COMPUTER ORGANIZATION & DESIGN

(20IC209P)

CPU DESIGN
D2DWALA_CPU

Submitted by:

Dilon	22BIT234D
Om Patel	22BIT248D
Ren Patel	22BIT249D
Het Virani	22BIT252D
Ved Vyas	22BIT253D

Department Name: ICT (H3)

Under The Guidance of:

Ganga Prasad Pandey
Ankur Changela
Manoj Yadav

Table of Content

AIM

To design a 16-bit Central Processing Unit (CPU) using Logisim.

- ✓ To design 16-bit ALU.
- ✓ To design Sequence Counter.
- ✓ To design Control Unit.
- ✓ To design various 16-bit registers.
- ✓ To design common bus.
- ✓ To design encoders for common bus and ALU.
- ✓ To understand working principle of CPU.

ASSUMPTIONS

Design of Basic Computer:

- ✓ A memory unit with 4096 words of 16 bits each.
- ✓ Five registers: AR, PC, DR, IR, TR
- ✓ flip-flop: D Flipflop as register
- ✓ Multiplexers & Demultiplexers : 2 to 1mux , 1 to 16 demux
- ✓ A 16-bit common bus.
- ✓ Priority Encoder: 3 to 8
- ✓ Control Logic Gates
- ✓ Adder & logic circuit connected to the input of AC

INTRODUCTION

PROCEDURE :

- ✓ Create 16-Bit Register
- ✓ Create 12-Bit Register
- ✓ Create Datapath
- ✓ Create ALU
- ✓ Create Timing and Control Unit
- ✓ Connect Common Bus with registers and ALU
- ✓ Connect all Registers and other logics
- ✓ Connect all the Circuits (Timing Unit, Common Bus and Logic Circuits)
- ✓ Connect the obtained circuit to RAM (Primary Memory)

WORKING PRINCIPLES

Registers:

- ✓ Registers are small, high-speed storage locations within the CPU used to hold data temporarily during processing.They store operands, intermediate results, memory addresses, and other important values.
- ✓ Registers are organized into categories such as general-purpose registers (for storing data), special-purpose registers (for storing status flags, instruction pointers, etc.), and control registers (for controlling CPU operations).
- ✓ In our 16-bit CPU design, we'll implement various registers to hold data, addresses, and control information.

Arithmetic Logic Unit (ALU):

- ✓ The ALU performs arithmetic and logical operations on data stored in registers or memory.
- ✓ Arithmetic operations include addition, subtraction, multiplication, and division.Logical operations include AND, OR, XOR, and NOT operations.
- ✓ The ALU takes input from registers or memory, performs the specified operation, and produces output.
- ✓ In our 16-bit CPU, the ALU will be responsible for executing arithmetic and logical operations on 16-bit data.

Control Logic:

- ✓ The control logic coordinates the operation of various CPU components to execute instructions.
- ✓ Control signals regulate activities such as fetching instructions from memory, decoding instructions, reading and writing data to registers or memory, and controlling the ALU.

- ✓ The control logic ensures that instructions are executed in the correct sequence and that data flows smoothly through the CPU.
- ✓ In our 16-bit CPU design, the control logic will be crucial for orchestrating the execution of instructions and coordinating data movement.

Instruction Decoding:

- ✓ Instruction decoding involves interpreting the binary representation of instructions fetched from memory.
- ✓ Each instruction consists of an opcode (operation code) and possibly one or more operands.The instruction decoder interprets the opcode to determine the type of operation to be performed and the operands involved.
- ✓ Based on the decoded instruction, the CPU initiates the appropriate actions, such as fetching additional operands, performing calculations, or branching to a different instruction.
- ✓ In our 16-bit CPU, efficient instruction decoding is essential for accurately executing a wide range of instructions.

Memory Access Units:

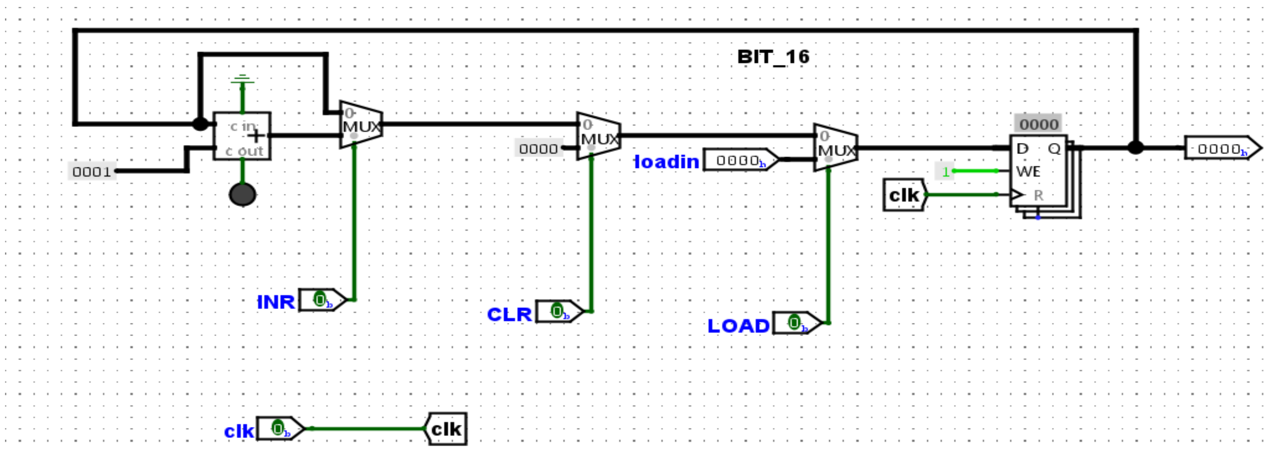
- ✓ Memory access units facilitate communication between the CPU and external memory devices.
- ✓ They handle operations such as reading instructions and data from memory, and writing data back to memory.Memory access units ensure efficient access to memory resources and proper synchronization with CPU operations.
- ✓ In our 16-bit CPU design, memory access units will enable the CPU to interact with program instructions and data stored in memory.

These components work together harmoniously within the CPU to execute instructions, manipulate data, and perform computations, enabling the CPU to fulfill its role as the core processing unit of a computing system.

