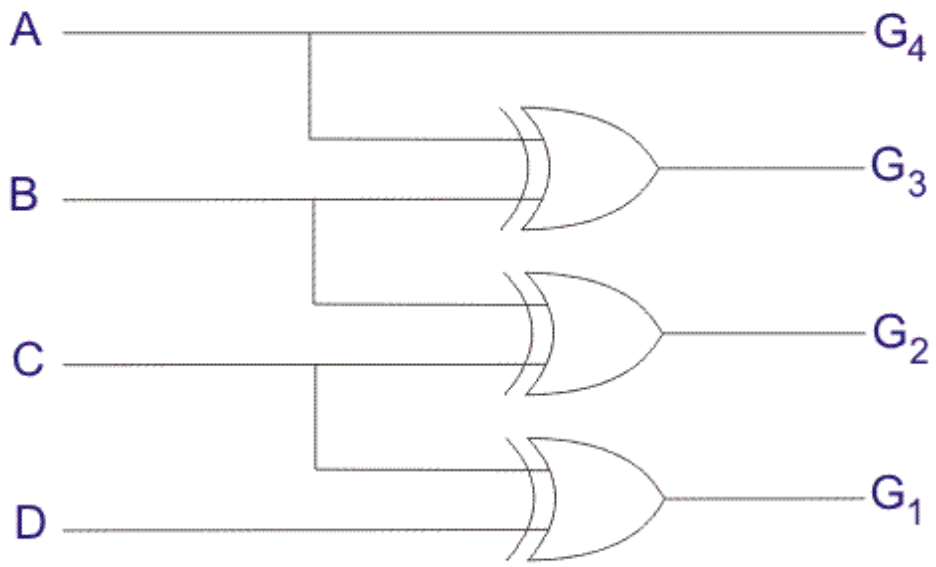
Decimal Number	4 bit Binary Number ABCD	4 bit Gray Code $G_1G_2G_3G_4$
0	0 0 0 0	0 0 0 0
1	0 0 0 1	0 0 0 1
2	0 0 1 0	0 0 1 1
3	0 0 1 1	0 0 1 0
4	0 1 0 0	0 1 1 0
5	0 1 0 1	0 1 1 1
6	0 1 1 0	0 1 0 1
7	0 1 1 1	0 1 0 0
8	1 0 0 0	1 1 0 0
9	1 0 0 1	1 1 0 1
10	1 0 1 0	1 1 1 1
11	1 0 1 1	1 1 1 0
12	1 1 0 0	1 0 1 0
13	1 1 0 1	1 0 1 1
14	1 1 1 0	1 0 0 1
15	1 1 1 1	1 0 0 0

How to Convert Binary to Gray Code:

1. The MSB (Most Significant Bit) of the gray code will be exactly equal to the first bit of the given binary number.

2. The second bit of the code will be exclusive-or (XOR) of the first and second bit of the given binary number, i.e if both the bits are same the result will be 0 and if they are different the result will be 1.

3. The third bit of gray code will be equal to the exclusive-or (XOR) of the second and third bit of the given binary number. Thus the binary to gray code conversion goes on. An example is given below to illustrate these steps.



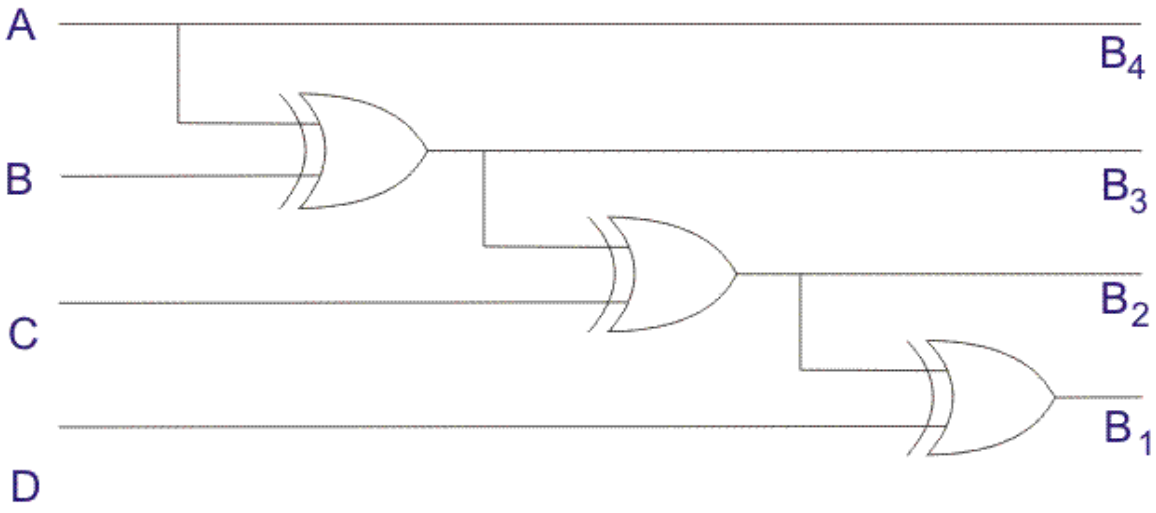
Logic Circuit for Binary to Gray Code Converter

Gray to Binary Code Converter:

In a gray to binary code converter, the input is gray code and output is its equivalent binary code.

4 bit Gray Code	4 bit Binary Code
A B C D	B_4 B_3 B_2 B_1
0 0 0 0	0 0 0 0
0 0 0 1	0 0 0 1
0 0 1 1	0 0 1 0
0 0 1 0	0 0 1 1
0 1 1 0	0 1 0 0
0 1 1 1	0 1 0 1
0 1 0 1	0 1 1 0
0 1 0 0	0 1 1 1
1 1 0 0	1 0 0 0
1 1 0 1	1 0 0 1
1 1 1 1	1 0 1 0
1 1 1 0	1 0 1 1
1 0 1 0	1 1 0 0
1 0 1 1	1 1 0 1
1 0 0 1	1 1 1 0
1 0 0 0	1 1 1 1

The gray code to binary converter circuit is shown below:



Logic Circuit for Gray to Binary Code Converter

Gray Code to Binary Conversion:

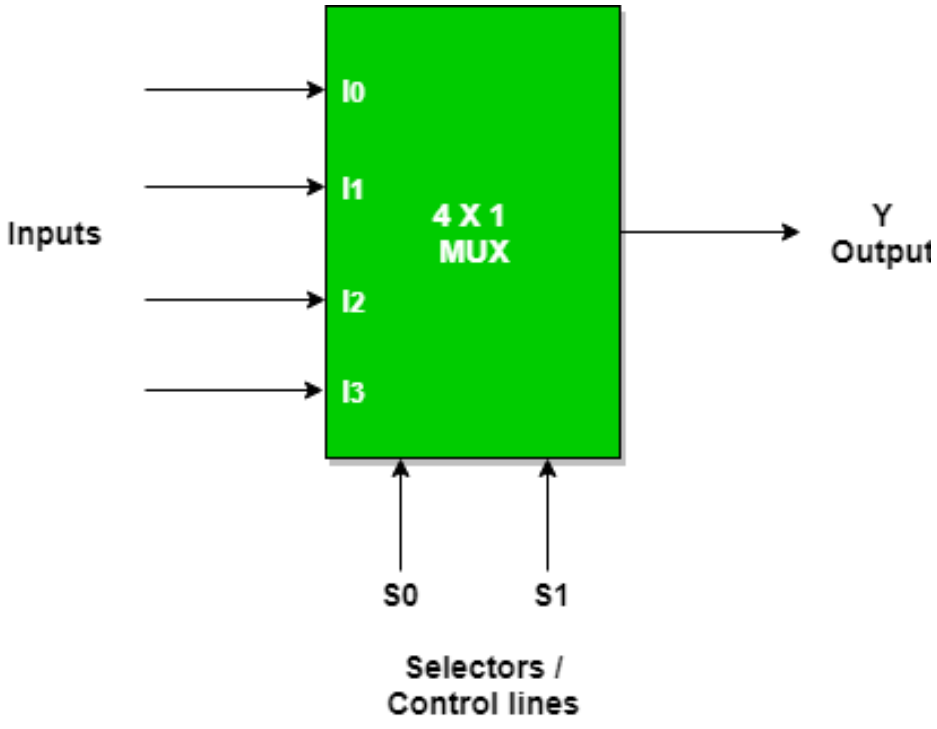
1. The MSB of the binary number will be equal to the MSB of the given gray code.

2. Now if the second gray bit is 0, then the second binary bit will be the same as the previous or the first bit. If the gray bit is 1 the second binary bit will alter. If it was 1 it will be 0 and if it was 0 it will be 1.

3. This step is continued for all the bits to do Gray code to binary conversion.

Multiplexers in Digital Logic:

It is a combinational circuit which have many data inputs and single output depending on control or select inputs.
For N input lines, log n (base2) selection lines, or we can say that for 2^n input lines, n selection lines are required.
Multiplexers are also known as "Data n selector, parallel to serial convertor, many to one circuit, universal logic circuit".
Multiplexers are mainly used to increase amount of the data that can be sent over the network within certain amount of time and bandwidth.

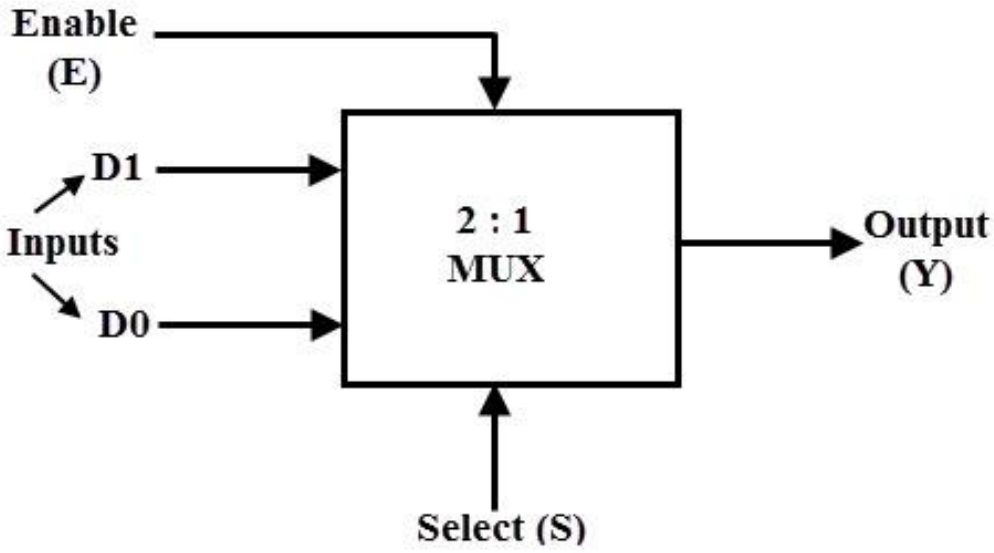


2 to 1 Multiplexer:

2 to 1 means that this multiplexer has 2 input channels and 1 output. 2 channels mean it has 1 control signal.

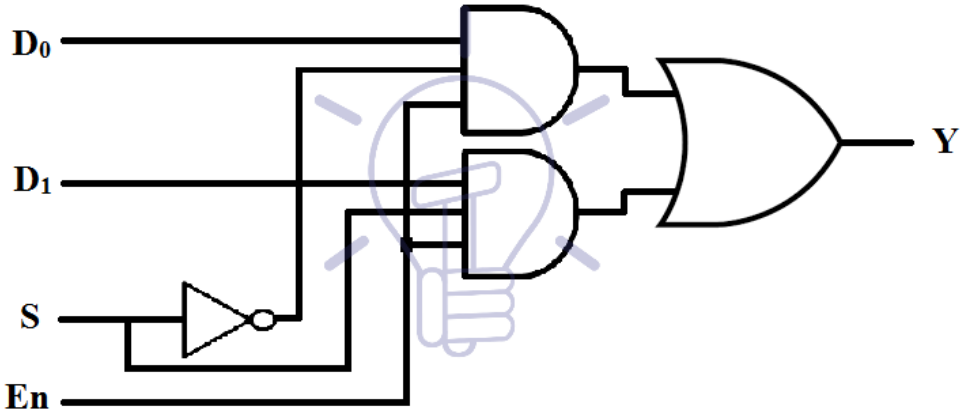
When the control signal is "0", the first channel is selected and the2nd channel is selected when the control signal is "1".

There is also an Enable bit used for enabling/disabling the circuit. When enable is high, MUX is enabled. When Enable pin is Low, MUX is disabled.



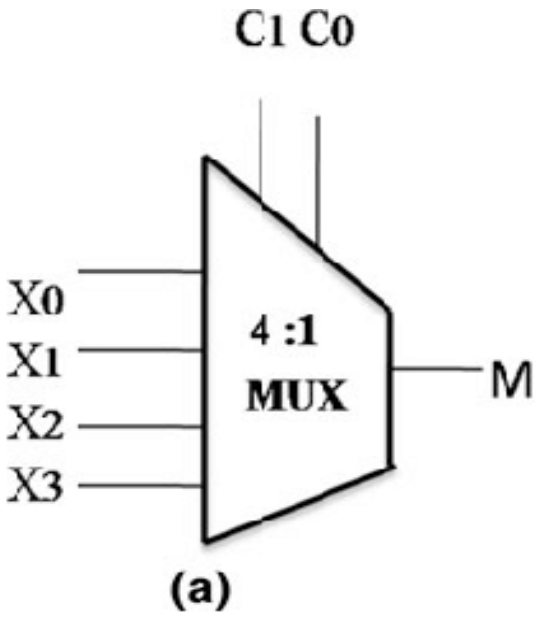
Select	Inputs		Output
0	0	0	0
0	0	1	1
1	1	0	1
1	1	1	1

A MUX need AND gates equal to the number of input channels, NOT gates equal to the number of Control signals and a single OR gate.
Implantation of Multiplexer using logic gates is given below.



4 to 1 Multiplexer:

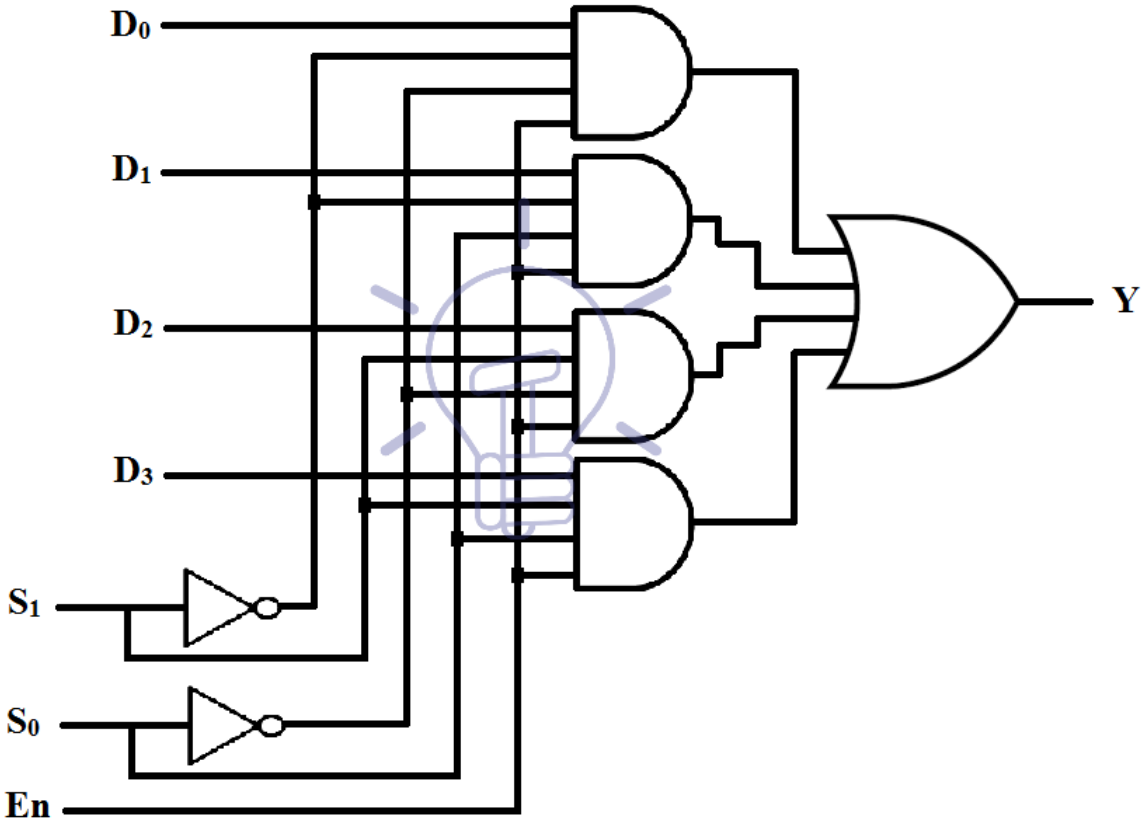
This multiplexer has 4 input channels, 1 output, and 2 control signals. Each binary combination of control signal will select one out of four input channels.



C1	C0	M
0	0	X0
0	1	X1
1	0	X2
1	1	X3

(b)

4 to 1 multiplexer implementation using logic gates is shown in the figure given below.



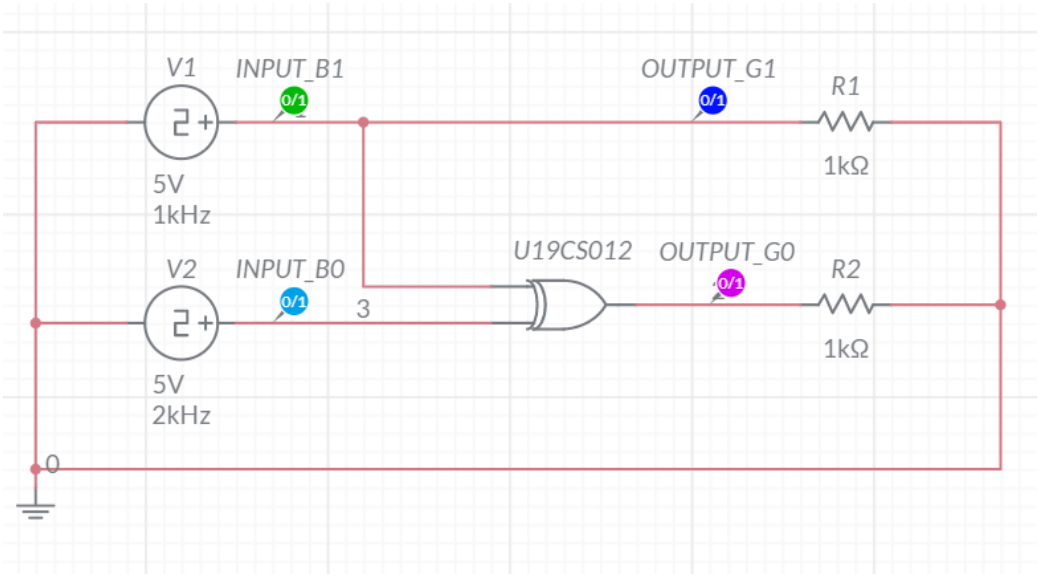
Application of Multiplexer:

In all types of digital system applications, multiplexers find its immense usage. Since these allows multiple inputs to be connected independently to a single output, these are found in variety of applications including data routing, logic function generators, control sequencers, parallel-to-serial converters, etc.

