Computer Organization & Design Laboratory (20IC209P)

Project Report Submitted to

Pandit Deendayal Energy University, Gandhinagar

*for*



Bachelor of Technology

*in*

Information and Communication Technology Department

Submitted by

DHANDHUKIYA HARSH (22BIT144)

Course Faculty

Dr. Gangaprasad Pandey

Dr. Ankur Changela

Dr. Manoj Yadav

 PANDIT DEENDAYAL ENERGY UNIVERSITY, GANDHINAGAR – 382007, India

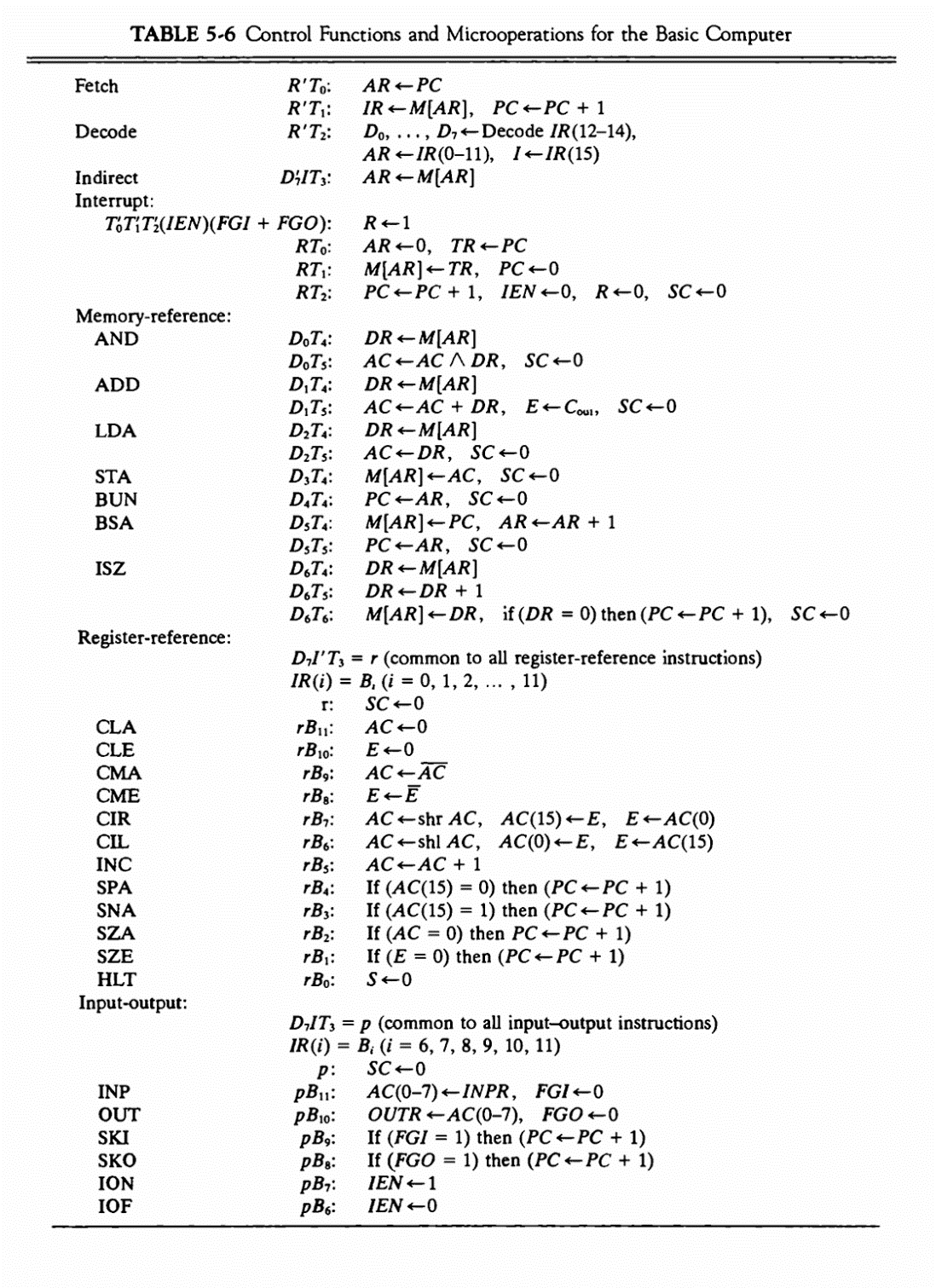
* **AIM: -**

To design Central Processing Unit – CPU to perform various logical and arithmetic operations.