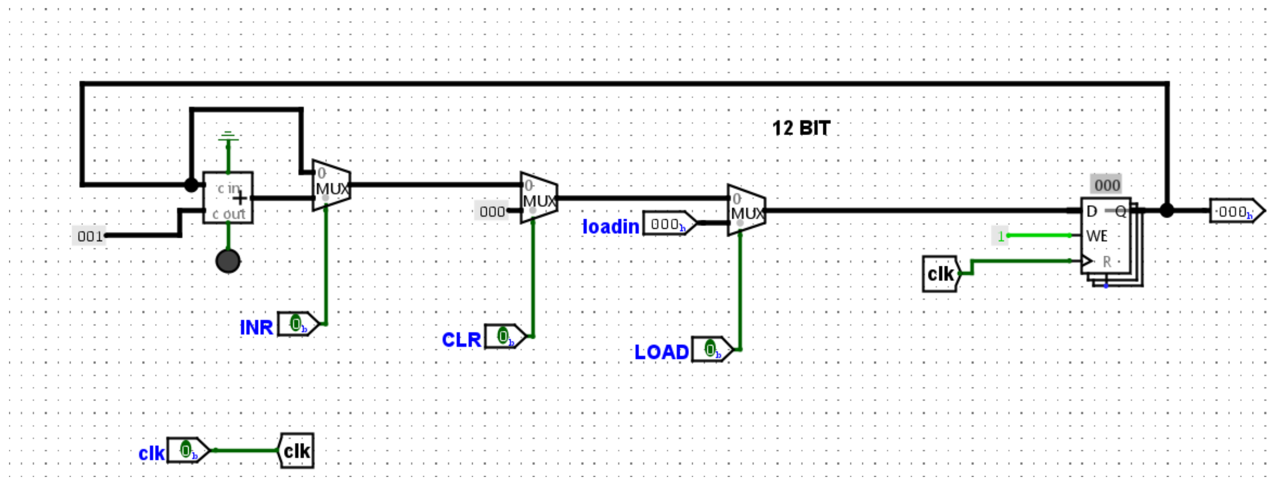
CIRCUIT AND ITS SUB-CIRCUITS

REGISTER

- ✓ 16-bit register: A 16-bit register is a storage unit capable of holding data with a size of 16 bits. It is typically used to temporarily store operands, addresses, or data during the execution of instructions within the CPU.



- ✓ 12-bit register: It stores data with a size of 12 bits. It is often utilized in scenarios where data with smaller bit widths are required, such as in specific arithmetic or logical operations.

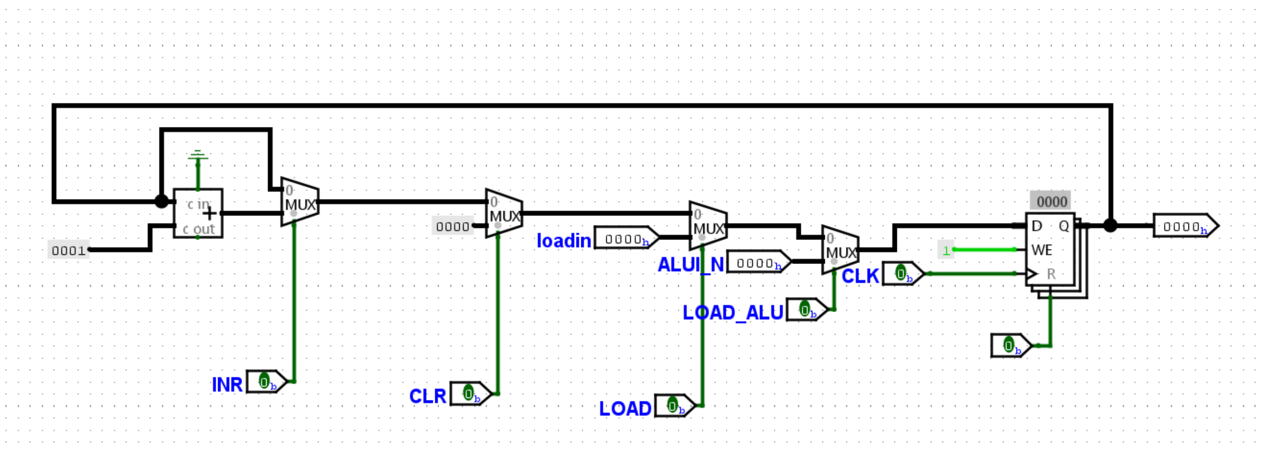


- ✓ AR Logic: The Arithmetic Register (AR) Logic handles arithmetic operations involving registers. It coordinates the fetching, storing, and manipulation of data within registers during arithmetic operations like addition, subtraction, multiplication, and division.

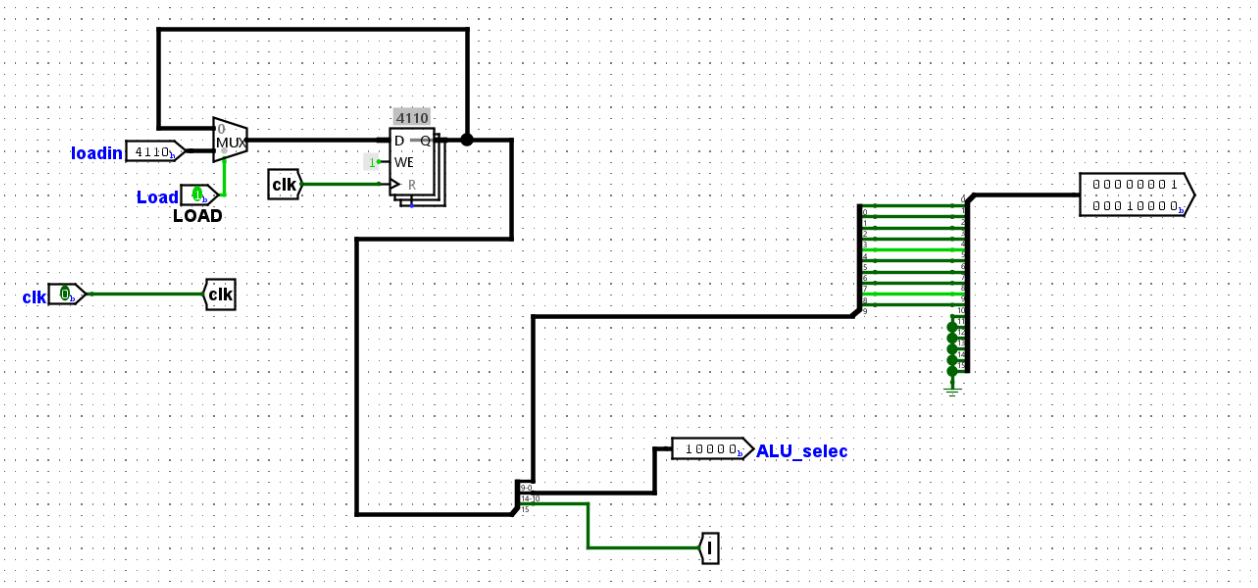
- ✓ PC Logic: The Program Counter (PC) Logic manages the program counter register, which stores the memory address of the next instruction to be fetched and executed. It controls the incrementing of the program counter and facilitates branching operations.