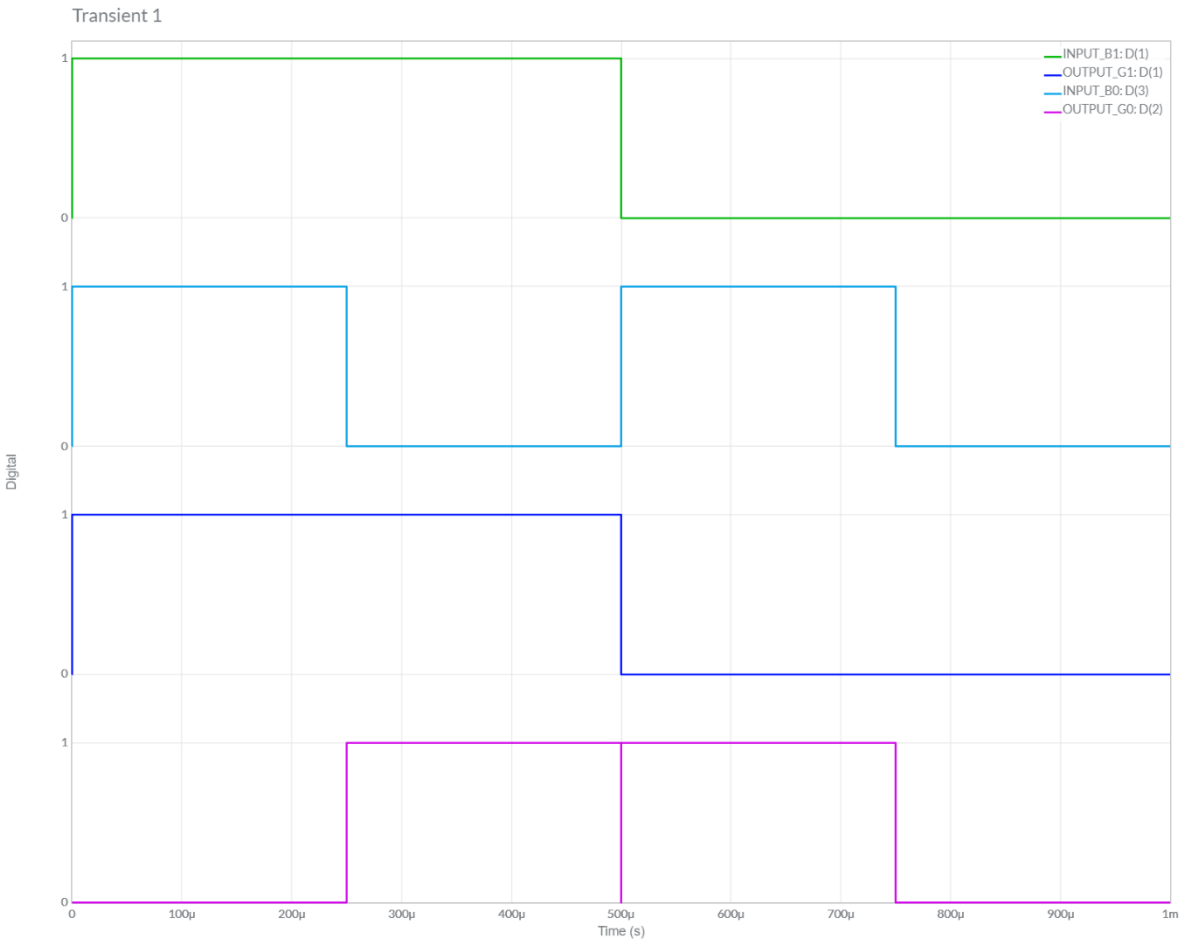
SIMULATION SCREENSHOTS

2 – BIT Binary to Gray code converter

CIRCUIT/CONNECTION DIAGRAMS (FROM MULTISIM)

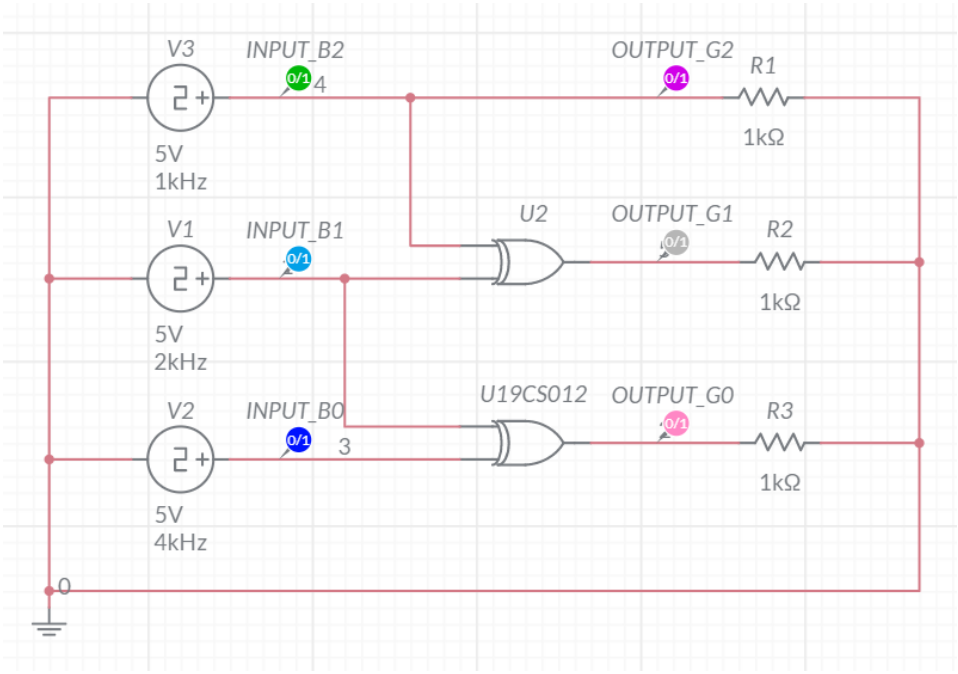


WAVEFORMS (FROM MULTISIM)

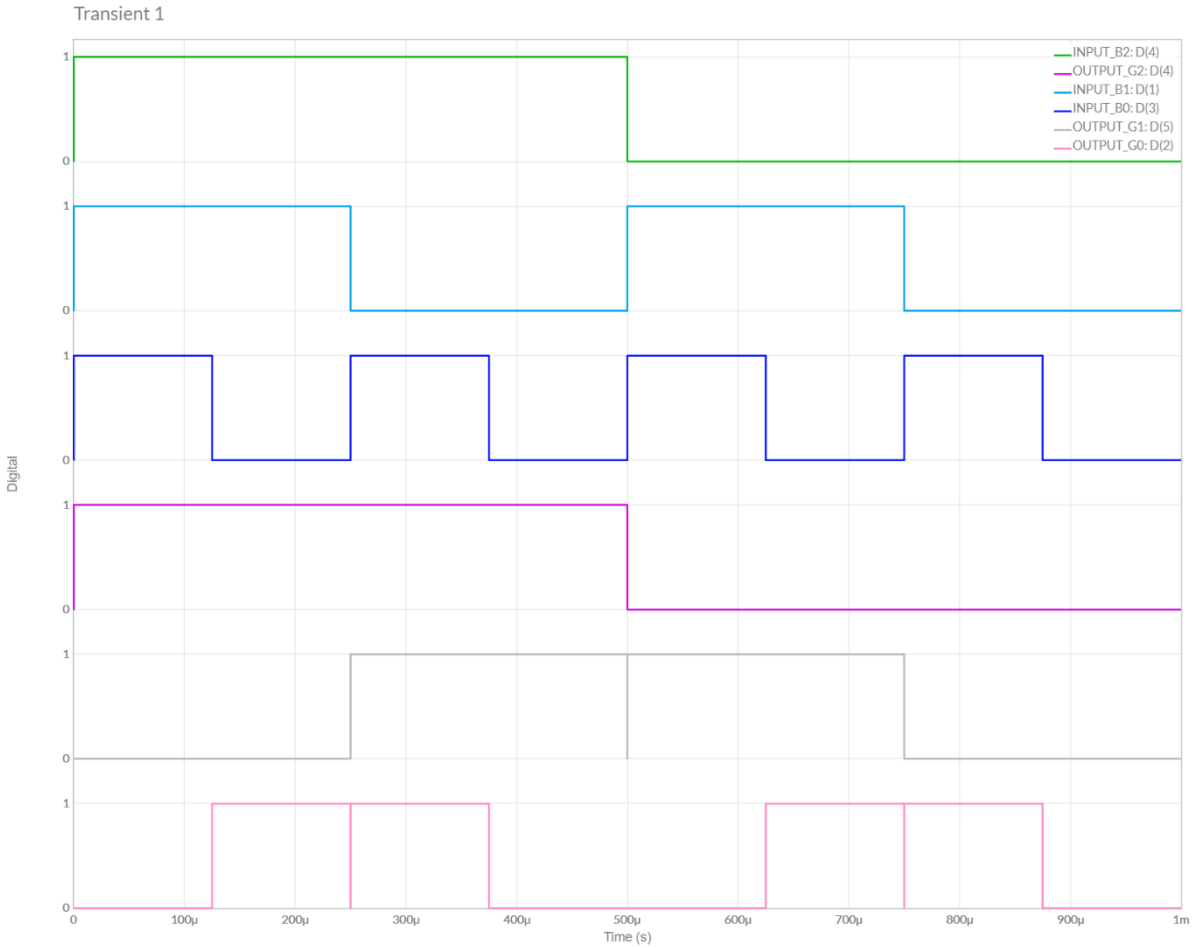


3 – BIT Binary to Gray code converter

CIRCUIT/CONNECTION DIAGRAMS (FROM MULTISIM)

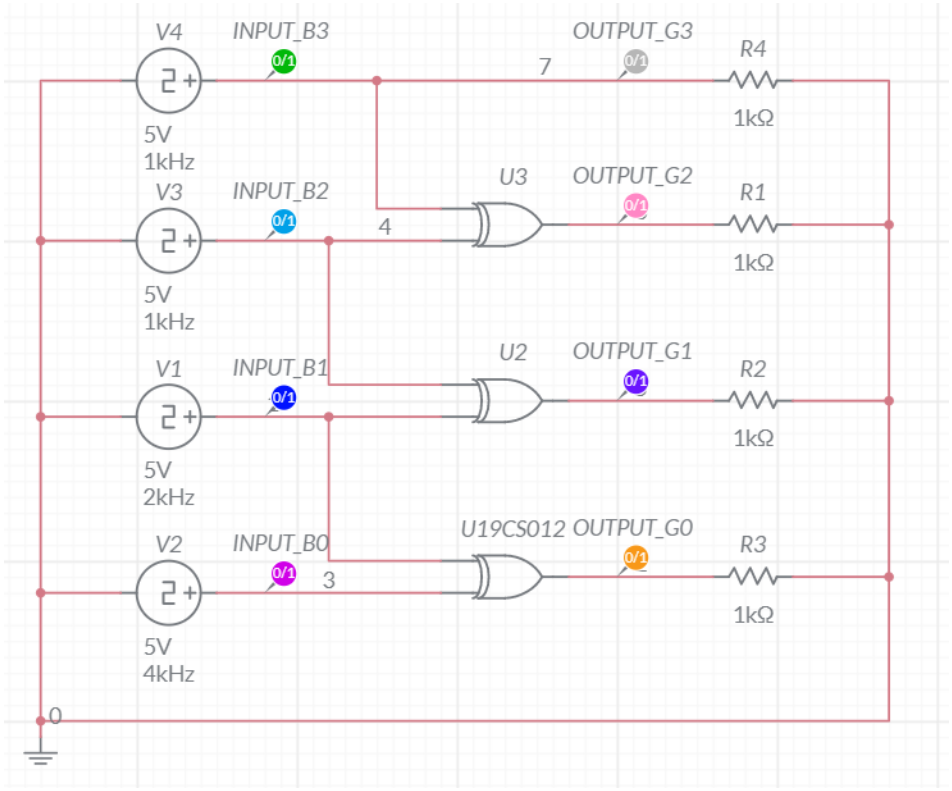


WAVEFORMS (FROM MULTISIM)

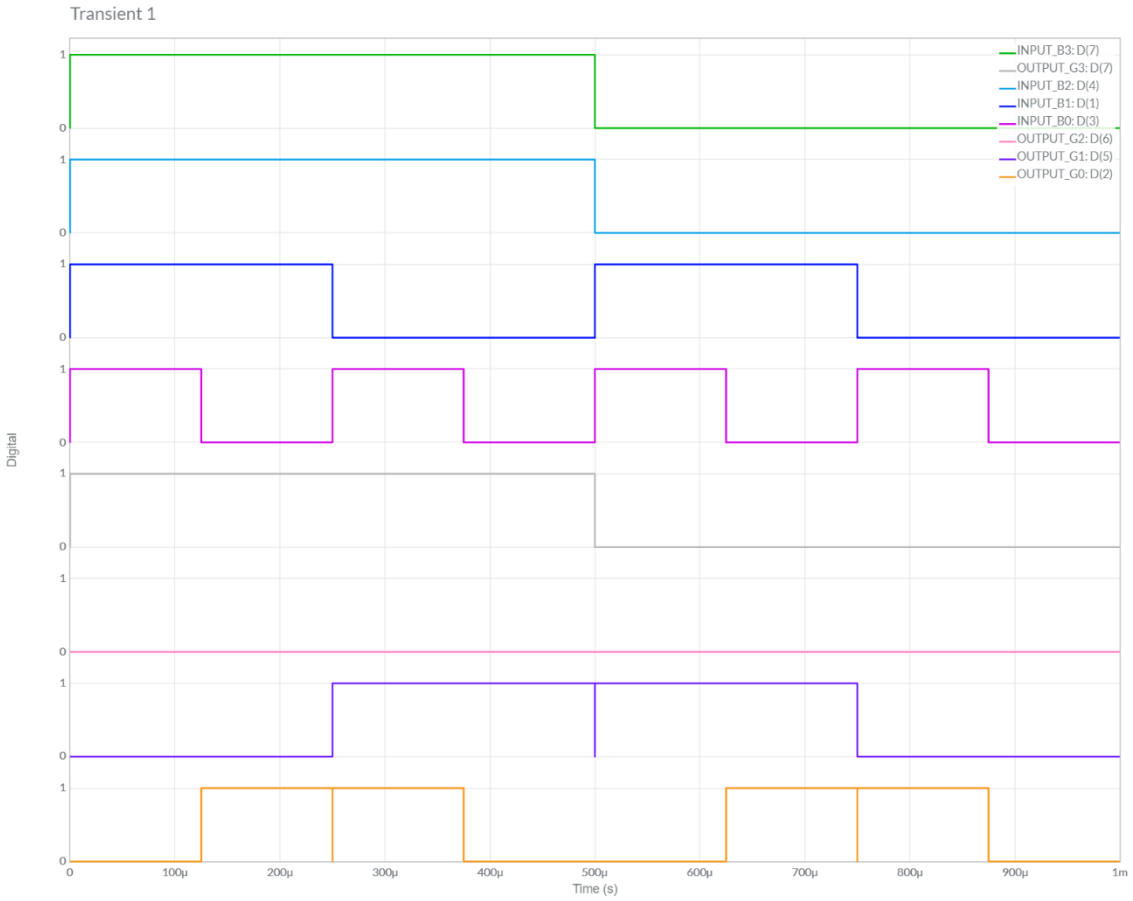


4 – BIT Binary to Gray code converter

CIRCUIT/CONNECTION DIAGRAMS (FROM MULTISIM)

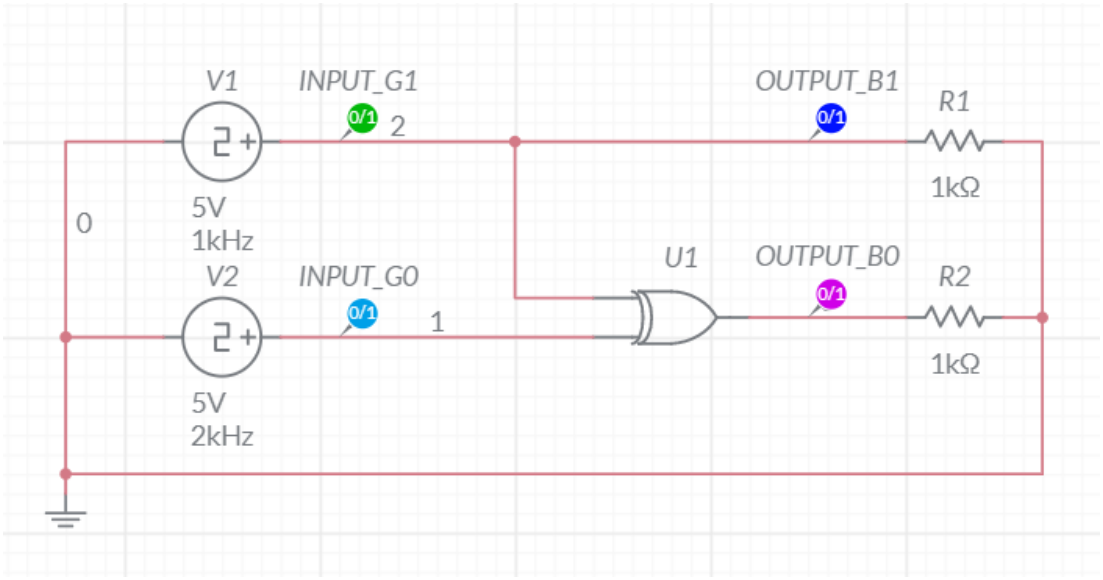


WAVEFORMS (FROM MULTISIM)

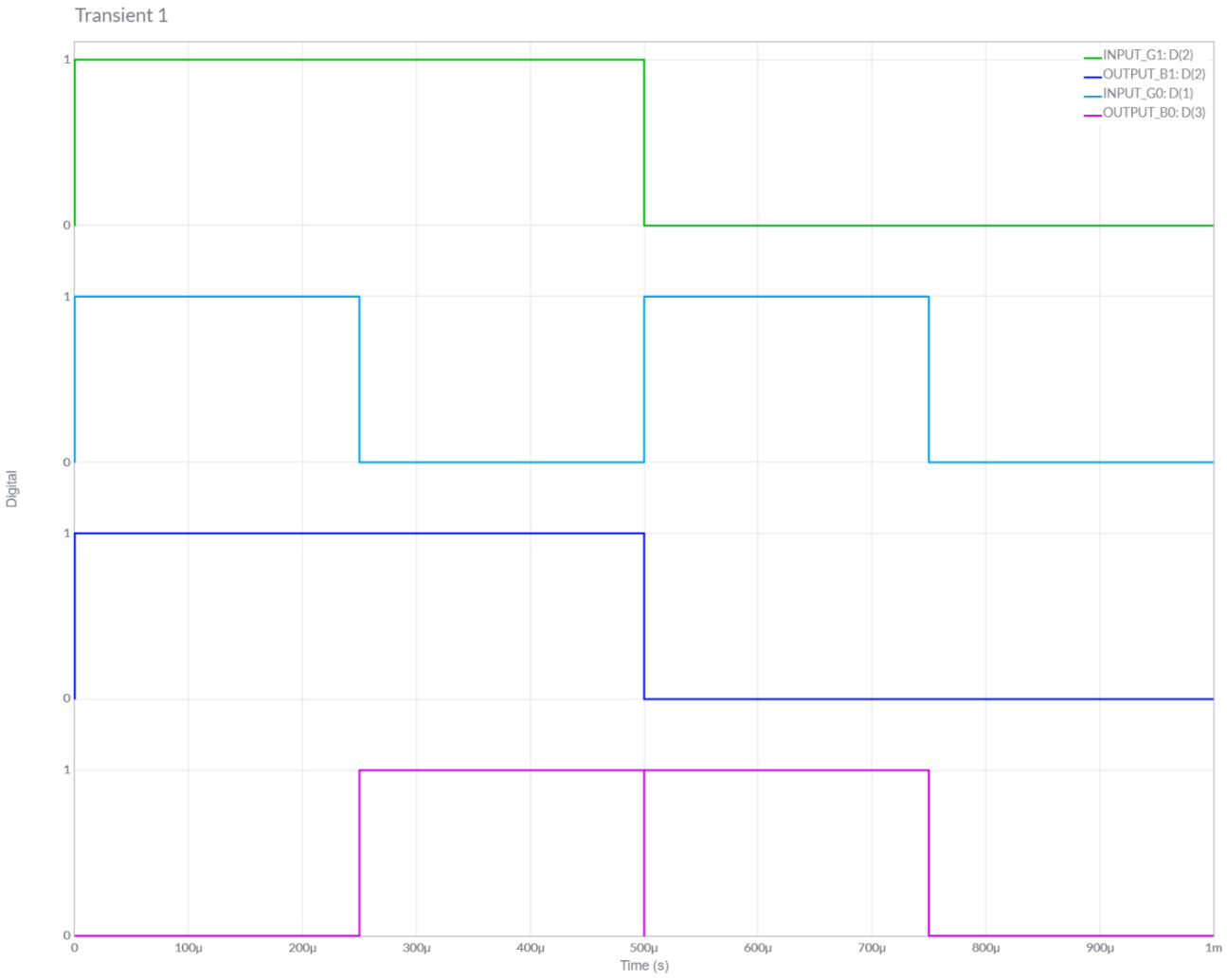


2 – BIT Gray to Binary code converter

CIRCUIT/CONNECTION DIAGRAMS (FROM MULTISIM)