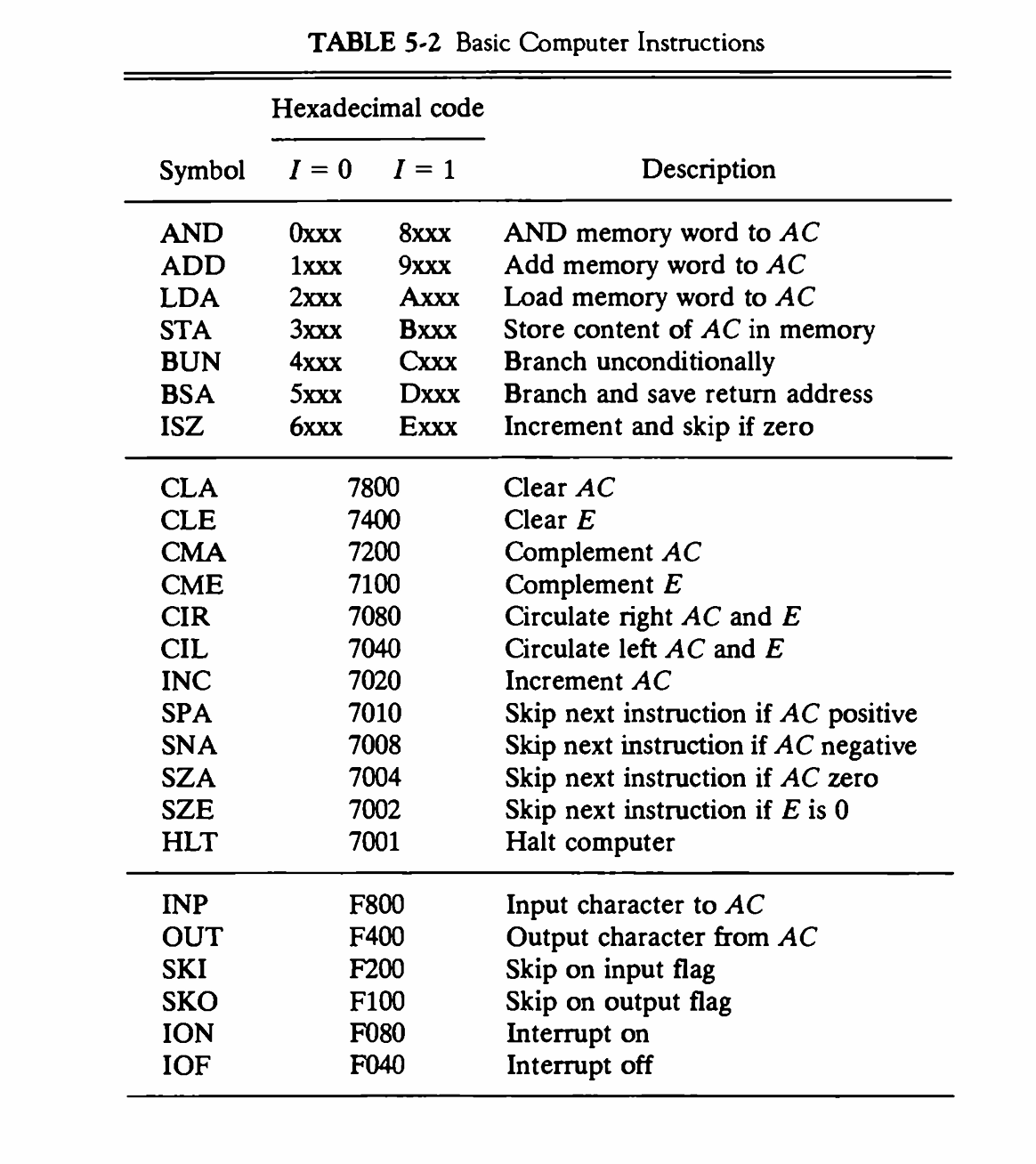
* **APPARATUS: -**

Logisim Software

* **SPECIFICATIONS: -**
* There are certain registers like TR, AC, DR and IR that is used in this circuit design is 16-bit registers.
* PC and AR both are 12-bit register itself.
* Memory which is used in this CPU is 4Kx16 (there is 12-bits of address lines using 12-bit registers).
* **THEORY: -**
* In broadly summarize, this entire circuit could be divided into three different parts or units which is known as **A**rithmetic **L**ogic **U**nit (ALU) based on its individual operations; Arithmetic Unit, Logical Unit and Shift Operations.
* The Arithmetic unit simply performs a basic mathematical operation in the CPU such as Addition, Subtraction, Add with carry, sub with carry, transfer, Increment and Decrement.
* The Logical units executes all of the logical operations occurs in the CPU like AND, OR, XOR & NOT as well.
* Last unit is Shift operations which simply performs the arithmetic shift. Logical shift and circular shift tasks in the CPU.
* In order to design this CPU, we used as well as refer several Register Transfer Language (RTL) instructions which is as per the following;