- ✓ DR Logic: The Data Register (DR) Logic governs the data register, responsible for holding data temporarily during data processing operations. It handles data transfer between the CPU and memory, as well as between other CPU components.

- ✓ AC Logic: The Accumulator (AC) Logic oversees the accumulator register, which serves as a temporary storage location for intermediate arithmetic and logical results. It plays a central role in arithmetic and logical operations performed by the CPU.



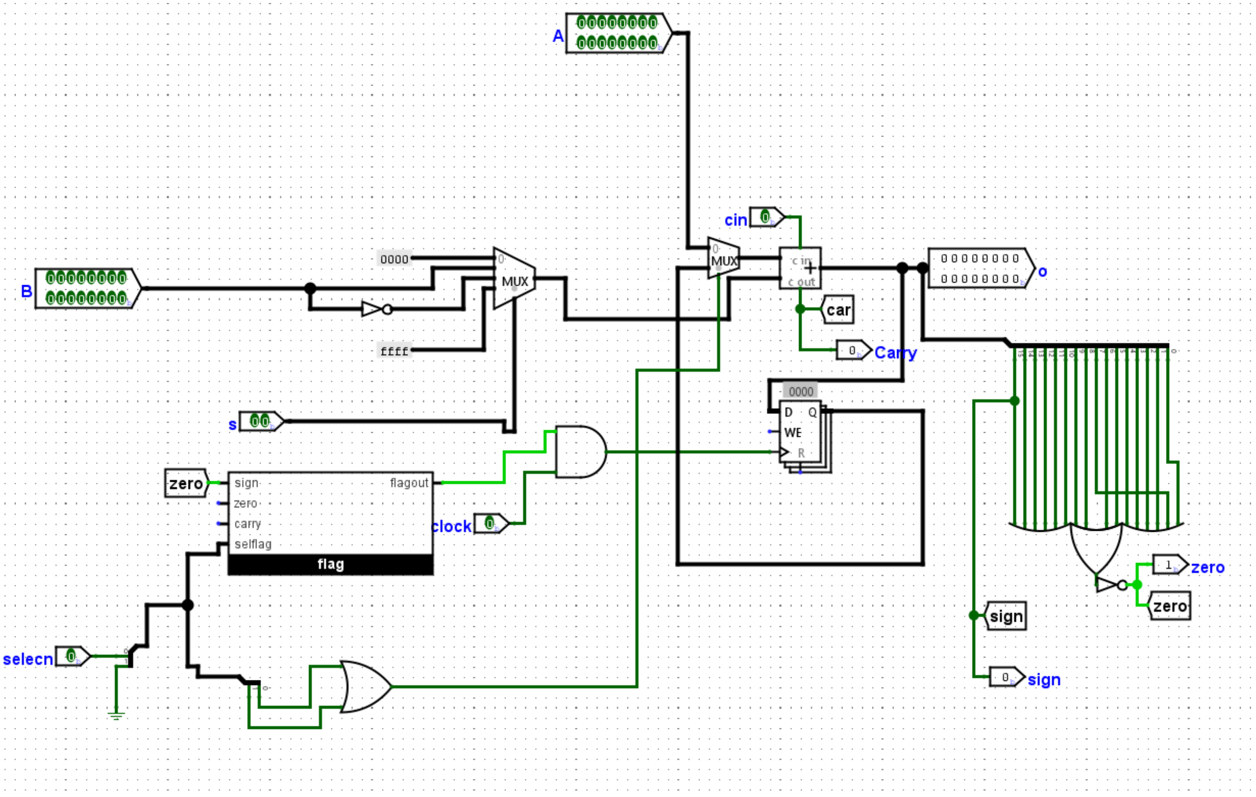
- ✓ IR Logic: The Instruction Register (IR) Logic manages the instruction register, which stores the current instruction being executed by the CPU. It facilitates the decoding and execution of instructions fetched from memory.



- ✓ TR Logic: The Temporary Register (TR) Logic controls temporary registers used for storing transient data during various CPU operations. These registers are often employed for holding intermediate results or facilitating data movement within the CPU.

ARITHMETIC LOGIC UNIT(ALU)

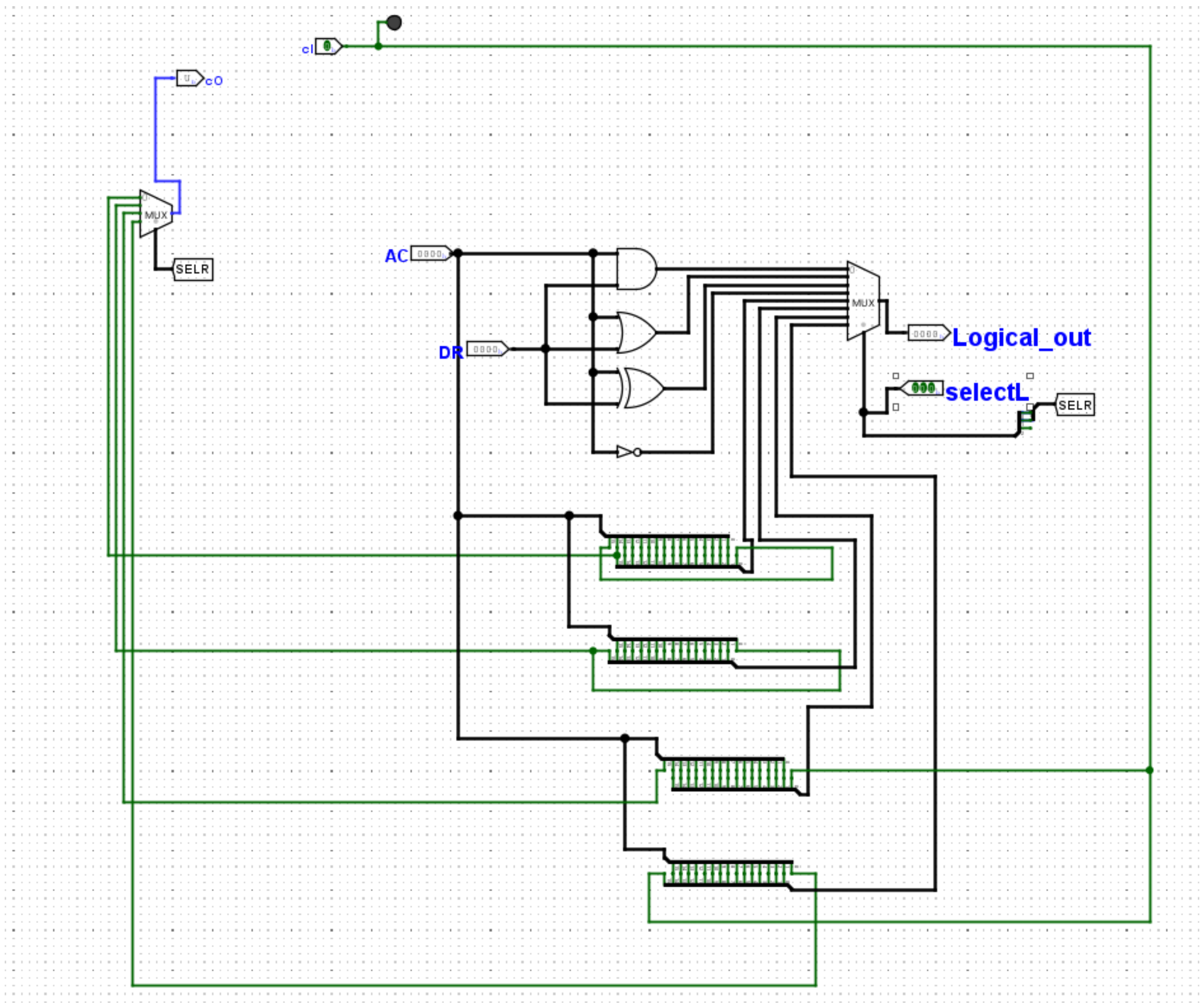
- ✓ Arithmetic Unit: The Arithmetic Unit within the ALU performs arithmetic operations such as addition, subtraction, multiplication, and division on binary numbers. It operates on operands provided by registers or memory locations and produces results based on the specified operation.



