CHITTAGONG UNIVERSITY OF ENGINEERING AND TECHNOLOGY

ELECTRONICS AND TELECOMMUNICATION ENGINEERING

VLSI TECHNOLOGY SESSIONAL

ETE 404

Experiment No. - 02

Schematic Modeling and Implementation of modified SAP Architecture in Logisim-2

Submitted by:

Tayba Busra

1908017

Submitted to:

Arif Istiaque Rupom

LECTURER

DEPARTMENT OF ETE, CUET

May 12, 2024

Objective

- To familiarize with the BUS in a microprocessor system.
- To familiarize with addressing and RAM.
- To familiarize with the program execution cycle.

Design Process

Decoder:

To decode encoded data and activate particular outputs in response to input signals, a decoder is a basic digital circuit component used in electronic systems. To choose one or more output lines, it decodes input signals that are binary or coded. The Decoder's design elements are:

- **Input Lines**: Input lines serve as the primary interface for providing control signals to the decoder. These lines carry encoded instructions or addresses that need to be decoded to activate specific outputs.
- **Decoder Logic**: The decoder logic interprets the input signals and generates corresponding outputs based on the provided instructions. It typically consists of a combinational logic circuit that maps each unique input combination to a specific output line.
- **Output Lines**: Output lines convey the decoded signals to the desired destination within the digital system. These lines are activated based on the input signals received by the decoder, facilitating the selection or activation of specific components or functionalities.
- **Enable Signals**: Enable signals to control the operation of the decoder, determining when it should be active or inactive. These signals may be used to enable or disable the decoding process based on certain conditions or external control signals.
- **Decoder Timing Circuitry**: Decoder timing circuitry ensures that the decoding process occurs synchronously with the rest of the system. It may include clocking mechanisms or timing control circuits to coordinate the timing of input signal sampling and output signal generation.