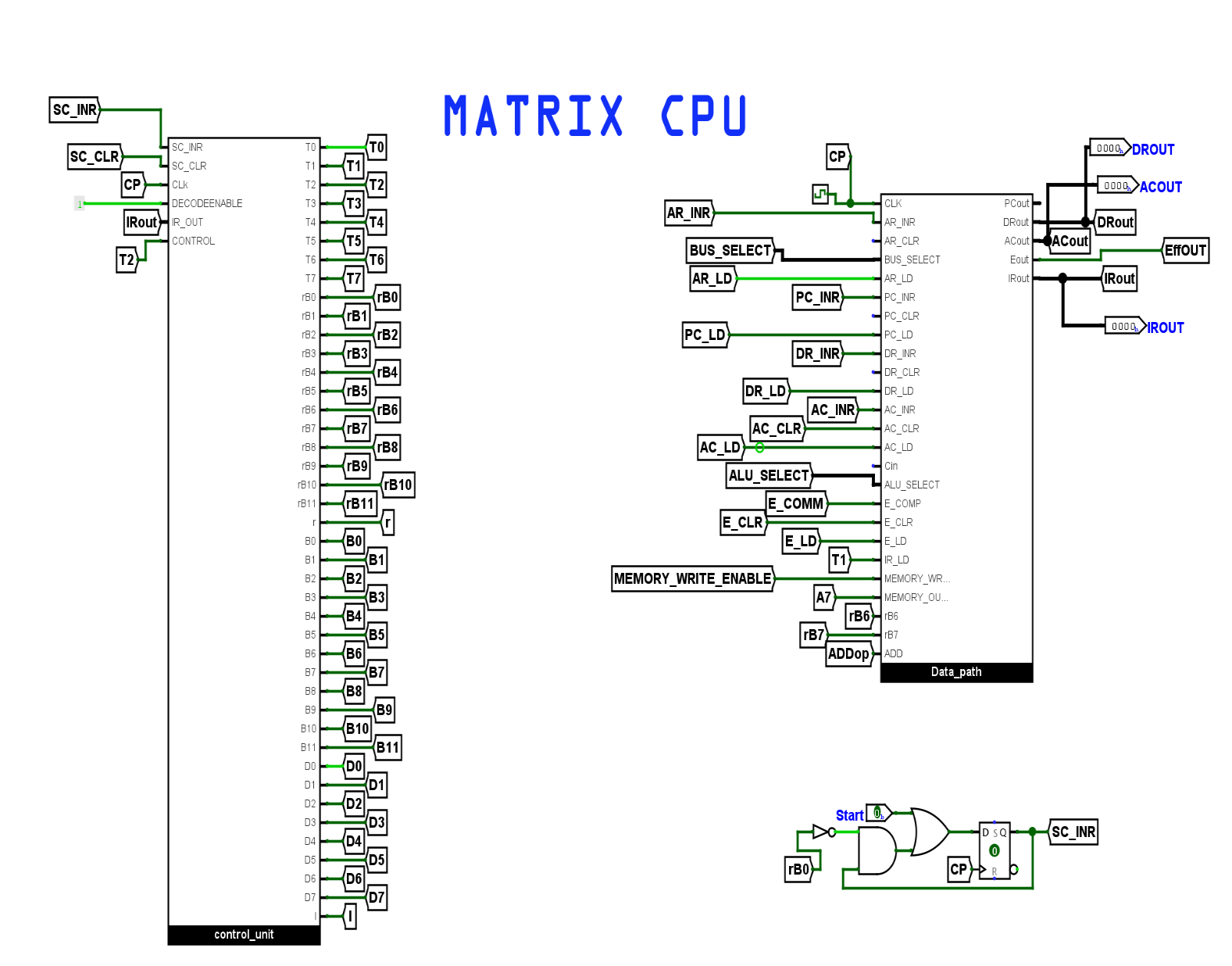
* It is outlining the different parts of the CPU and the instructions it can carry out. Here is a breakdown of the table along with how it relates to your CPU design:



* **Control Unit:**
* The control unit is responsible for fetching, decoding, and executing instructions.
* Based on the opcode (operation code) in the instruction register (IR), the control unit generates various control signals to other parts of the CPU.
* In the image, the table shows the micro-operations for each instruction which are the basic building blocks for the control unit.
* **Data Path:**
* The data path is the circuitry that performs the ALU operations and data transfers directed by the control unit.
* The data path includes the arithmetic logic unit (ALU), registers, and memory
* The ALU performs arithmetic and logical operations on data.
* Registers store data for temporary use.
* Memory stores data and instructions.
* **Instruction set architecture (ISA):**
* This refers to the set of instructions that your CPU can understand and execute. The table shows a limited ISA with instructions for Load (LDA), Store (STA), Add (ADD), and Branch (BUN) operations.
* **Instruction Fetch:**
* The CPU retrieves instructions from the computer's memory using the Instruction Pointer (IP) or Program Counter (PC).
* **Instruction Decode:**
* It decodes the fetched instruction to determine the operation and operands involved.
* **Operand Fetch:**
* If needed, the CPU retrieves operands from memory or registers.
* **Execution:**
* It performs the operation based on the decoded instruction and fetched operands.
* **Write Back:**
* If necessary, the CPU stores the result of the operation.
* **Repeat:**
* The CPU repeats these steps for the next instruction, following the program's flow.
* Throughout these steps, the CPU interacts with other components like memory, registers, and input/output devices. Components like the ALU, Control Unit, and Register File help manage data and instructions.

The CPU operates based on instructions stored in memory, executing them sequentially until the program is completed or terminated.