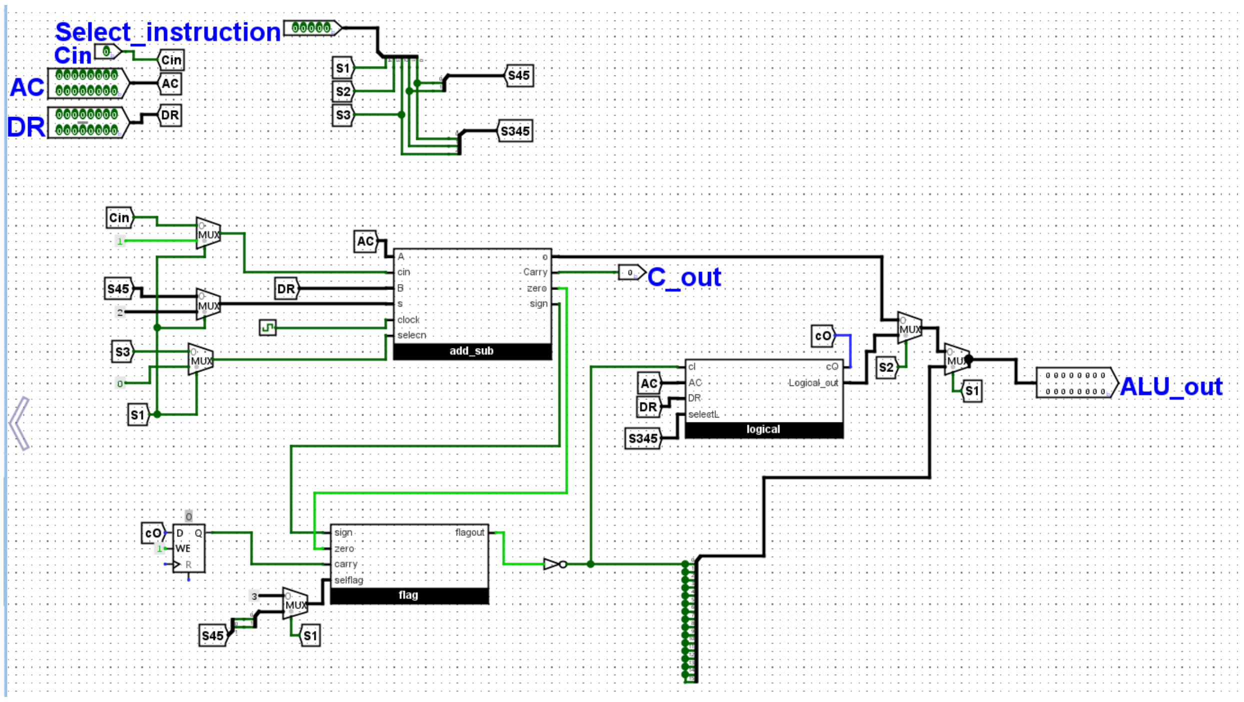
- ✓ Logical Unit: The Logical Unit handles logical operations like AND, OR, XOR, and NOT. It performs bitwise logical operations on binary data, manipulating individual bits based on the logical operation specified in the instruction.

- ✓ Shift Unit: The Shift Unit is responsible for shifting binary data left or right by a specified number of bits. It facilitates operations such as logical shifts, arithmetic shifts, and rotate operations, enabling data manipulation at the bit level.

LOGICAL AND SHIFT UNIT



- ✓ Main ALU circuit: The Main ALU Circuit comprises the core arithmetic and logical processing unit within the CPU. It integrates the Arithmetic Unit, Logical Unit, and Shift Unit to perform a wide range of arithmetic, logical, and data manipulation operations.