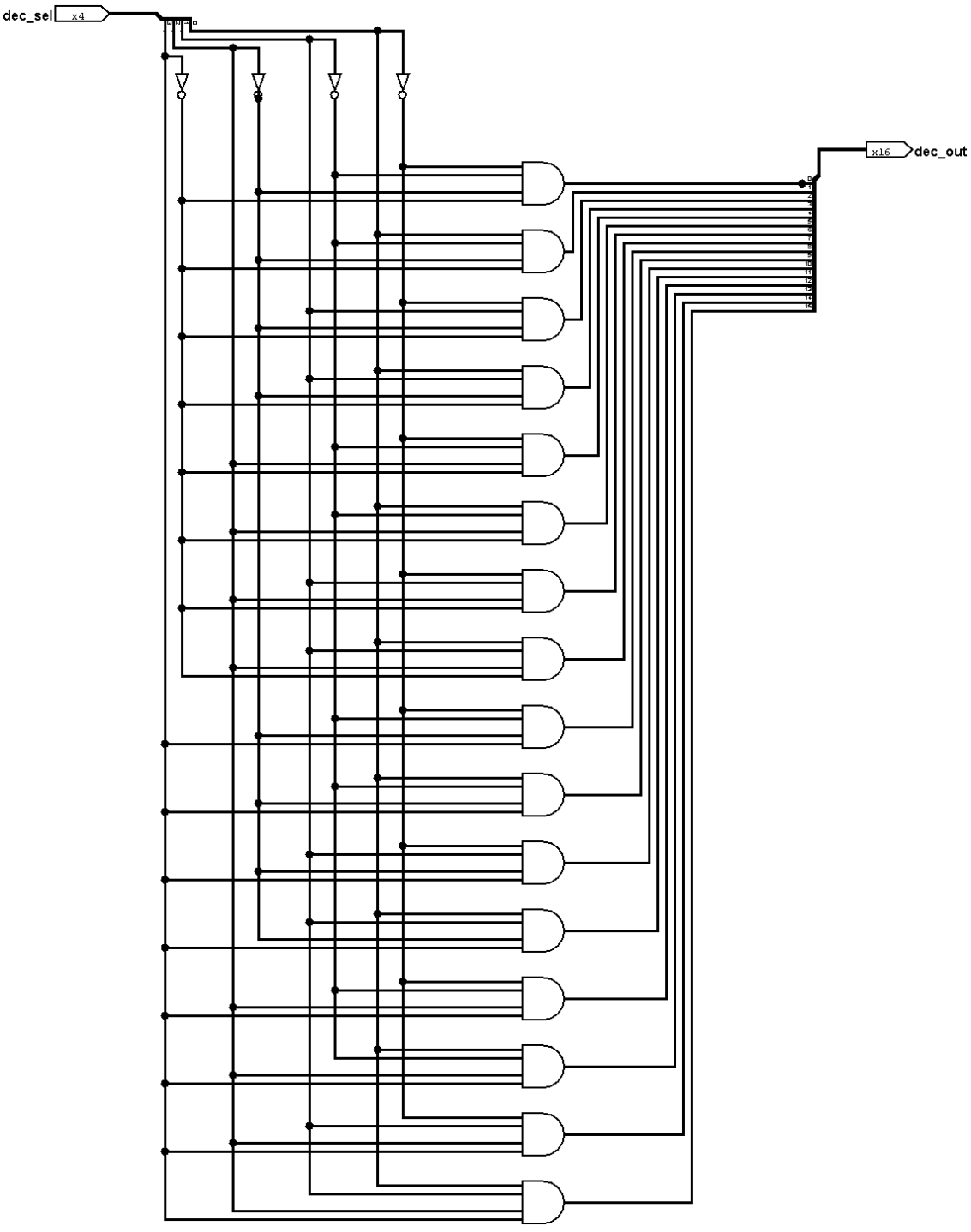


Figure 1: Schematic of 4 to 16 Decoder

Figure 1 shows the schematic diagram of the decoder which is designed. It takes binary or coded input signals and decodes them to select one or more output lines.

SRAM cell:

For fast data storage and retrieval in digital devices, Static Random Access Memory (SRAM) is a type of volatile memory. SRAM is faster and more dependable for some applications than Dynamic RAM (DRAM) because it doesn't require periodic refreshes. The design components of SRAM are:

- **Storage Elements**: The core of the SRAM cell consists of two cross-coupled inverters that form a bistable latch. This bistable configuration allows the cell to store a single bit of data.
- **Bit Lines**: Two complementary lines, known as the bit line (BL) and bit line complement (BL'), are used for data input and output. During a read or write operation, the bit lines carry the data to and from the SRAM cell.
- **Access Transistors**: As access transistors, two n-type MOSFETs are often utilized. These transistors connect the bit lines and storage nodes. When engaged, the word line connects the internal storage nodes to the external bit lines, controlling the access transistors and facilitating read and write operations.
- **Word Line**: a control line that joins the access transistors' gates. The word line activates the access transistors, which permits the writing and reading of data to and from the SRAM cell. During these operations, the word line makes sure that just the chosen SRAM cell is accessible.

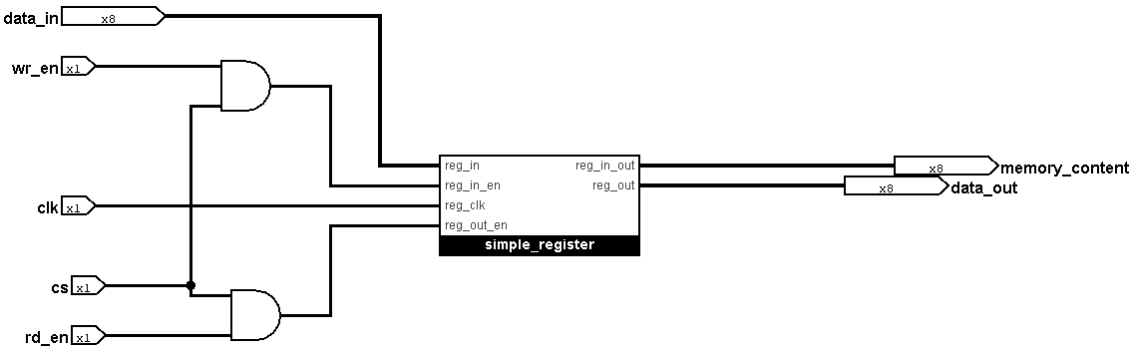


Figure 2: Schematic of a single SRAM cell

Figure 2 shows the schematic diagram of the SRAM which is designed. SRAM cell relies on a combination of access transistors, latch transistors, bit lines, word lines, and power connections to store and retrieve data efficiently.

