

Course code and name:	B37DF Computer architecture and embed system
Type of assessment:	Individual
Coursework Title:	Lab1
Student Name:	MA XUNCHI
Student ID Number:	H00392669

Declaration of authorship. By signing this form:

- **I declare** that the work I have submitted for individual assessment OR the work I have contributed to a group assessment, is entirely my own. I have NOT taken the ideas, writings or inventions of another person and used these as if they were my own. My submission or my contribution to a group submission is expressed in my own words. Any uses made within this work of the ideas, writings or inventions of others, or of any existing sources of information (books, journals, websites, etc.) are properly acknowledged and listed in the references and/or acknowledgements section.
- I confirm that I have read, understood and followed the University's Regulations on plagiarism as published on the [University's website](#), and that I am aware of the penalties that I will face should I not adhere to the University Regulations.
- I confirm that I have read, understood and avoided the different types of plagiarism explained in the University guidance on [Academic Integrity and Plagiarism](#)

Student Signature (type your name): MA XUNCHI

Date: 10-3-2023

Lab 1 – Design of Programmable Logic Devices (PLDs) using Logisim

Part 1: Introduction

In this lab, we are going to look at how to design and implement three different type of Programmable Logic Devices (PLDs). Namely, Programmable Read Only Memory (PROM), Programmable Array Logic (PAL) and Programmable Logic Array (PLA). PLDs are integrated circuits (ICs) consisting of an array of AND and OR gates. You are also going to look at how to implement various Boolean logic functions based on the requirement using PLDs.

Part 2: Design of Programmable Read Only Memory (PROM)

For the first task, we are required to design a Programmable Read Only Memory (PROM). The structure of a PROM is shown in Figure 1.

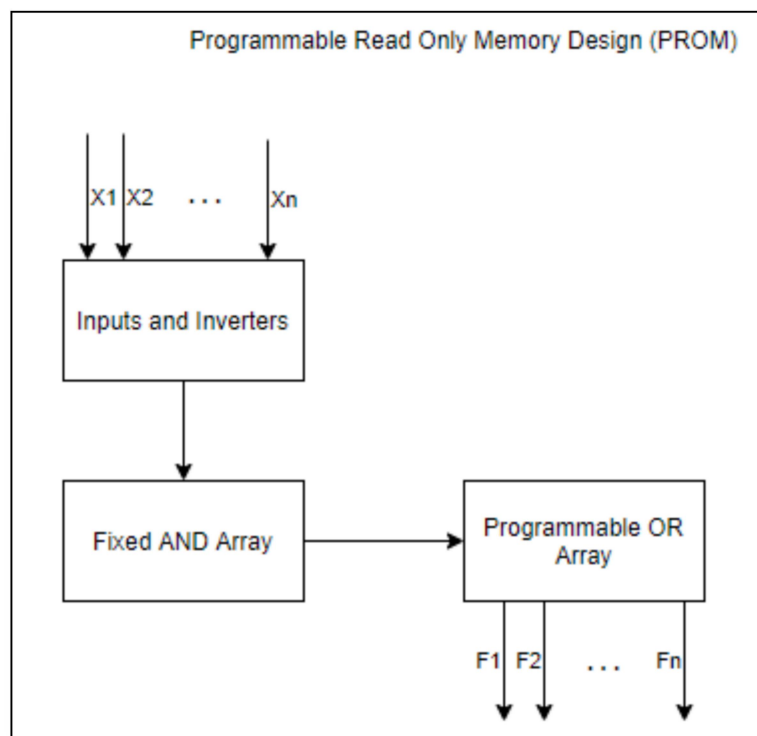


Figure 1: PROM Structure

Most notably, the PROM consists of a **fixed** AND array and a **programmable** OR array. We are going to look at two ways to implement the PROM. The first method is using AND gates and Inverters for the fixed AND array and the second method is a more convenient way using x to n decoders for the fixed AND array depending on the number of inputs x.

First, we are going to design a 2 input PROM using AND gates and inverters but without any outputs for the time being to have a base to work on. The inputs are 'A' and 'B'. In Logisim, create a workspace as shown in Figure 2.

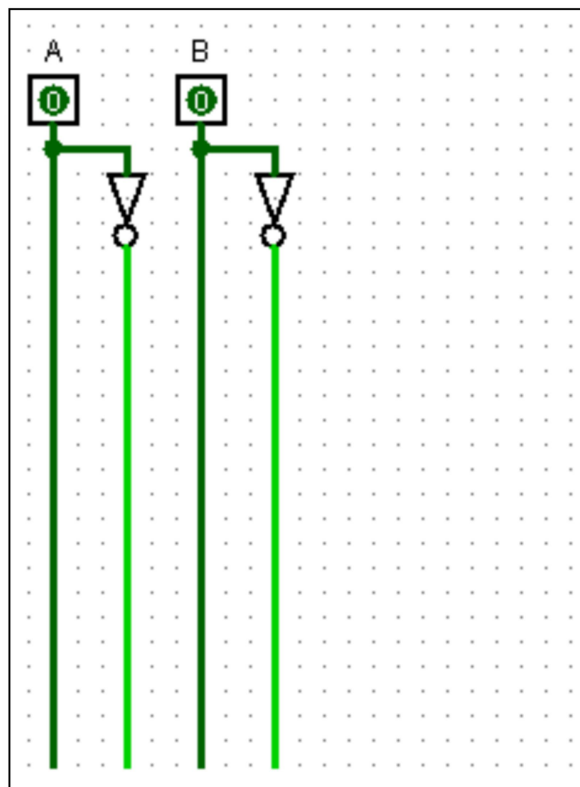


Figure 2: 2 input PROM workspace

The AND array is a fixed side of the PROM, all the possible input combinations need to be fixed and accordingly the functions can be programmed using the OR array.

Task 2.1: PROM design with two inputs

- (a) Add an AND gate with n inputs (one for each input, so two in this case) to your circuit for each possible input combination and connect these input combinations to each gate, respectively.

Next, label the section that consists of the Fixed AND array and Programmable OR array respectively while ensuring to label the AND gates with the corresponding input combination.

Finally, ensure there are wires extended from each AND gate to make room for the programmable OR gate connections.

[6 marks]

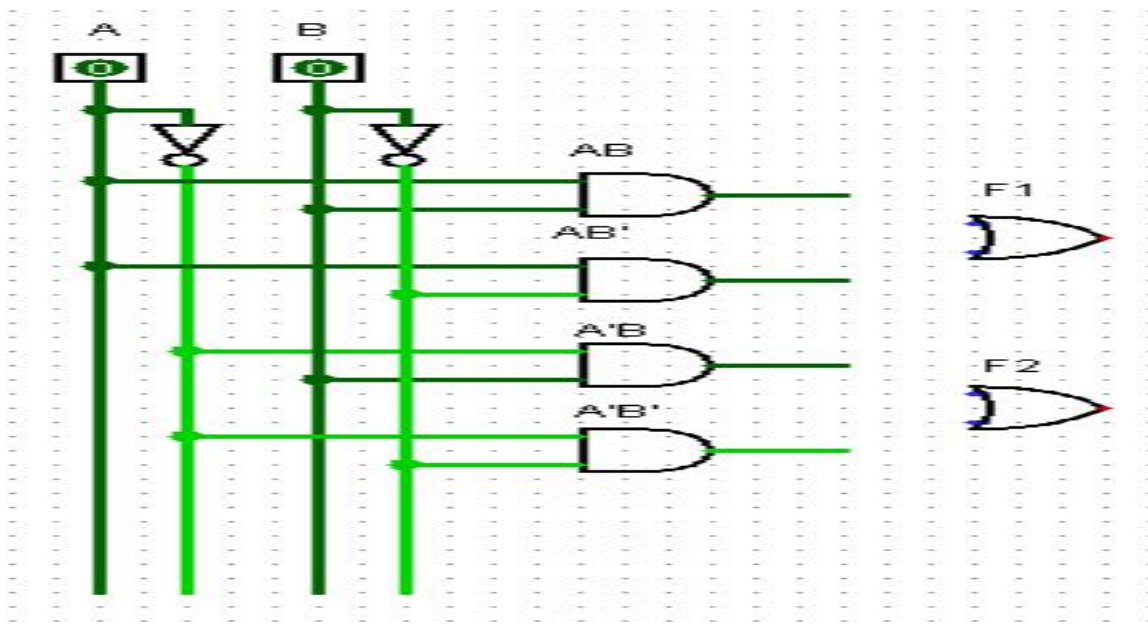


Figure 3: PROM Design for Task 2.1 (a) using Logisim without inputs

- (b) Implement the following Boolean functions using PROM

- $F1(A, B) = A \text{ XOR } B$
- $F2(A, B) = A \text{ XNOR } B$

Create a truth table for F1 and F2 and identify logic expressions for each function in terms of inputs A and B. Show all working done to arrive at your answers.