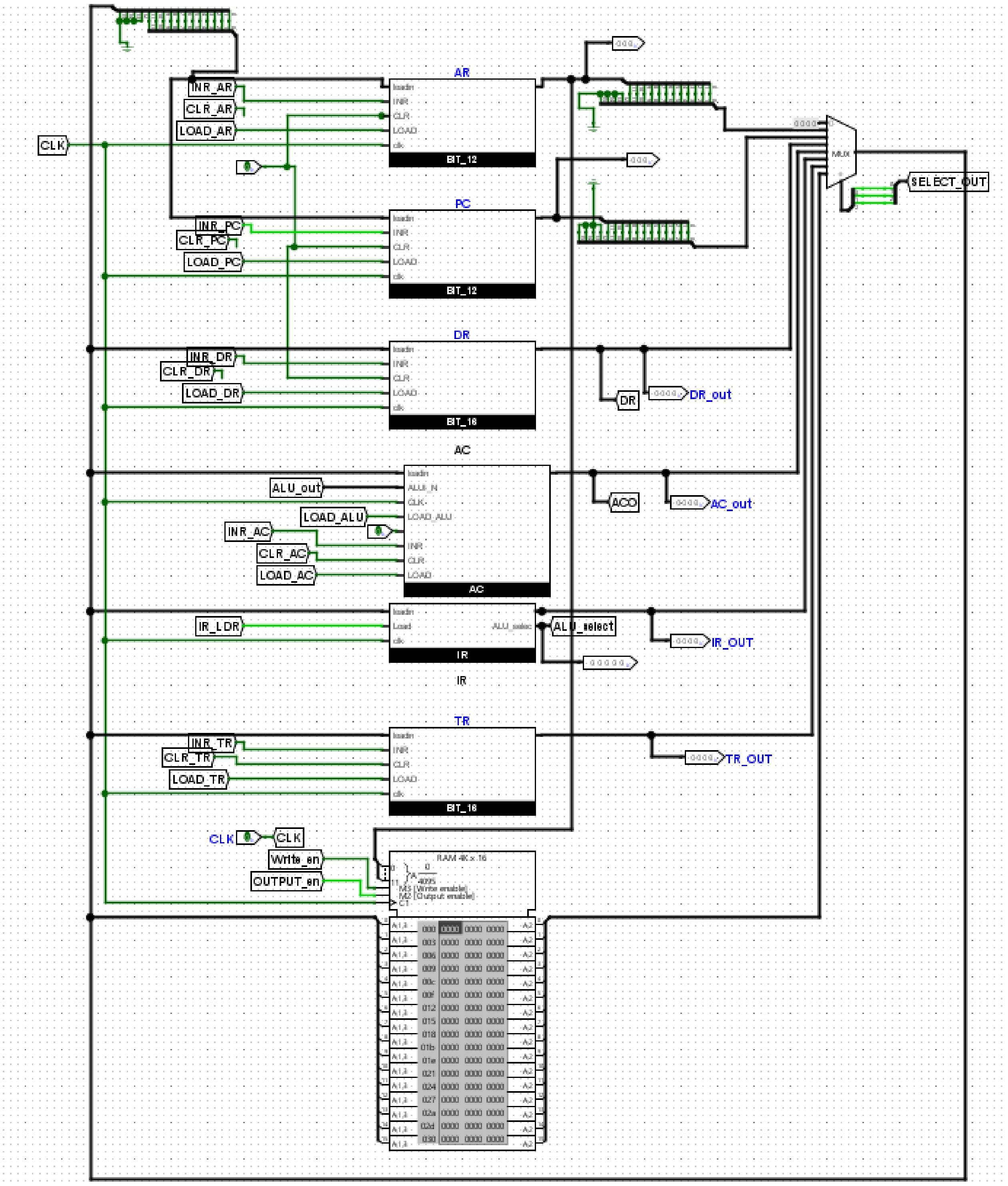


- ✓ ALU Logic: The ALU Logic coordinates the operation of the Arithmetic Logic Unit, generating control signals and directing data flow within the ALU. It ensures proper sequencing and execution of arithmetic, logical, and shift operations based on the instructions provided.

S1	S2	S3	S4	S5	Cin	Micro instruction
0	0	0	0	0	0	Transfer A
0	0	0	0	0	1	A+1
0	0	0	0	1	0	A+B
0	0	0	0	1	1	A+B+1
0	0	0	1	0	0	A+B’(A-B-1)
0	0	0	1	0	1	A+B’+1(A-B)
0	0	0	1	1	0	A-1
0	0	0	1	1	1	Transfer A
0	0	1	1	1	0	Do until zero
0	1	0	0	0	X	A && B
0	1	0	0	1	X	A || B
0	1	0	1	0	X	A ^ B
0	1	0	1	1	X	NOT A
0	1	1	0	0	X	Rotate left without carry
0	1	1	0	1	X	Rotate right without carry
0	1	1	1	0	X	Rotate left with carry
0	1	1	1	1	X	Rotate Right with carry
1	0	X	0	1	X	A<=B
1	0	X	1	0	X	A=B

COMMON BUS SYSTEM

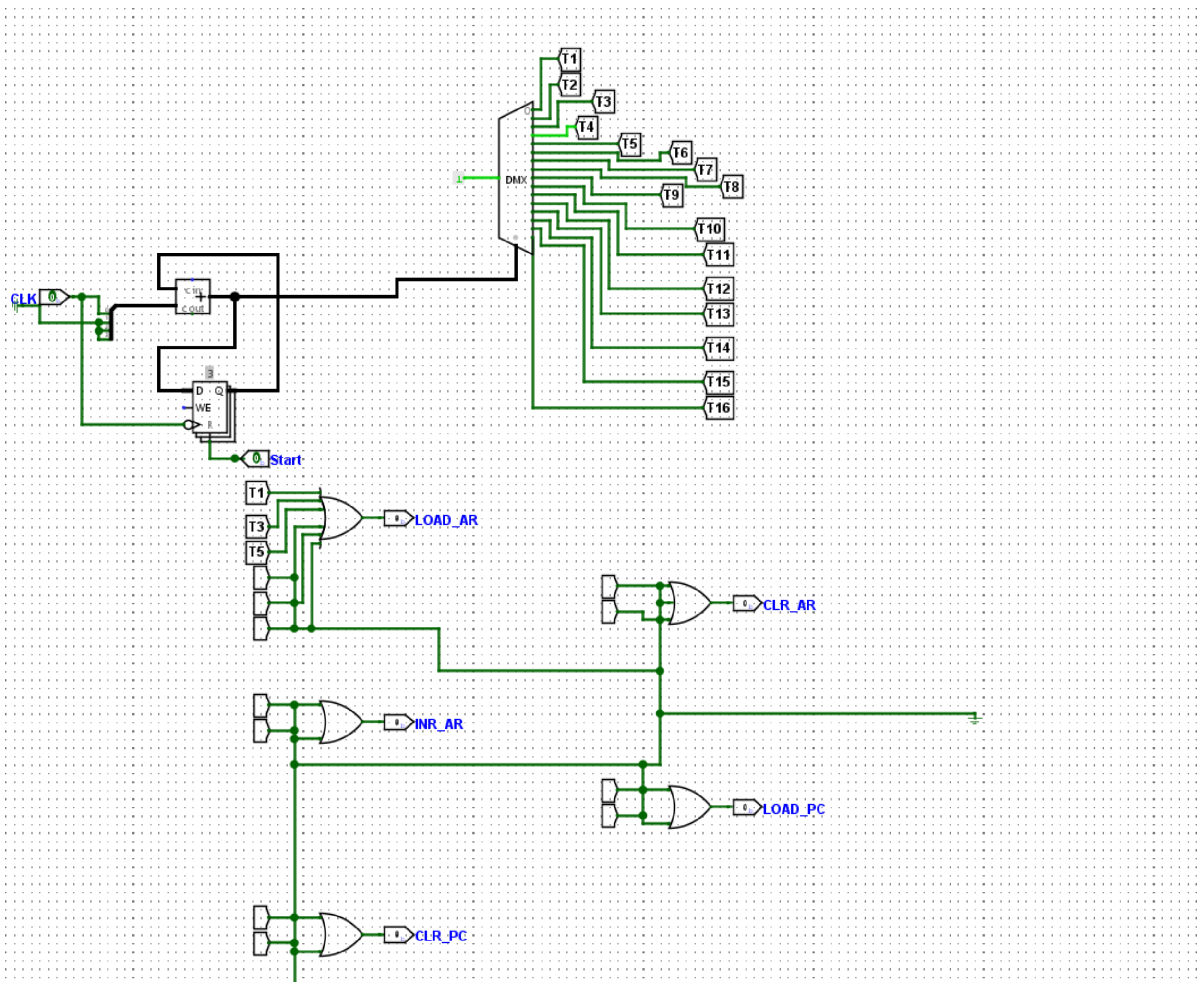
- ✓ Common Bus Circuit: The Common Bus Circuit provides a shared communication pathway for transferring data between various CPU components. It allows for the efficient exchange of data among registers, the ALU, memory access units, and other components connected to the CPU.

