INFORMATICS INSTITUTE OF TECHNOLOGY

Foundation Certificate in Higher Education

Module: DOC325-Digital Circuits and Logic Design

Module Leader: Ms. Aathika Salam

Assignment Type: Group Coursework

Submission date: 03rd December 2023

Group: E

Group ID: E2

Student ID	Student Name
20230297	**Mr. Anusigan Sivananthan**
20230176	**Mr. Lithila Mahagedara**
20222090	**Mr. N. T. Welagedara**
20230229	**Mr. M.H.A Daneesha Hansaka**
20230148	**Ms. Senumi Dahanayake**

I. Acknowledgment

In the preparation of our group coursework, a few respected individuals helped us to complete the group course work and those individuals deserve our appreciation. We would like to convey our exceptional thanks to Ms. Chathuri Udagedara, Our Tutorial instructor. She gave us clear guidance which helped us to complete our group coursework. Also we would like to thank convey our exceptional thanks to our Lecturer Ms.Aathika Salam for the guidance. The group coursework was completed with the efforts of our group members Anusigan, Lithila, Daneesha, Namina, and Senumi who interactively participated and helped each other to complete the group coursework on time. We would like to express our gratitude all those who helped us to accomplish this report.

II. Table of Contents

I. Acknowledgment 1
II. Table of Contents 2
III. List of Tables 3
IV. List of Figures 3
1. Question 1- Real-World Digital Circuit Application and Incident Analysis 4
1.1 Problem and Solution 4
1.2 Inputs and Outputs 5
1.3 Truth table for the defined problem 5
1.4 Output in SOP and POS form 6
1.4.1 Output (X) 6
1.4.2 Output (Y) 7
1.5 Simplification using K-map and Boolean algebra laws 8
1.5.1 For Output (X) 8
1.5.2 For Output (Y) 9
1.6 Logic circuit of the solution 10
1.7 Logic circuit Simulation 11
2.Question 2- Multiplexer/Demultiplexer 12
2.1 Schematic Diagram and Truth tables 12
2.2 Expression for the outputs 16
2.3 Logic circuit Simulation 17
2.3.1 Multiplexer 17
2.3.2 Demultiplexer 18
3. Adders 18
3.1 simulation of full adder using half adders and how it works 18
3.2 Creation of 'Ripple Carry Adder' and its capability 23
4.References 25
5.Workload matrix 26

III. List of Tables

Table 1-Inputs of the solution 5
Table 2-Outputs of the solution 5
Table 3-Truth table of the problem 5
Table 4- Minterm Maxterm Table of Output (X) 6
Table 5- Minterm Maxterm Table of Output (Y) 7
Table 6-Truth table of Multiplexer 13
Table 7-Truth table of Demultiplexer 15
Table 8-Truth table for half adder 18
Table 9-Truth Table for full adder 21

IV. List of Figures

Figure 1-Logic circuit design of the solution 10
Figure 2-Logic circuit simulation of the solution 11
Figure 3-Schematic Diagram 16X1 Multiplexer 12
Figure 4-Logic circuit for 16X1 Multiplexer 12
Figure 5-Schematic Diagram of 1X16 Demultiplexer 14
Figure 6-Logic circuit of 16X1 Demultiplexer 14
Figure 7-Logic circuit simulation of 16X1 Multiplexer 17
Figure 8-Logic circuit simulation of 1X16 Demultiplexer 18
Figure 9-Logic circuit diagram of half adder 19
Figure 10-Logic circuit simulation for half adder 20
Figure 11-Logic circuit for Full adder 22
Figure 12-Logic circuit simulation of Full adder 22
Figure 13-Block Diagram of Ripple carry adder 24
Figure 14-Logic circuit simulation for 6-bit ripple carry adder 24

1. Question 1- Real-World Digital Circuit Application and Incident Analysis

1.1 Problem and Solution

Problem

Agricultural lands require a proper irrigation system to maintain the soil moisture and to maintain the healthiness of crops. Statistics states that Lack of irrigation in summer times made the agriculture lands drought and Excess irrigation resulted in the destruction of agricultural lands and decreased the productivity of crops.

Actual Scenario

Sri Lanka planted 1.3 million acres (526,091 hectares) for the summer harvest, according to the agriculture ministry.

"We have lost at least 80,000 metric tonnes of paddy as per the latest data and it could be more," said Buddhi Marambe, professor of crop science at Sri Lanka's Peradeniya University.Experts say they are worried that if Sri Lanka does not receive the rains it needs next March because of the continuing El Nino, the country will be left with scant reserve stocks and will have to resort to large-scale, expensive imports.Rice is the staple food of the country's 22 million people and its biggest crop. According to government data, two million people in the country are rice farmers out of 8.1 million people engaged in fishing and agriculture in the largely rural economy.The drought has also wiped out the small chilli, peanut and banana plants Seneviratne's wife, WM Makamma, 62 grows to feed her family.The paddy loss could be as much as 75,000 acres (30,500 hectares), according to Agriculture Minister Mahinda Amaraweera, while other experts say full losses could be even higher as estimates are yet to be completed. (Aljazeera,2023)

Due to a critical shortage of seeds and a lack of water for irrigation, the second 2017 paddy harvest – known as Yale, due to be harvested in August and September – is forecast at 1.2 million tonnes, 24 percent below last year's level. (WFP,2017)

Solution

We came up with a 3 input and 2 output combinational circuit solution which can be implemented to create a Smart Irrigation System Control. This system considers soil moisture level, Rainfall, Temparature as inputs to activate the irrigation system and also irrigation can be activated through manual override input. Another alarm signal will be activated if the soil moisture level and Temperature of the environment are below threshold.

1.2 Inputs and Outputs

Inputs	Boolean Variable
Soil moisture level below threshold (Below 500kPa)	S
Temperature of the environment above threshold (Above 32^0C)	T
No Rainfall	R
Manual override (Activates the irrigation system manually)	M

Table 1-Inputs of the solution

Outputs	Boolean Variable
Irrigation Activation	X
Alarm Activation	Y

Table 2-Outputs of the solution

1.3 Truth table for the defined problem

M	S	R	T	SRT	X	Y
0	0	0	0	0	0	0
0	0	0	1	0	0	0
0	0	1	0	0	0	0
0	0	1	1	0	0	0
0	1	0	0	0	0	0
0	1	0	1	0	0	1
0	1	1	0	0	0	0
0	1	1	1	1	1	1
1	0	0	0	0	1	0
1	0	0	1	0	1	0
1	0	1	0	0	1	0
1	0	1	1	0	1	0
1	1	0	0	0	1	0
1	1	0	1	0	1	1
1	1	1	0	0	1	0
1	1	1	1	1	1	1