* **CIRCUIT DIAGRAM: -**

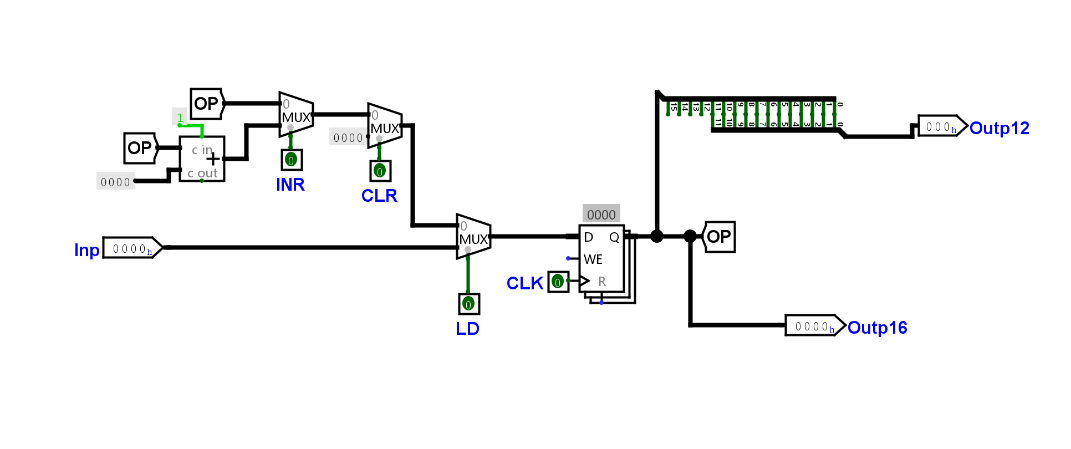


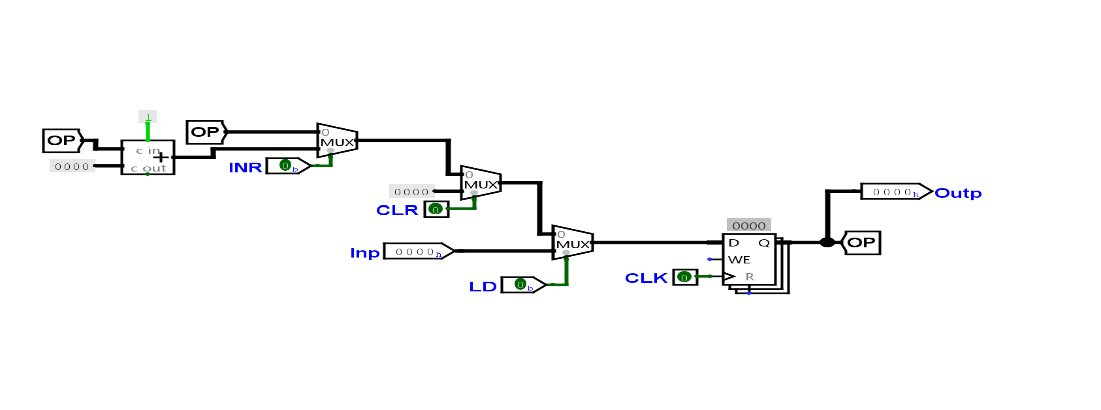
* **CIRCUIT DESCREPTION: -**

1. **12-bit & 16-bit Register**

A 12-bit register is a digital circuit component capable of storing and manipulating 12 binary bits, each representing 0 or 1. It is typically made up of 12 flip-flops or memory cells organized sequentially, allowing for separate read and write operations at the bit level. It can store up to 12 bits of data and represent decimal values from 0 to 4095. 12-bit registers are widely used in digital systems for data storing, buffering, and arithmetic operations. They are ideal for moderate precision needs.

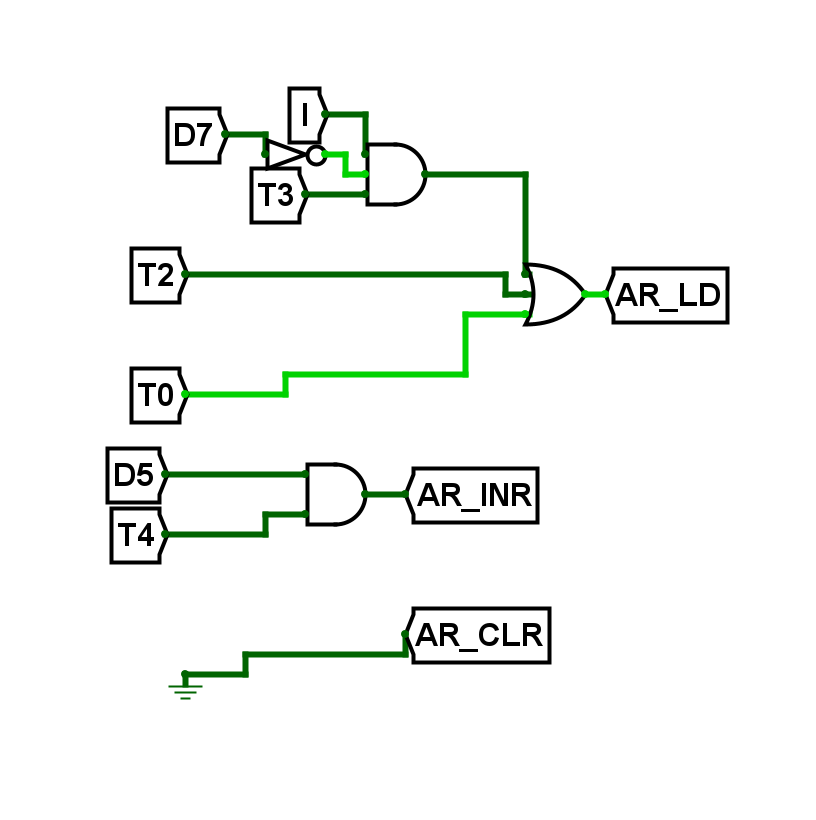
A 16-bit register stores and manipulates 16 binary values, each representing a 0 or 1. It usually consists of 16 flip-flops or memory cells arranged Sequentially stores and retrieves 16 bits of data. This register is useful in digital systems for storing data, buffering, and doing arithmetic. It can store 16 bits of data and represent decimal numbers from 0 to 65535. 16-bit registers are commonly used in computer systems and microprocessors for managing data and instructions, with a reasonable precision requirement.