F_1	A	B	F_1	F_2	A	B	F_2
	0	0	0		0	0	1
	0	1	1		0	1	0
	1	0	1		1	0	0
	1	1	0		1	1	1

F_1	\overline{A}	B		F_2	\overline{A}	B	
	0	0	1		0	1	0
	1	1	0		1	0	1

$F_1 = \overline{A}B + A\overline{B}$	$F_2 = AB + \overline{A}\overline{B}$
---------------------------------------	---------------------------------------

[6 marks]

Figure 4: Truth Table for F1 and F2 in Task 2.1 (b) along with Simplified logic expressions

- (c) Using Logisim implement Task 2.1 (b). In the programmable OR array, use the necessary outputs from four AND gates in conjunction with two OR gates to implement F1 and F2. Label them appropriately.

[8 marks]

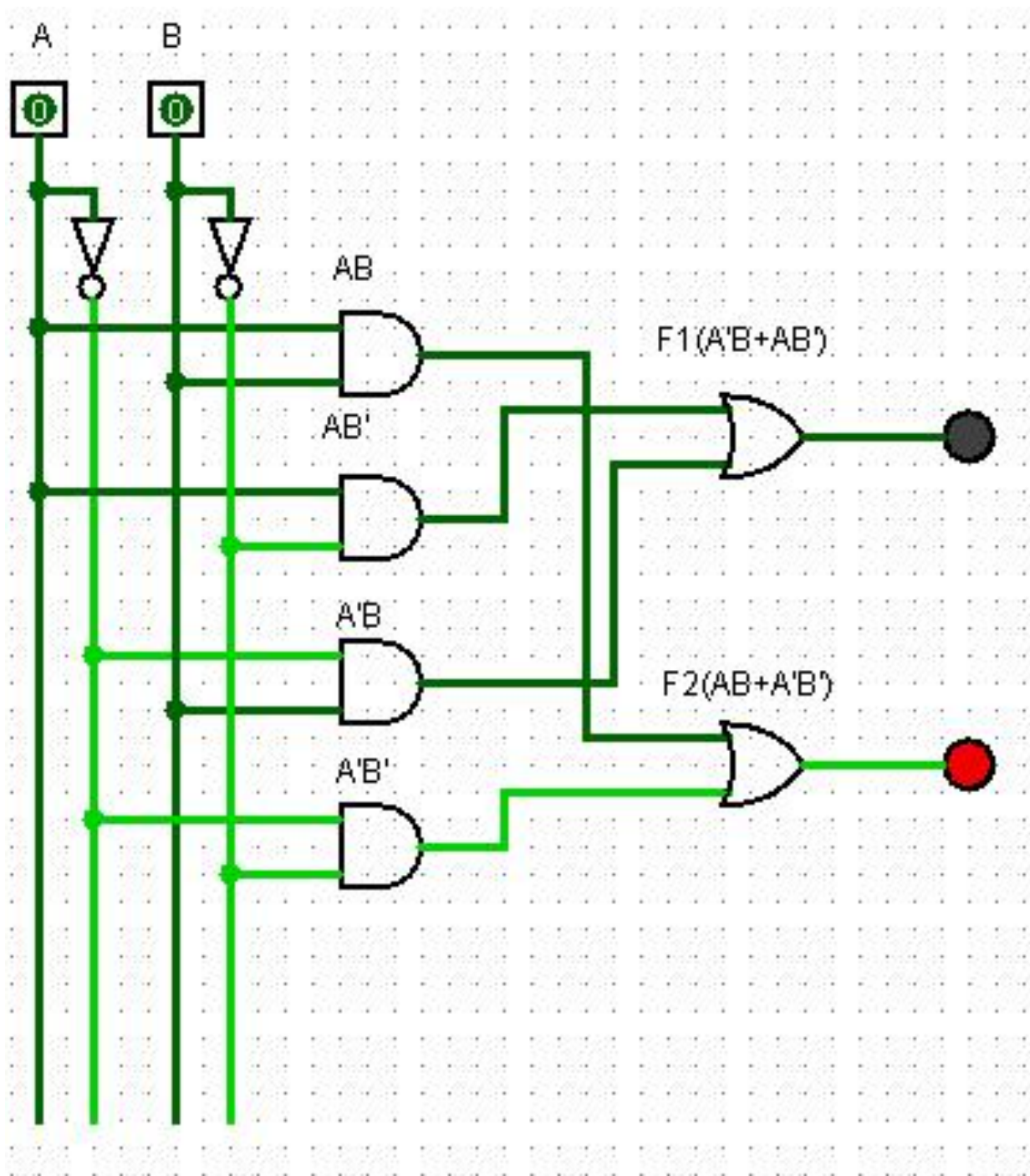


Figure 5 PROM Implementation of Task 2.1 (b) using Logisim

Task 2.2: PROM design with decoders

PROM designs can also be made conveniently using x to n decoders for the fixed AND array.

- a) Use a 3-8 decoder to create a 3 input PROM design without any outputs (use A, B and C as inputs). Showcase your design in the space given. Label the fixed AND array and programmable OR array appropriately.

[6 marks]