Random Access Memory (RAM) :

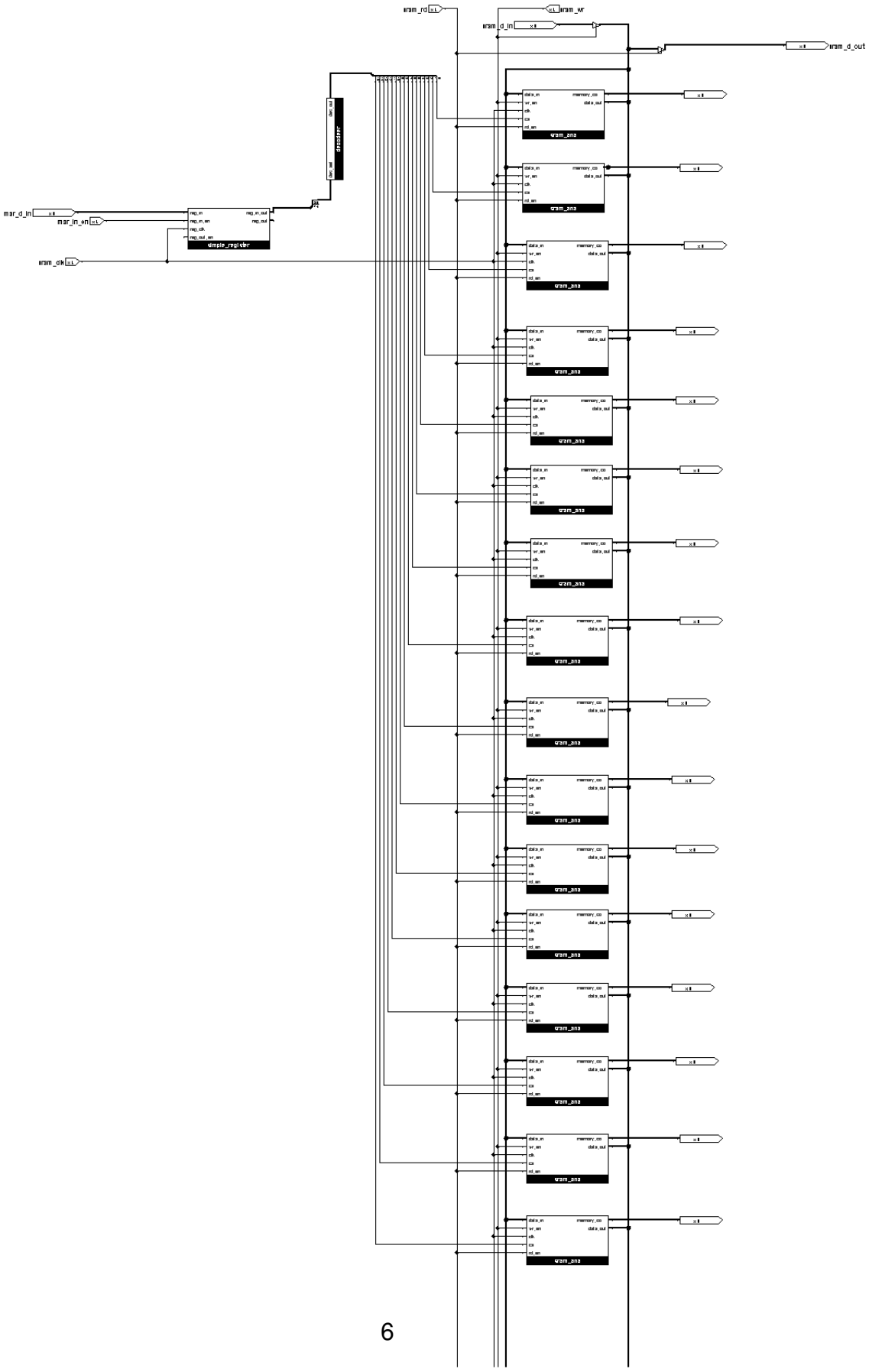
Computers and other electronic devices employ Random Access Memory (RAM), a form of volatile memory, to store data that is being used or processed by the CPU at the moment. It is distinguished by its capacity to read and write quickly, which is essential for the effective completion of tasks and programs. It can also access data in any order. RAM's design elements are as follows:

- **Memory Cells**: The basic unit of RAM where data is stored. There are different types of RAM cells, such as SRAM (Static RAM) and DRAM (Dynamic RAM), each with different designs. Memory cells store individual bits of data. In SRAM, each cell typically uses six transistors (6T) to store a bit, while in DRAM, each cell uses one transistor and one capacitor to store a bit.
- **Bit Lines**: Conductive lines that run vertically through the memory array and connect to the memory cells. Bit lines are used to read from and write data to the memory cells. In SRAM, there are typically two complementary bit lines (BL and BL'), while in DRAM, a single bit line is used.
- **Access Transistors**: Transistors used to connect the memory cell to the bit lines. In SRAM, access transistors (usually two per cell) control the connection between the storage nodes and the bit lines.

In DRAM, a single access transistor controls the connection between the capacitor and the bit line.