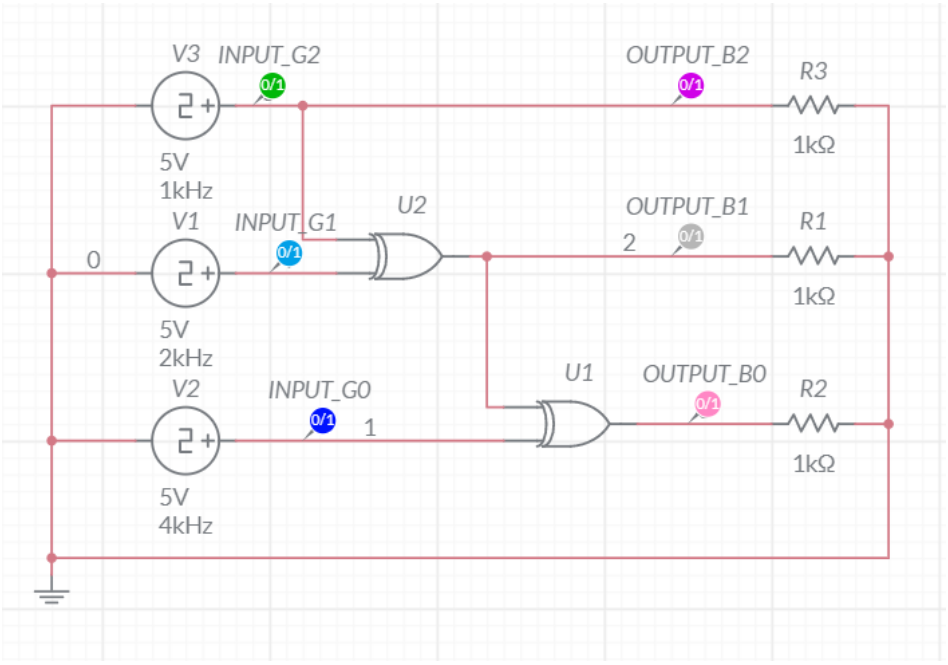


WAVEFORMS (FROM MULTISIM)

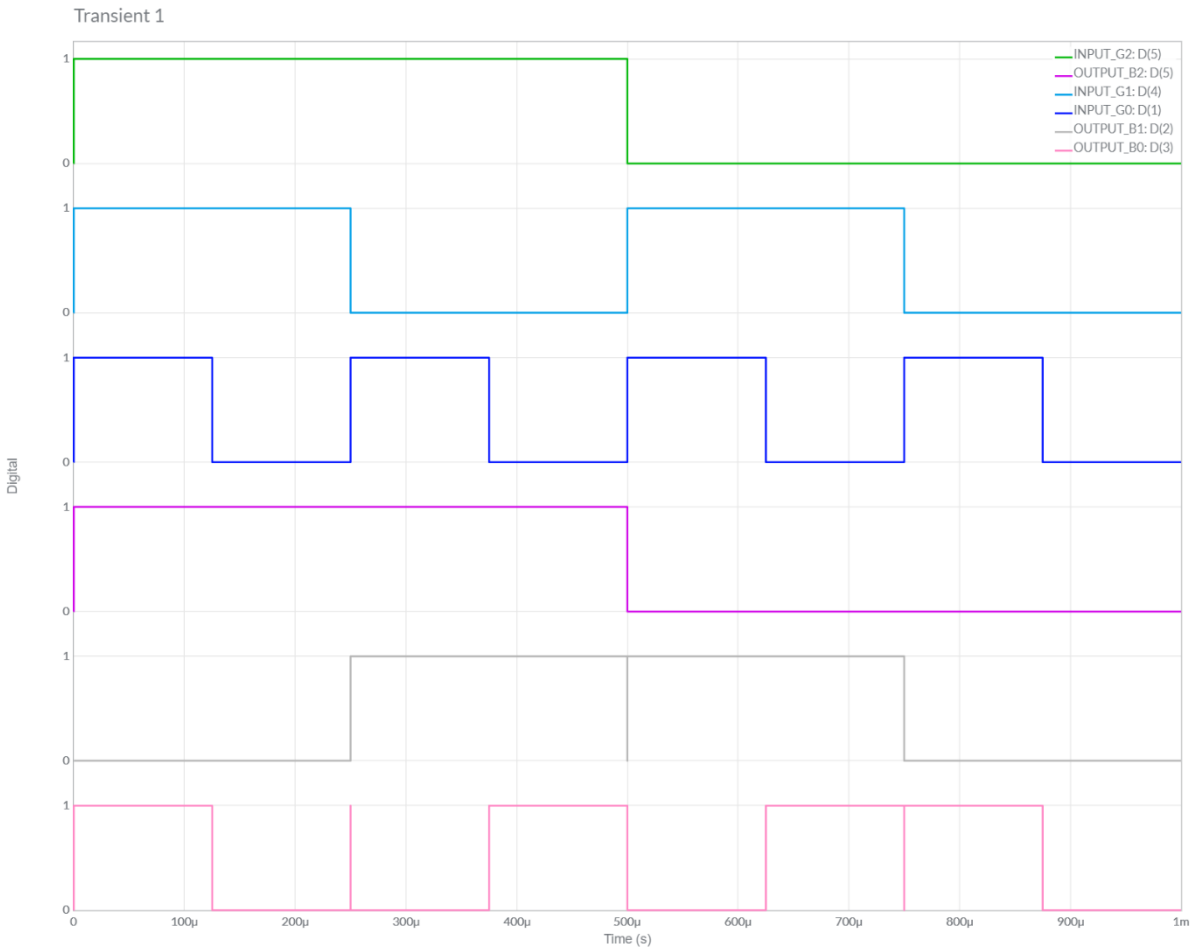


3 – BIT Gray to Binary code converter

CIRCUIT/CONNECTION DIAGRAMS (FROM MULTISIM)

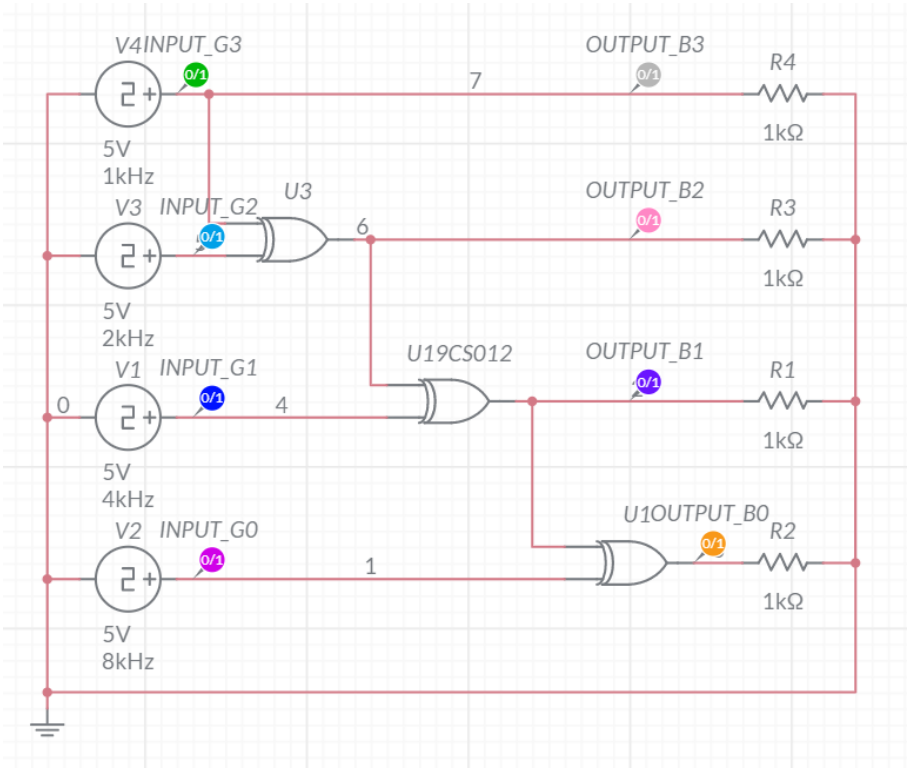


WAVEFORMS (FROM MULTISIM)

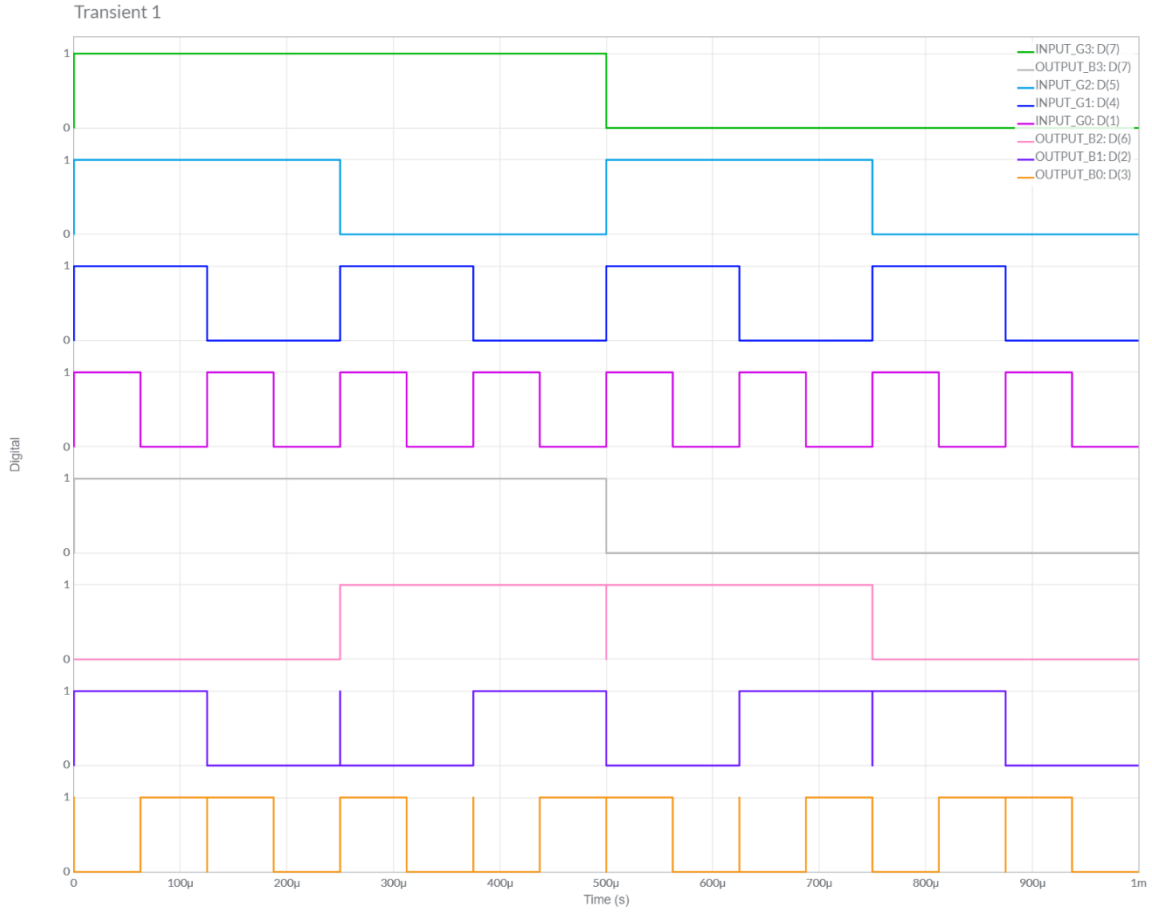


4 – BIT Gray to Binary code converter

CIRCUIT/CONNECTION DIAGRAMS (FROM MULTISIM)

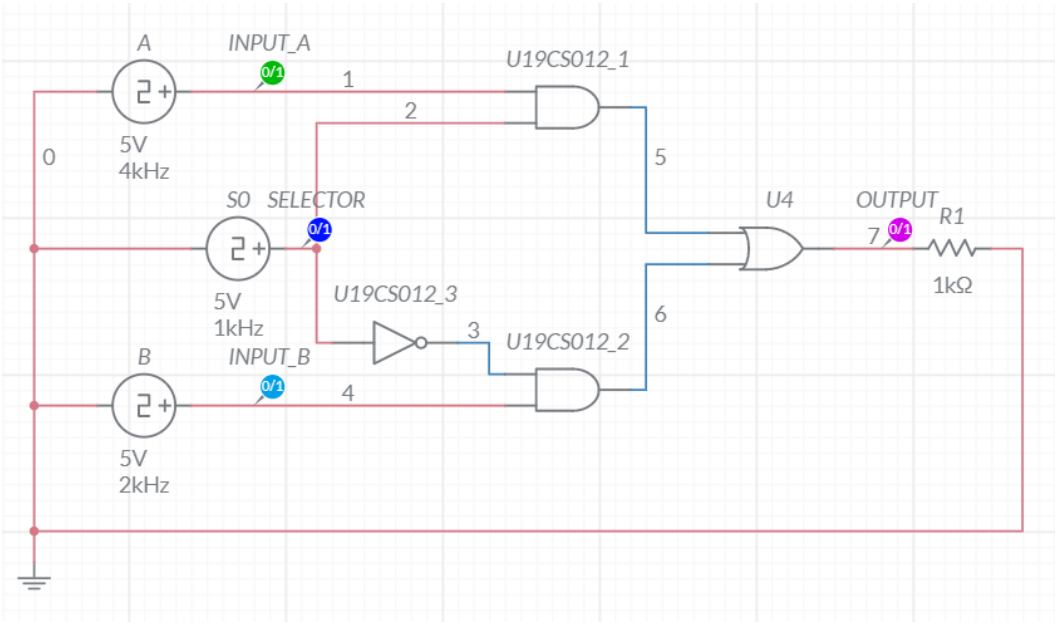


WAVEFORMS (FROM MULTISIM)

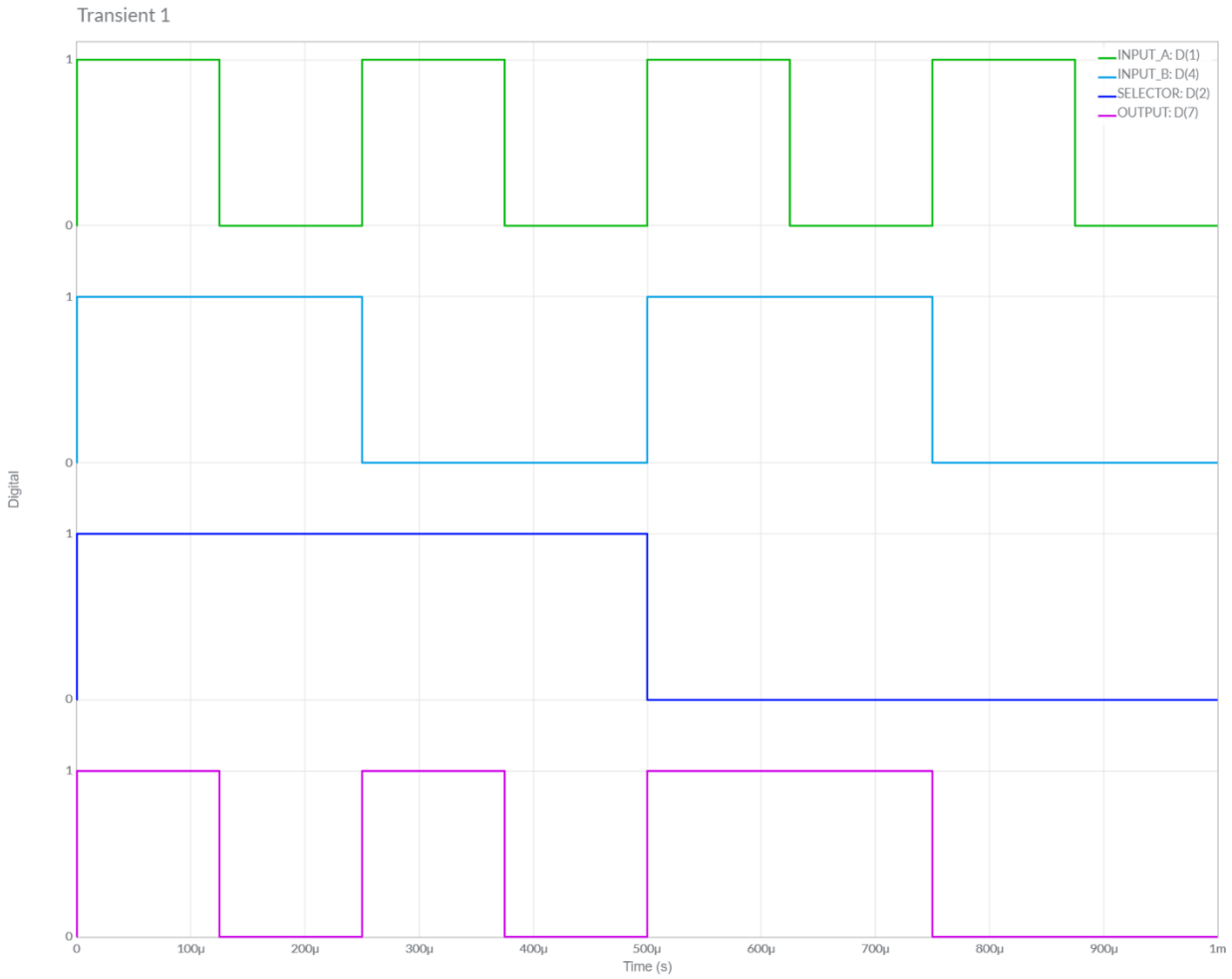


2 : 1 MULTIPLEXOR

CIRCUIT/CONNECTION DIAGRAMS (FROM MULTISIM)

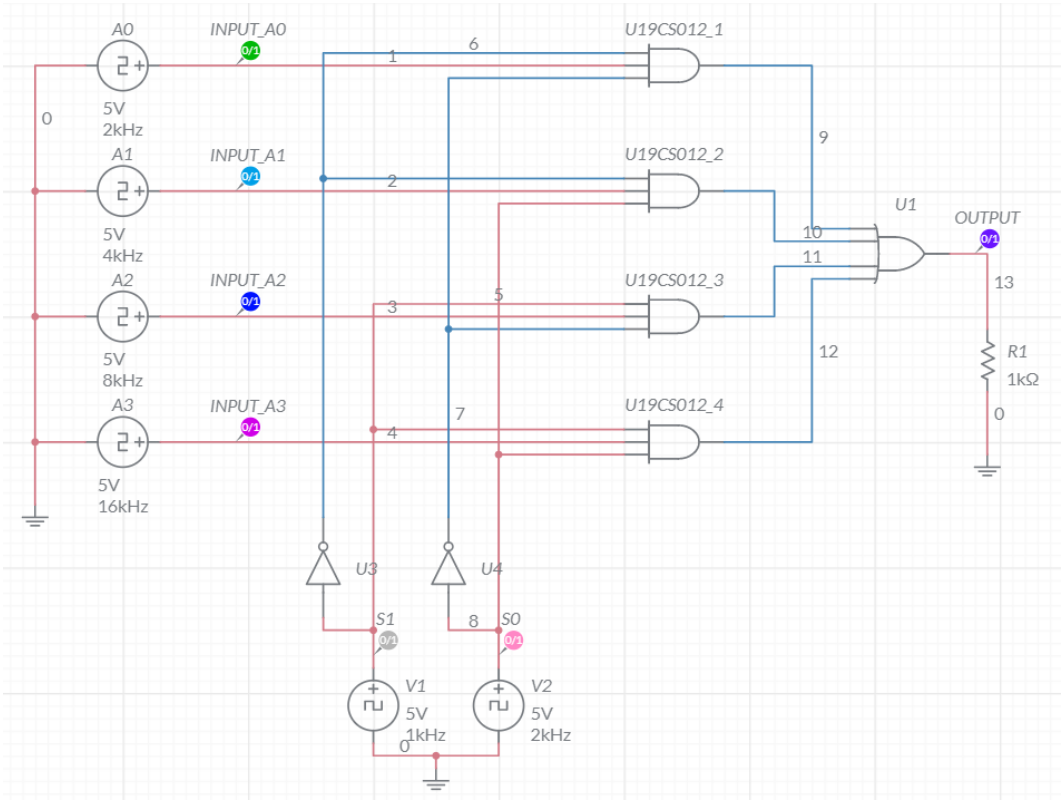


WAVEFORMS (FROM MULTISIM)