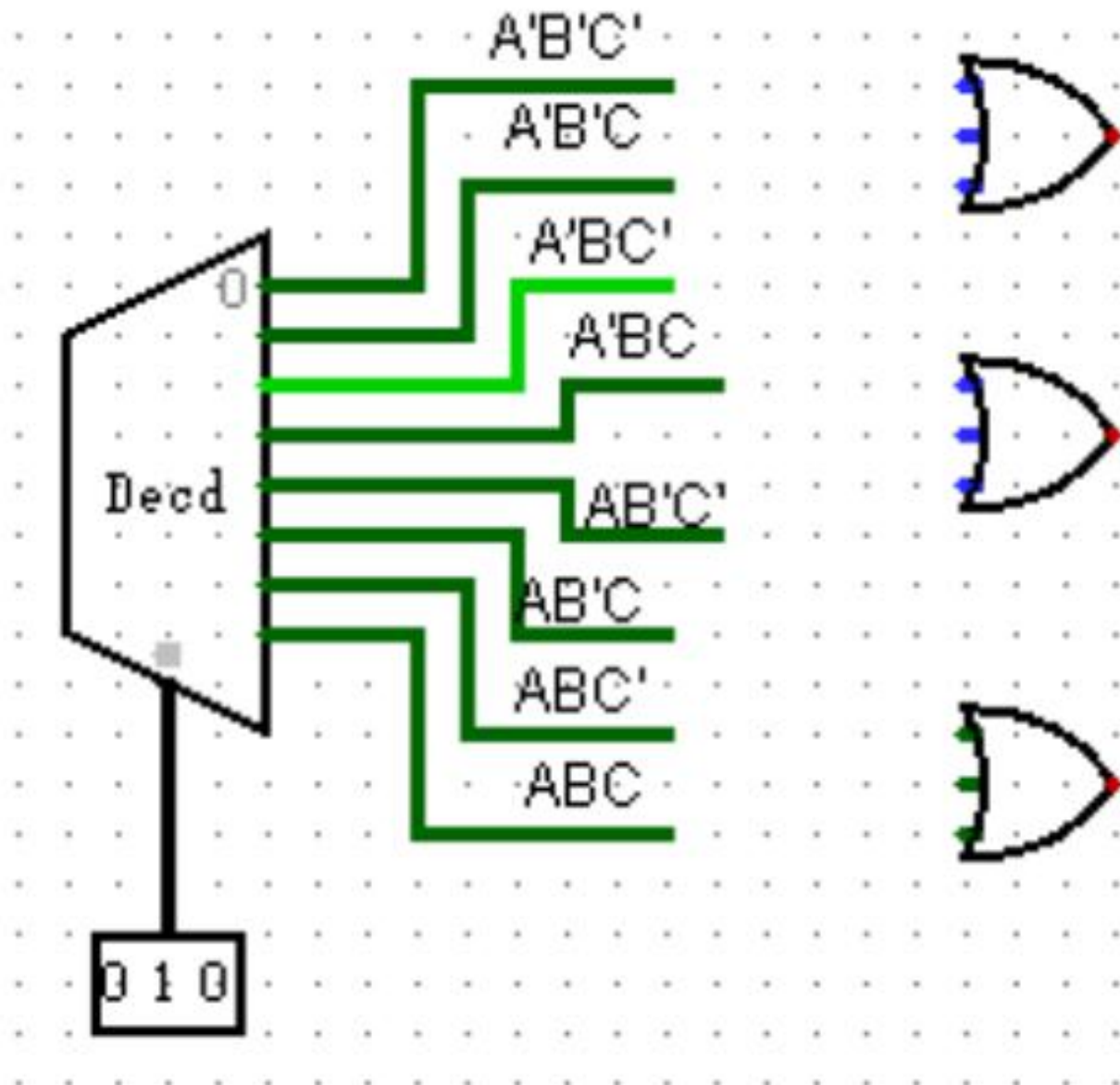


Figure 6: PROM design for Task 2.2 (a) using 3 to 8 Decoder

- (b) Use this design to implement the following Boolean functions:

- $F1(A, B, C) = (A+B) * (B'+C)$
- $F2(A, B, C) = \prod M(1,3,4,7)$
- $F3(A, B, C) = \sum m(0,1,2,5,7)$

Create a truth table for F₁, F₂ and F₃ and identify the logic expressions for each function in terms of inputs A, B and C. Show all working done to arrive at your answers.

[10 marks]

F_1	A	B	C	F_1	F_2	A	B	C	F_2	F_3	A	B	C	F_3
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	0	1	1	
0	1	0	0	0	0	1	0	0	1	0	1	0	1	
0	1	1	0	1	0	1	1	0	0	0	1	1	0	
1	0	0	1	1	1	0	0	1	0	1	0	0	0	
1	0	1	1	1	1	0	1	1	1	1	0	1	1	
1	1	0	1	0	1	1	0	1	1	1	1	0	0	
1	1	1	1	1	1	1	1	1	0	1	1	1	1	

F ₁	$\begin{matrix} AB \\ C \end{matrix}$	00	01	11	10
0	0	0	0	0	1
1	1	0	1	1	1

$$\begin{aligned}
 F_1 &= AB' + AC + BC \\
 &= BC + AB' \\
 &= ABC + A'B'C + AB'C + AB'C'
 \end{aligned}$$

F ₂	$\begin{matrix} AB \\ C \end{matrix}$	00	01	11	10
0	0	1	1	1	0
1	1	0	0	0	1

$$F_2 = A'B'C' + A'BC' + AB'C' + ABC'$$

F ₃	$\begin{matrix} AB \\ C \end{matrix}$	00	01	11	10
0	0	1	1	0	0
1	1	1	0	1	1

$$F_3 = A'B'C' + AB'C' + ABC' + AB'C + ABC$$

Figure 7: PROM design for Task 2.2 (b) using 3-8 decoder (Truth table, working and simplified logic expressions)

(c) Implement Task 2.2 (b) using Logisim. In the programmable OR array, use the necessary outputs from the 3 to 8 Decoder in conjunction with three OR gates to implement F1, F2 and F3. Label them appropriately.

[8 marks]

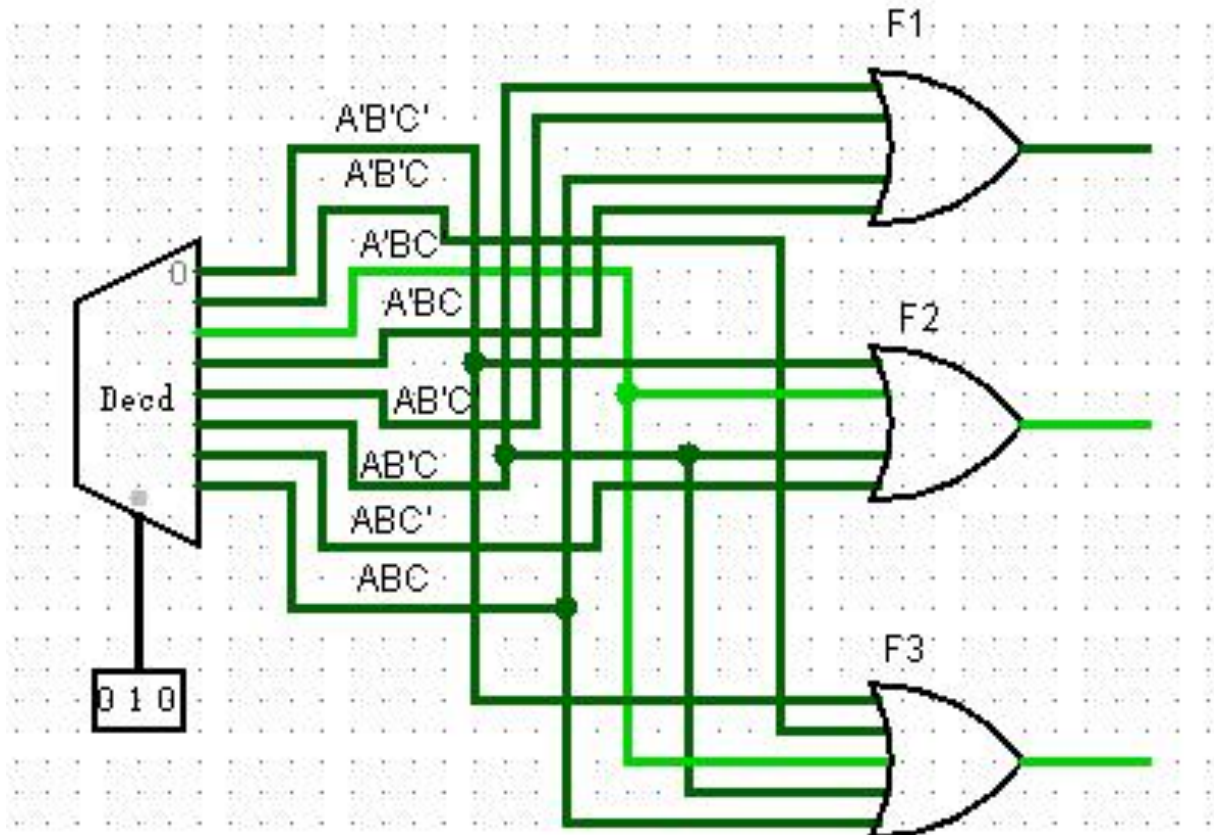


Figure 8: Implementation of Task 2.2 (b) using 3 to 8 decoder PROM with Logisim

Part 3: Design of Programmable Array Logic (PAL)

In this section, we are going to look at how to design a Programmable Array Logic (PAL) and implement various Boolean logic functions using a PAL. The structure of a PAL is shown in Fig. 9.

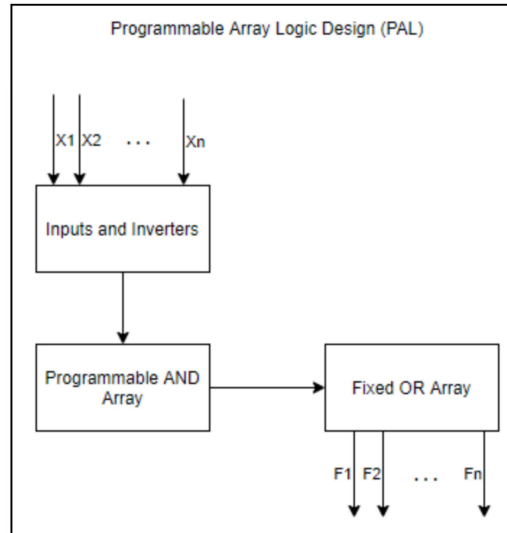


Figure 9: Programmable Array Logic (PAL) Structure

The difference between a PROM and a PAL is that a PROM consists of a programmable OR array and fixed AND array while a PAL consists of a programmable AND array and a fixed OR array. An example of a four AND gate and two OR gate implementation is shown in Figure 10.