- **Decoders**: Address decoders that select the appropriate word lines and bit lines based on the input address. Decoders translate the binary address input into the selection signals that activate the correct word line (row decoder) and bit line (column decoder) to access the desired memory cell.
- **Control Logic**: Logic circuits that manage the timing and control signals for the read, and write. Control logic generates the necessary signals to coordinate the operations of the memory cells, bit lines, word lines, sense amplifiers, and precharge circuits.

Figure 3 shows Random Access Memory (RAM) is a type of volatile memory used in computers and electronic devices to store data that is actively being used or processed by the CPU


sram_rd
sram_wr
sram_d_in
sram_d_out
mar_d_in
mar_ln_en
sram_clk
simple_register
decoder
sram_ena

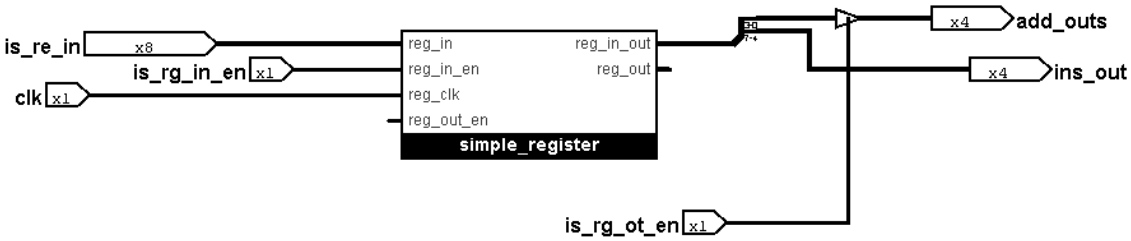


Figure 4: Schematic of Instruction Register

.

Instruction Register:

One essential part of a computer's central processing unit (CPU) is the instruction register (IR). The instruction that is now being executed or decoded is momentarily stored there. The IR ensures that instructions are handled in an efficient and organized way, which is crucial for CPU control and operation.

- **Temporary Storage**: The IR temporarily stores the current instruction fetched from the memory. This allows the CPU to decode and execute the instruction while the next instruction is being fetched.
- **Decoding Instructions**: The contents of the IR are sent to the instruction decoder, which interprets the instruction and generates the necessary control signals to carry out the specified operation.
- **Control Unit Integration**: The IR works closely with the control unit of the CPU. The control unit uses the information in the IR to determine the sequence of micro-operations needed to execute the instruction.
- **Synchronization**: The IR helps synchronize the stages of instruction execution, ensuring that the CPU executes each instruction in the correct sequence and timing.

Figure 4 shows The Instruction Register is vital for the proper functioning of the CPU, acting as a temporary holding area for the instruction currently being executed.

Microprosseor :

A microprocessor is an integrated circuit (IC) chip that houses the central processing unit (CPU) of a computer. It functions as the computer's brain, processing data and carrying out commands. All contemporary computing devices, including smartphones, embedded systems in appliances, and cars, depend on microprocessors to function. A microprocessor's main design elements are as follows:

- **Arithmetic Logic Unit (ALU)**: The ALU is the core component responsible for performing arithmetic and logical operations.Executes mathematical calculations like addition, subtraction, multiplication, and division, as well as logical operations such as AND, OR, XOR, and NOT.
- **Control Unit (CU)**: The CU directs the operation of the processor, coordinating the activities of the ALU, registers, and other components. Fetches instructions from memory decodes them, and generates control signals to execute them. Manages the sequence of instruction execution and orchestrates data flow within the CPU.
- **Registers**: Small, fast storage locations within the CPU. Temporarily hold data, instructions, and addresses that the CPU needs to access quickly. Key registers include the program counter (PC), instruction register (IR), accumulator (ACC), and general-purpose registers.
- **Bus Interface**: The communication system that transfers data between different components of the computer.Includes address bus, data bus, and control bus. Facilitates data transfer between the CPU, memory, and I/O devices
- **Instruction Decoder**: A circuit that interprets the instructions fetched by the control unit.Decodes machine language instructions into control signals that drive the execution of operations by the ALU and other components.
- **Clock Generator**: Generates the timing signals that synchronize the operations of the microprocessor.Provides a constant clock signal to ensure that all parts of the CPU operate in a coordinated manner, maintaining timing and synchronization.
- **Program Counter (PC)**: A register that holds the address of the next instruction to be executed. Increments after each instruction fetch to point to the subsequent instruction, ensuring sequential execution of the program.