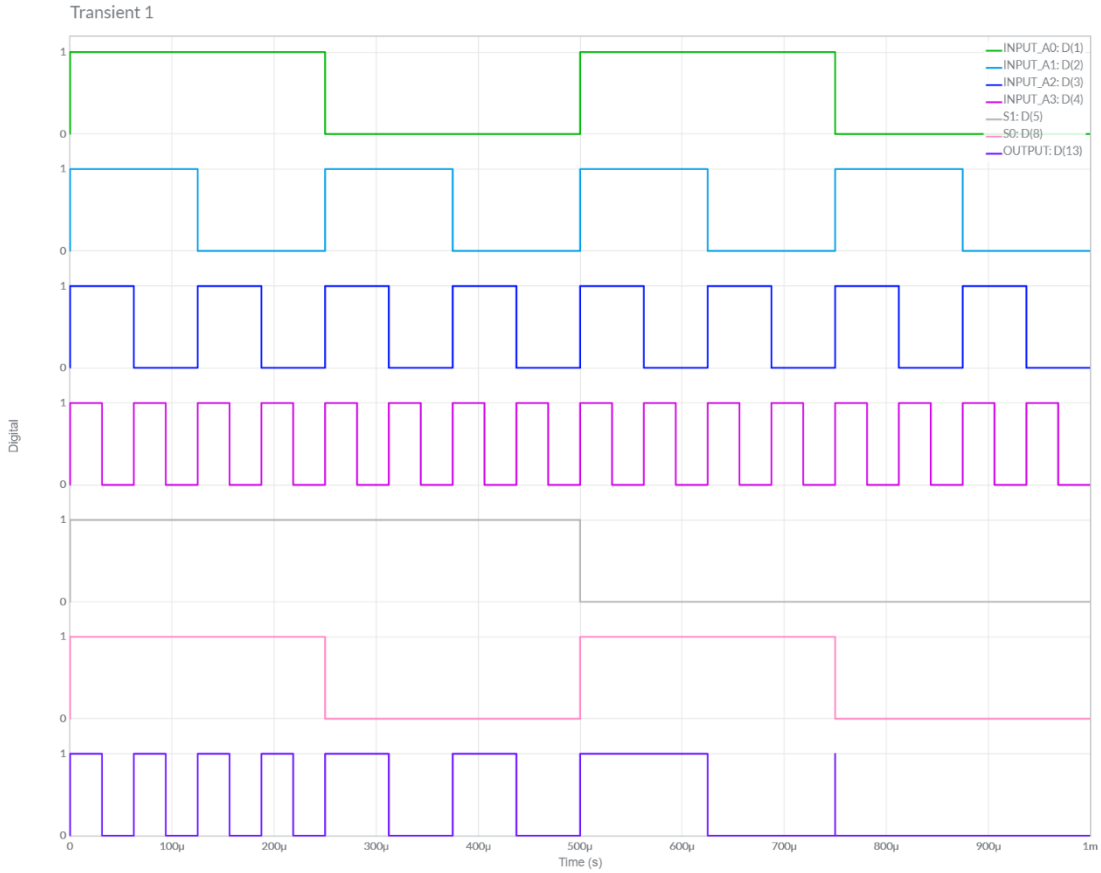


4 : 1 MULTIPLEXOR

CIRCUIT/CONNECTION DIAGRAMS (FROM MULTISIM)



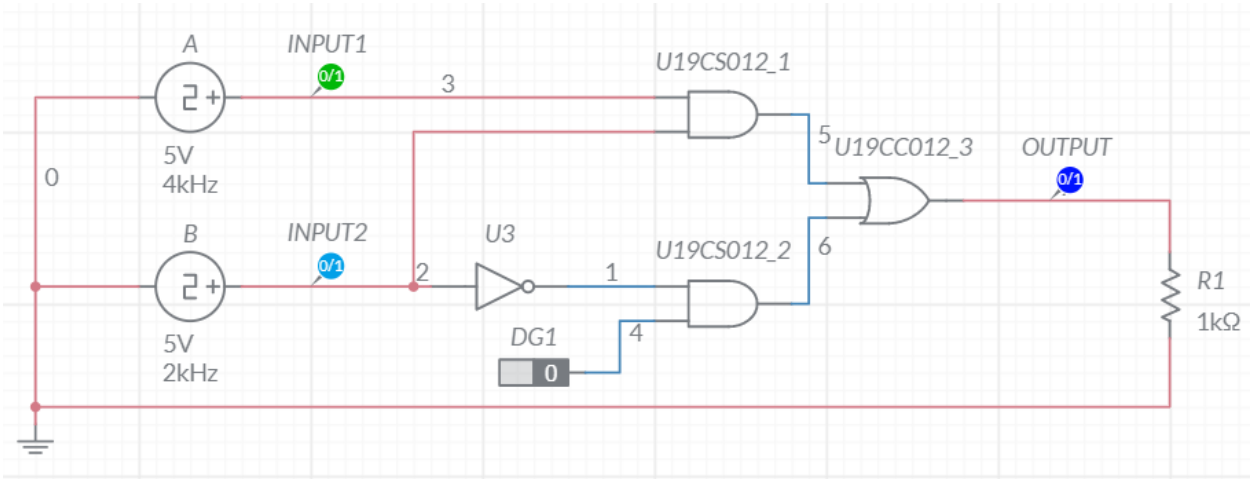
WAVEFORMS (FROM MULTISIM)



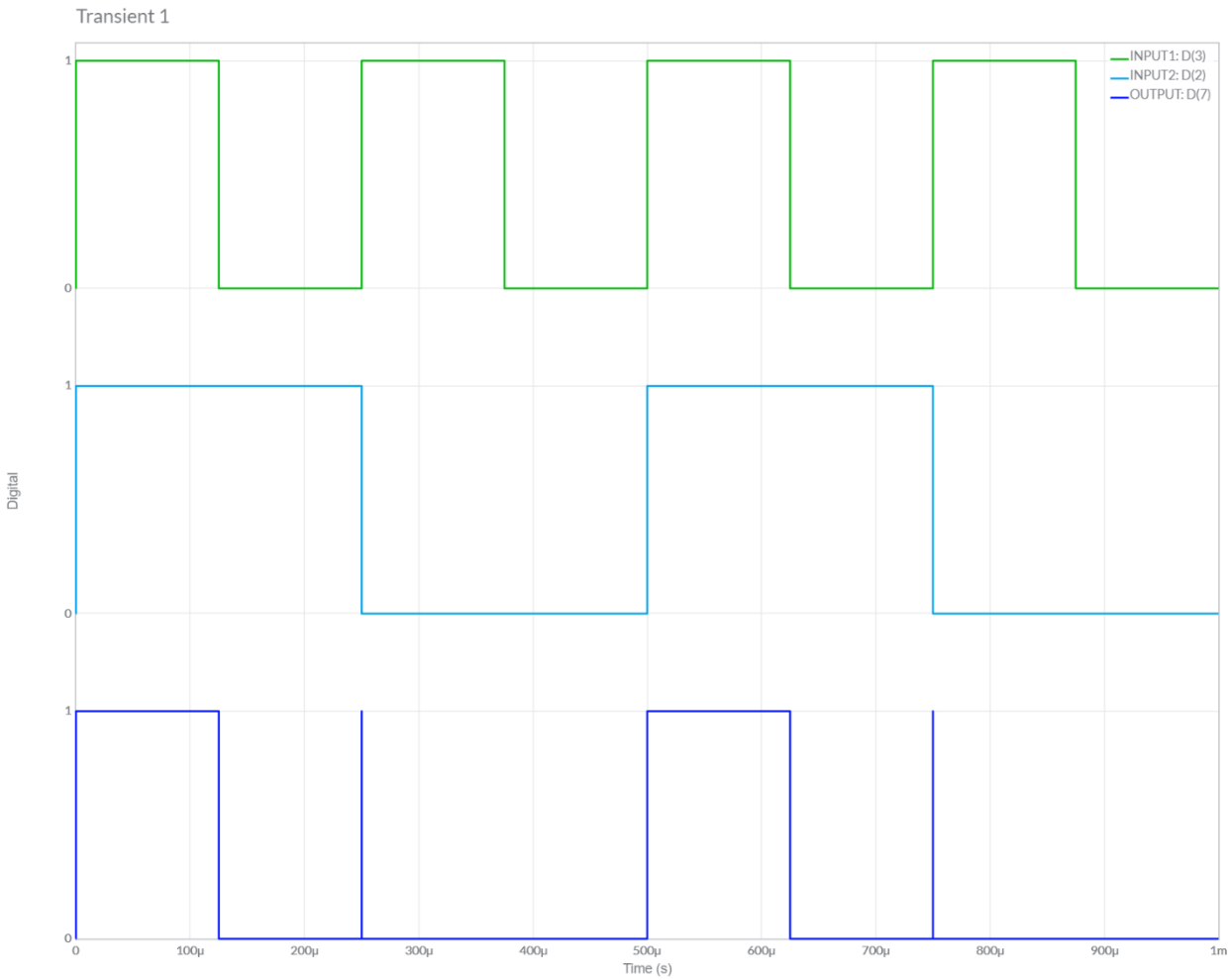
BASIC GATES USING 2:1 MULTIPLEXER

AND GATE

CIRCUIT/CONNECTION DIAGRAMS (FROM MULTISIM)

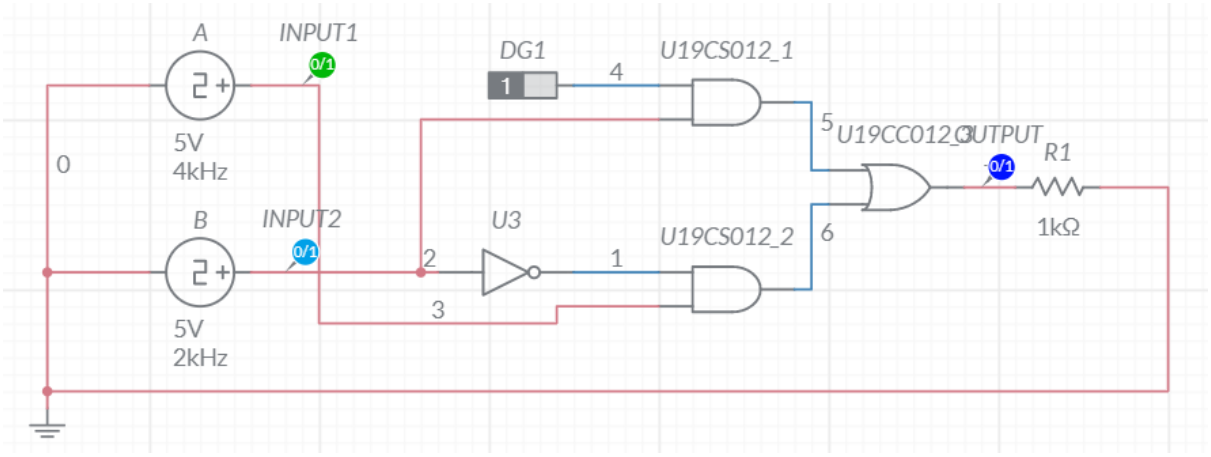


WAVEFORMS (FROM MULTISIM)

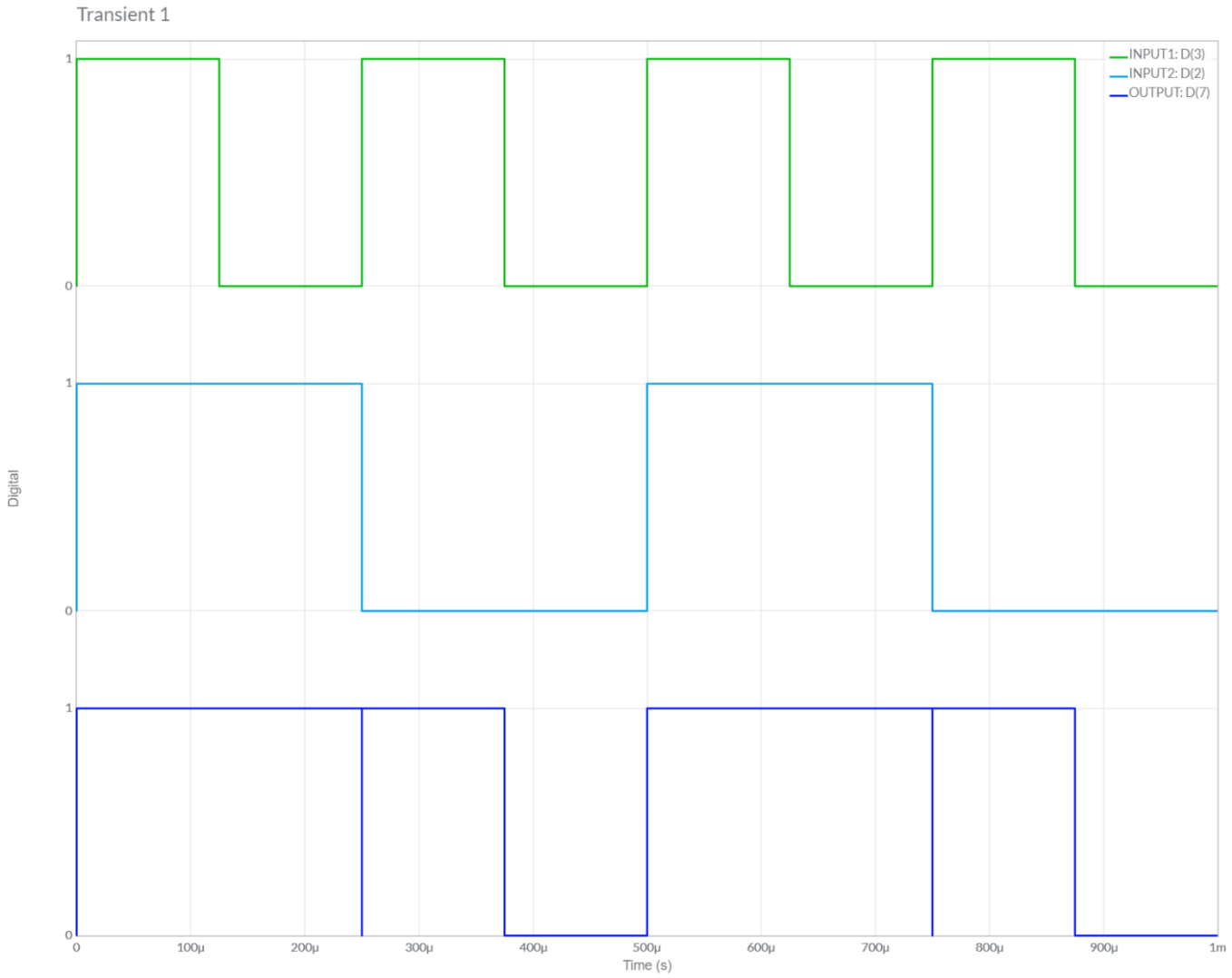


OR GATE

CIRCUIT/CONNECTION DIAGRAMS (FROM MULTISIM)

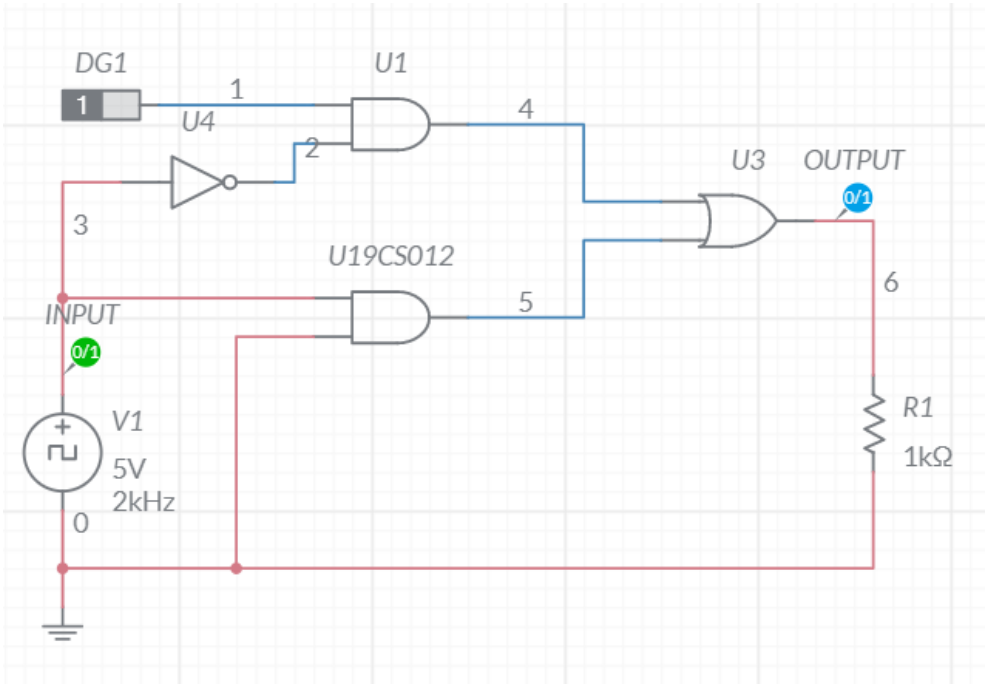


WAVEFORMS (FROM MULTISIM)

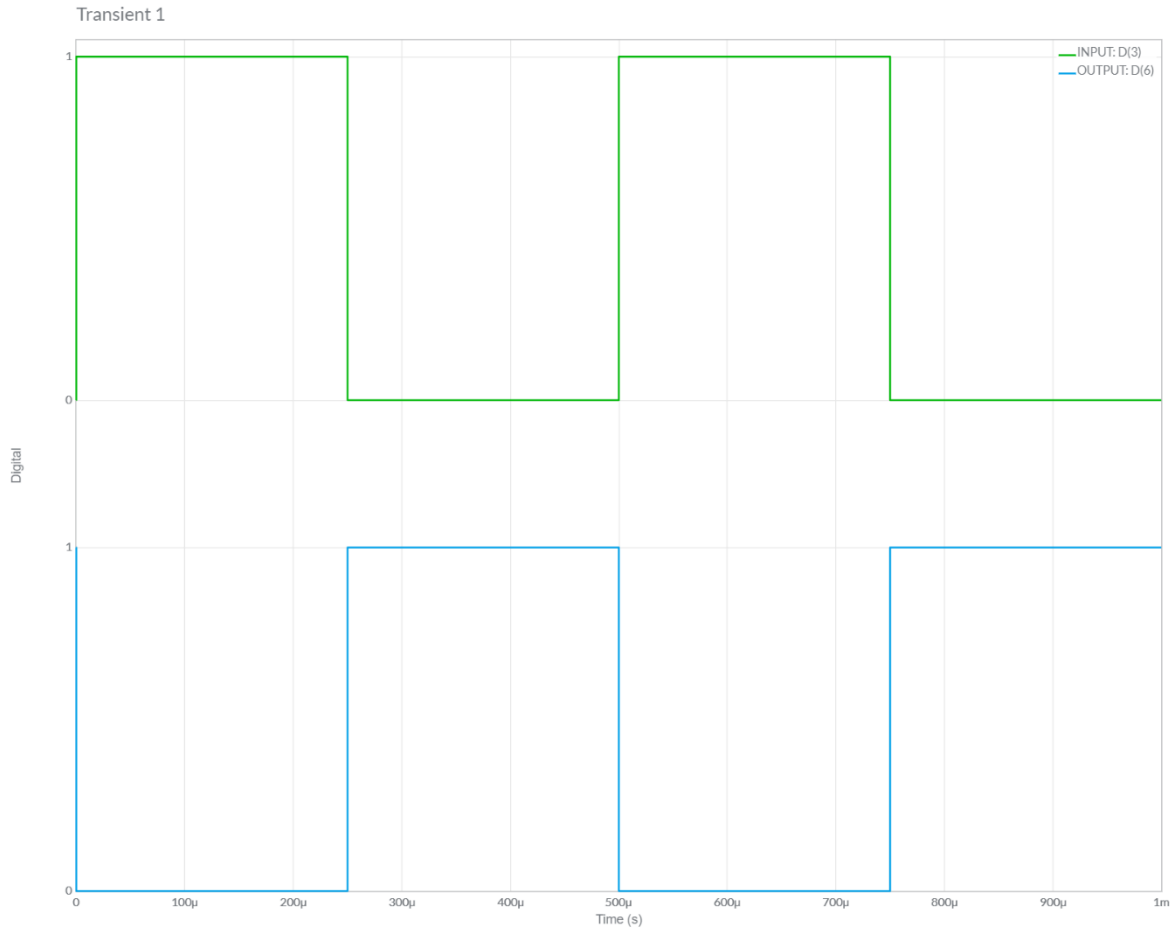


NOT GATE

CIRCUIT/CONNECTION DIAGRAMS (FROM MULTISIM)