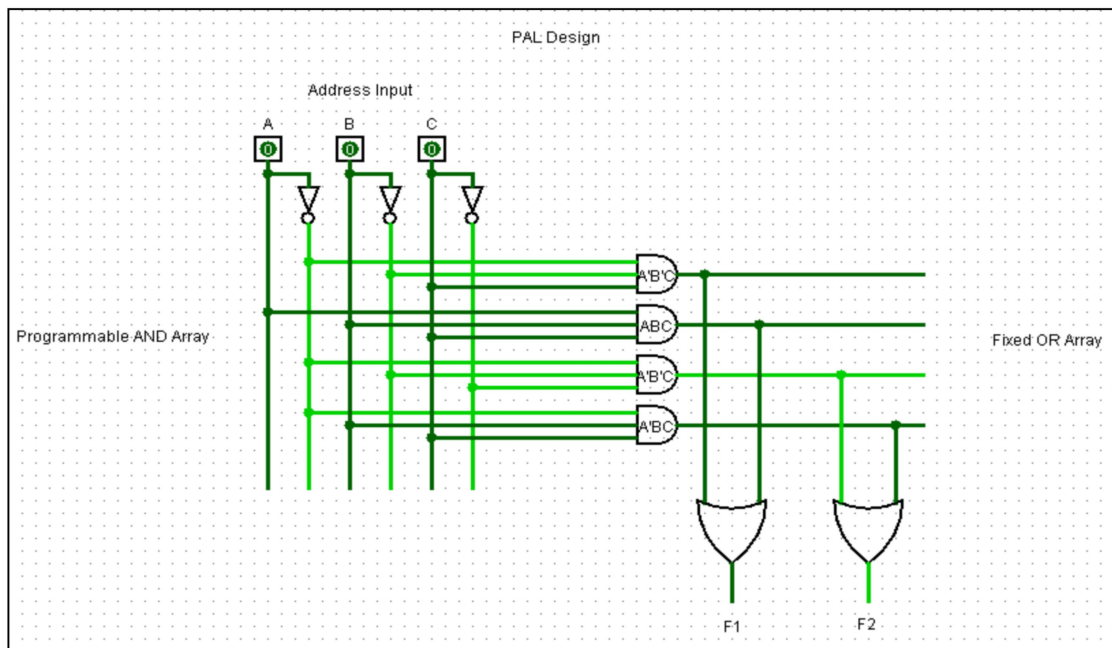


Figure 10 : Example of PAL design in Logisim

Task 3.1: Designing a PAL circuit in Logisim

Design a PAL with four inputs (A, B, C and D) and space for four outputs (F1, F2, F3 and F4). Use 12 Four - input AND gates for the programmable AND array. Use four 3-input OR gates connected to three AND gate outputs each, respectively. Label appropriately. You do not have to connect any of the inputs to the AND gates.

[6 marks]

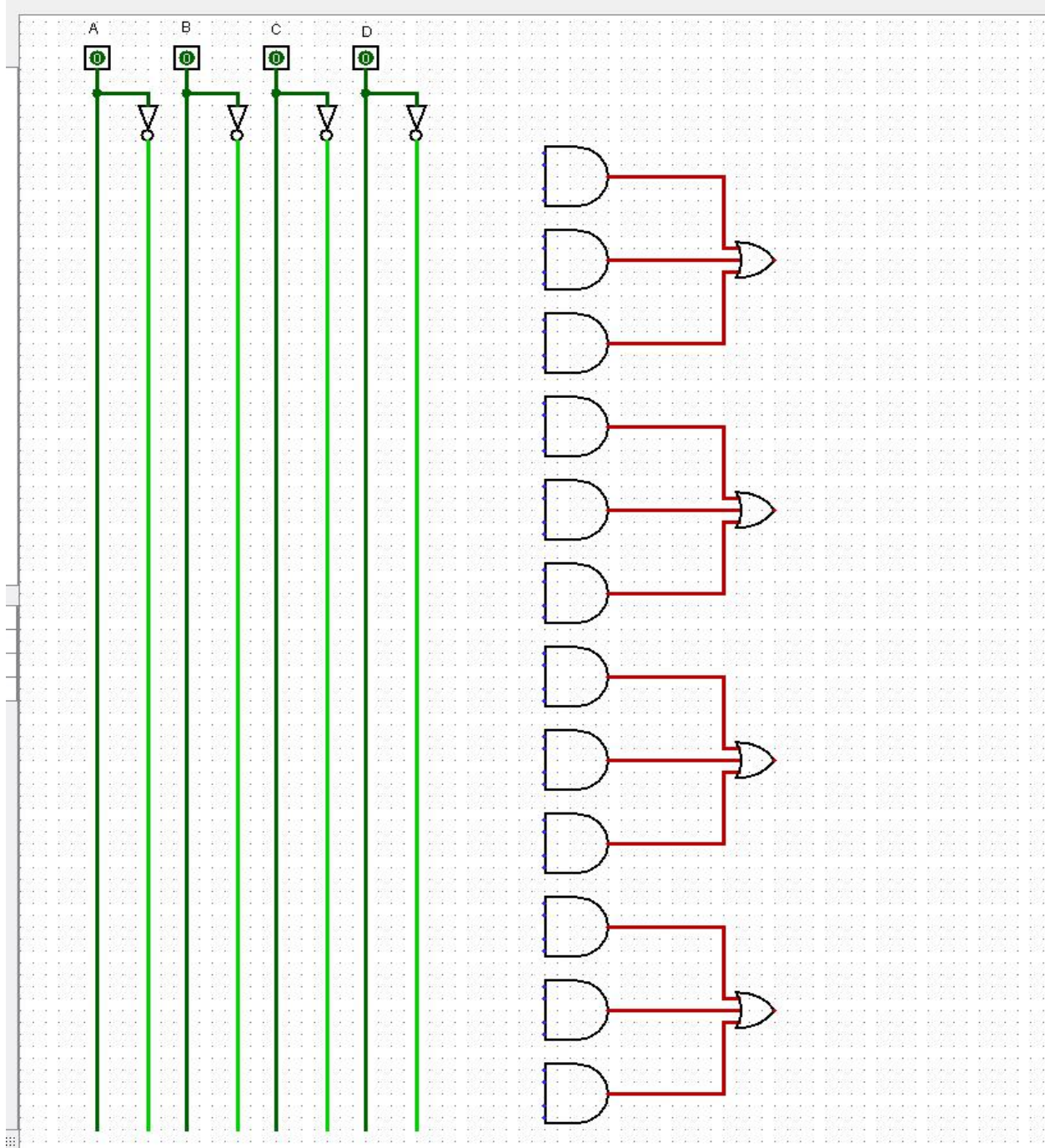


Figure 11: PAL Design for Task 3.1 using Logisim

Task 3.2: Implementing Boolean functions using PAL

Using the PAL design, implement the following Boolean functions:

- $F_1(A, B, C, D) = \sum m(2, 12, 13)$
- $F_2(A, B, C, D) = \sum m(7, 8, 9, 10, 11, 12, 13, 14, 15)$
- $F_3(A, B, C, D) = \sum m(0, 2, 3, 4, 5, 6, 7, 8, 10, 11, 15)$
- $F_4(A, B, C, D) = \sum m(1, 2, 8, 12, 13)$

(a) Simplify the four functions into minimum number of terms keeping in mind that no function can exceed the use of three AND gates. Create a PAL programming table to assist you. Mark the interconnection in Fig. 13. Use crosses to mark the appropriate terms for each AND gate input. Show all your working.

