Switching Circuit and Logic Design Laboratory Report-2

by

Group-11

Devendra Palod (*20CS10024*)

Subhajyoti Halder (*20CS10064*)

Anuj Kakde (*20CS30005*)

Deepiha S. (*20CS30015*)

Department of Computer Science and Engineering

Indian Institute of Technology Kharagpur

Spring Semester, 2021-22

1 Part-1 (BCD)

1.1 Problem Statement

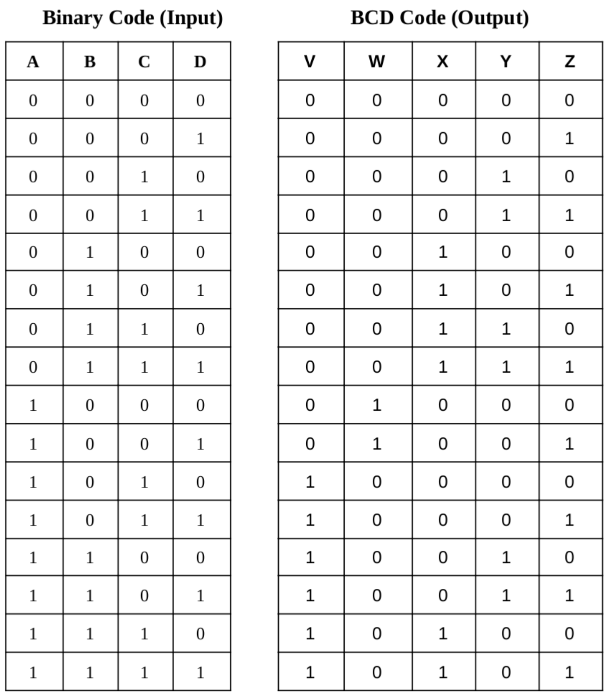
Develop circuits to convert from 4-bit binary to 2-digit BCD.

1.2 Solution Description

The BCD stands for Binary Coded Decimal Number. In BCD code, each digit of the decimal number is represented as its equivalent binary number. So, the LSB and MSB of the decimal numbers are represented as its binary numbers.

First, we converted the binary number into decimal. And then, the decimal number into BCD.

1.3 Logic Expression



Binary Code (Input)				BCD Code (Output)				
A	B	C	D	V	W	X	Y	Z
0	0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0	1
0	0	1	0	0	0	0	1	0
0	0	1	1	0	0	0	1	1
0	1	0	0	0	0	1	0	0
0	1	0	1	0	0	1	0	1
0	1	1	0	0	0	1	1	0
0	1	1	1	0	0	1	1	1
1	0	0	0	0	1	0	0	0
1	0	0	1	0	1	0	0	1
1	0	1	0	1	0	0	0	0
1	0	1	1	1	0	0	0	1
1	1	0	0	1	0	0	1	0
1	1	0	1	1	0	0	1	1
1	1	1	0	1	0	1	0	0
1	1	1	1	1	0	1	0	1

Figure 1: Truth Table for 4-bit Binary and 2-digit BCD Codes

The logical expression for each variable are as follows:
for V:

$$V = AB\bar{C}\bar{D} + AB\bar{C}D + ABCD + ABC\bar{D} + A\bar{B}CD + A\bar{B}C\bar{D}$$

$$\begin{gathered} = \mathrm{ABC}(\bar{D}+D) + AB\bar{C}(\bar{D}+D) + A\bar{B}C(\bar{D}+D) \\ = \mathrm{ABC} + \mathrm{AB}\bar{C} + A\bar{B}C \\ = \mathrm{AB(C+}\bar{C}) + A\bar{B}C \\ = \mathrm{AB} + \mathrm{A}\bar{B}C \\ = \mathrm{A(B+}\bar{B}C) \\ = \mathrm{A(B+C)(B+}\bar{B}) \\ \mathbf{V = AB + AC} \end{gathered}$$

for W:

$$W = A\bar{B}\bar{C}\bar{D} + A\bar{B}\bar{C}D$$

$$= \mathrm{A}\bar{B}\bar{C}(\bar{D}+D)$$
$$\mathbf{W} = \mathrm{A}\bar{B}\bar{C}$$

for X:

$$X = \bar{A}B\bar{C}\bar{D} + \bar{A}B\bar{C}D + \bar{A}BCD + \bar{A}BC\bar{D} + ABCD + ABC\bar{D}$$

$$= \bar{A}B\bar{C}(\bar{D}+D) + \bar{A}BC(D+\bar{D}) + ABC(D+\bar{D})$$
$$= \bar{A}B\bar{C} + \bar{A}BC + ABC$$
$$= \bar{A}B(\bar{C}+C) + ABC$$
$$= \bar{A}B + ABC$$
$$= \mathrm{B}(\bar{A} + AC)$$
$$= \mathrm{B}(\bar{A}+C)(A+\bar{A})$$
$$\mathbf{X} = \bar{A}B + BC$$

for Y:

$$Y = \bar{A}\bar{B}CD + \bar{A}\bar{B}C\bar{D} + \bar{A}BCD + \bar{A}BC\bar{D} + AB\bar{C}\bar{D} + AB\bar{C}D$$

$$= \bar{A}\bar{B}C(D+\bar{D}) + \bar{A}BC(D+\bar{D}) + AB\bar{C}(D+\bar{D})$$
$$= \bar{A}\bar{B}C + \bar{A}BC + AB\bar{C}$$
$$= \bar{A}C(\bar{B}+B) + AB\bar{C}$$
$$\mathbf{Y} = \bar{A}C + AB\bar{C}$$

for Z:

$$\mathbf{Z} = \mathbf{D}$$

1.4 Circuit Diagram

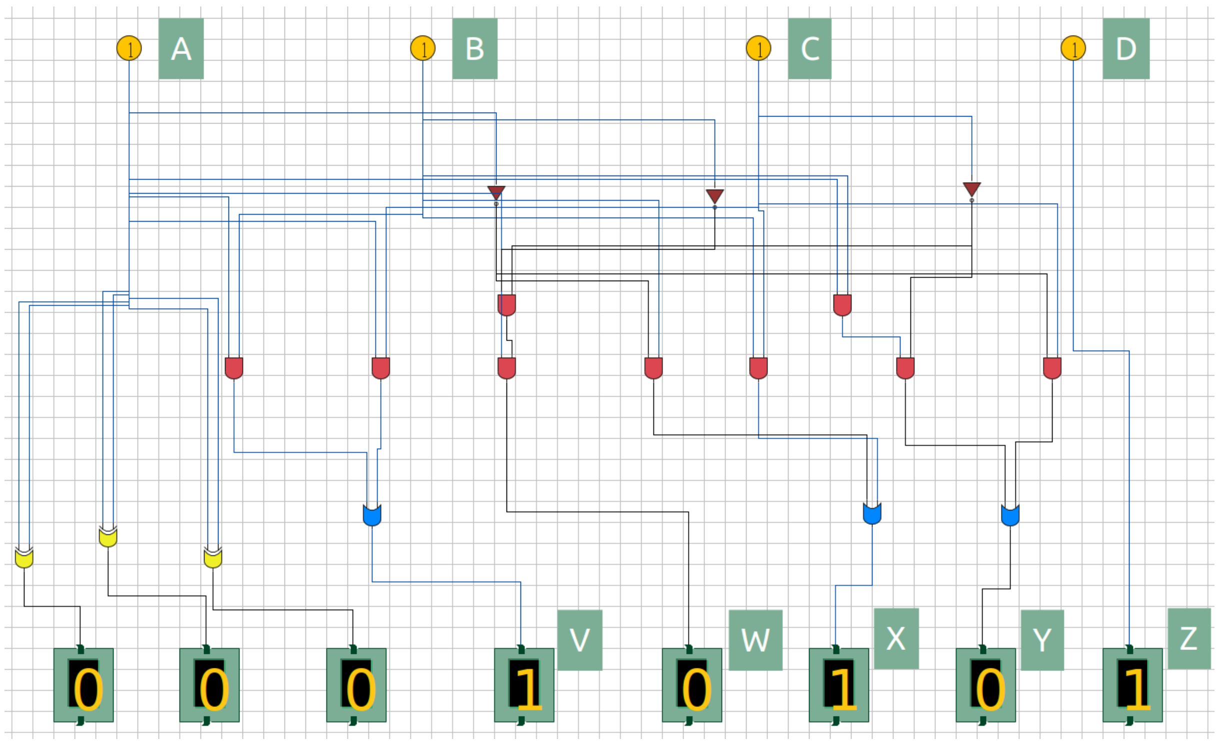


Figure 2: 4-bit Binary to 2-digit BCD Converter

2 Part-2 (Gray)

2.1 Problem Statement

Develop circuits to convert from 4-bit Gray to 4-bit binary and vice-versa

2.2 Solution Description

2.2.1 4-bit Binary to 4-bit Gray

The Binary to Gray code converter is a logical circuit that is used to convert the binary code into its equivalent Gray code. In the Gray code, the MSB will always be the same as the 1'st bit of the given binary number.

In order to perform the 2nd bit of the gray code, we perform the exclusive-or (XOR) of the 1'st and 2nd bit of the binary number. It means that if both the bits are different, the result will be one else the result will be 0.

In order to get the 3rd bit of the gray code, we need to perform the exclusive-or (XOR) of the 2nd and 3rd bit of the binary number. The process remains the same for the 4th bit of the Gray code.

2.2.2 4-bit Gray to 4-bit Binary

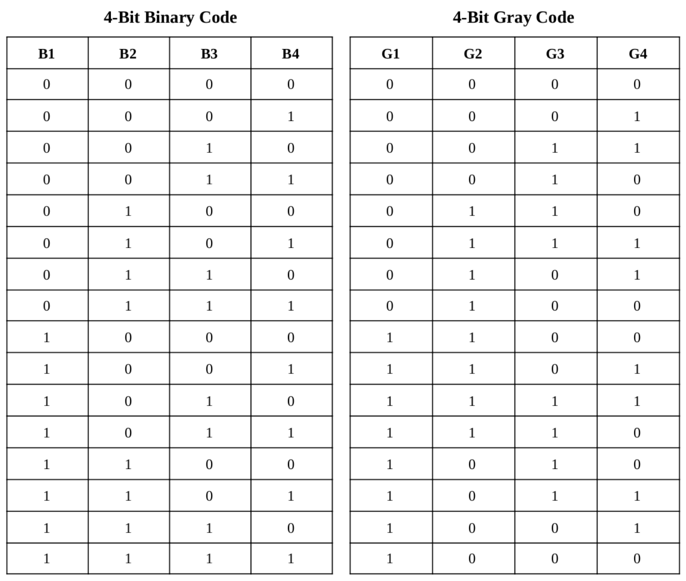
The Gray to Binary code converter is a logical circuit that is used to convert the gray code into its equivalent binary code. Just like binary to gray, in gray to binary, the 1st bit of the binary number is similar to the MSB of the Gray code.

In order to perform the 2nd bit of the binary code, we perform the exclusive-or (XOR) of the 1st and 2nd bit of the gray number.

In order to get the 3rd bit of the binary code, we need to perform the exclusive-or (XOR) of the 1st, 2nd and 3rd bit of the gray number or simply the exclusive-or (XOR) of the 2nd bit of binary number which we have found and 3rd bit of the gray number.

In order to get the 4th bit of the binary code, we need to perform the exclusive-or (XOR) of the 3rd bit of binary number which we have found and 4th bit of the gray number.

2.3 Logic Expression



4-Bit Binary Code				4-Bit Gray Code			
B1	B2	B3	B4	G1	G2	G3	G4
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	1
0	0	1	1	0	0	1	0
0	1	0	0	0	1	1	0
0	1	0	1	0	1	1	1
0	1	1	0	0	1	0	1
0	1	1	1	0	1	0	0
1	0	0	0	1	1	0	0
1	0	0	1	1	1	0	1
1	0	1	0	1	1	1	1
1	0	1	1	1	1	1	0
1	1	0	0	1	0	1	0
1	1	0	1	1	0	1	1
1	1	1	0	1	0	0	1
1	1	1	1	1	0	0	0

Figure 3: Truth Table for Binary and Gray Codes

The logical expression for each variable are as follows:

2.3.1 For Binary to Gray Conversion:

$$G_1 = B_1$$

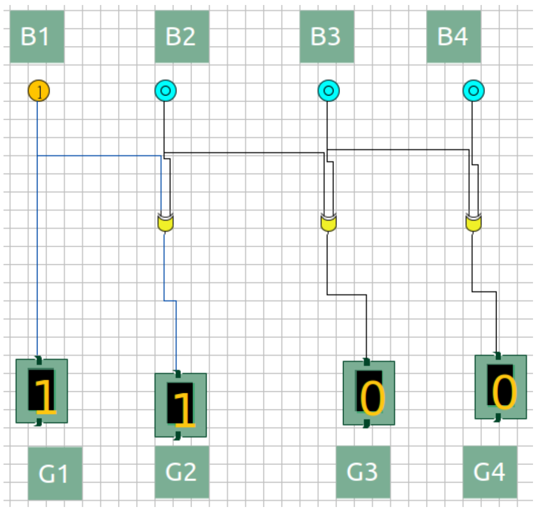
$$\mathrm{G}_i = B_i \oplus B_{i-1} \qquad for \;\; i = 2,3,4$$

2.3.2 For Gray to Binary Conversion:

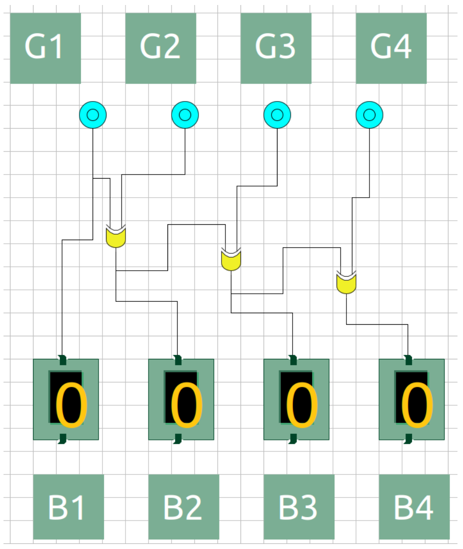
$$B_1 = G_1$$

$$\mathrm{B}_i = (G_1 \oplus ... \oplus G_{i-1}) \oplus G_i = B_{i-1} \oplus G_i$$

2.4 Circuit Diagram



(a) 4-bit Binary to 4-bit Gray Convertor



(b) 4-bit Gray to 4-bit Binary Convertor

3 Part-3 (Excess-3)

3.1 Problem Statement

1. Develop a half adder for handling two bits
2. Develop a full adder using half adders and any additional logic
3. Develop a ripple carry adder needed for this assignment using full adders
4. Develop circuits to convert from excess-3 to 4-bit binary and vice-versa

3.2 Solution Description

3.2.1 Half Adder

A half adder is an adder which adds two binary digits together, resulting in a sum and a carry.

3.2.2 Full Adder

Full Adder is the adder which adds three inputs and produces two outputs. The first two inputs are X and Y and the third input is an input carry as Z. The output carry is designated as CARRY and the normal output is designated as SUM.

2 Half Adders and a OR gate is required to implement a Full Adder.

3.2.3 Ripple Carry Adder

A ripple carry adder is a logic circuit in which the carry-out of each full adder is the carry in of the succeeding next most significant full adder. It is called a ripple carry adder because each carry bit gets rippled into the next stage.

3.2.4 4-bit Binary to 4-bit Excess-3

Excess-3 codes can be obtained by adding 3 to each decimal digit then it can be represented by using 4 bit binary number for each digit. An Excess-3 equivalent of a given binary binary number is obtained using the following steps: Find the decimal equivalent of the given binary number. Add +3 to each digit of decimal number. Convert the newly obtained decimal number back to binary number to get required excess-3 equivalent.

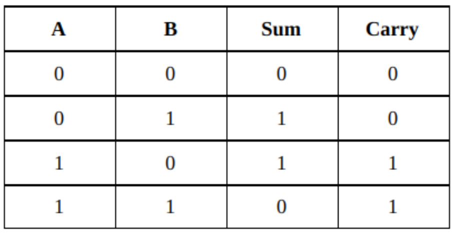
3.3 Logic Expression

3.3.1 Half Adder

$$\begin{aligned} \text{Sum} &= AB' + A'B \\ &= A \text{ xor } B \\ \text{Carry} &= AB \end{aligned}$$