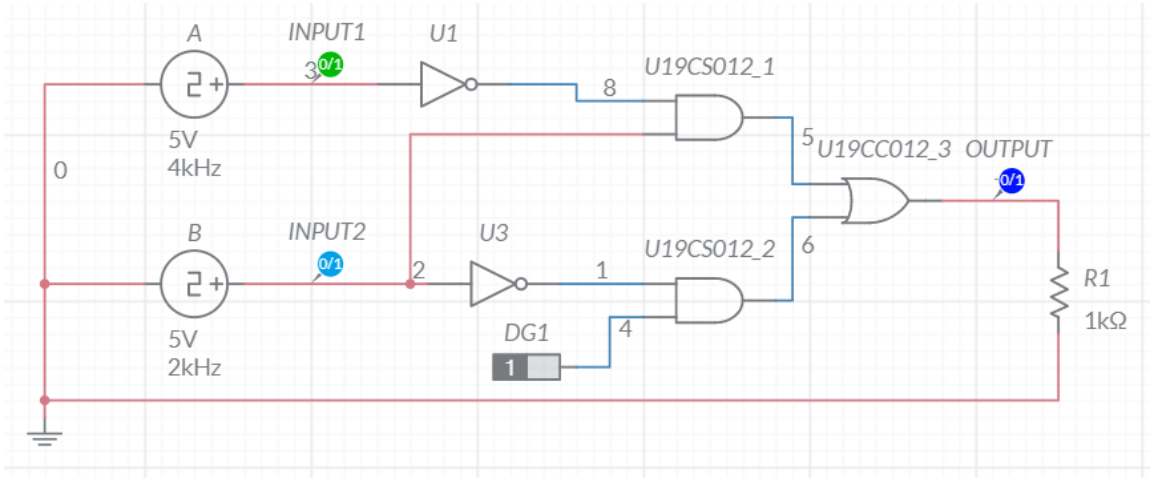


WAVEFORMS (FROM MULTISIM)

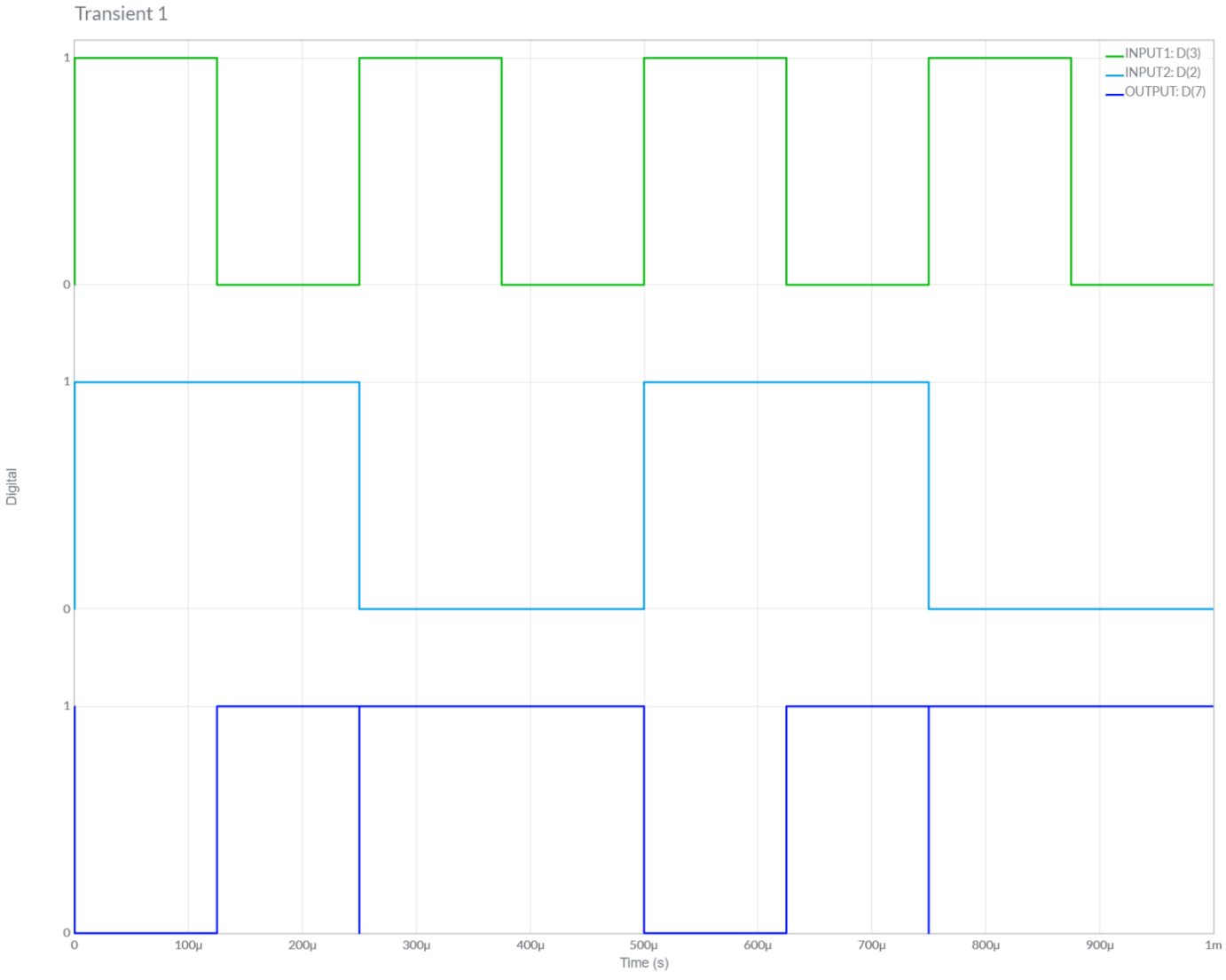


NAND GATE

CIRCUIT/CONNECTION DIAGRAMS (FROM MULTISIM)

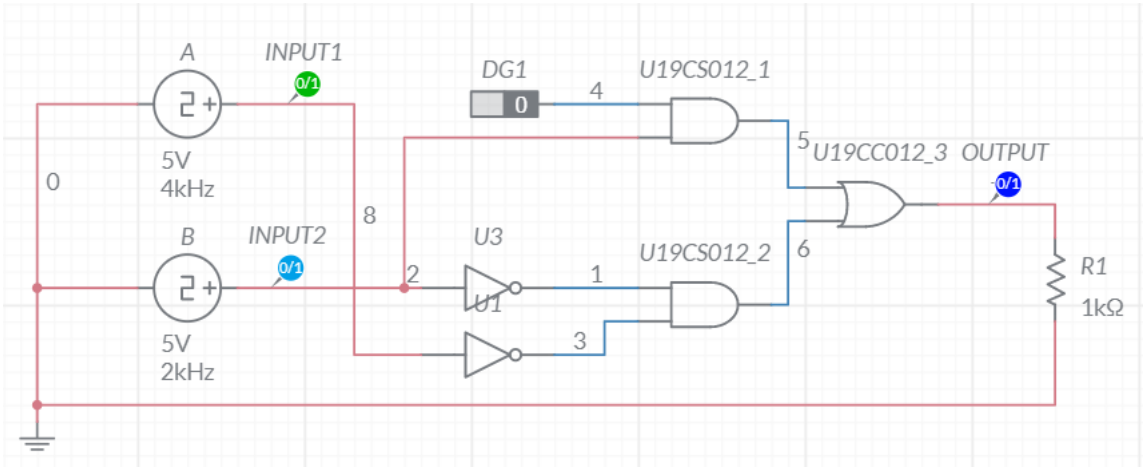


WAVEFORMS (FROM MULTISIM)

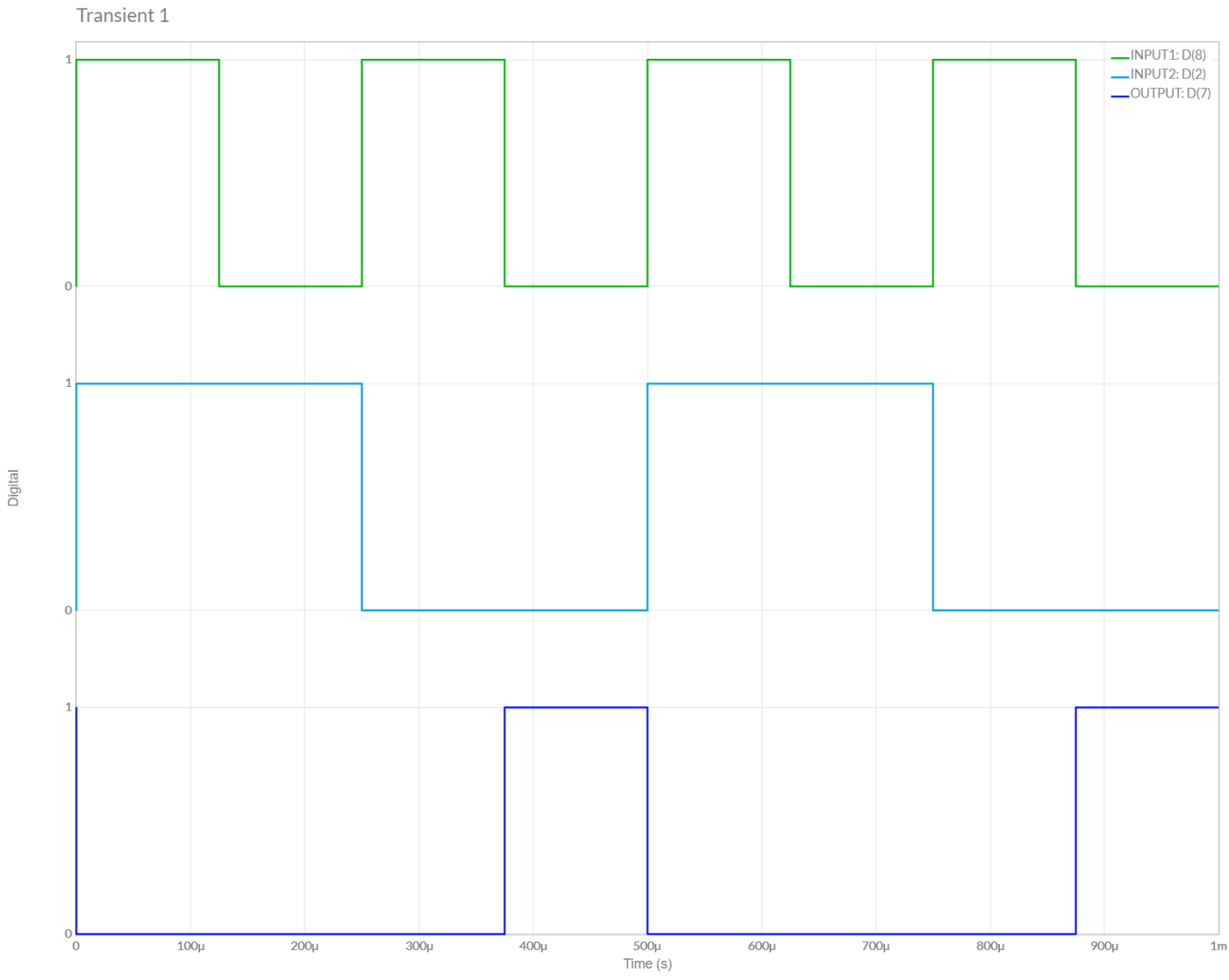


NOR GATE

CIRCUIT/CONNECTION DIAGRAMS (FROM MULTISIM)

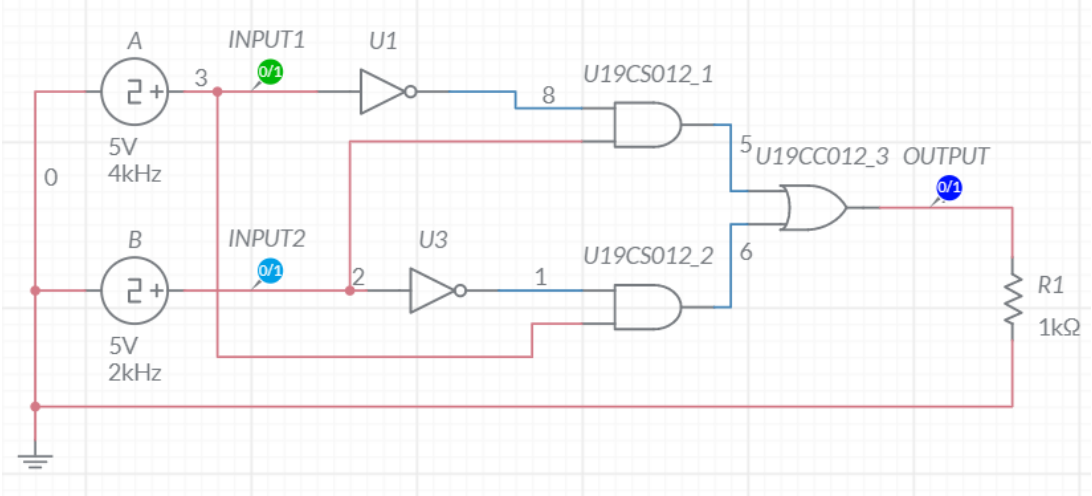


WAVEFORMS (FROM MULTISIM)

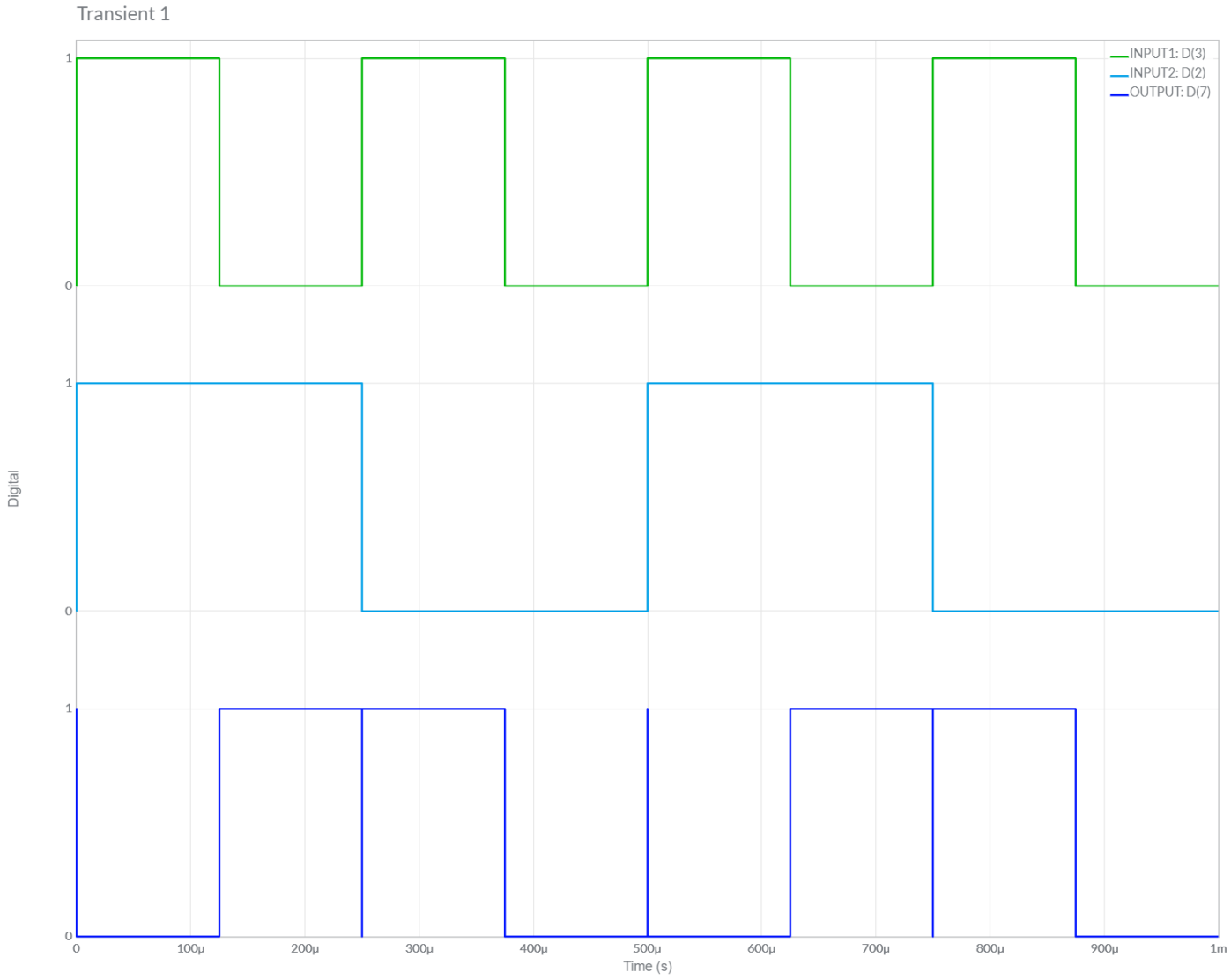


XOR GATE

CIRCUIT/CONNECTION DIAGRAMS (FROM MULTISIM)

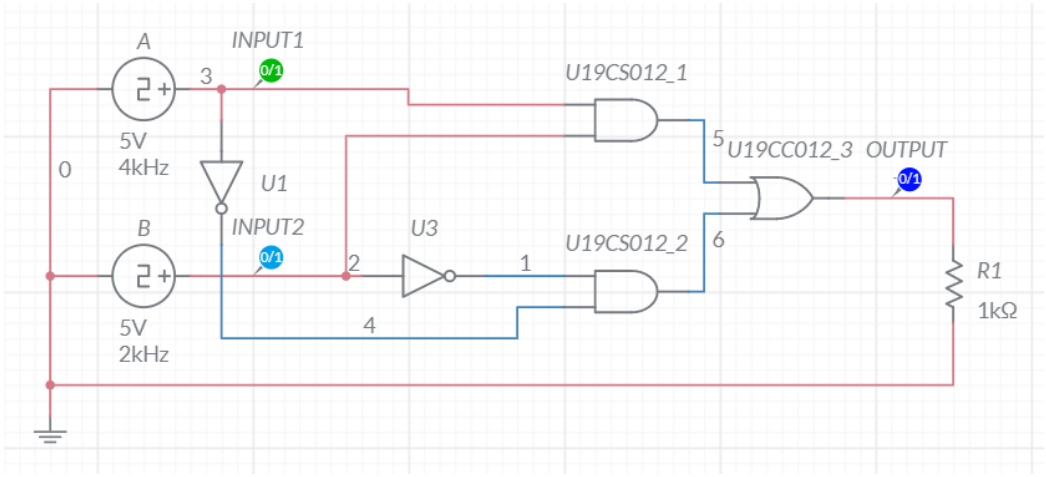


WAVEFORMS (FROM MULTISIM)

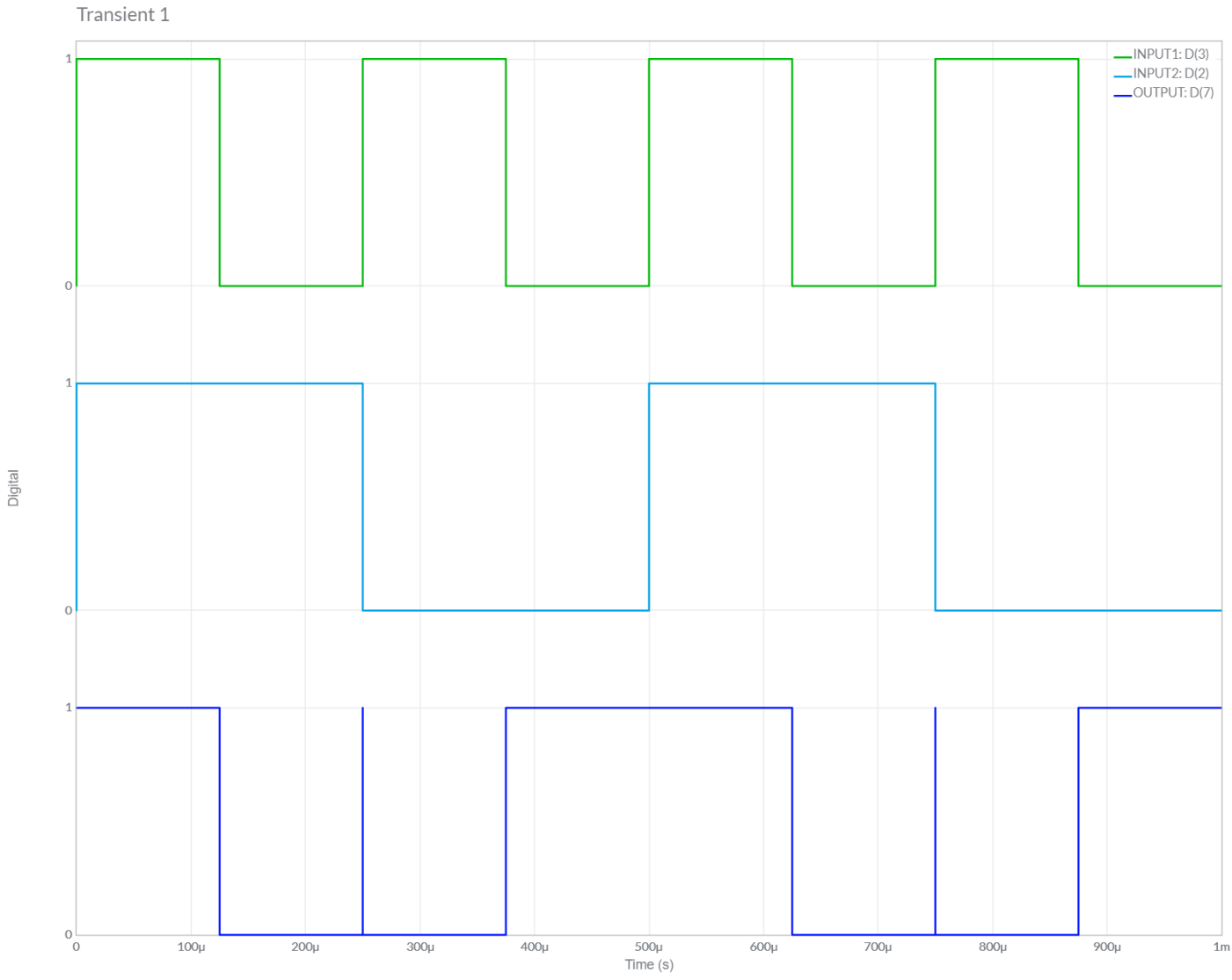


XNOR GATE

CIRCUIT/CONNECTION DIAGRAMS (FROM MULTISIM)