[16 marks]

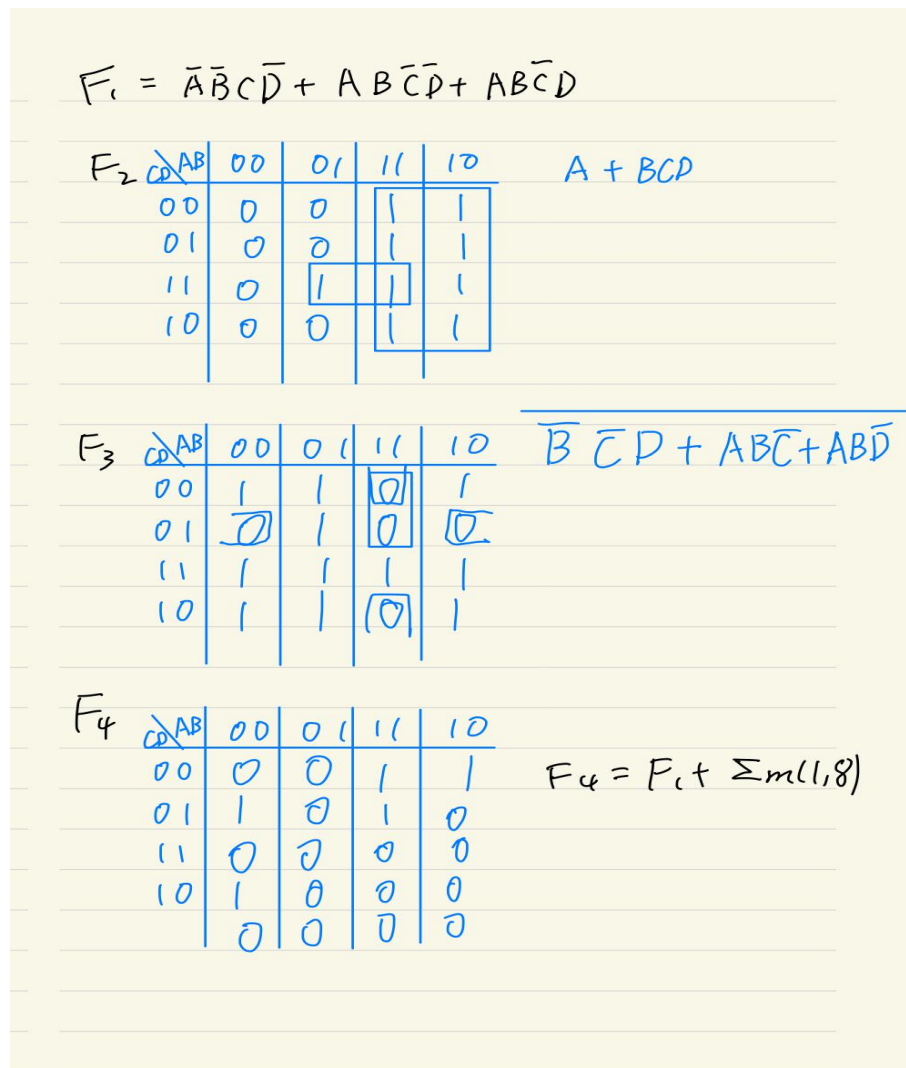


Figure 12: Working for PAL Design (K-map simplification) Task 3.2 (a)

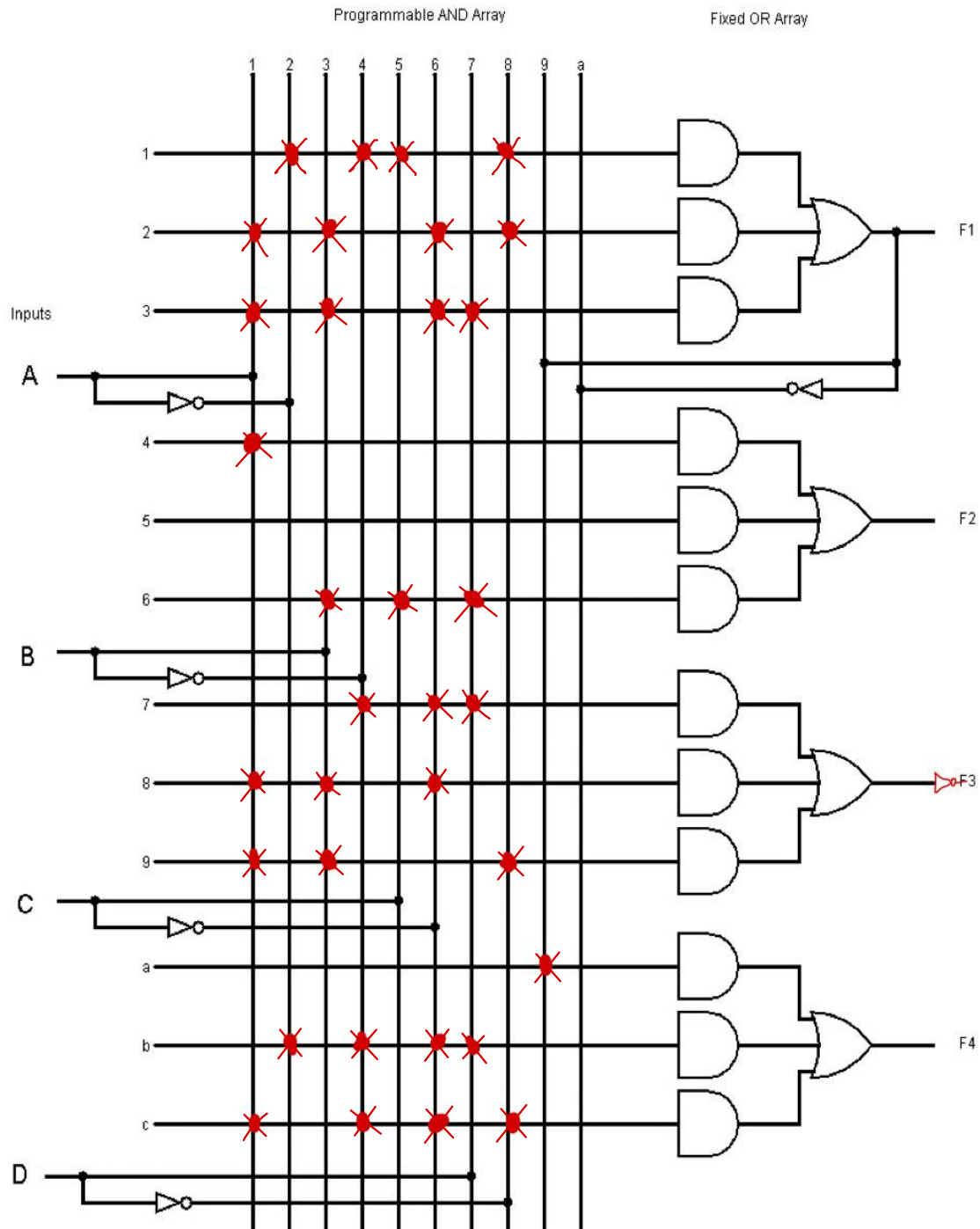


Figure 13: PAL implementation of Task 3.2 (a)

(b) Using Logisim, implement the schematic results you have obtained using the PAL design you have created in Task 3.1.

[10 marks]