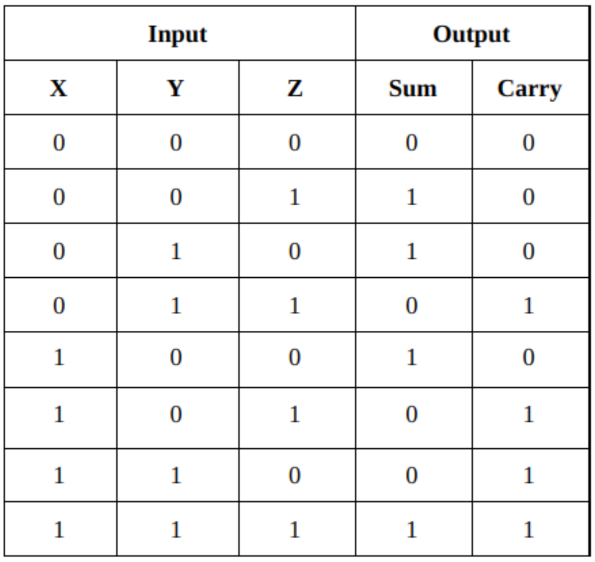
Figure 5: Logic Expression for Half Adder Circuit

3.3.2 Full Adder



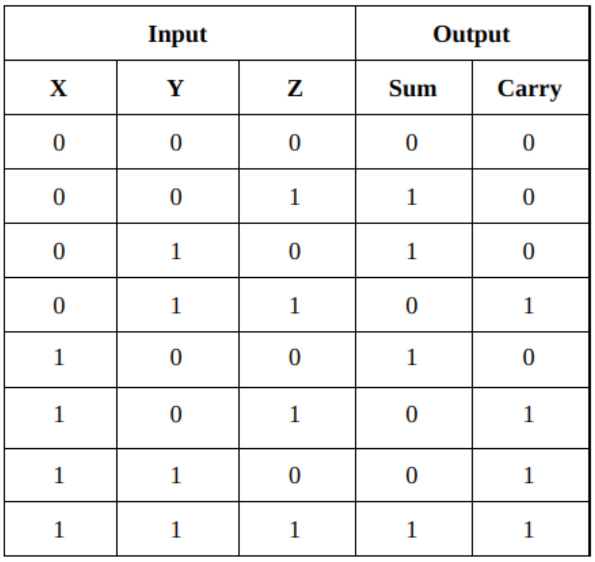
A	B	Sum	Carry
0	0	0	0
0	1	1	0
1	0	1	1
1	1	0	1

Figure 6: Truth Table for Half Adder Circuit



Input			Output	
X	Y	Z	Sum	Carry
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Figure 7: Logic Expression for Full Adder Circuit

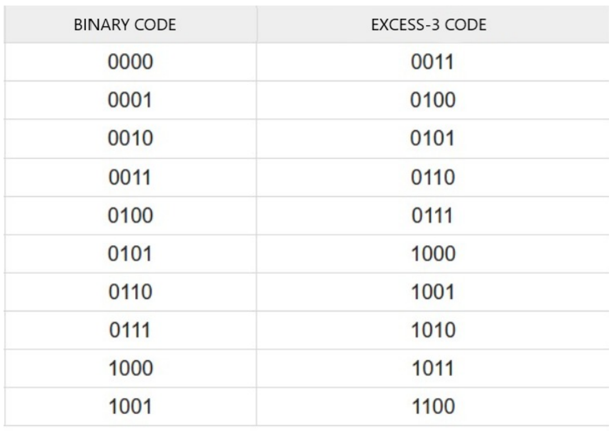


Input			Output	
X	Y	Z	Sum	Carry
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Figure 8: Truth Table for Full Adder Circuit

A1	A2	A3	A4	B4	B3	B2	B1	S4	S3	S2	S1	Carry
0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	1	0	0	1	0	0	0	0
1	0	0	0	1	0	0	0	0	0	0	0	1
1	0	1	0	1	0	1	0	0	1	0	0	1
1	1	0	0	1	1	0	0	1	0	0	0	1
1	1	1	0	1	1	1	0	1	1	0	0	1
1	1	1	1	1	1	1	1	1	1	1	0	1