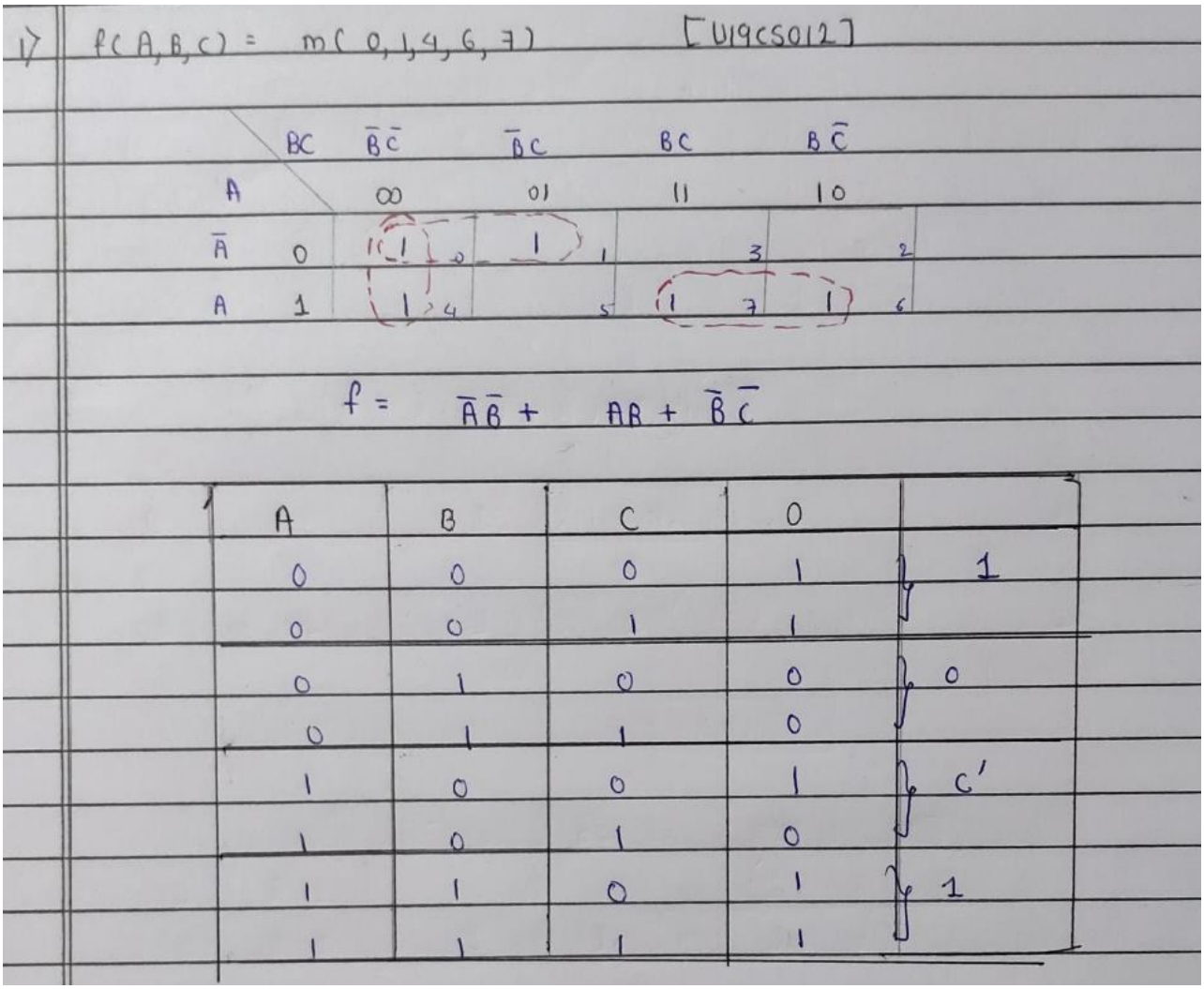


WAVEFORMS (FROM MULTISIM)

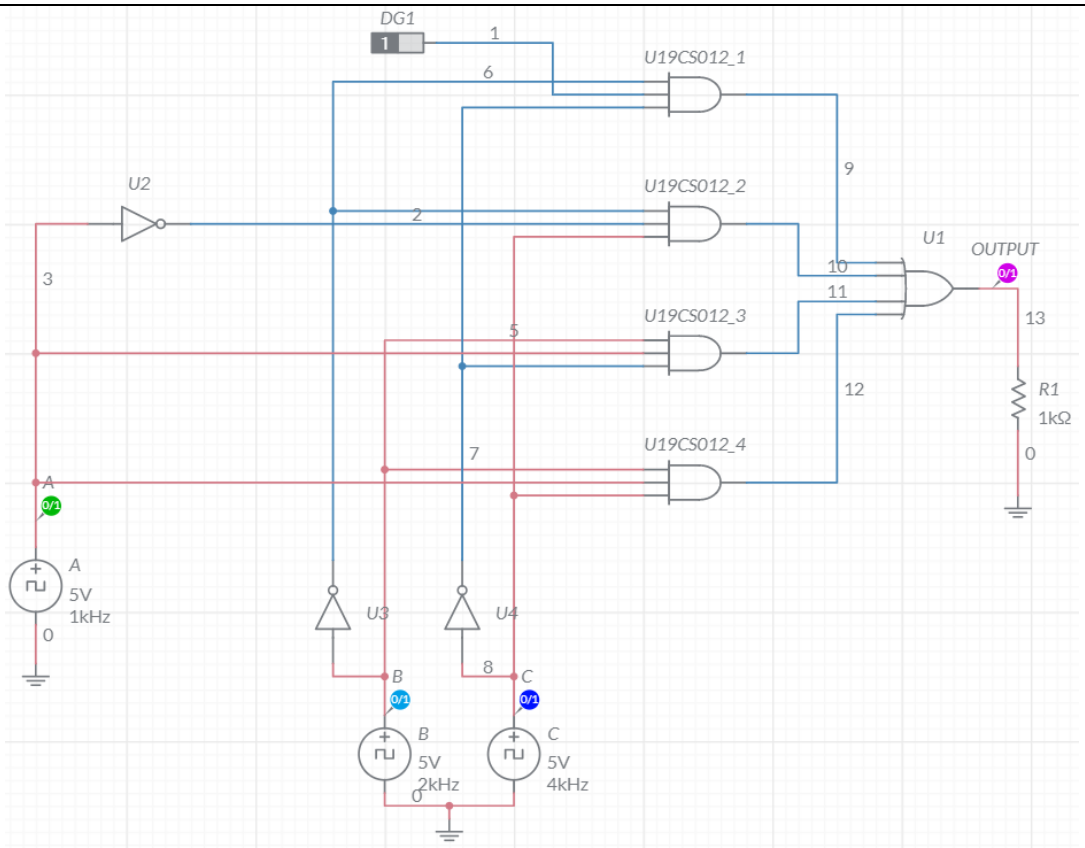


Function Implementation using Multiplexers

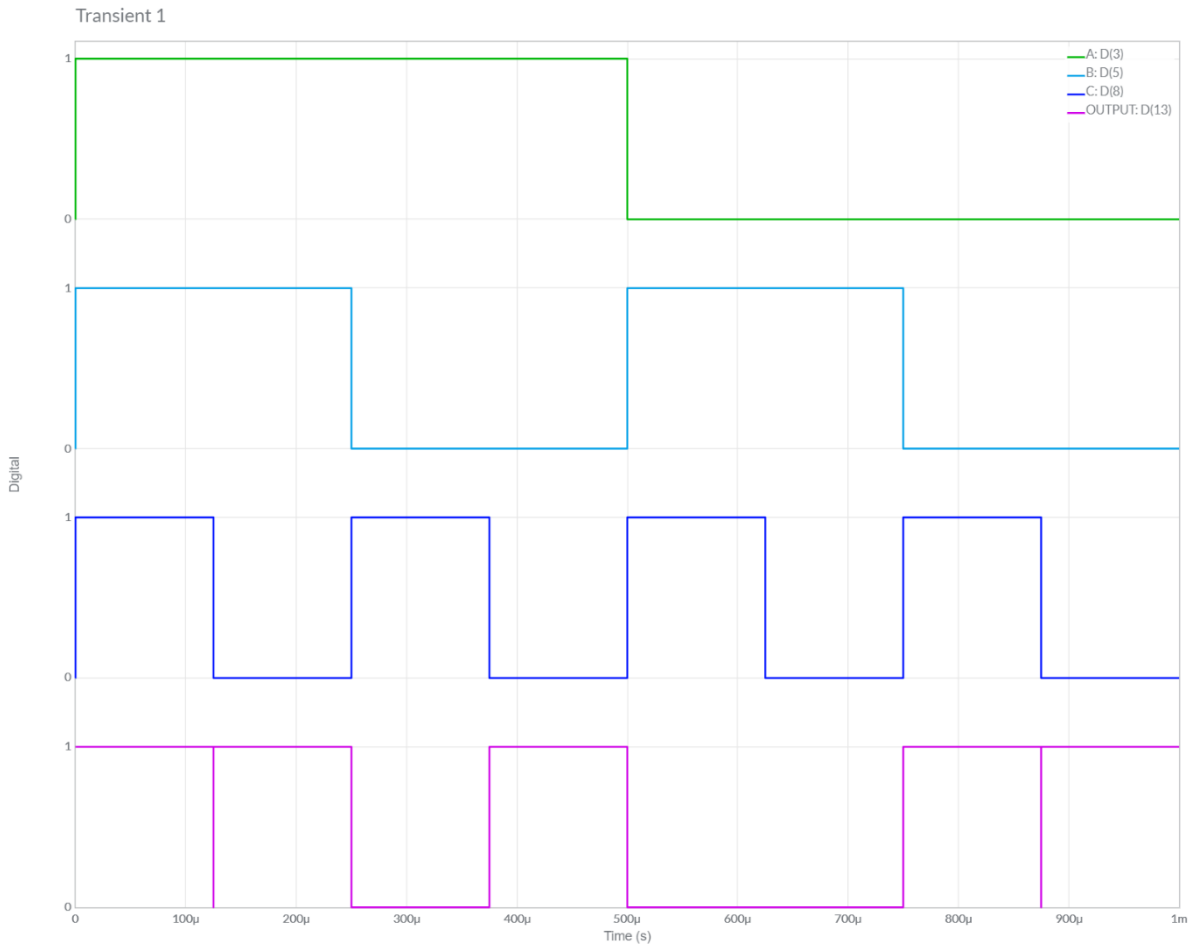
1) f(A, B, C) = m(0,1,4,6,7)



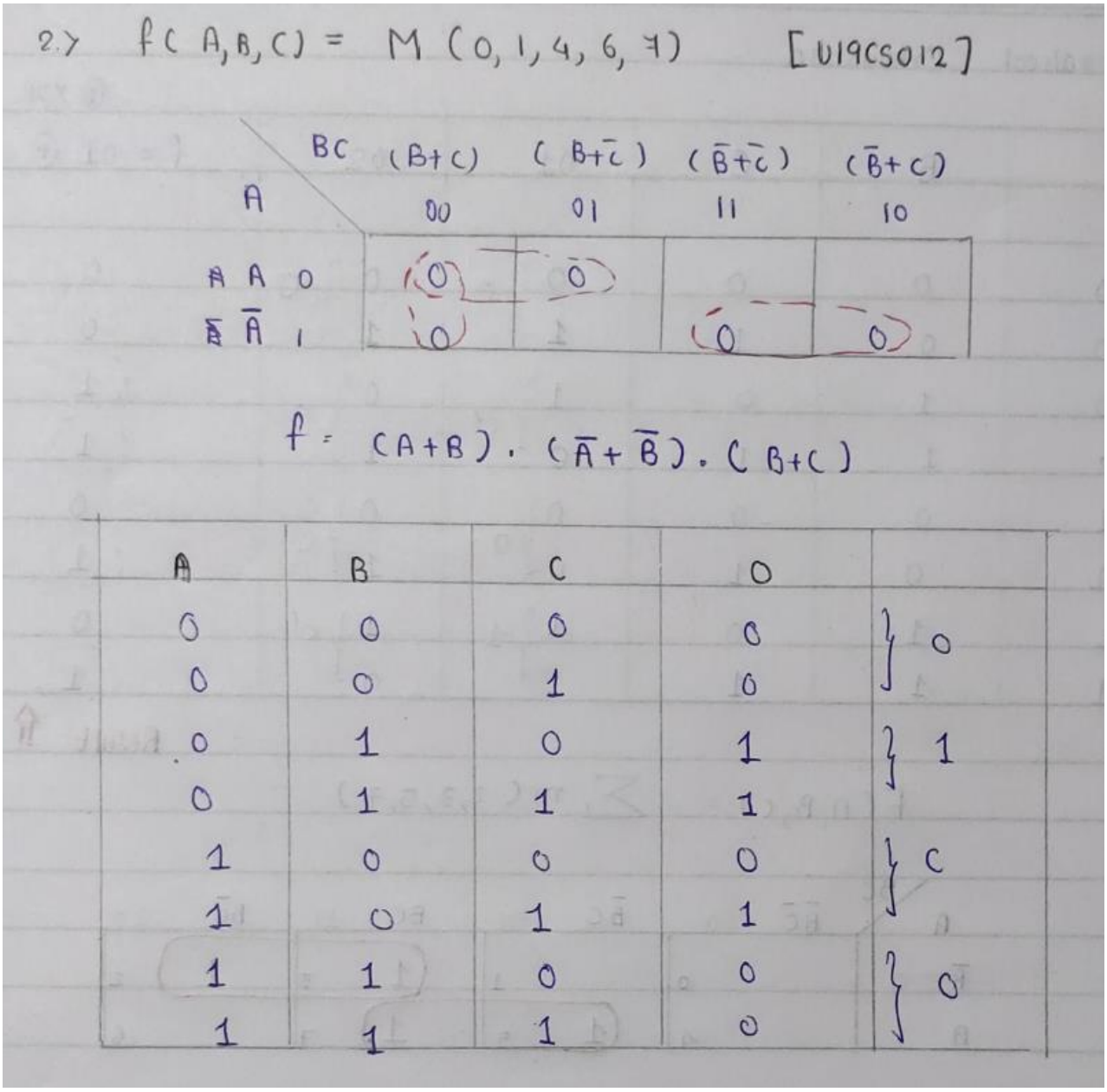
CIRCUIT/CONNECTION DIAGRAMS (FROM MULTISIM)



WAVEFORMS (FROM MULTISIM)

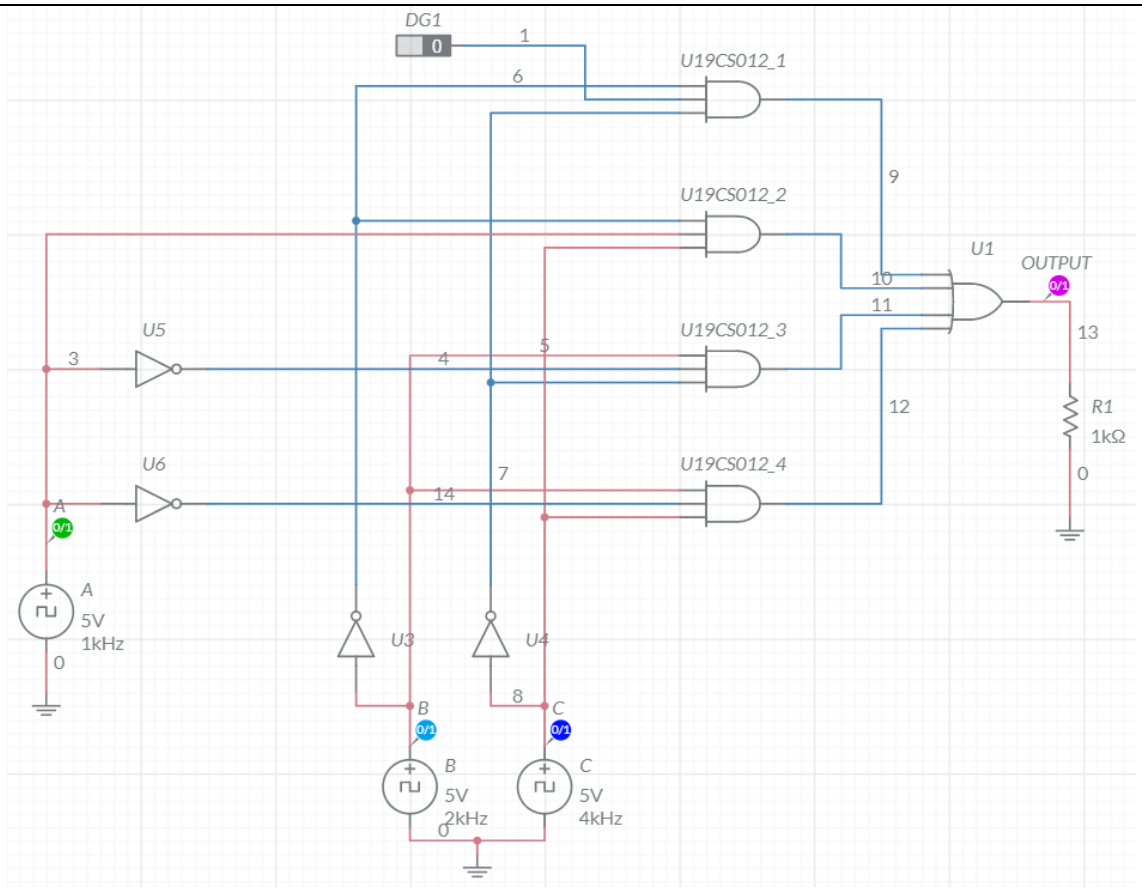


2) f(A, B, C) = M(0,1,4,6,7)