Table 3-Truth table of the problem

1.4 Output in SOP and POS form

1.4.1 Output (X)

M	S	R	T	X	MINTERM	MAXTERM
0	0	0	0	0	M’.S’.R’.T’	M+S+R+T
0	0	0	1	0	M’.S’.R’.T	M+S+R+T’
0	0	1	0	0	M’.S’.R.T’	M+S+R’+T
0	0	1	1	0	M’.S’.R.T	M+S+R’+T’
0	1	0	0	0	M’.S.R’.T’	M+S’+R+T
0	1	0	1	0	M’.S.R’.T	M+S’+R+T’
0	1	1	0	0	M’.S.R.T’	M+S’+R’+T
0	1	1	1	1	M’.S.R.T	M+S’+R’+T’
1	0	0	0	1	M.S’.R’.T’	M’+S+R+T
1	0	0	1	1	M.S’.R’.T	M’+S+R+T’
1	0	1	0	1	M.S’.R.T’	M’+S+R’+T
1	0	1	1	1	M.S’.R.T	M’+S+R’+T’
1	1	0	0	1	M.S.R’.T’	M’+S’+R+T
1	1	0	1	1	M.S.R’.T	M’+S’+R+T’
1	1	1	0	1	M.S.R.T’	M’+S’+R’+T
1	1	1	1	1	M.S.R.T	M’+S’+R’+T’

Table 4- Minterm Maxterm Table of Output (X)

SOP: (M’.S.R.T) + (M.S’.R’.T’) + (M.S’.R’.T) + (M.S’.R.T’) + (M.S’.R.T) + (M.S.R’.T’) + (M.S.R’.T) + (M.S.R.T’) + (M.S.R.T)

POS: (M+S+R+T) . (M+S+R+T’) . (M+S+R’+T) . (M+S+R’+T’) . (M+S’+R+T) . (M+S’+R+T’) . (M+S’+R’+T)

1.4.2 Output (Y)

M	S	R	T	Y	MINTERM	MAXTERM
0	0	0	0	0	M’.S’.R’.T’	M+S+R+T
0	0	0	1	0	M’.S’.R’.T	M+S+R+T’
0	0	1	0	0	M’.S’.R.T’	M+S+R’+T
0	0	1	1	0	M’.S’.R.T	M+S+R’+T’
0	1	0	0	0	M’.S.R’.T’	M+S’+R+T
0	1	0	1	1	M’.S.R’.T	M+S’+R+T’
0	1	1	0	0	M’.S.R.T’	M+S’+R’+T
0	1	1	1	1	M’.S.R.T	M+S’+R’+T’
1	0	0	0	0	M.S’.R’.T’	M’+S+R+T
1	0	0	1	0	M.S’.R’.T	M’+S+R+T’
1	0	1	0	0	M.S’.R.T’	M’+S+R’+T
1	0	1	1	0	M.S’.R.T	M’+S+R’+T’
1	1	0	0	0	M.S.R’.T’	M’+S’+R+T
1	1	0	1	1	M.S.R’.T	M’+S’+R+T’
1	1	1	0	0	M.S.R.T’	M’+S’+R’+T
1	1	1	1	1	M.S.R.T	M’+S’+R’+T’

Table 5- Minterm Maxterm Table of Output (Y)

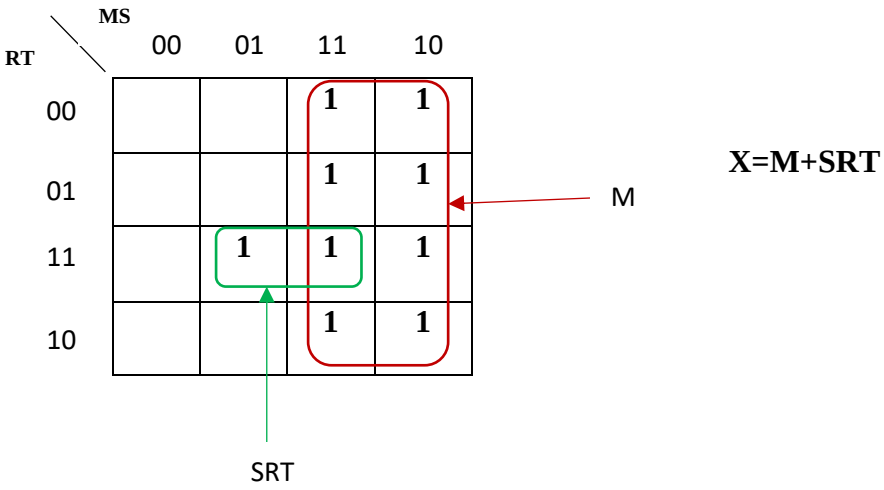
SOP: (M’.S.R’.T) + (M’.S.R.T) + (M.S.R’.T) + (M.S.R.T)

POS: (M+S+R+T) . (M+S+R+T’) . (M+S+R’+T) . (M+S+R’+T’) . (M+S’+R+T) . (M+S’+R’+T) . (M’+S+R+T) . (M’+S+R+T’) . (M’+S+R’+T) . (M’+S+R’+T’) . (M’+S’+R+T) . (M’+S’+R’+T)

1.5 Simplification using K-map and Boolean algebra laws

1.5.1 For Output (X)

Using K-map



X=M+SRT

Using Boolean algebra laws

(M'.S.R.T) + (M.S'.R'.T') + (M.S'.R'.T) + (M.S'.R.T') + (M.S'.R.T) +(M.S.R'.T') + (M.S.R'.T) + (M.S.R.T') + (M.S.R.T)

SRT (M'+M) + (M.S'.R'.T') + (M.S'.R'.T) + (M.S'.R.T') + (M.S'.R.T) + (M.S.R'.T') + (M.S.R'.T) + (M.S.R.T') (Distributive Law)

SRT (1) + (M.S'.R'.T') + (M.S'.R'.T) + (M.S'.R.T') + (M.S'.R.T) + (M.S.R'.T') + (M.S.R'.T) + (M.S.R.T') (Complement Law)

SRT + (M.S'.R'.T') + (M.S'.R'.T) + (M.S'.R.T') + (M.S'.R.T) + (M.S.R'.T') + (M.S.R'.T) + (M.S.R.T') (Identity Law)

SRT + M.S'(R.T'+R') + (M.S'.R.T) + (M.S.R'.T') + (M.S.R'.T) + (M.S.R.T') (Distributive Law)

SRT + MS'R'(1) + MS'RT'+MS'RT+MSR'T'+MSR'T+MSRT' (Complement Law)

SRT + MS'R' + MS'RT'+MS'RT+MSR'T'+MSR'T+MSRT' (Identity Law)

SRT + MS'(RT'+R') + MS'RT +MSR'T' + MSR'T +MSRT' (Distributive Law)

SRT + M.S' (T'+R') + (M.S'.R.T) + (M.S.R'.T') + (M.S.R'.T) + (M.S.R.T') (Absorption Law)

M.S'(T'+R') + R.T (M.S'+S) + (M.S.R'.T') + (M.S.R'.T) + (M.S.R.T') (Distributive Law)

M.S'(T'+R') + R.T (M+S) + (M.S.R'.T') + (M.S.R'.T) + (M.S.R.T') (Absorption Law)