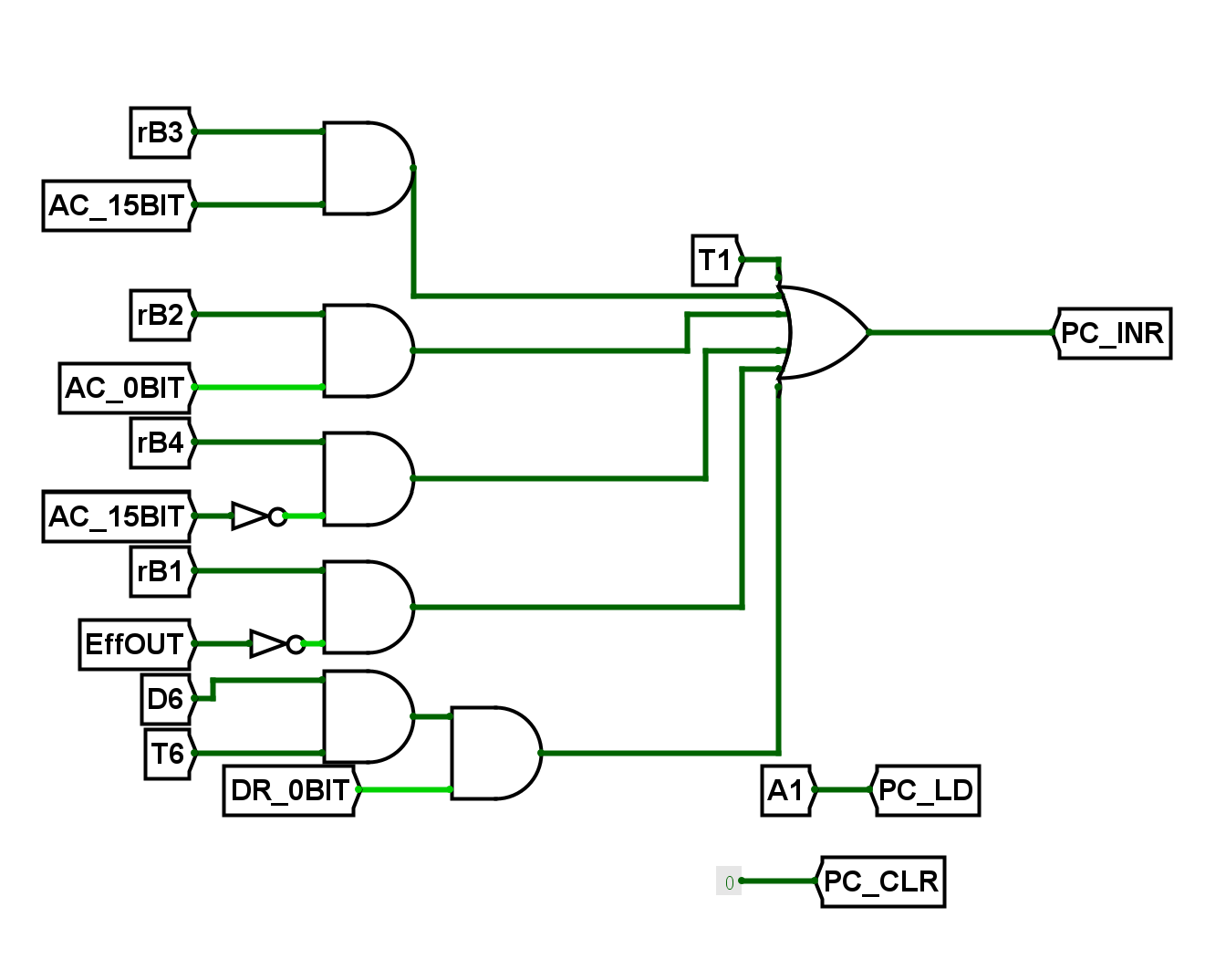
1. **AR**

An address register is a digital circuit component in computers that maintains memory addresses for data or instructions. It identifies the memory location where data or instructions are stored. A CPU uses the address register to access memory data and instructions. Address registers are commonly employed in microprocessors and computing systems to retrieve data or instructions from memory during the fetch phase. They are essential for memory addressing, enabling the CPU to effectively locate and handle stored data.