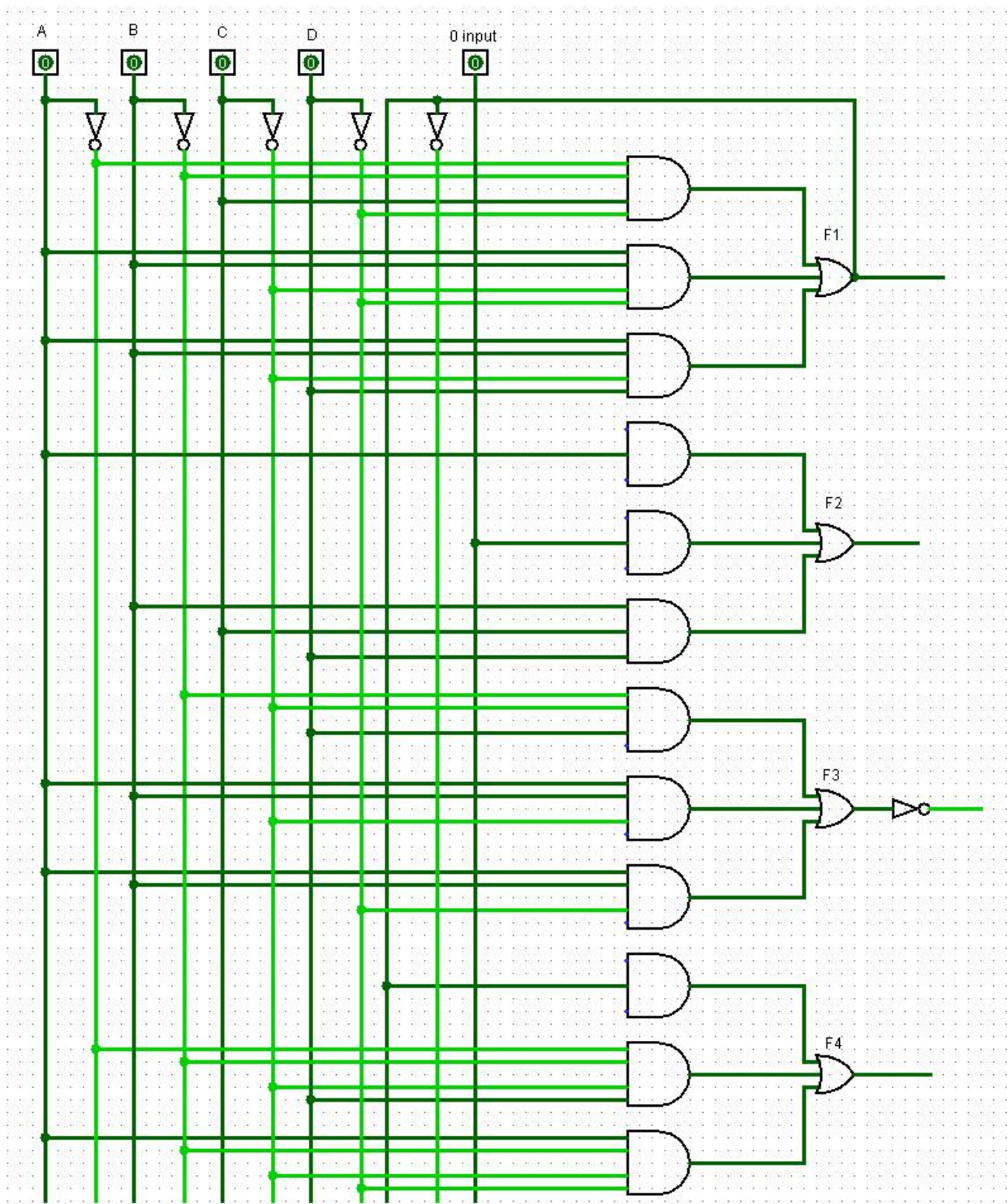


Figure 14: PAL implementation for Task 3.2 (b) using Logisim

Part 4: Design of Programmable Logic Array (PLA)

In this section, we are going to look at how to design a Programmable Logic Array (PLA) and implement various Boolean logic functions using a PLA. The structure of a PLA is shown in Figure 15.

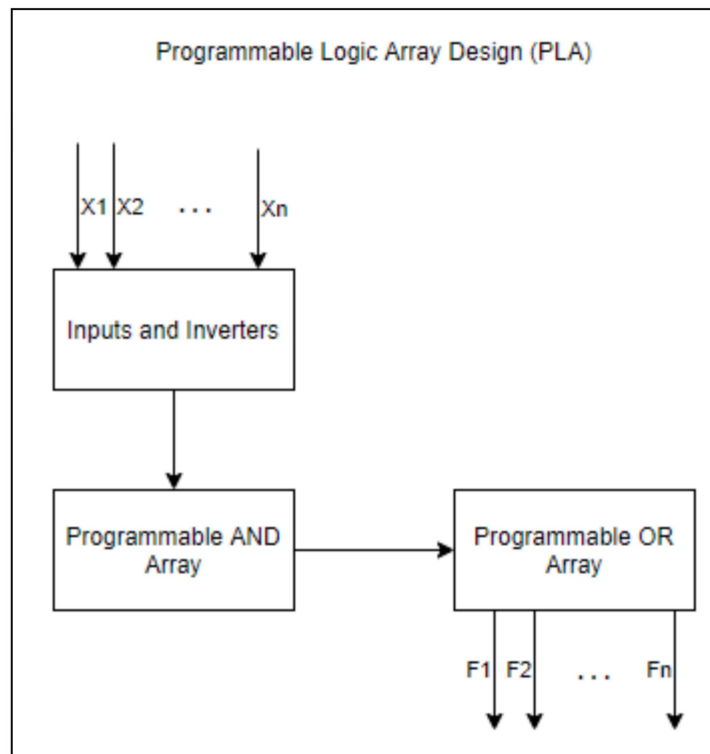


Figure 15: PLA Structure

In a PLA, both the AND array and OR array are programmable. The PLA design is highly dependent on the functions that are being implemented.

For example, let us see how we could design the following Boolean functions using a PLA:

- $F1(A, B, C) = AB + AC'$
- $F2(A, B, C) = AB' + BC + AC'$

F1 consists of two product terms and F2 consists of three product terms. The number of AND gates required for the programmable AND array depends on the number of unique product terms. In this example, F1 and F2 has one common term, **AC'**. Hence, there are four unique product terms. The PLA design should then have **four** AND gates.

The number of OR gates depend on the number of functions being implemented. In this case, there are two functions and so **two** OR gates are required. The number of inputs on each OR gate are decided by the number of sum of product (SOP) terms. In this example, F1 has two terms and F2 has three terms. Hence, F1 requires a **two** input OR

gate and F2 requires a **three** input OR gate. Figure 16 shows how this would be implemented in Logisim.

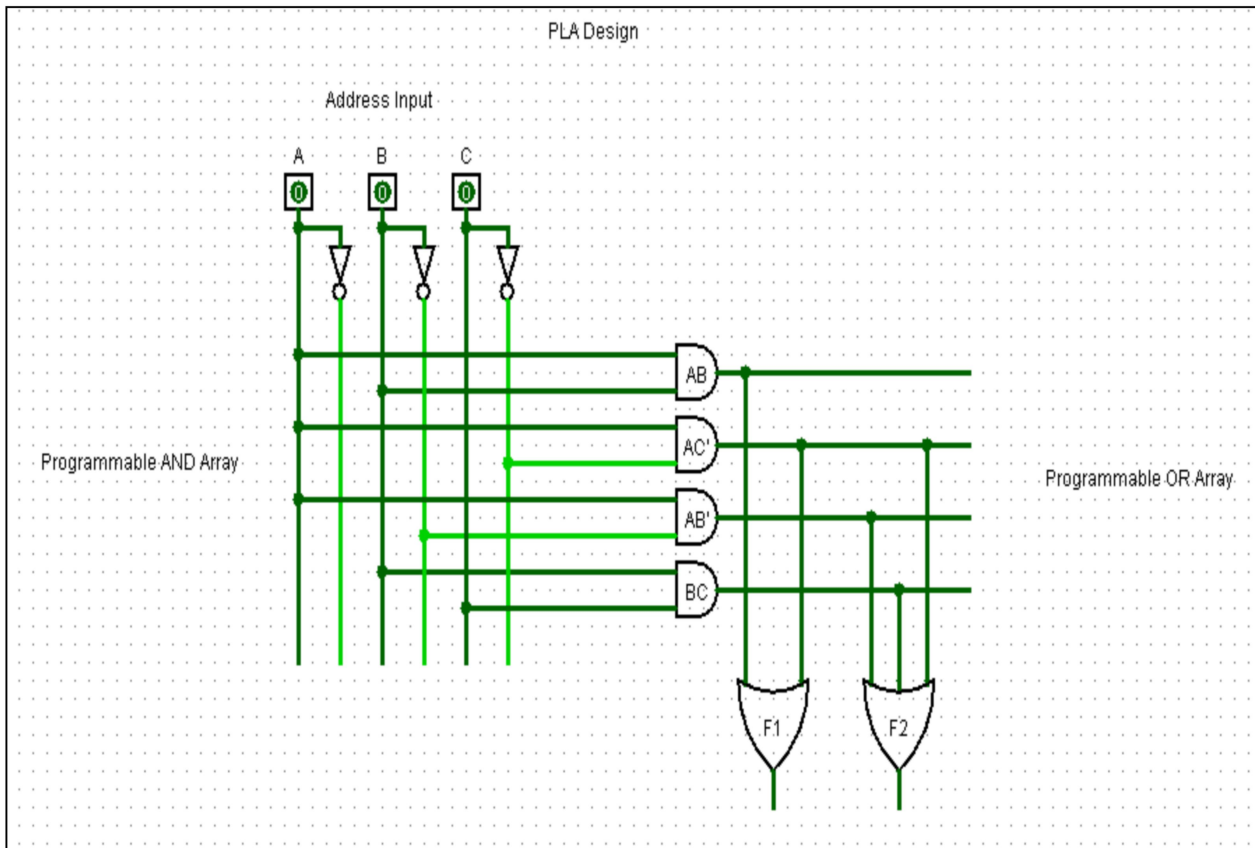


Figure 16: PLA Design Example

Task 4.1: PLA design using simple Boolean Logic

Implement the following Boolean functions using PLA:

- $F_1(A, B, C) = (AB + AC + BC)'$
- $F_2(A, B, C) = A'B' + A'C' + ABC$

a) Create a truth table for F_1 and F_2 and identify simplified logic expressions using K-map for each function in terms of inputs A, B and C. Use a PLA programming table. Show all working done to arrive at your answers. Mark the interconnection in Fig. 18.