M.S’(T’+R’) + R.T (M+S) + M.S.R’(T’+T) + M.S.R.T’ (Distributive Law)

M.S’(T’+R’) + R.T(M+S) +M.S.R’(1) + M.S.R.T’ (Complement Law)

M.S’(T’+R’) + R.T(M+S) + M.S.R’ + M.S.R.T’ (Identity Law)

M.S’(T’+R’) + R.T (M+S) + M.S (R.T’+R’) (Distributive Law)

M.S’(T’+R’) + R.T(M+S) + M.S (T’+R’) (Absorption Law)

M(T’+R’) (S’+S) + R.T(M+S) (Distributive Law)

M(T’+R’) (1) + R.T (M+S) (Complement Law)

M(T’+R’) + R.T (M+S) (Identity Law)

(M.T’) + (M.R’) + R.T (M+S) (Distribution)

(M.T’) + (M.R’) + (R.T.M) + (R.T.S) (Distribution)

(M.R’) + M(R.T+T’) + (R.T.S) (Distributive Law)

(M.R’) + M(R+T’) + (R.T.S) (Absorption Law)

(M.R’) + (M.R) + (M.T’) + (R.T.S) (Distribution)

M(R’+R) + (M.T’) + (R.T.S) (Distributive Law)

M (1) + (M.T’) + (R.T.S) (Complement Law)

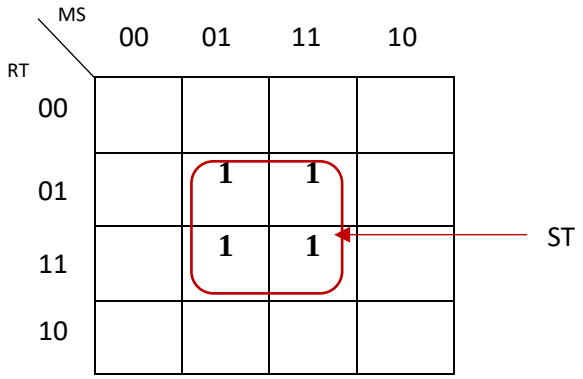
M + (M.T’) + (R.T.S) (Identity Law)

M+(SRT) (Absorption Law)

X= M+SRT

1.5.2 For Output (Y)

Using K-map



Y=ST

Using Boolean algebra laws

(M'.S.R.T) + (M'.S.R.T) + (M.S.R'.T) + (M.S.R.T)

M'.S.T (R'+R) + (M.S.R'.T) + (M.S.R.T) (Distributive Law)

M'.S.T (1) + (M.S.R'.T) + (M.S.R.T) (Complement Law)

(M'.S.T) + (M.S.R'.T) + (M.S.R.T) (Identity Law)

S.T (M.R'+M') + (M.S.R.T) (Distributive Law)

S.T (R'+M') + (M.S.R.T) (Absorption Law)

(S.T.R') + (S.T.M') + (M.S.R.T) (Distribution)

(S.T.M') + ST(M.R+R') (Distributive Law)

(S.T.M') + S.T (M+R') (Absorption Law)

(S.T.M') + (S.T.M) + (S.T.R') (Distribution)

S.T(M'.M) + (S.T.R') (Distributive Law)

S.T (1) + (S.T.R') (Complement Law)

(S.T) + (S.T.R') (Identity Law)

(S.T) (Absorption Law)