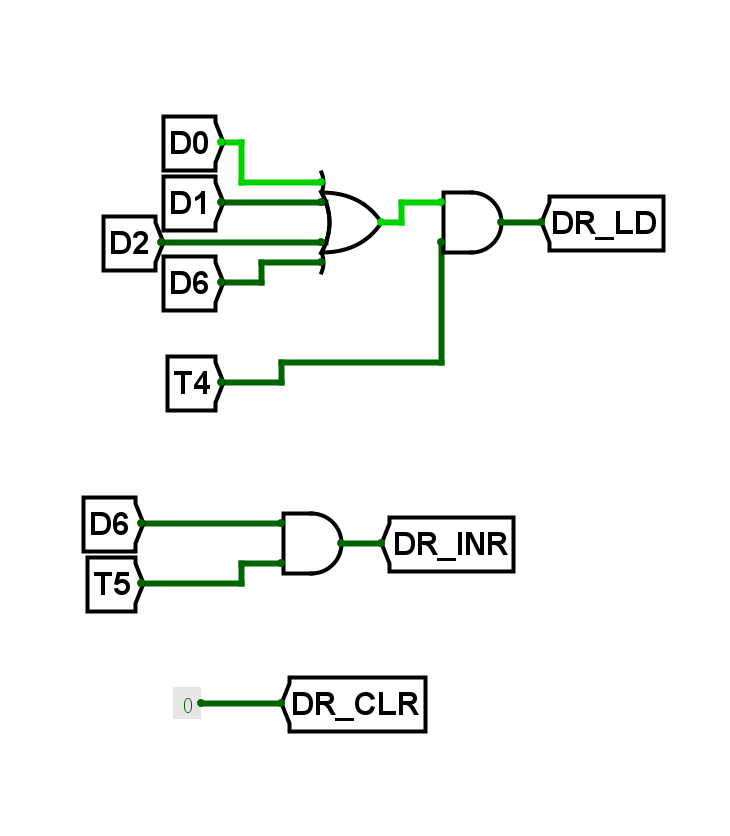
1. **PC**

The program counter, also known as the instruction pointer, can also be referred to as the instruction address register, instruction counter, or simply the instruction itself. A sequencer is a processor register that indicates a computer's current program sequence. This contains the address of the following instruction. This is a 12-bit register that stores the address of the next instruction.



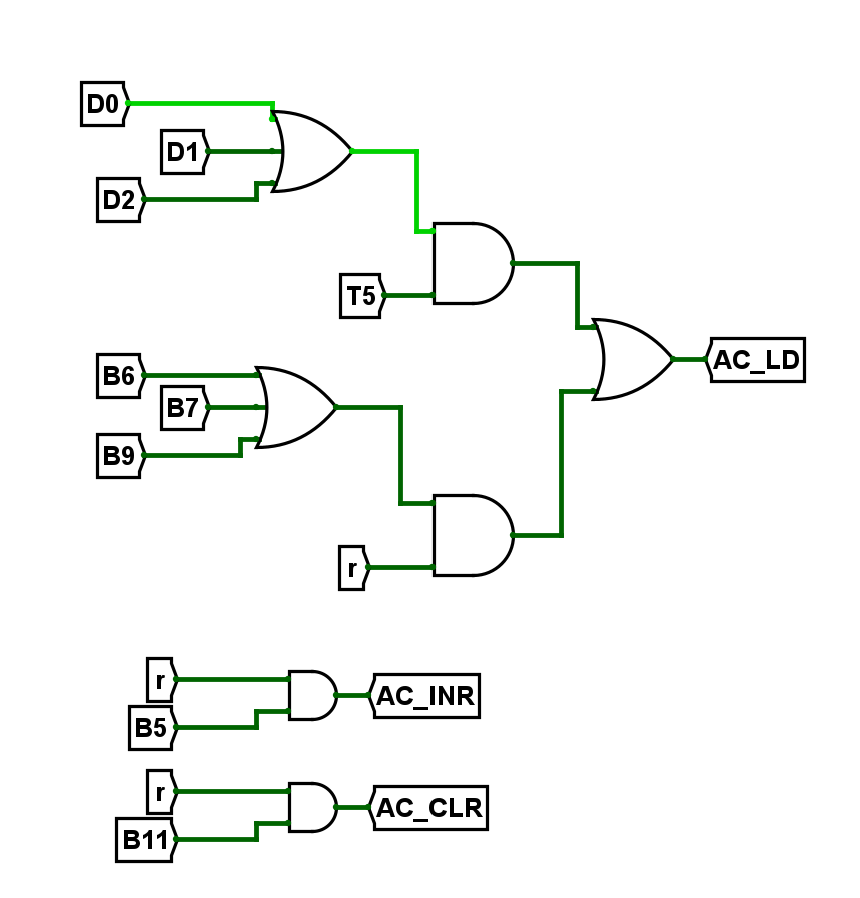
1. **DR**

A data register is a digital circuit component used in computers to store data. A register stores CPU-processed data. Microprocessors utilize data registers. Data can be temporarily stored on other computing systems while it is being read from memory, calculated, or altered. Binary values can represent data, including numbers, characters, and other information. Data registers store data temporarily during instruction execution and are essential for CPU data processing. Data contained in a register can be manipulated and transferred to other registers or memory locations.



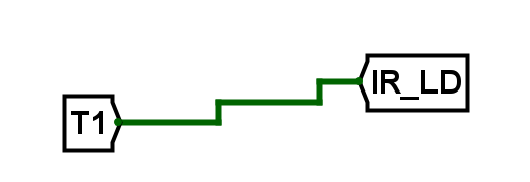
1. **AC**

An accumulator is a CPU register that performs arithmetic and logic operations. This is a temporary storage area for intermediate results or operands used during instruction execution. The accumulator serves as a working register for arithmetic and logic operations. Data is received from memory or input/output devices, processed by the CPU, and then saved in the accumulator or transferred to other registers or memory locations.