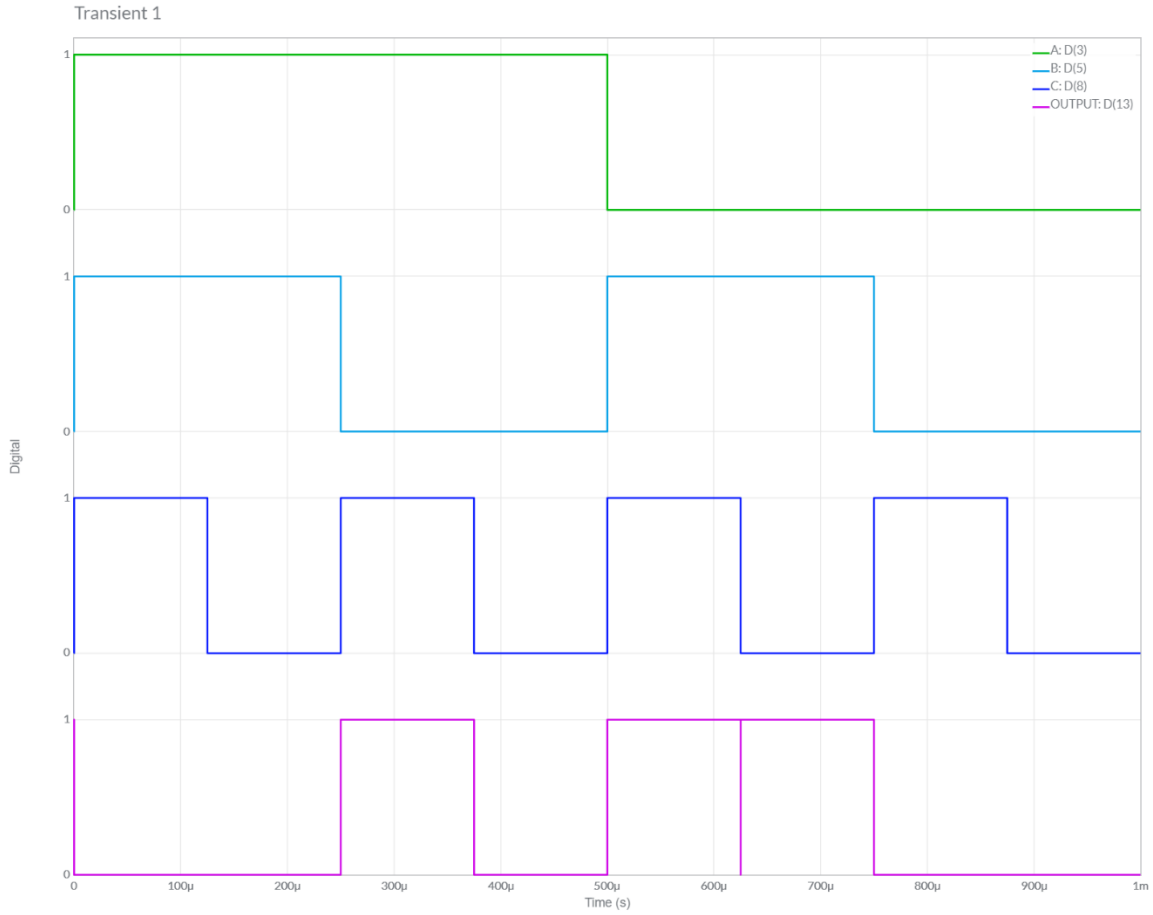


A	B	C	O	
0	0	0	0	} 0
0	0	1	0	
0	1	0	1	} 1
0	1	1	1	
1	0	0	0	} C
1	0	1	1	
1	1	0	0	} 0
1	1	1	0	

CIRCUIT/CONNECTION DIAGRAMS (FROM MULTISIM)



WAVEFORMS (FROM MULTISIM)

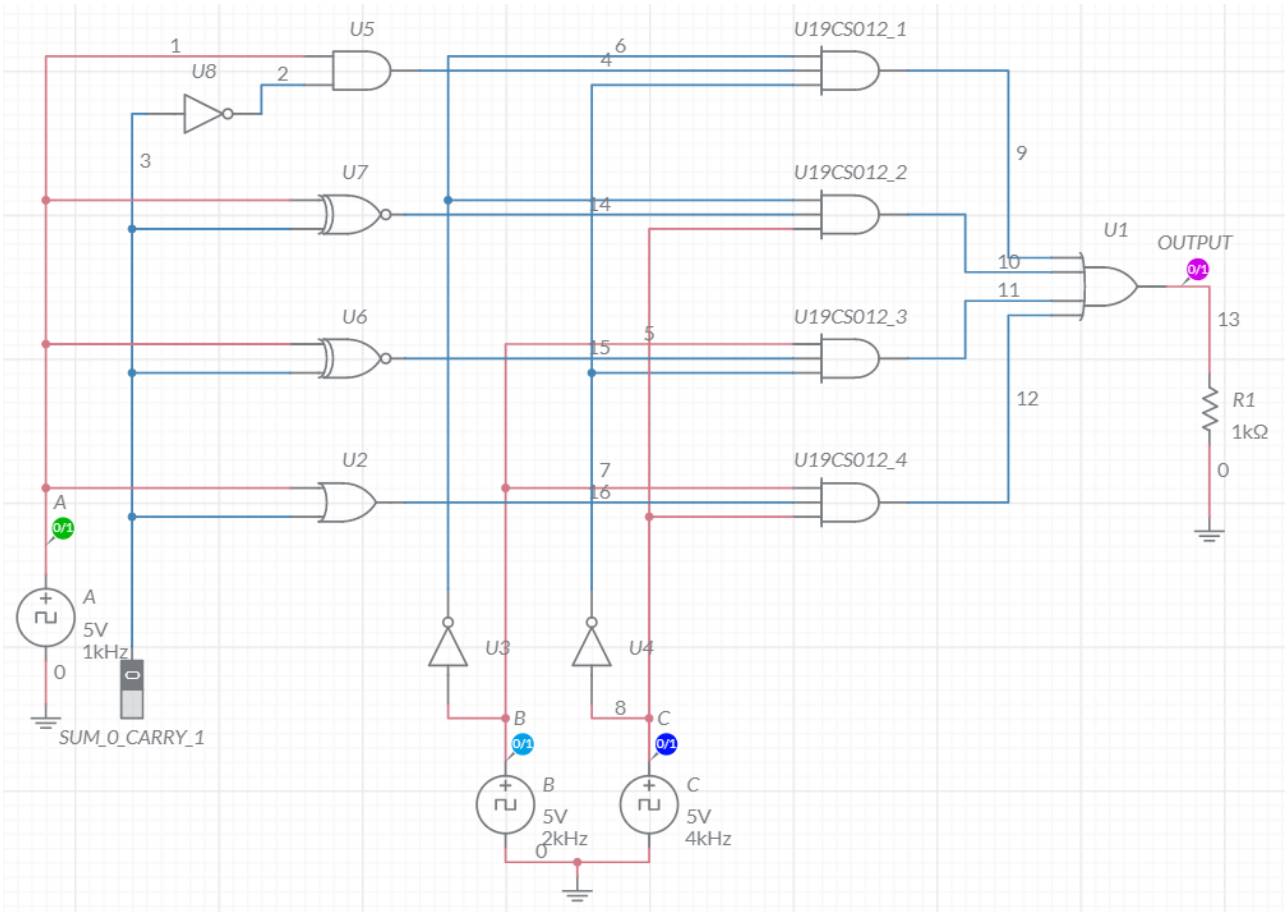


3) Full Adder

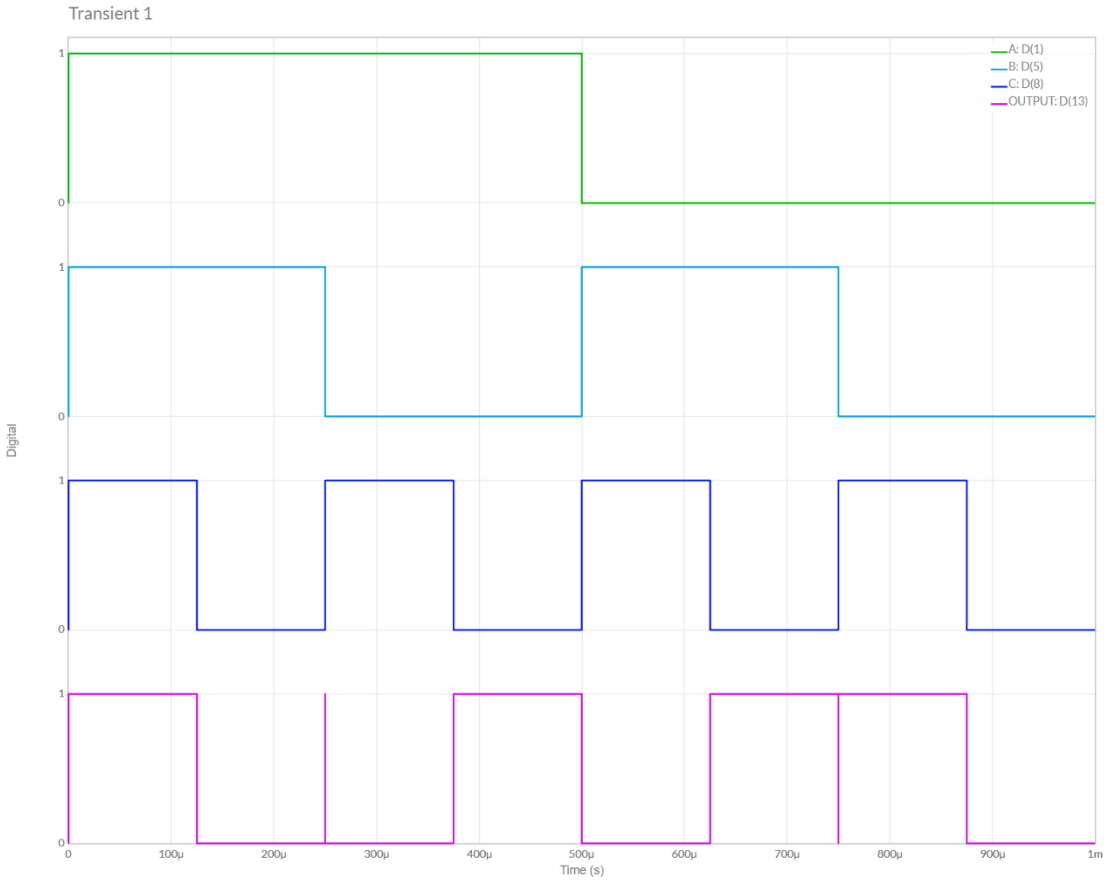
3.) Full Adder [U19CS012]

A	B	C_{in}	Sum	Carry
0	0	0	0 } C	0 } 0
0	0	1	1	0
0	1	0	1 } C'	0 } C
0	1	1	0	1
1	0	0	1 } C'	0 } C
1	0	1	0	1
1	1	0	0 } C	1 } (1)
1	1	1	1	1

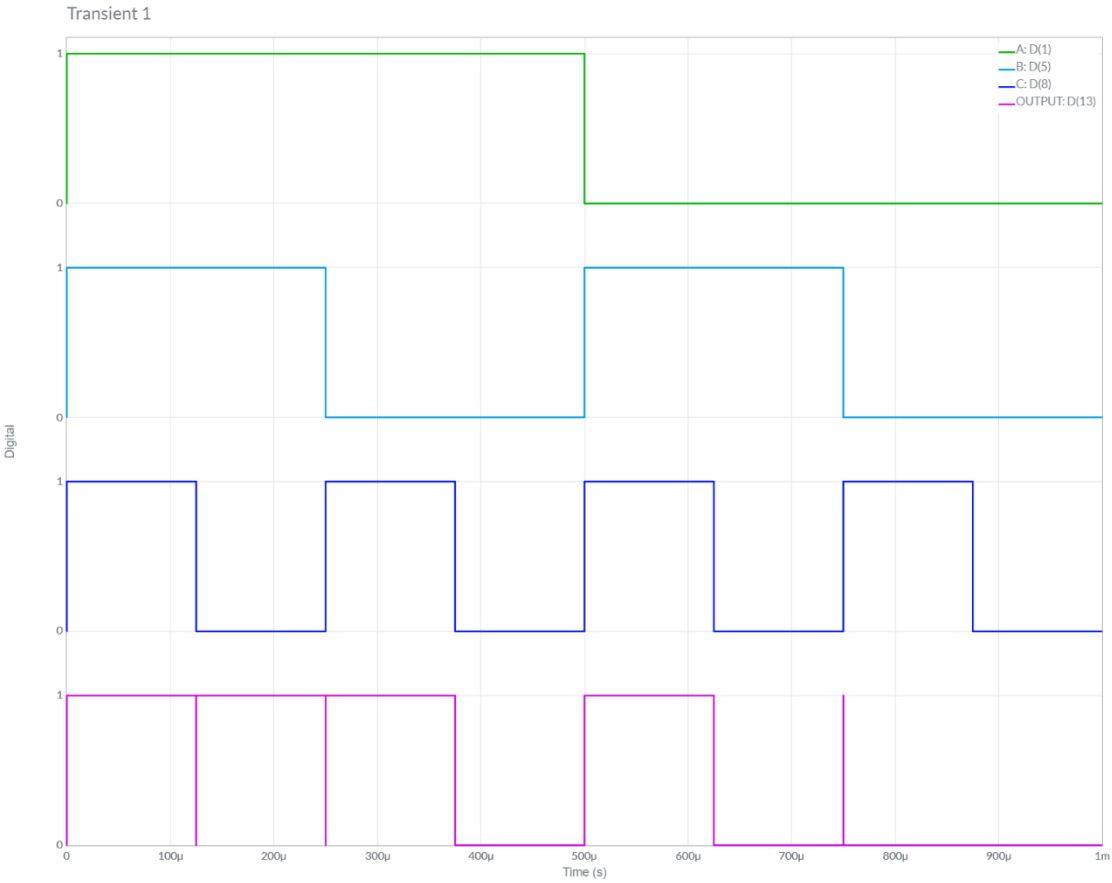
CIRCUIT/CONNECTION DIAGRAMS (FROM MULTISIM)



WAVEFORMS (SUM)



WAVEFORMS (CARRY)



CONCLUSIONS

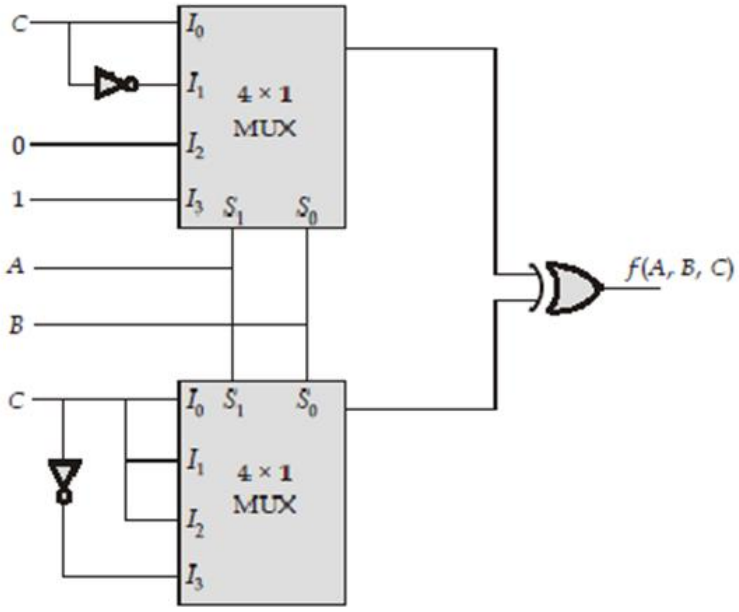
1.) In this Experiment, We have studied about **Code-Converters**[Both Binary to Gray and Gray to Binary], **Multiplexors** (2x1 & 4x1), Realized **All Basic Gates using 2x1** Multiplexor and Some **Functional Implementation** using Multiplexor on Multisim.

2.) We **Verified** the Theoretical Knowledge gained above by *implementing the above Circuits* in Multisim [verified it with their Truth Table] and successfully Got the **Desired Output**. Hence the *Experiment has Been Completed Successfully.*