Y= ST

1.6 Logic circuit of the solution

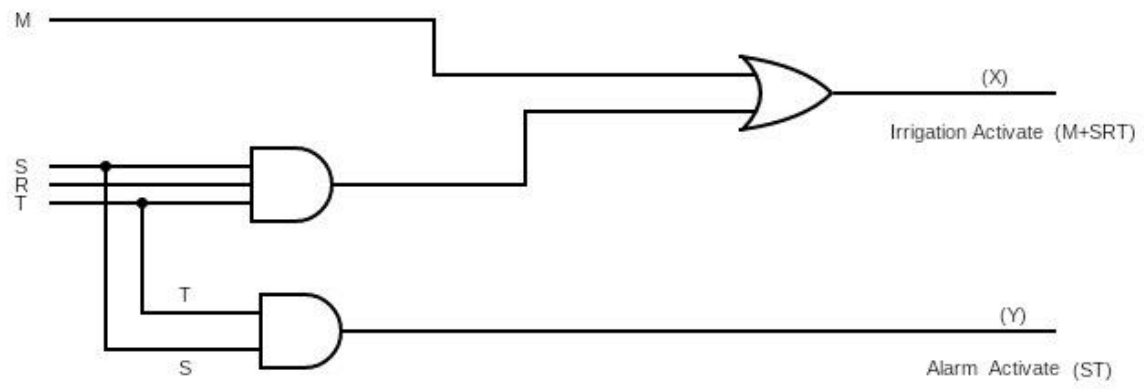


Figure 1-Logic circuit design of the solution

1.7 Logic circuit Simulation

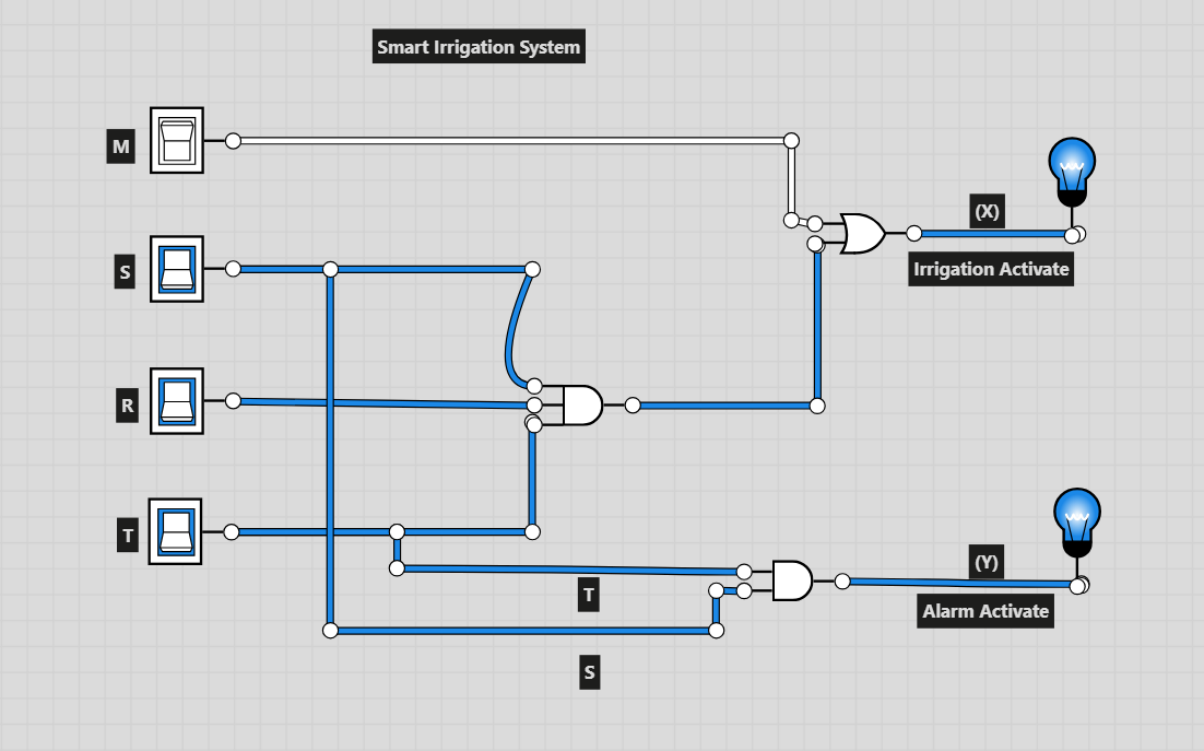


Figure 2-Logic circuit simulation of the solution

2.Question 2- Multiplexer/Demultiplexer

2.1 Schematic Diagram and Truth tables

For Multiplexer

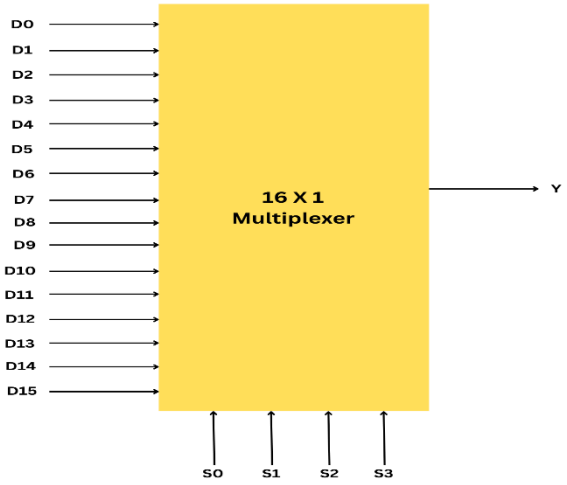


Figure 3-Schematic Diagram 16X1 Multiplexer

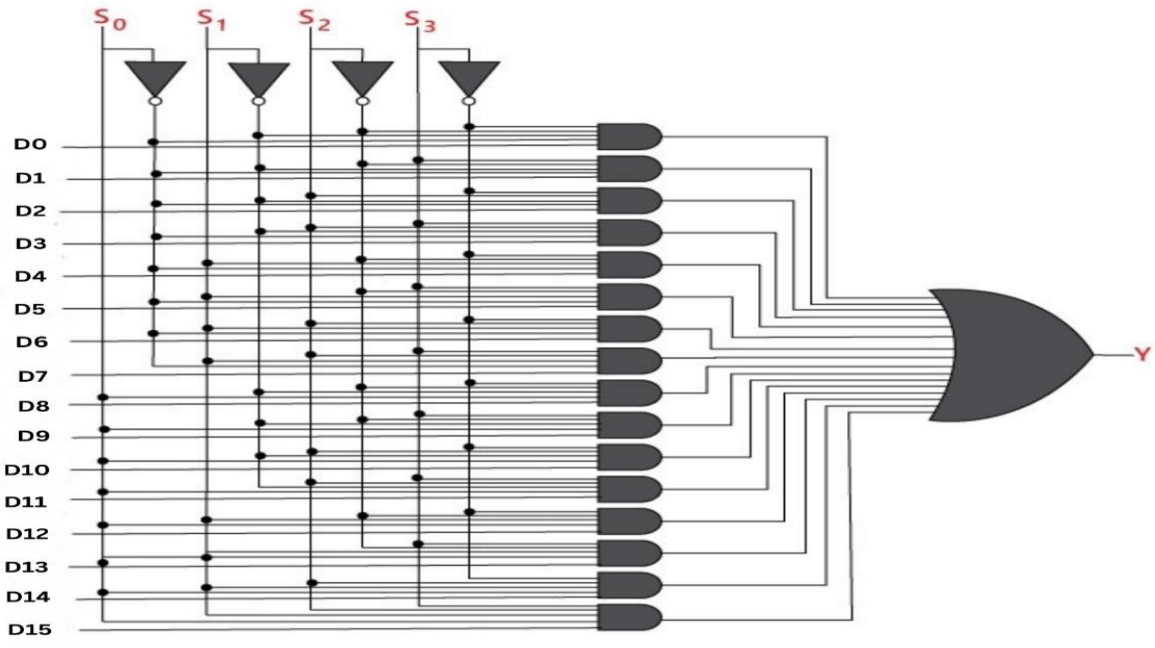


Figure 4-Logic circuit for 16X1 Multiplexer

S_0	S_1	S_2	S_3	Y
0	0	0	0	D_0
0	0	0	1	D_1
0	0	1	0	D_2
0	0	1	1	D_3
0	1	0	0	D_4
0	1	0	1	D_5
0	1	1	0	D_6
0	1	1	1	D_7
1	0	0	0	D_8
1	0	0	1	D_9
1	0	1	0	D_{10}
1	0	1	1	D_{11}
1	1	0	0	D_{12}
1	1	0	1	D_{13}
1	1	1	0	D_{14}
1	1	1	1	D_{15}