1. **IR**

This register stores all instructions for decoding and connects to the control and timing unit to decode the four MSBs. This is a 16-bit register, as this CPU's data size is 16 bits. The most significant bit (MSB) is 'I' (Mode bit), followed by three instruction bits (14-12) and twelve address bits (11-0). The instruction register stores the opcode (operation code) and other information, such as operands or addressing modes.

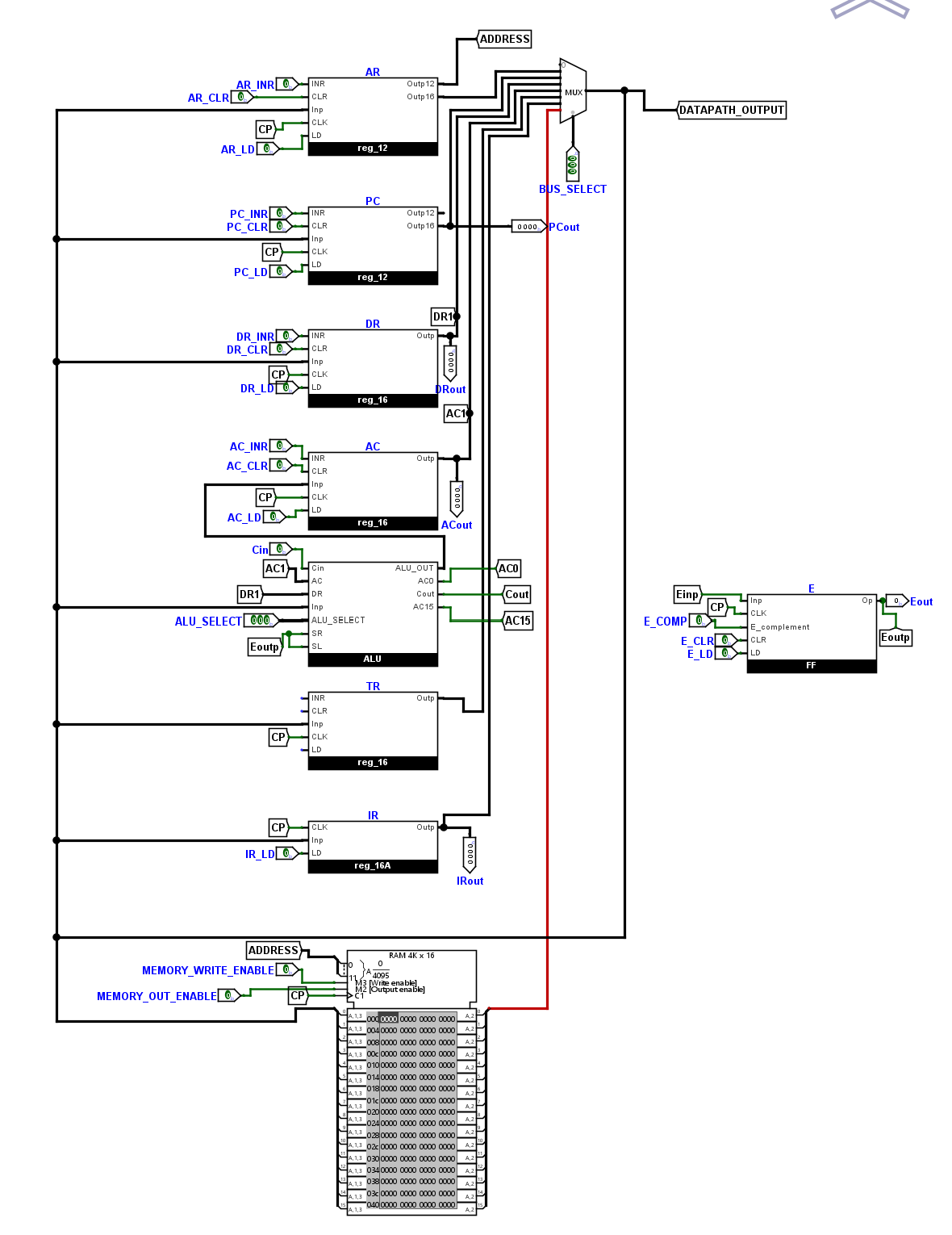


1. **TR**

Only temporary registers can be read and written many times within a single instruction. Temporary registers support single-write and triple-read access. Instructions can have up to three temporary registers as input source operands. This register stores temporary data, such as the next program instruction, between interrupt cycles. However, this register is not used in this project due to the absence of interrupts. This register is 16-bit in capacity, matching the CPU's data size.