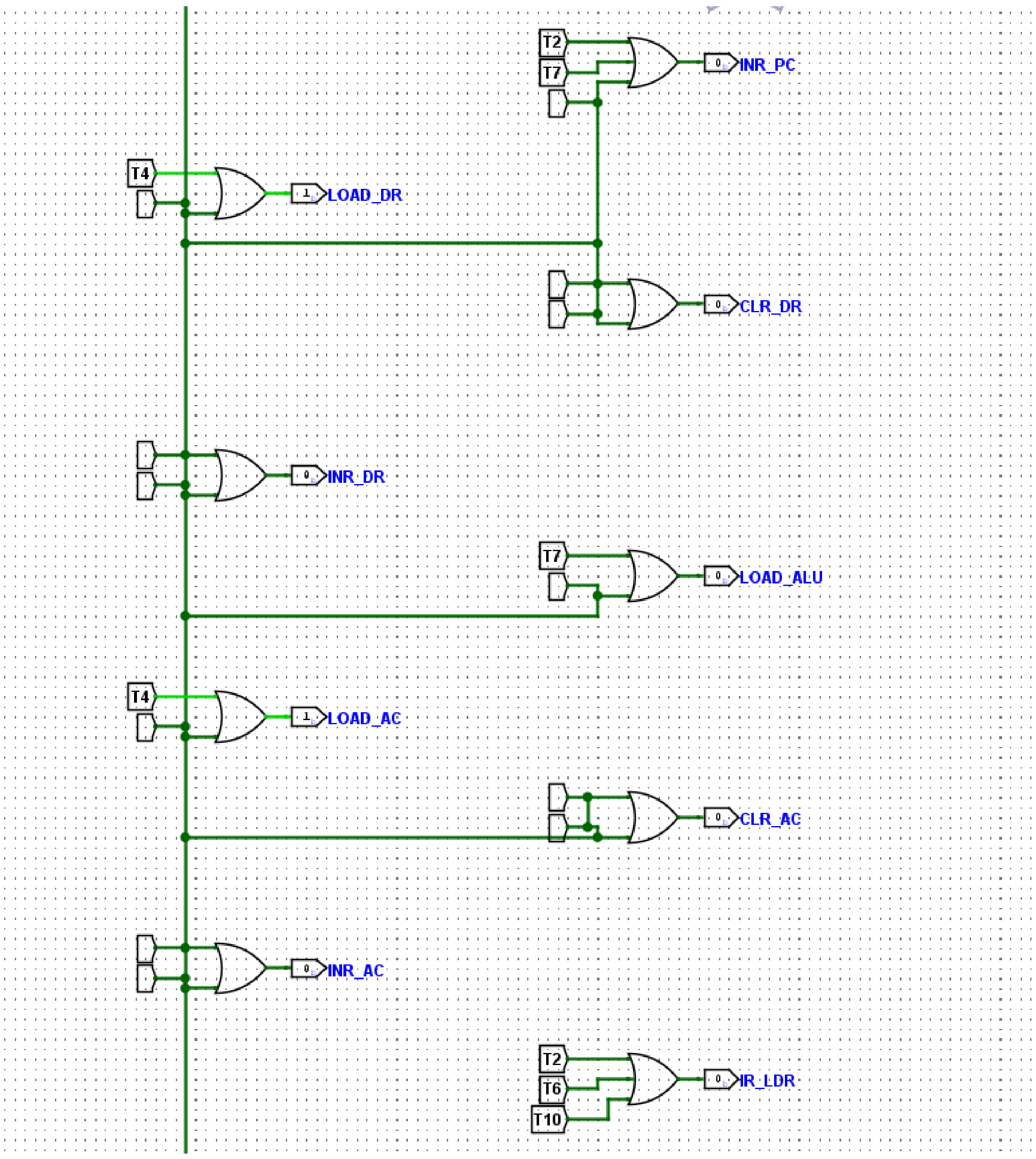


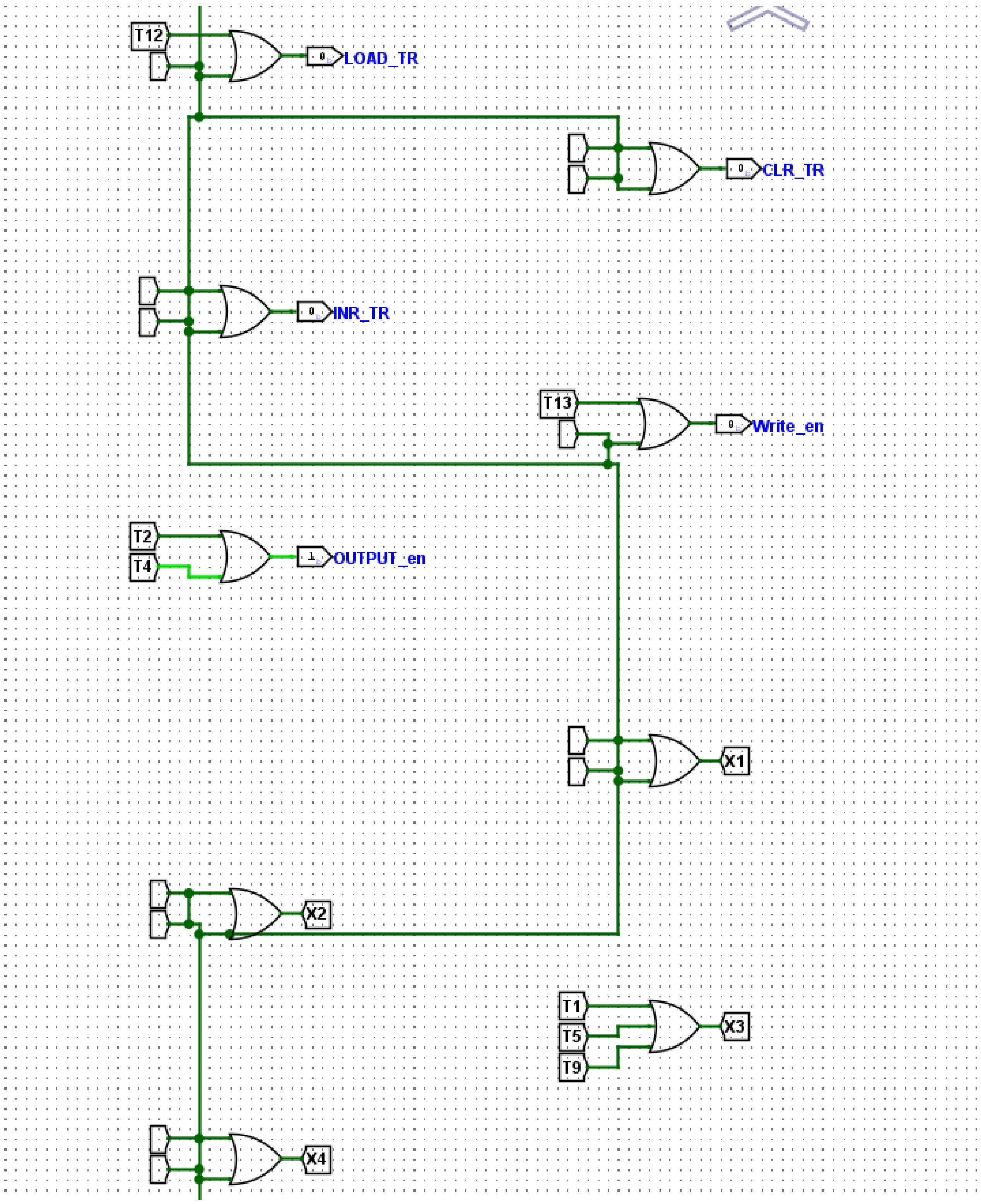
- ✓ Bus Logic: The Bus Logic governs data flow on the common bus, regulating the timing and direction of data transfers. It manages bus arbitration, ensuring that only one component accesses the bus at a time to prevent data collisions and ensure data integrity.