Table 6-Truth table of Multiplexer

For Demultiplexer

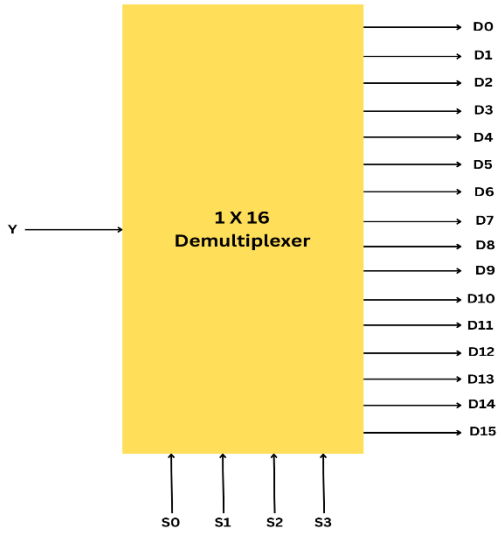


Figure 5-Schematic Diagram of 1X16 Demultiplexer

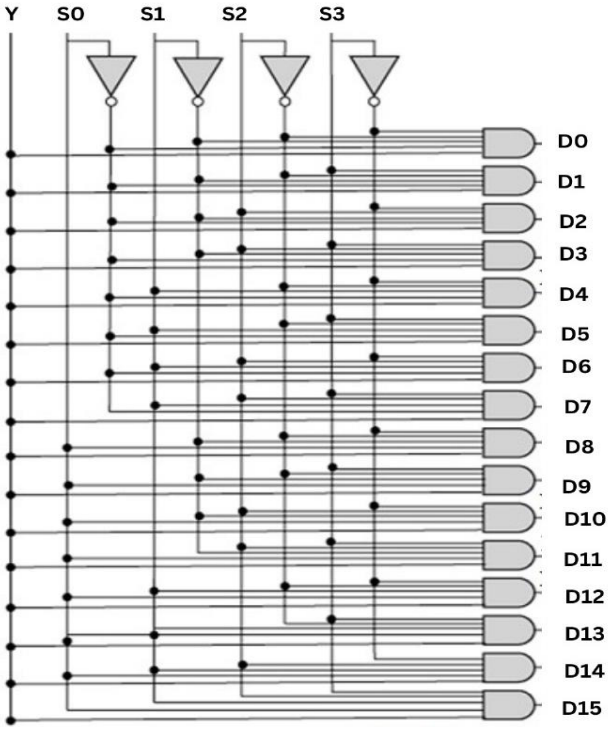


Figure 6-Logic circuit of 16X1 Demultiplexer

S_0	S_1	S_2	S_3	D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7	D_8	D_9	D_{10}	D_{11}	D_{12}	D_{13}	D_{14}	D_{15}
0	0	0	0	Y	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	0	Y	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	Y	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	1	0	0	0	Y	0	0	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	Y	0	0	0	0	0	0	0	0	0	0	0
0	1	0	1	0	0	0	0	0	Y	0	0	0	0	0	0	0	0	0	0
0	1	1	0	0	0	0	0	0	0	Y	0	0	0	0	0	0	0	0	0
0	1	1	1	0	0	0	0	0	0	0	Y	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	Y	0	0	0	0	0	0	0
1	0	0	1	0	0	0	0	0	0	0	0	0	Y	0	0	0	0	0	0
1	0	1	0	0	0	0	0	0	0	0	0	0	0	Y	0	0	0	0	0
1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	Y	0	0	0	0
1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Y	0	0	0
1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	Y	0	0
1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Y	0
1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Y

Table 7-Truth table of Demultiplexer

2.2 Expression for the outputs

Multiplexer

$Y= S_0'.S_1'.S_2'.S_3'.D_0 + S_0'.S_1'.S_2'.S_3.D_1 + S_0'.S_1'.S_2.S_3'.D_2 + S_0'.S_1'.S_2.S_3.D_3 + S_0'.S_1.S_2'.S_3'.D_4 + S_0'.S_1.S_2'.S_3.D_5 + S_0'.S_1.S_2.S_3'.D_6 + S_0'.S_1.S_2.S_3.D_7 + S_0.S_1'.S_2'.S_3'.D_8 + S_0.S_1'.S_2'.S_3.D_9 + S_0.S_1'.S_2.S_3'.D_{10} + S_0.S_1'.S_2.S_3.D_{11} + S_0.S_1.S_2'.S_3'.D_{12} + S_0.S_1.S_2'.S_3.D_{13} + S_0.S_1.S_2.S_3'.D_{14} + S_0.S_1.S_2.S_3.D_{15}$