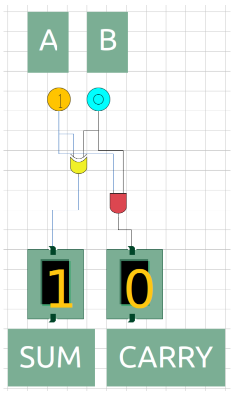
Figure 9: Truth Table for Ripple Carry Adder Circuit



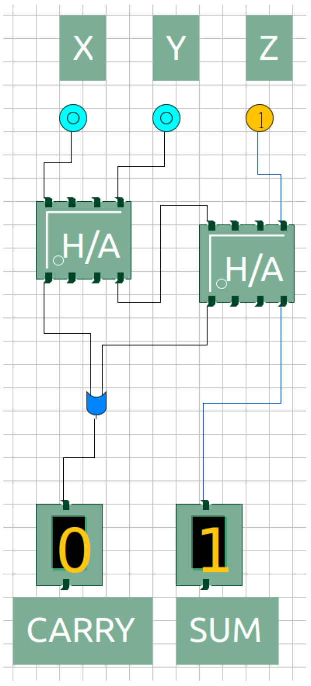
BINARY CODE	EXCESS-3 CODE
0000	0011
0001	0100
0010	0101
0011	0110
0100	0111
0101	1000
0110	1001
0111	1010
1000	1011
1001	1100

Figure 10: Truth Table for BCD Codes to Excess-3

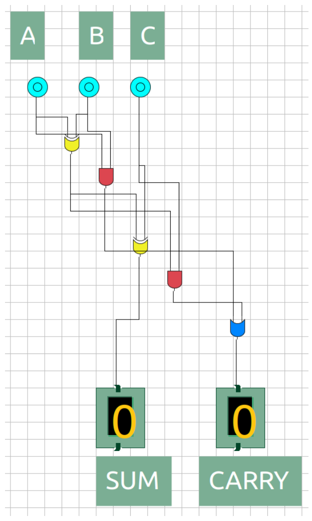
3.4 Circuit Diagram



(a) Half-Adder



(b) Full (Using Half-Adder)



(c) Full Adder

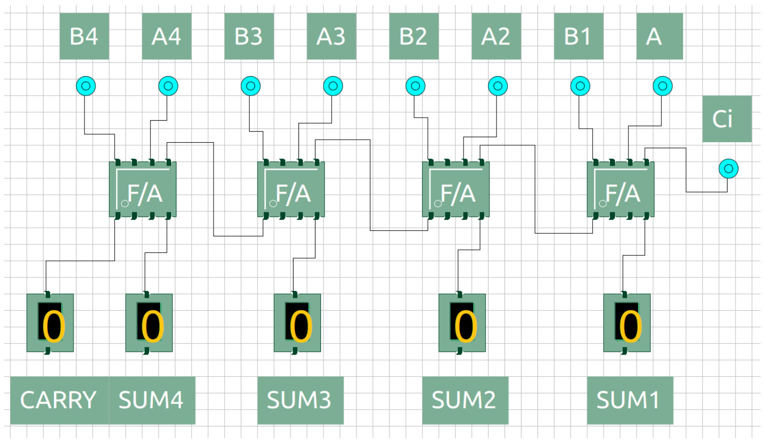
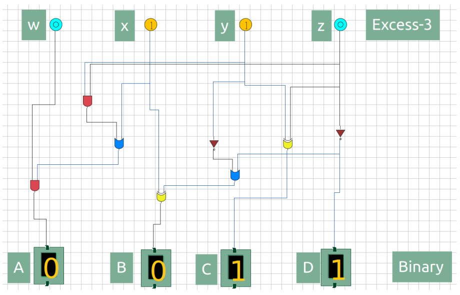
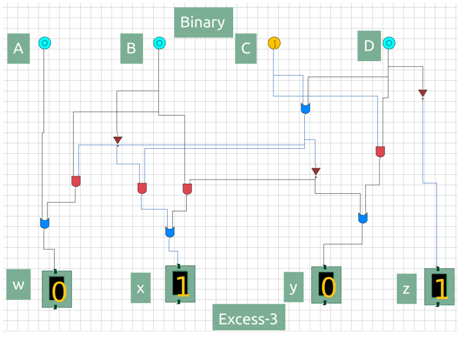


Figure 12: Ripple Carry Adder



(a) Excess-3 to Binary



(b) Binary to Excess-3