ASSIGNMENT-12

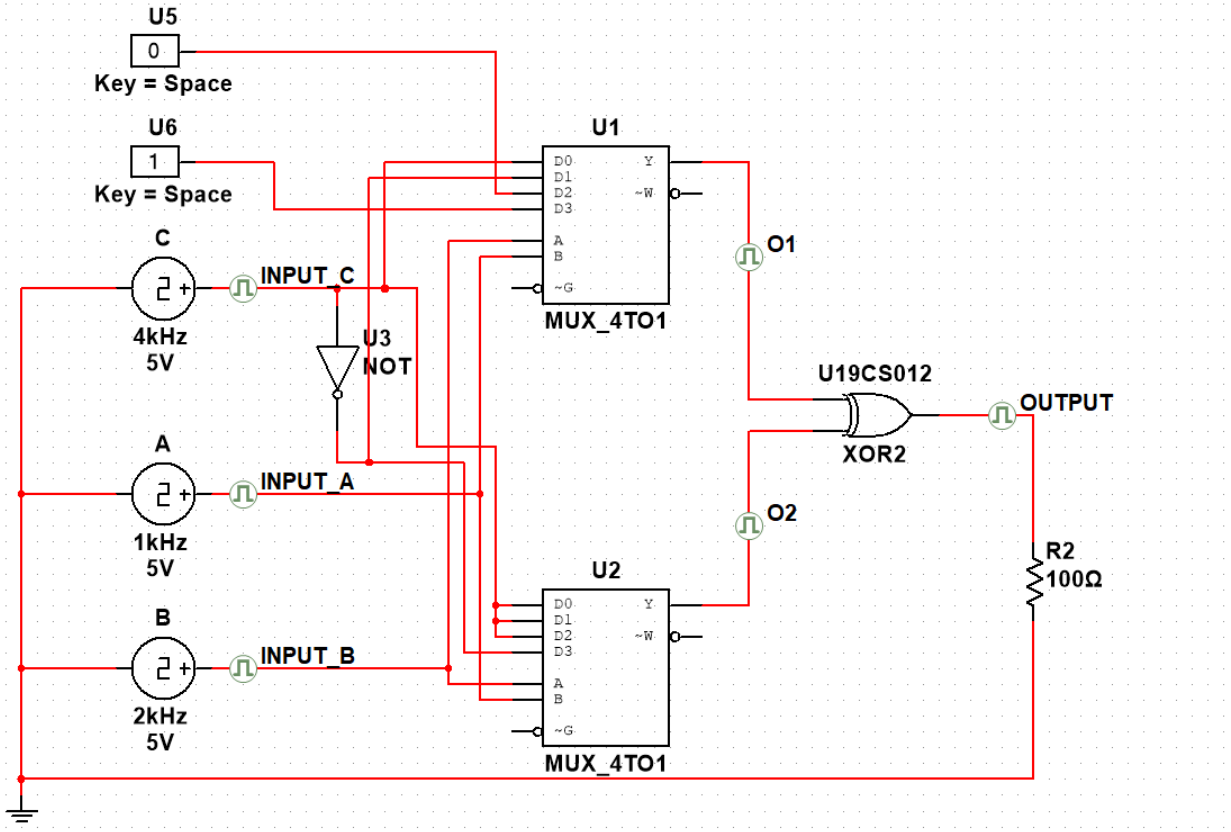
U19CS012

1. Solve for output Function/Functions. Also verify the same using Multisim.

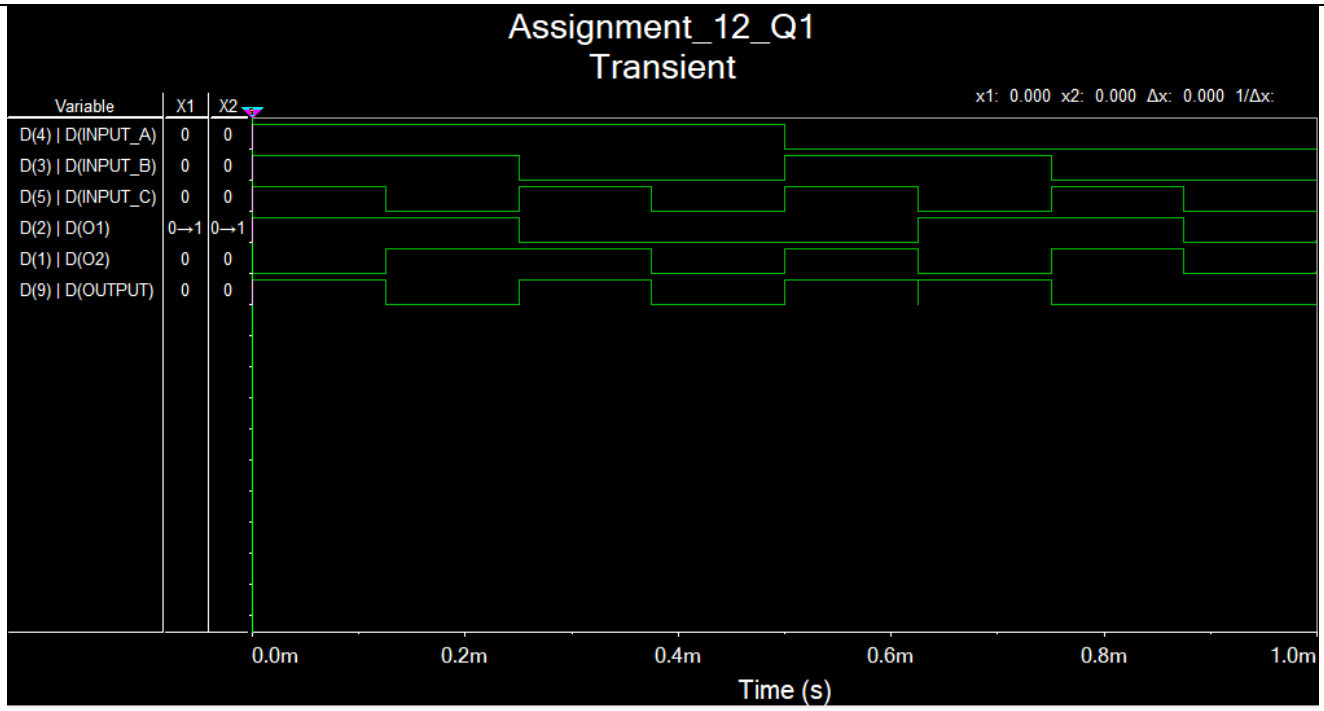


"A" Batch Question

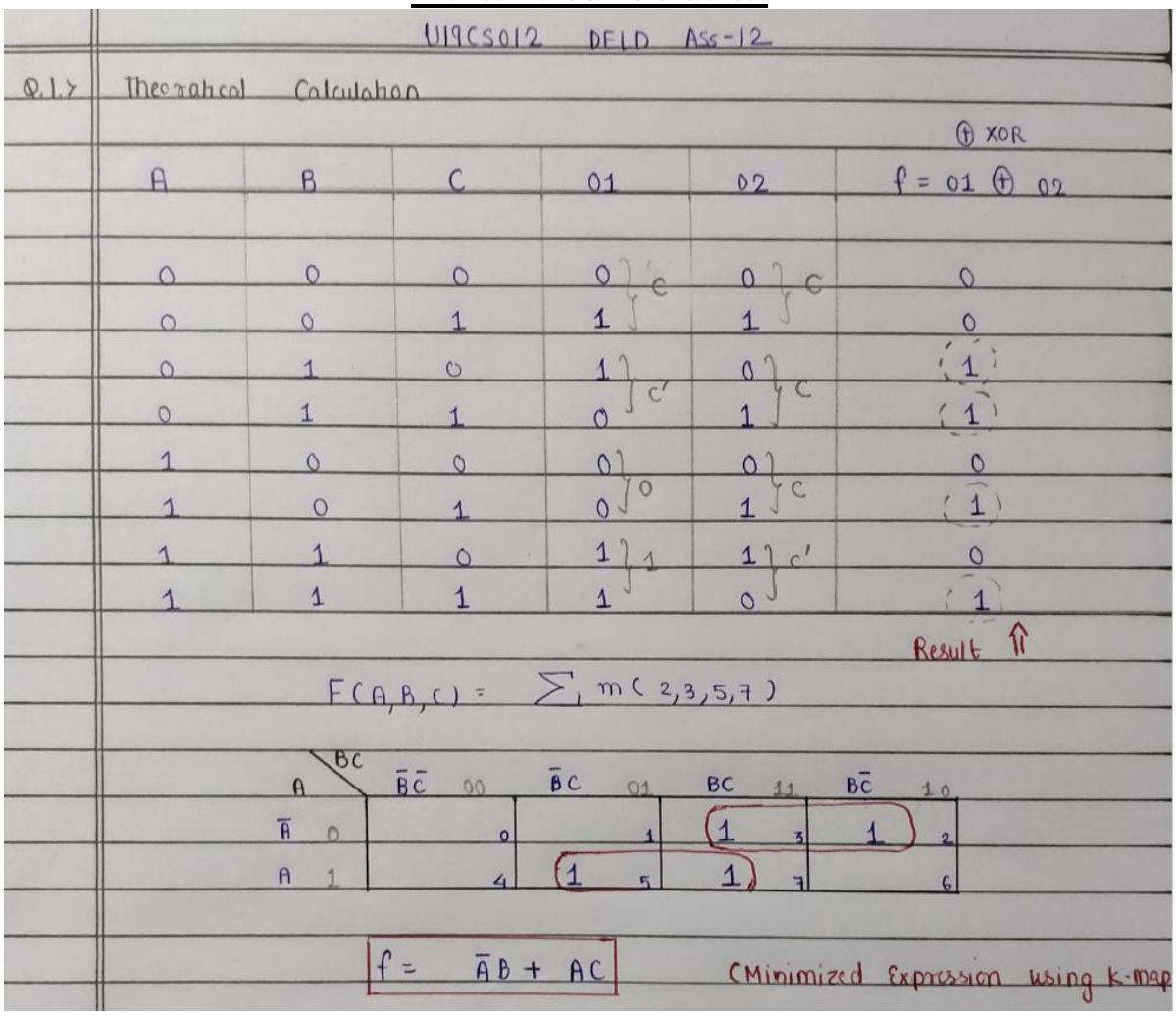
Circuit Diagram [Multisim Implementation]



Grapher Image [Transient]



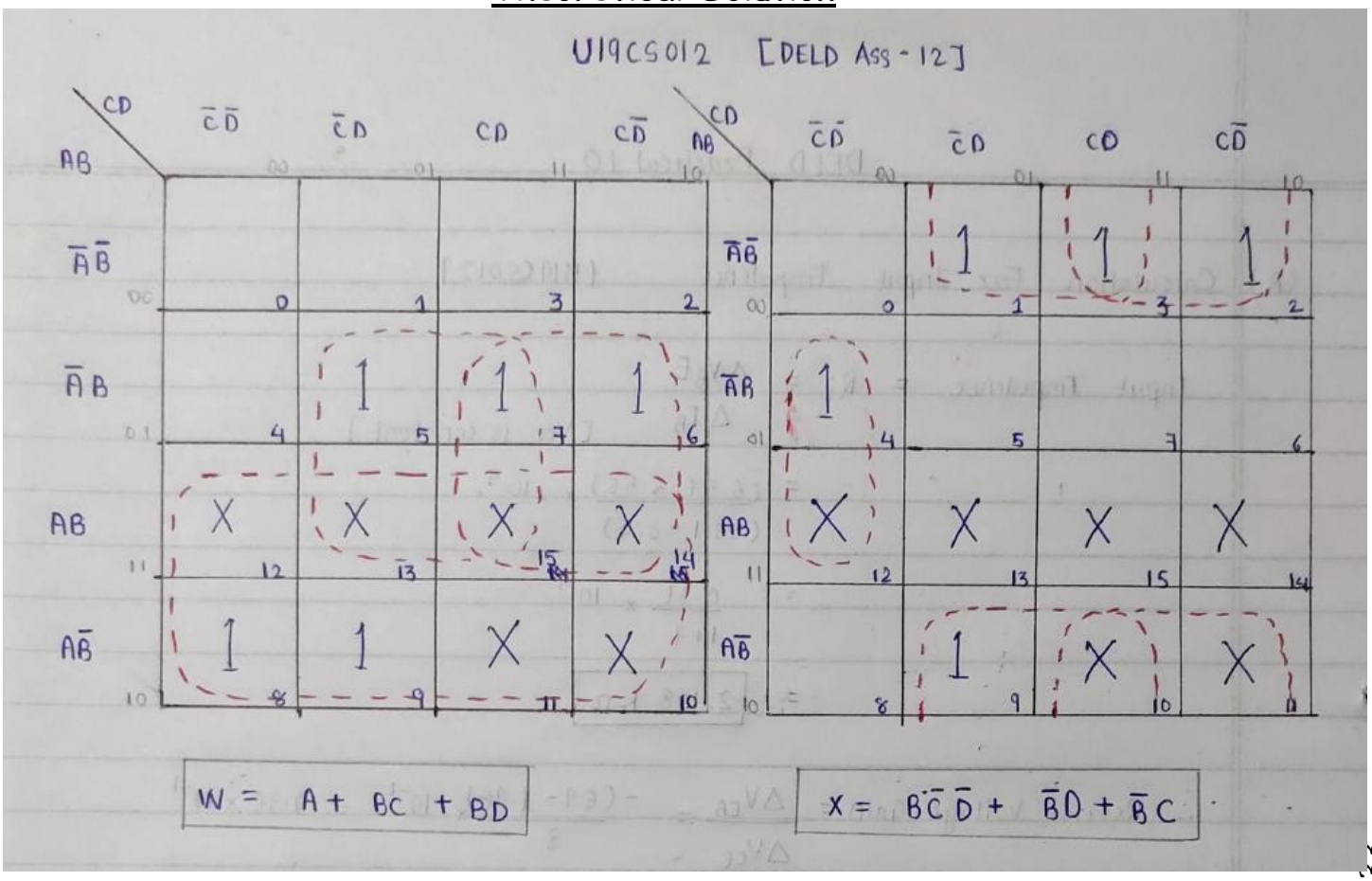
Theoretical Solution

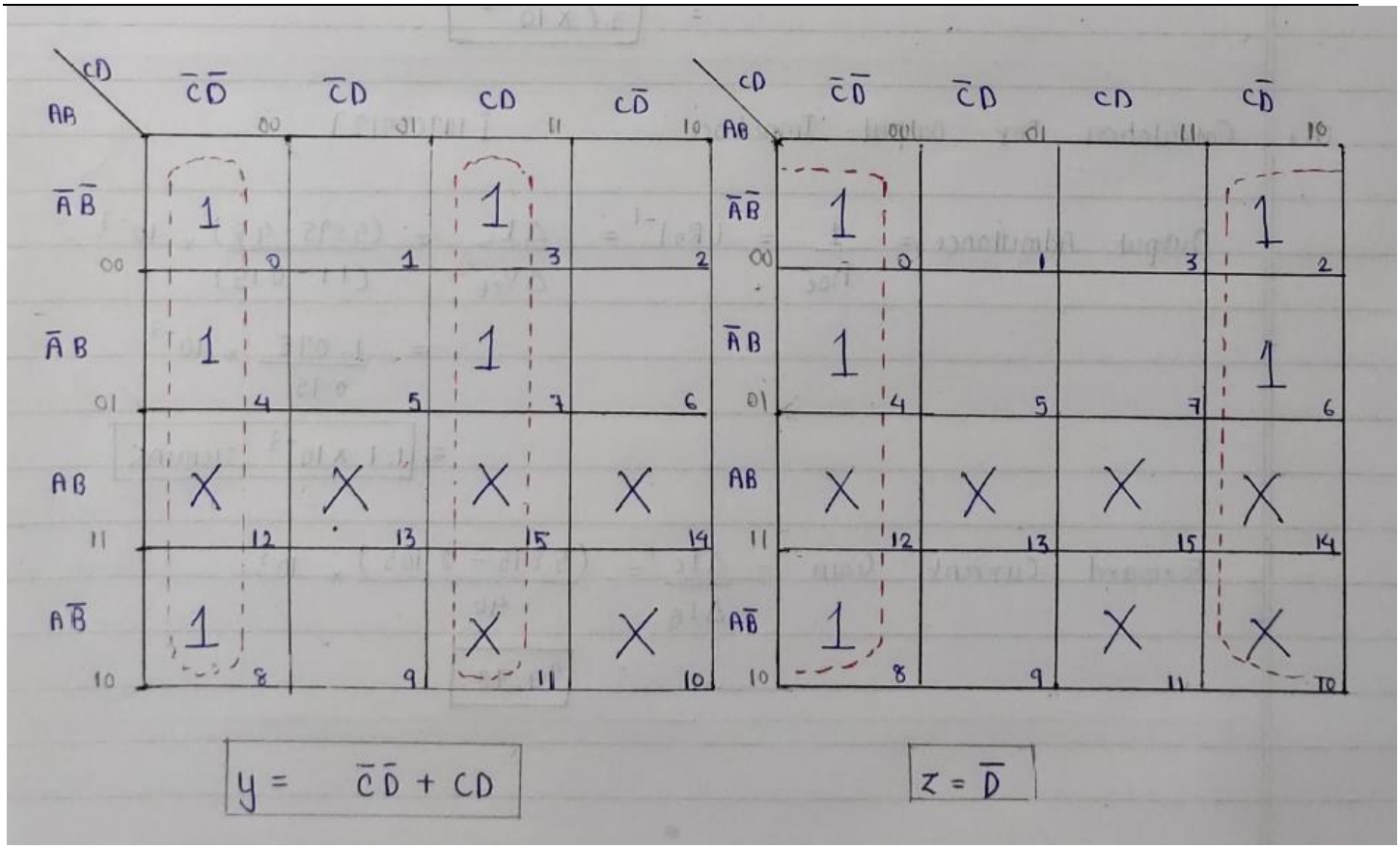


2. Design, implement and verify using Multisim: BCD to Excess – 3 Code Converter

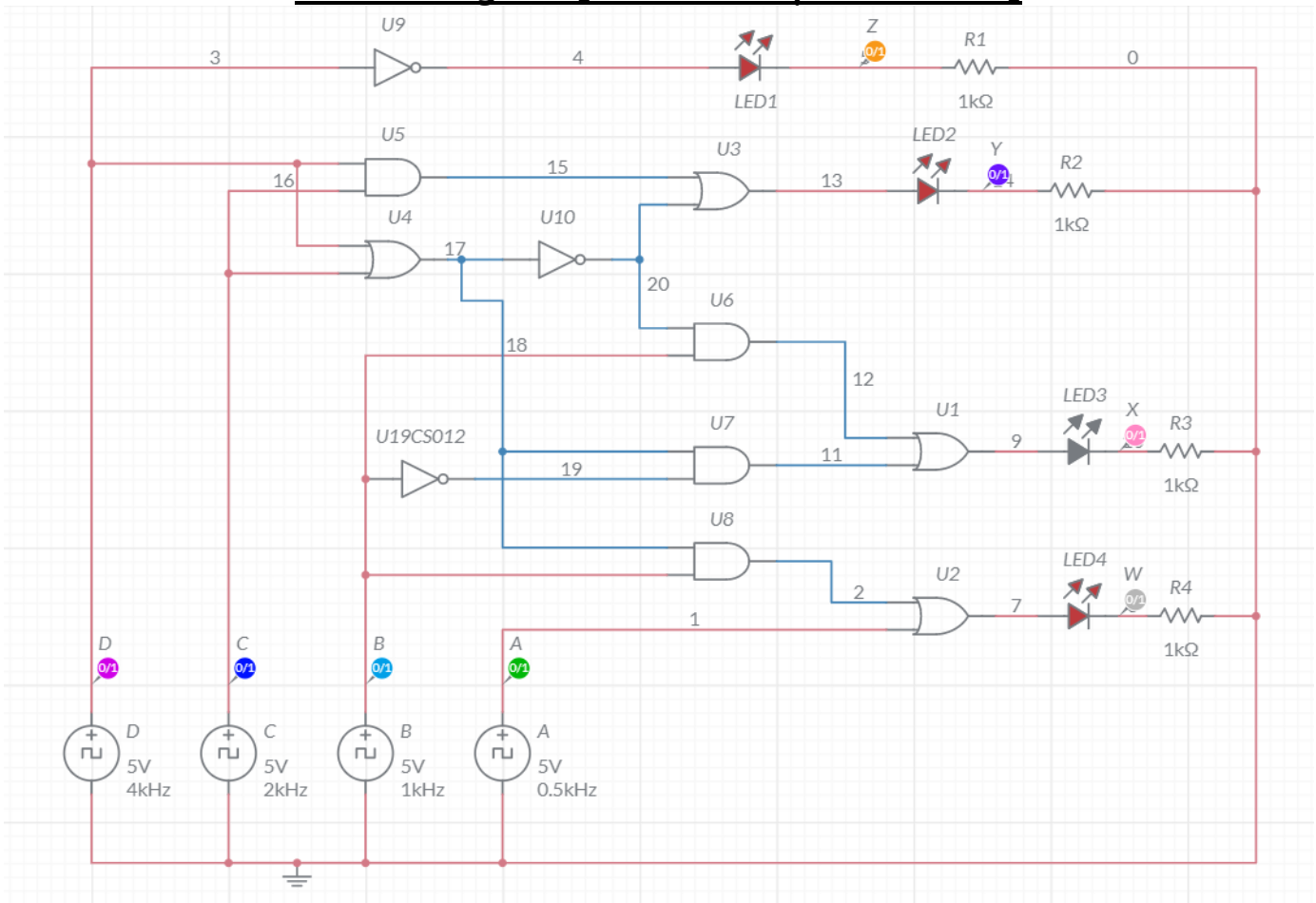
BCD(8421)				Excess-3			
A	B	C	D	w	x	y	z
0	0	0	0	0	0	1	1
0	0	0	1	0	1	0	0
0	0	1	0	0	1	0	1
0	0	1	1	0	1	1	0
0	1	0	0	0	1	1	1
0	1	0	1	1	0	0	0
0	1	1	0	1	0	0	1
0	1	1	1	1	0	1	0
1	0	0	0	1	0	1	1
1	0	0	1	1	1	0	0
1	0	1	0	X	X	X	X
1	0	1	1	X	X	X	X
1	1	0	0	X	X	X	X
1	1	0	1	X	X	X	X
1	1	1	0	X	X	X	X
1	1	1	1	X	X	X	X

Theoretical Solution

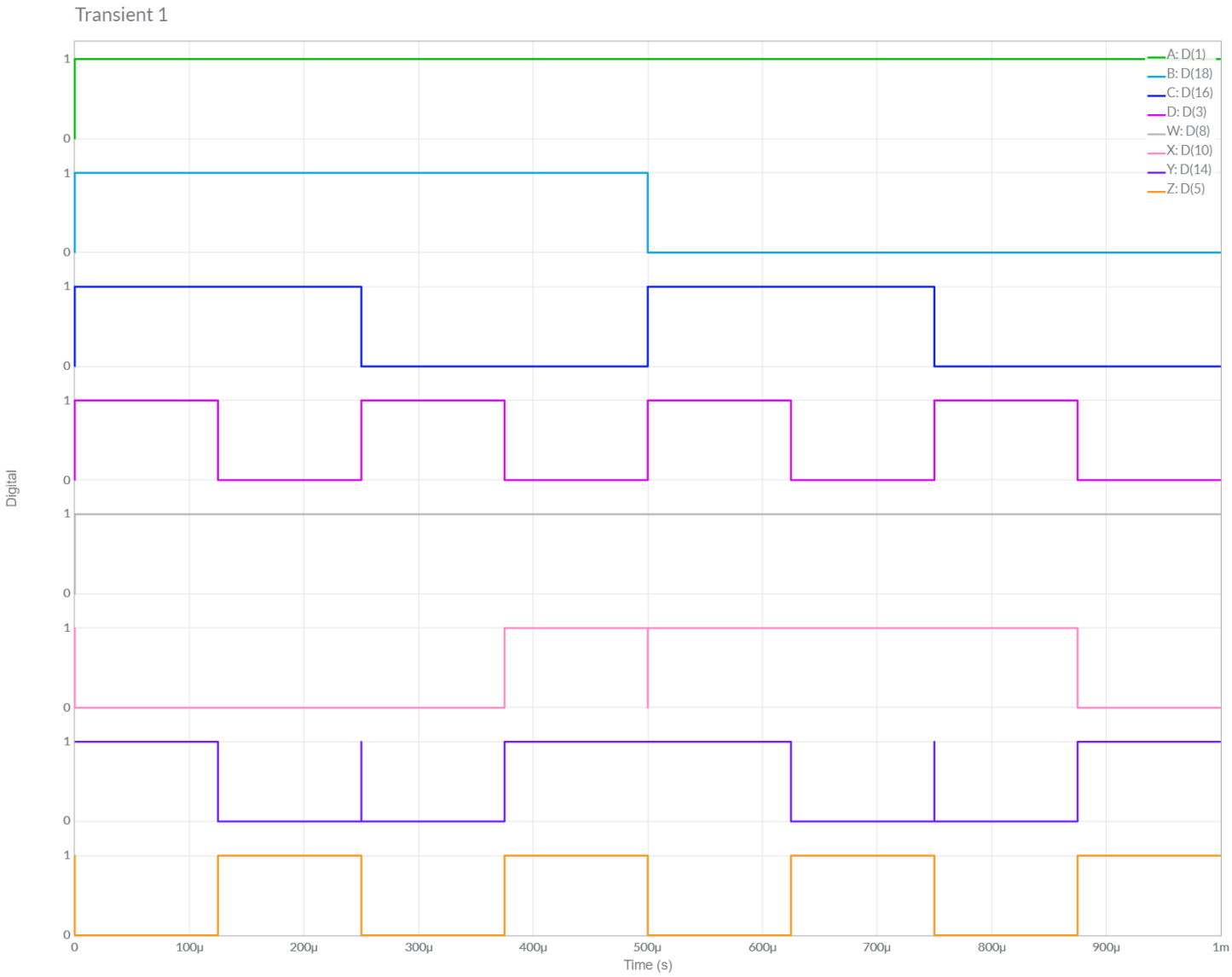




Circuit Diagram [Multisim Implementation]



Grapher Image [Transient]



D.) CONCLUSION:

We have Successfully Implemented *Particular Circuit [A'B + AC]* and *BCD to Excess-3 Convertor* and **verified** our **MULTISIM Outputs** and **Results** from *Theoretical Calculations.*

Hence Results Both Theoretical Calculation and Multisim Implementation have been verified to be ***same*** *and The Experiment has Been Successfully Performed.*