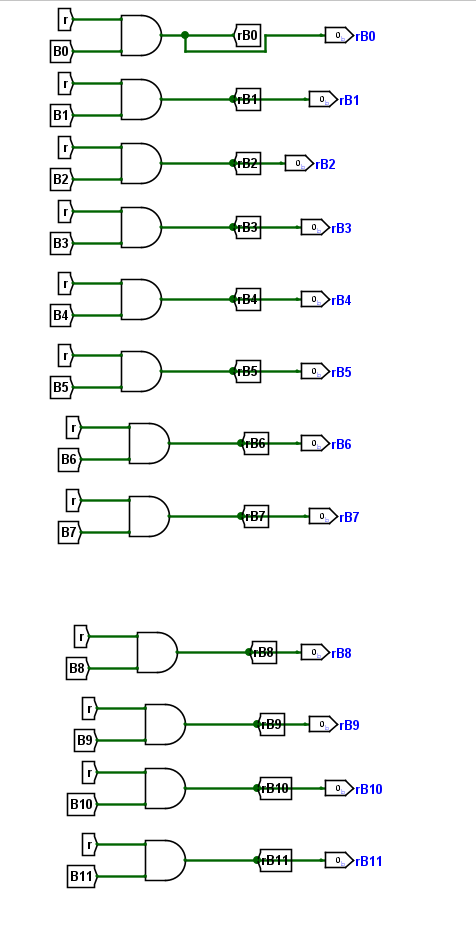
1. **Common Bus**

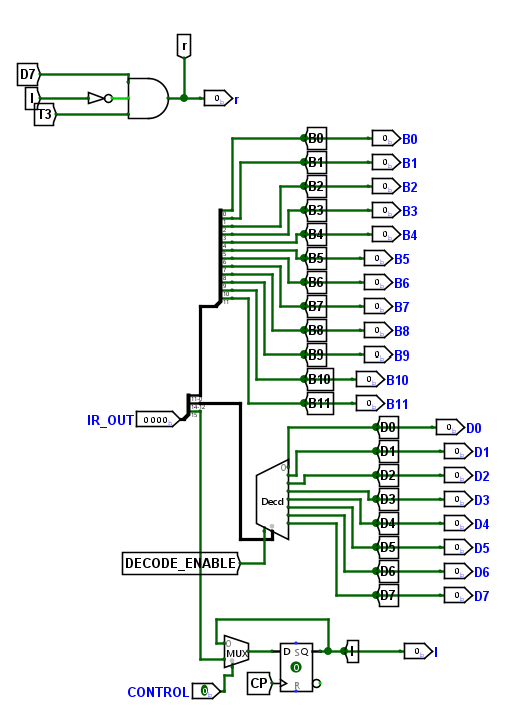
The 16-bit common bus is governed by selection inputs S2, S1, and S0. Each bus input is associated with a decimal number, indicating the binary configuration required for the corresponding register selection. These binary configurations, represented by Boolean variables x1 through x7, activate specific gate structures to choose the register or memory for the bus. For instance, when x1 equals 1, the binary sequence S2S1S0 must be 001, resulting in the selection of the output of AR for the bus. This setup facilitates efficient memory transfer between registers and memory within the CPU. Compared to using separate lines for each register's input and output, the common bus reduces complexity and cost. A decoder is employed to determine which register's data will be present on the common bus.

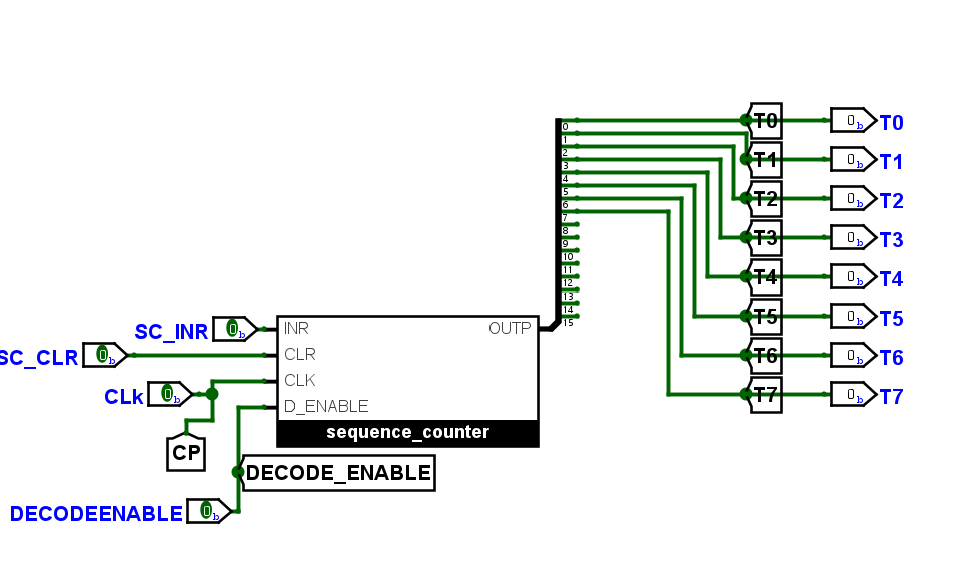


1. **Timing and Control Unit**

The Timing and Control Unit decodes IR data and generates control signals. This includes a 4-bit sequence counter and a decoder that creates timing signals. The 'I' in the MSB of IR indicates the mode. Bits 14 to 12 determine the instruction, which is then sent into a decoder to form a control signal. In memory reference instructions, bits 11 through 0 are address bits, but in register reference instructions, they are control signals. The generated signals are used in the hardwired control unit, which includes control logic for various registers.







* **CONCLUSION: -**

This project explains fundamental aspects essential for discerning hardware digital processing units, encompassing the replacement of microoperations by computer registers. Additionally, it elucidates the functionality of pivotal components such as the Arithmetic Logic Unit (ALU), registers, and multiplexers. Consequently, it can be asserted that the design of the 16-bit CPU exhibits success, having effectively addressed and integrated these foundational elements.