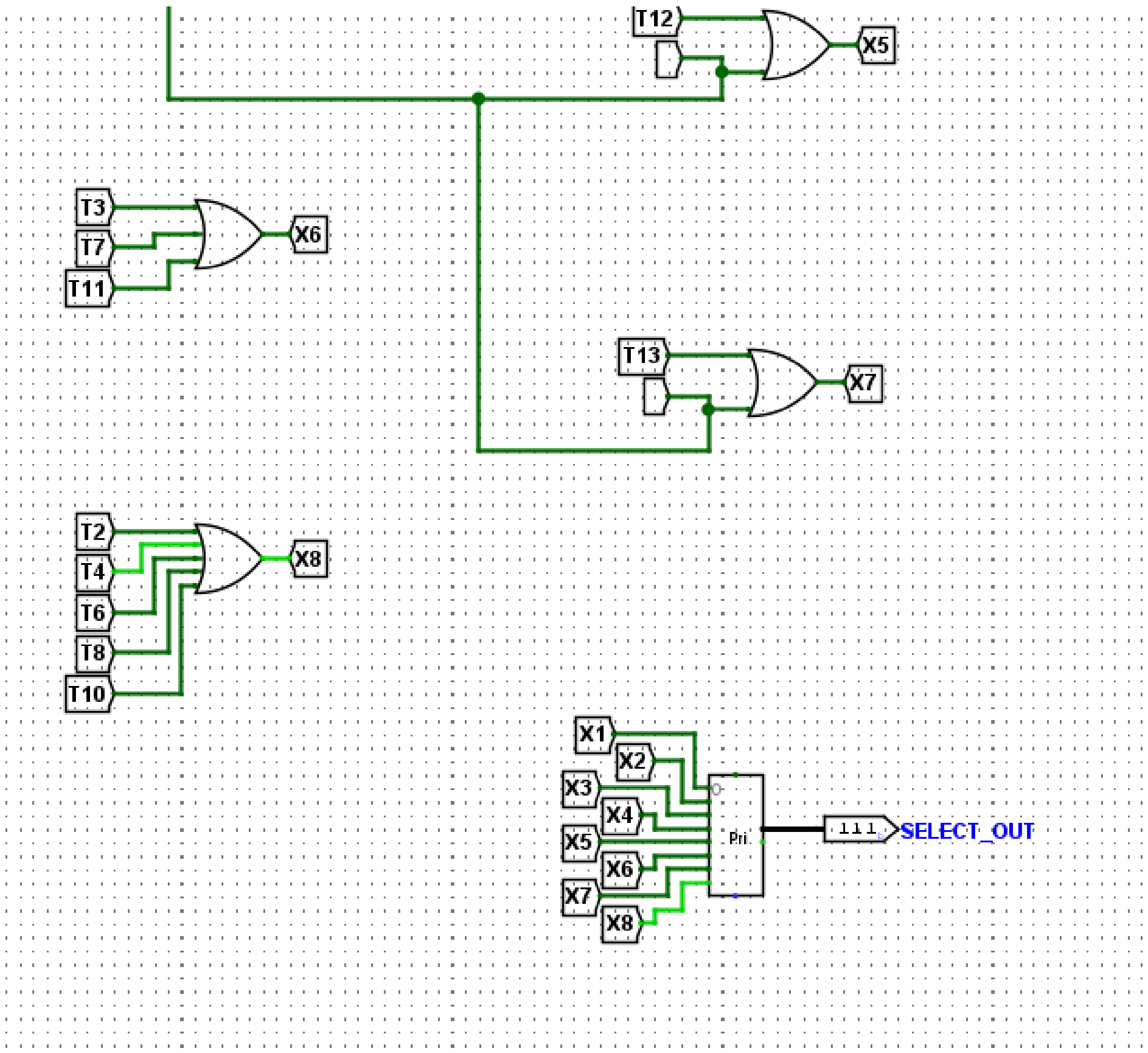
CONTROL UNIT

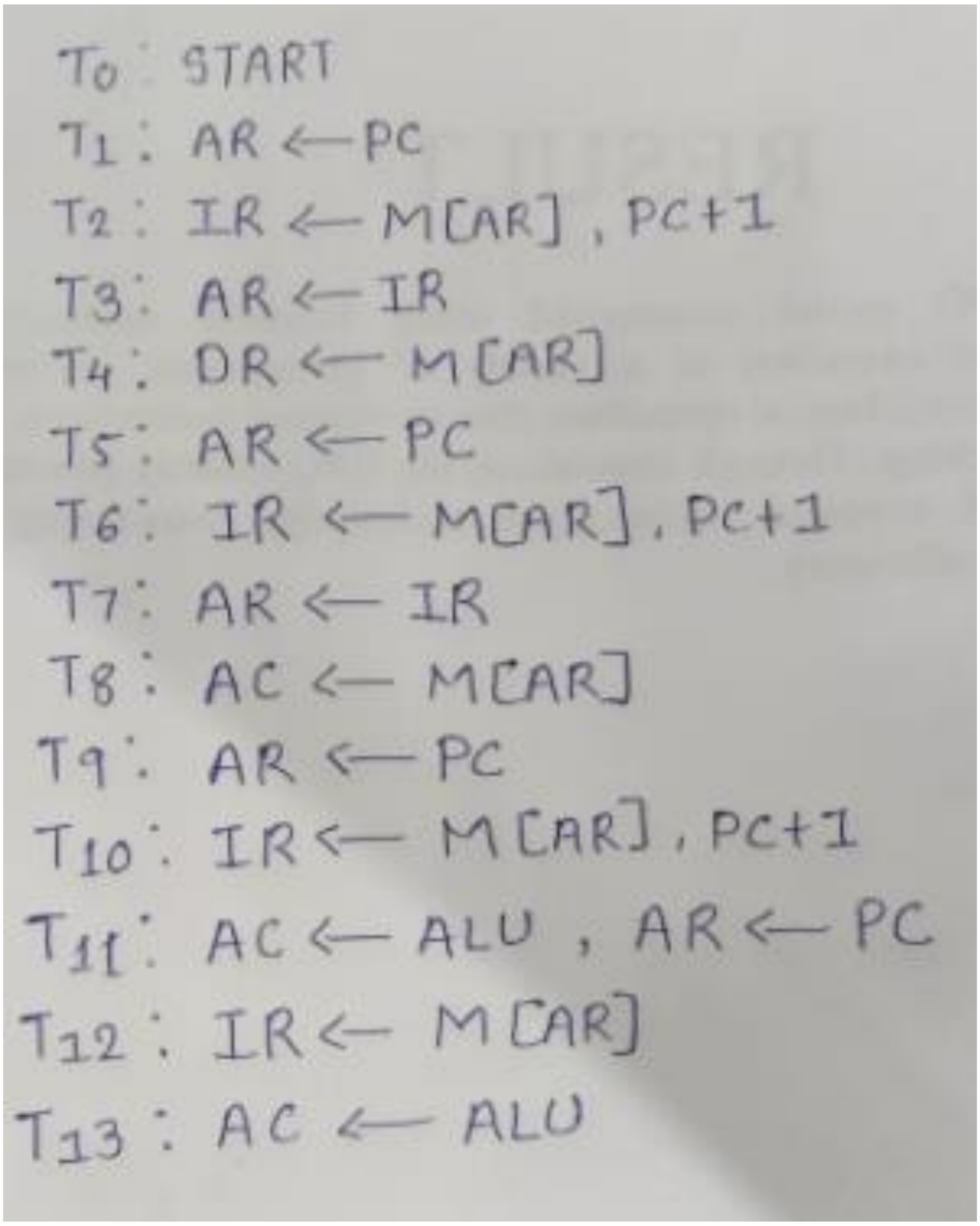
- ✓ Timing and Control unit circuit: The Timing and Control Unit Circuit synchronizes the operation of CPU components by generating timing signals and control signals. It coordinates the execution of instructions, controlling the sequencing of fetch, decode, execute, and writeback stages in the CPU pipeline.




T2
T7
INR_PC
T4
LOAD_DR
CLR_DR
INR_DR
T7
LOAD_ALU
T4
LOAD_AC
CLR_AC
INR_AC
T2
T6
T10
IR_LDR


T12
LOAD_TR
CLR_TR
INR_TR
T13
Write_en
T2
T4
OUTPUT_en
X1
X2
T1
T5
T9
X3
X4


T12
X5
T3
T7
T11
X6
T13
X7
T2
T4
T6
T8
T10
X8
X1
X2
X3
X4
X5
X6
X7
X8
Pri
111
SELECT_OUT



T_0 : START

T_1 : $AR \leftarrow PC$

T_2 : $IR \leftarrow M[AR]$, $PC+1$

T_3 : $AR \leftarrow IR$

T_4 : $DR \leftarrow M[AR]$

T_5 : $AR \leftarrow PC$

T_6 : $IR \leftarrow M[AR]$, $PC+1$

T_7 : $AR \leftarrow IR$

T_8 : $AC \leftarrow M[AR]$

T_9 : $AR \leftarrow PC$

T_{10} : $IR \leftarrow M[AR]$, $PC+1$

T_{11} : $AC \leftarrow ALU$, $AR \leftarrow PC$

T_{12} : $IR \leftarrow M[AR]$

T_{13} : $AC \leftarrow ALU$

RESULT

The 16-bit CPU model constructed using Logisim successfully demonstrated the execution of a variety of instructions, including arithmetic operations, logical operations, data movement instructions, and conditional branching. Through simulation, the CPU reliably processed 16-bit data and executed instructions accurately, showcasing its functionality and efficiency.

STATEMENT OF SKILL DEVELOPMENT ACHIEVED

Dilon Brahmbhatt:

- I have learned about how to develop the ALU from scratchand itwas very fun as well as helpful in understanding the various anddeep concepts regarding the ALU design.

Ved Vyas:

- I have also gone into deeper concepts of ALU designing which performs different operations like logical, arithmetic, shift. Moreover, I investigated the synchronization between the ALU and other CPU components, ensuring smooth communication and seamless operation.

Het Virani

- I have learned the concept about data bus designing and how the different 16 bit and 12 bit registers work together as well as why they are so important.

Om Patel

- I understood and designed the timing sequence for the CPU which helps to automate the task with the help of clock pulse selecting output automatically.

Ren Patel

- I gave effort in designing the control unit for the CPU and gained a lot of insight about the working of the CPU.