Demultiplexer

$D_0 = S_0'.S_1'.S_2'.S_3'.Y$

$D_1= S_0'.S_1'.S_2'.S_3.Y$

$D_2 = S_0'.S_1'.S_2.S_3'.Y$

$D_3 = S_0'.S_1'.S_2.S_3.Y$

$D_4 = S_0'.S_1.S_2'.S_3'.Y$

$D_5 = S_0'.S_1.S_2'.S_3.Y$

$D_6 = S_0'.S_1.S_2.S_3'.Y$

$D_7 = S_0'.S_1.S_2.S_3.Y$

$D_8 = S_0.S_1'.S_2'.S_3'.Y$

$D_9 = S_0.S_1'.S_2'.S_3.Y$

$D_{10}= S_0.S_1'.S_2.S_3'.Y$

$D_{11}= S_0.S_1'.S_2.S_3.Y$

$D_{12} = S_0.S_1.S_2'.S_3'.Y$

$D_{13} = S_0.S_1.S_2'.S_3.Y$

$D_{14} = S_0.S_1.S_2.S_3'.Y$

$D_{15}= S_0.S_1.S_2.S_3.Y$

2.3 Logic circuit Simulation

2.3.1 Multiplexer

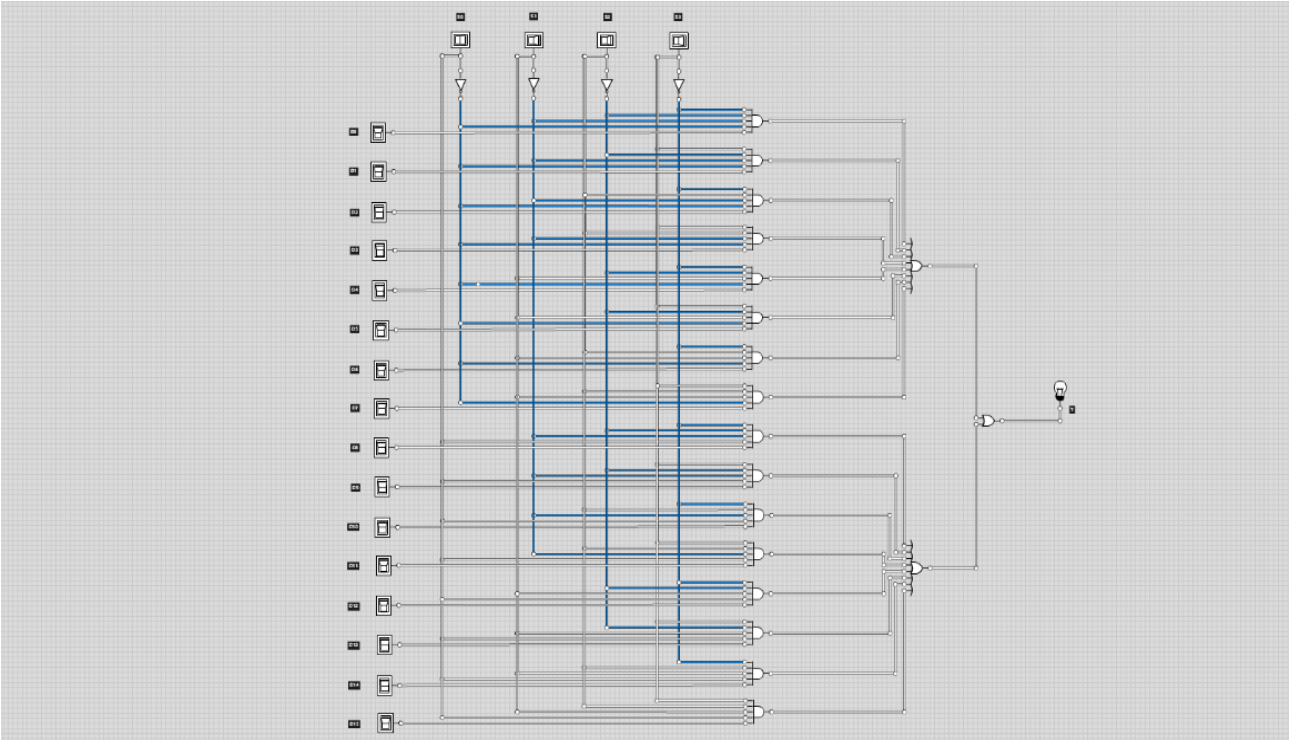


Figure 7-Logic circuit simulation of 16X1 Multiplexer

2.3.2 Demultiplexer

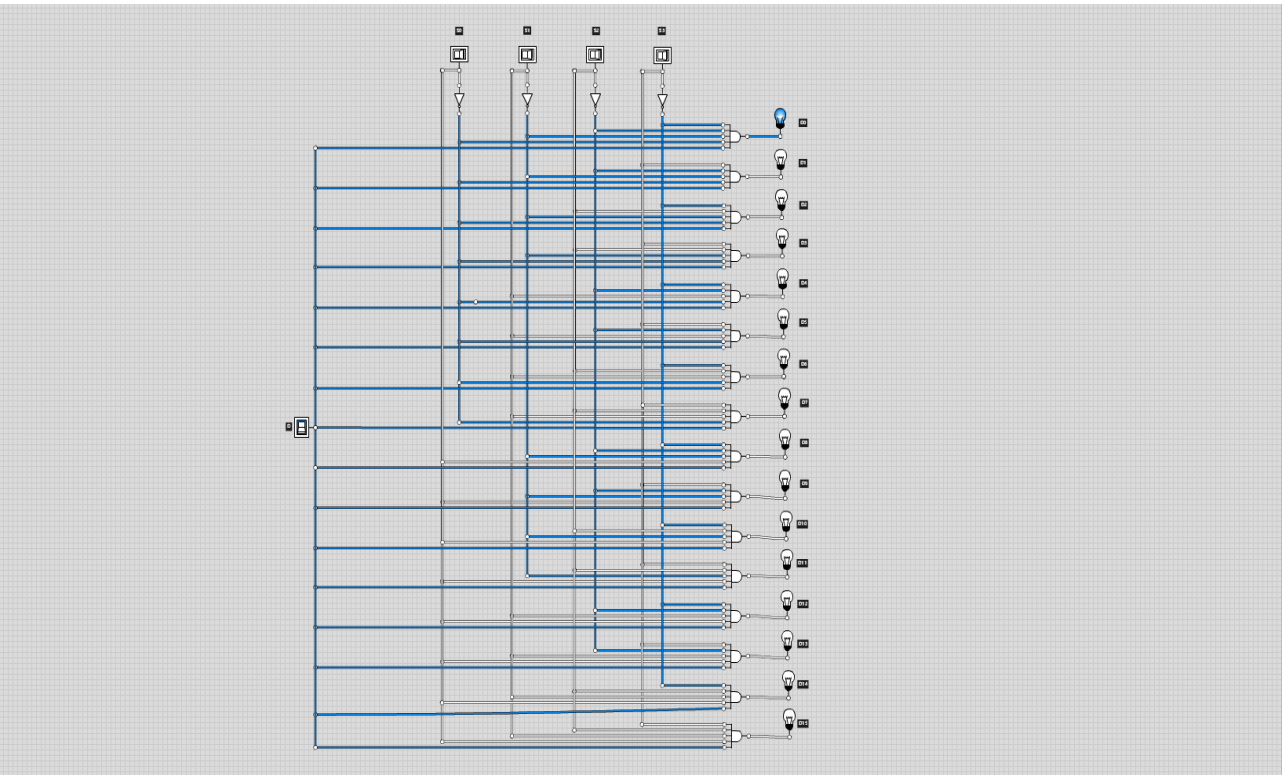


Figure 8-Logic circuit simulation of 1X16 Demultiplexer

3. Adders

3.1 simulation of full adder using half adders and how it works

Half adder

A half adder is a digital logic circuit that performs binary addition of two single-bit binary numbers. It has two inputs, A and B, and two outputs, SUM and CARRY. The SUM output is the least significant bit (LSB) of the result, while the CARRY output is the most significant bit (MSB) of the result, indicating whether there was a carry-over from the addition of the two inputs. The half adder can be implemented using basic gates such as XOR and AND gates.

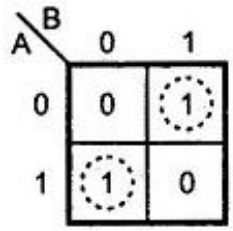
Truth table for half adder

A	B	C (Carry)	S (Sum)
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

Table 8-Truth table for half adder

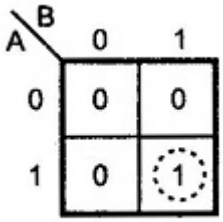
Kmap for half adder

For output sum (s)



SUM=AB' + A'B

For output carry (c)