[8+6 marks]

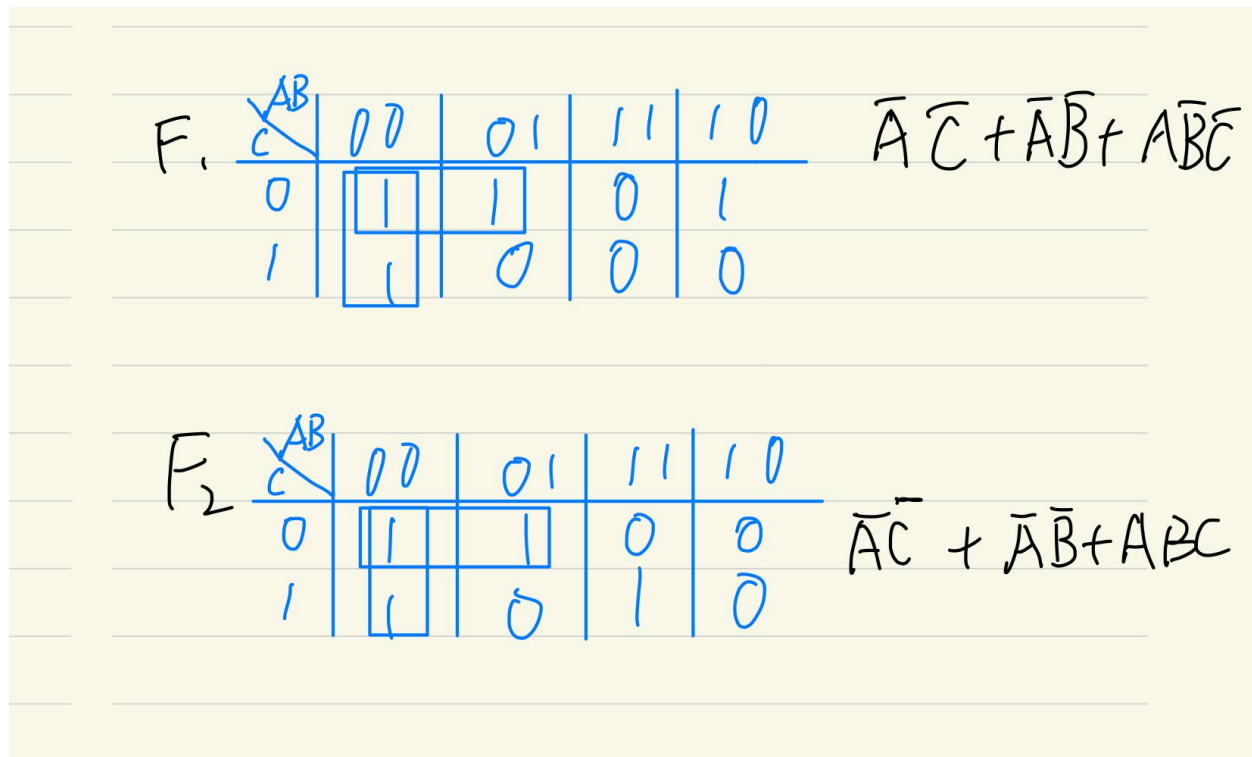


Figure 16. Working for PLA Design (K-map simplification) for Task 4.1

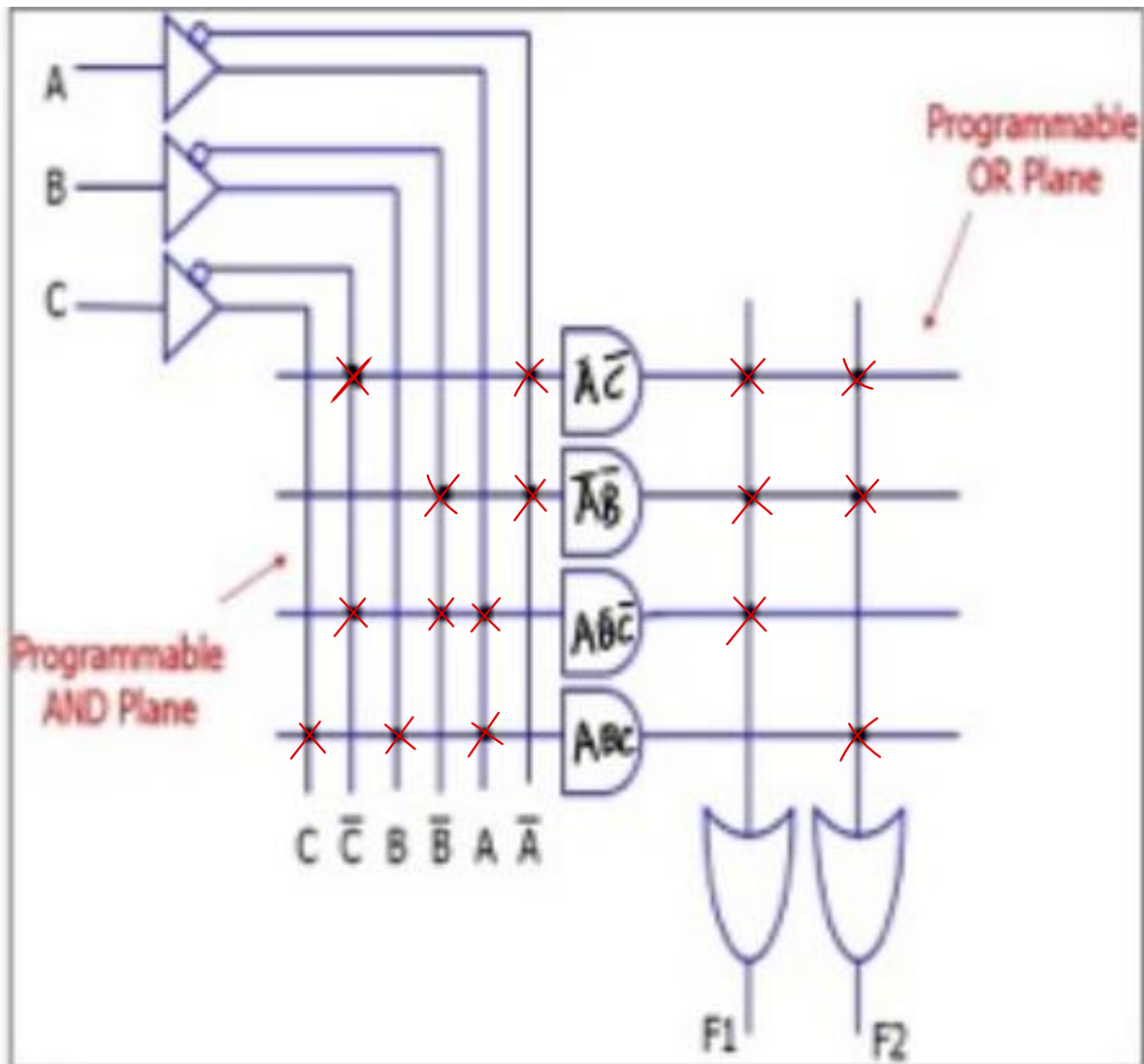


Figure 18: PLA implementation for Task 4.1

b) Use four AND three input AND gates and two OR gates to implement the PLA design in Logisim for F1 and F2.