CARRY=AB

Logic circuit Diagram for Half adder

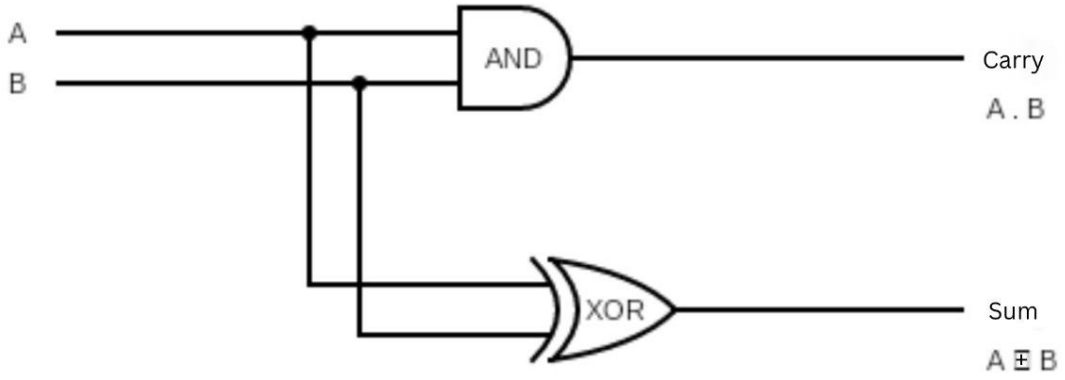


Figure 9-Logic circuit diagram of half adder

Logic circuit Simulation for half adder

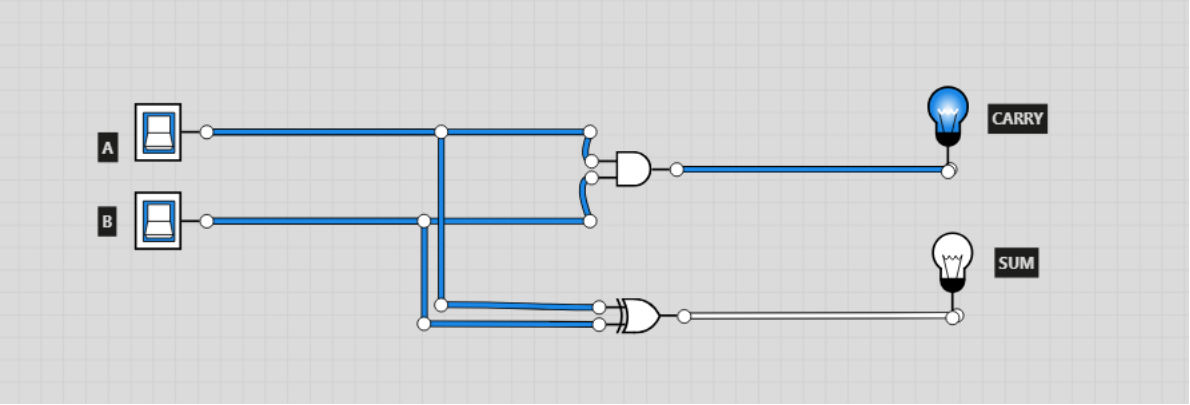


Figure 10-Logic circuit simulation for half adder

Full adder

A full adder serves as a digital circuit designed for the addition of three binary digits: A, B, and a carry input (Cin) representing the carry from the prior stage. Its primary outputs are the sum (S) and carry out (Cout). The construction of a full adder involves the integration of two half adders.

1. First Half Adder :

- Utilizes XOR and AND gates with inputs A and B.
- XOR gate generates the intermediate sum, S1.
- AND gate produces the intermediate carry, C1.

2. Second Half Adder :

- Takes the intermediate sum S1 from First Half Adder and incorporates the Carry input Cin.
- XOR gate yields the final sum, S.
- AND gate produces the final carry, Cout.

By combining these two half adders, a complete full adder is obtained. The intermediate carry (C1) from the initial half adder acts as the carry input (Cin) for the subsequent half adder. This configuration allows for the simultaneous addition of three inputs (A, B, and Cin), resulting in the production of the final sum (S) and carry out (Cout). This modular structure facilitates the scalability of the adder for handling multiple bits in a binary number by cascading numerous full adders, where the carry out of one adder becomes the carry input for the next.

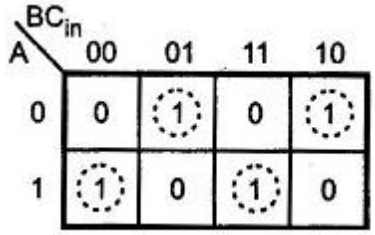
Truth Table of full adder

A	B	C_{in}	C_{out}	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

Table 9-Truth Table for full adder

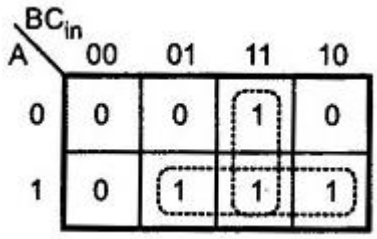
K map for full adder

For output sum



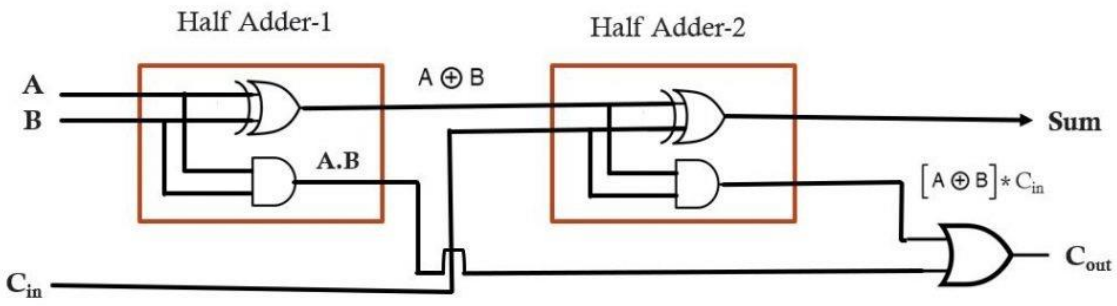
SUM=A'B'C_{in} + A'BC'$_{in}$ + AB'C'$_{in}$ +ABC_{in}

For output Carry(out)



C_{out}= AB + AC_{in} + BC_{in}

Logic circuit of full adder



Logic circuit Simulation of full adder

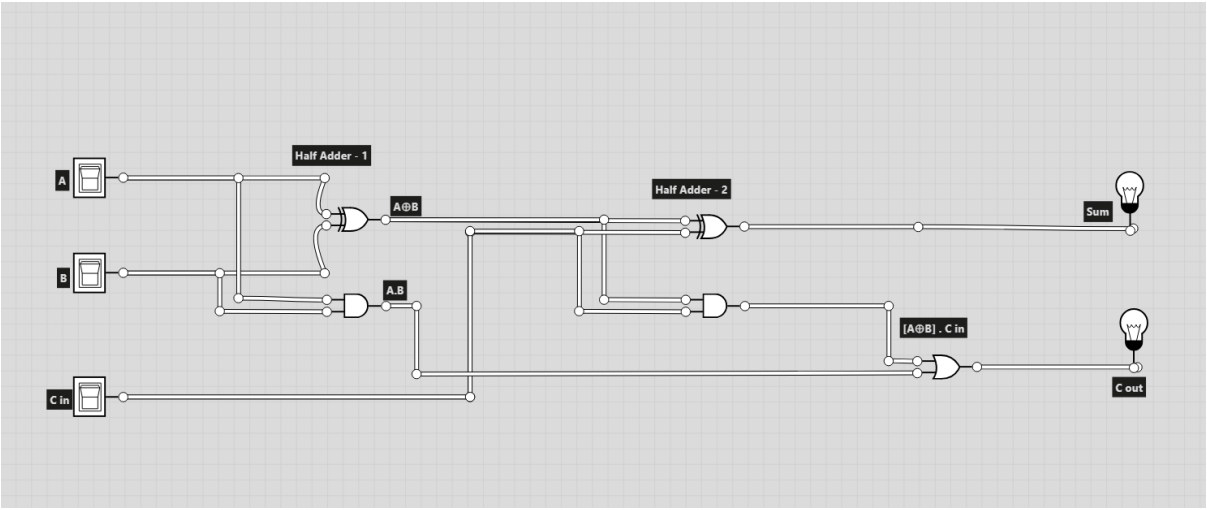


Figure 12-Logic circuit simulation of Full adder

3.2 Creation of 'Ripple Carry Adder' and its capability

A Ripple Carry Adder is a digital circuit designed to add two binary numbers by cascading multiple full adders together. Each full adder in the series processes a pair of bits from the input numbers, and the carry-out from one full adder becomes the carry-in for the next, creating a ripple effect through the adder.

Construction of a 6-bit Ripple Carry Adder:

Consider the addition of two 6-bit binary numbers: A5 A4 A3 A2 A1 A0 and B5 B4 B3 B2 B1 B0. The result C5 C4 C3 C2 C1 C0 will be a 7-bit binary number.

The Ripple Carry Adder is constructed by connecting six full adders in series.

1. **Full Adder 1 (FA1):**
    - Inputs: A0, B0, Cin (initially set to 0)
    - Outputs: S0 (Sum bit), C0 (Carry-out)
2. **Full Adder 2 (FA2):**
    - Inputs: A1, B1, C0 (carry-out from FA1)
    - Outputs: S1 (Sum bit), C1 (Carry-out)
3. **Full Adder 3 (FA3):**
    - Inputs: A2, B2, C1(carry-out from FA2)
    - Outputs: S2 (Sum bit), C2 (Carry-out)
4. **Full Adder 4 (FA4):**
    - Inputs: A3, B3, C2 (carry-out from FA3)
    - Outputs: S3 (Sum bit), C3 (Carry-out)
5. **Full Adder 5 (FA5):**
    - Inputs: A4, B4, C3 (carry-out from FA4)
    - Outputs: S4 (Sum bit), C4 (Carry-out)
6. **Full Adder 6 (FA6):**
    - Inputs: A5, B5, C4 (carry-out from FA5)

Outputs: S5 (Sum bit), C5 (Final Carry-out)

Each full adder processes a pair of bits from the input numbers and the carry-out from the previous stage.

The sum bit (Si) is generated for each stage, and the carry-out (Ci+1) is propagated to the next stage.

Block Diagram

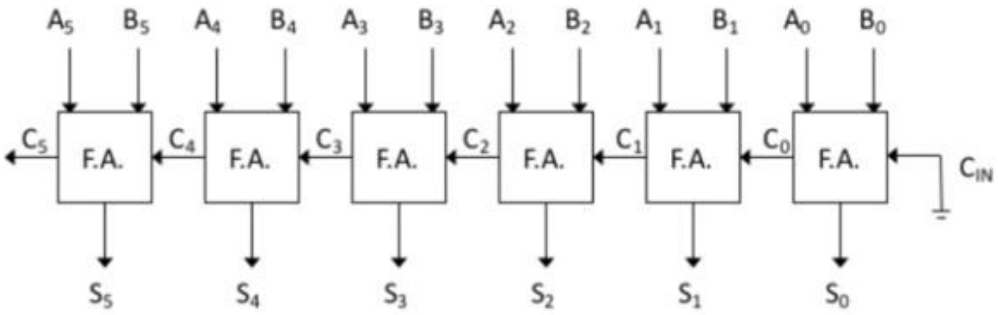


Figure 13-Block Diagram of Ripple carry adder

Simulation Diagram of Ripple carry adder

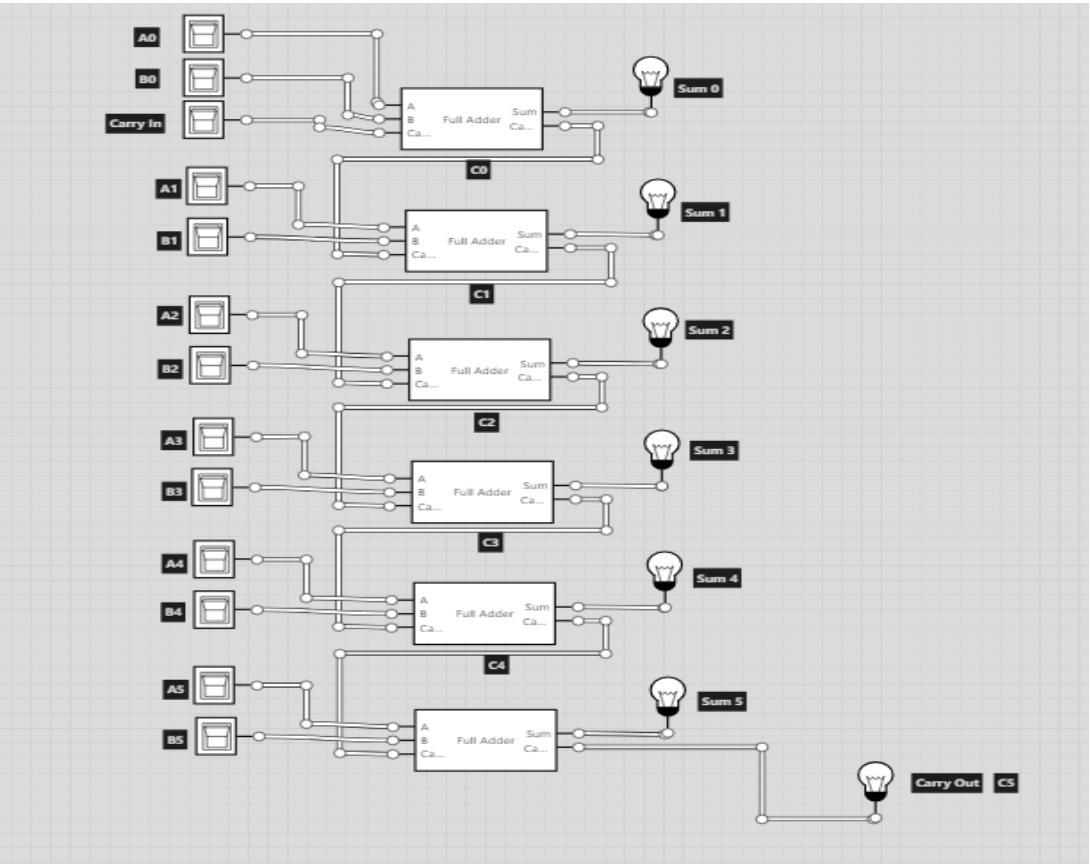


Figure 14-Logic circuit simulation for 6-bit ripple carry adder

4.References

Al Jazeera. (n.d.). *Photos: Drought hits livelihood of Sri Lankan farmers*. [online] Available at: https://www.aljazeera.com/gallery/2023/8/31/photos-harvest-is-so-bad-drought-hits-livelihood-of-sri-lankan-farmers#:~:text=Sri%20Lanka%20planted%201.3%20million [Accessed 20 Nov. 2023].

Wfp.org. (2017). *Sri Lanka's Food Production Hit By Extreme Drought Followed By Floods | World Food Programme*. [online] Available at: https://www.wfp.org/news/sri-lanka%E2%80%99s-food-production-hit-extreme-drought-followed-floods. [Accessed 20 Nov.2023]

5.Workload matrix

Tasks Allocated	Anusigan Sivananthan (20230297)	Lithila Mahagedara (20230176)	N.T Welagedara (20222090)	M.H.A Daneesha Hansaka (20230229)	Senumi Dahanayake (20230148)
Real world digital circuit application in agriculture sector	✓				
Incident Analysis	✓				
Multiplexers			✓		
Demultiplexers		✓			
Adders					✓
Simulation files for all questions				✓	
Report making	✓				
Proof Reading	✓	✓	✓	✓	✓
Logic circuit diagrams and Schematic diagrams				✓	

NOTE: Mr.R.A.T.M.Gunawardana (20230869)-No Contribution