[10 marks]

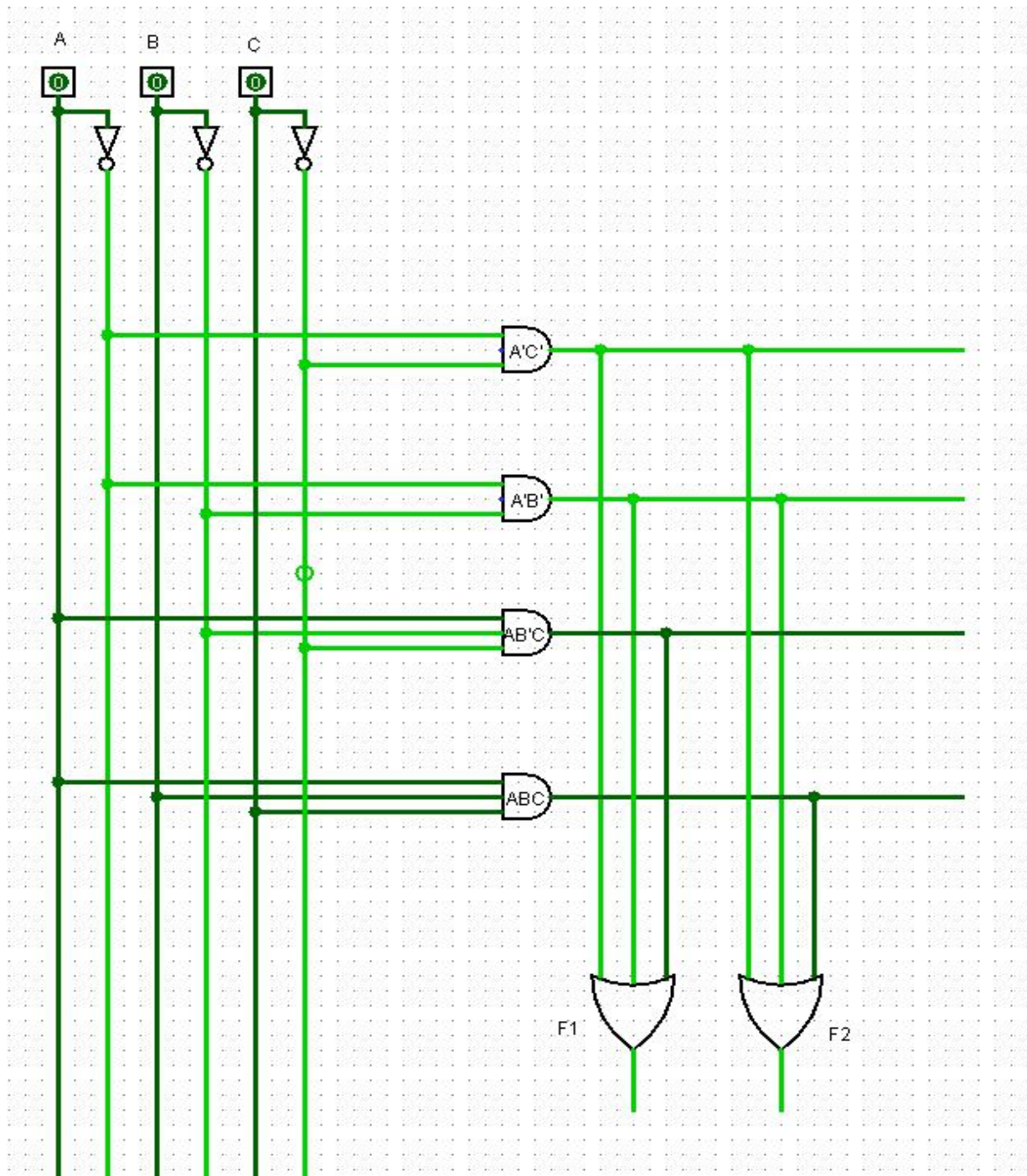


Figure 17: PLA implementation for Task 4.1 using Logisim

- b) Use four AND three input AND gates and two OR gates to implement the PLA design in Logisim for F1 and F2.

[10 marks]

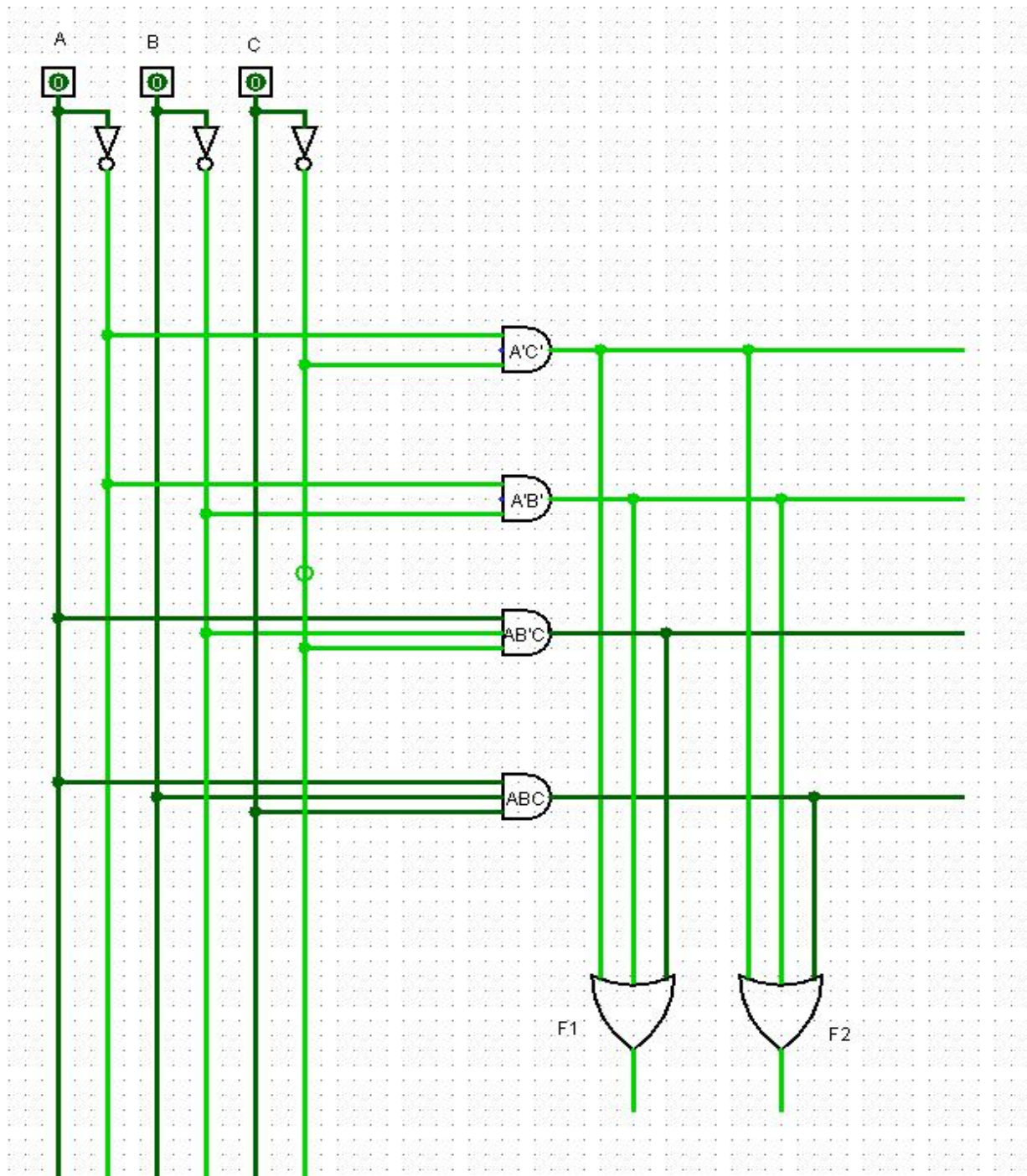


Figure 17: PLA implementation for Task 4.1 using Logisim