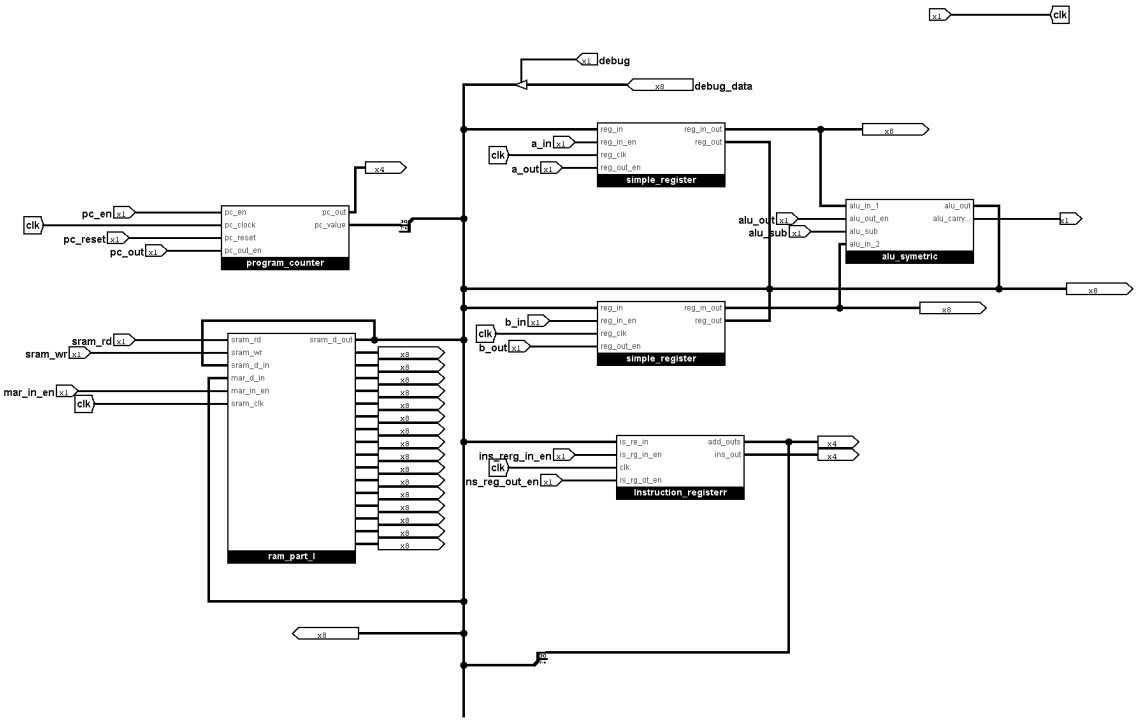


Figure 5: Schematic of a microprocessor

- **Instruction Register (IR)**: Holds the instruction currently being executed. Temporarily stores the fetched instruction before it is decoded and executed.
- **Load/Store Unit**: Manages the transfer of data between the CPU registers and the main memory. Executes load (read) and store (write) operations to move data to and from the memory hierarchy.

Fetch Decode Execute:

Each computer's CPU can have different cycles based on different instruction sets, but will be similar to the following cycle:

- **Fetch Stage:** The next instruction is fetched from the memory address that is currently stored in the program counter and stored into the instruction register. At the end of the fetch operation, the PC points to the next instruction that will be read at the next cycle.
- **Decode Stage:** During this stage, the encoded instruction presented in the instruction register is interpreted by the decoder.Read the effective address: In the case of a memory instruction (direct or indirect), the execution phase will be during the next clock pulse. If the instruction has an indirect address, the effective address is read from main memory, and any required data is fetched from main memory to be processed and then placed into data registers (clock pulse: T3). If the instruction is direct, nothing is done during this clock pulse. If this is an I/O instruction or a register instruction, the operation is performed during the clock pulse.
- **Execute Stage:** The control unit of the CPU passes the decoded information as a sequence of control signals to the relevant functional units of the CPU to perform the actions required by the instruction, such as reading values from registers, passing them to the ALU to perform mathematical or logic functions on them, and writing the result back to a register. If the ALU is involved, it sends a condition signal back to the CU. The result generated by the operation is stored in the main memory or sent to an output device. Based on the feedback from the ALU, the PC may be updated to a different address from which the next instruction will be fetched.

Behaviour Analysis of a microprocessor :

Fect Decode Execute analysis: After the design is ready, Firstly programmed the RAM with instructions code. The instruction: 0001 1010 as LDA 10 or load the register A with memory contents of location 10. This instruction has two parts. The Opcode: 0001 meaning load A and the operand 1010 meaning 10.

Program the Ram
STEP:

- • Turn on debug pin
- • Pulse the pcreset pin.

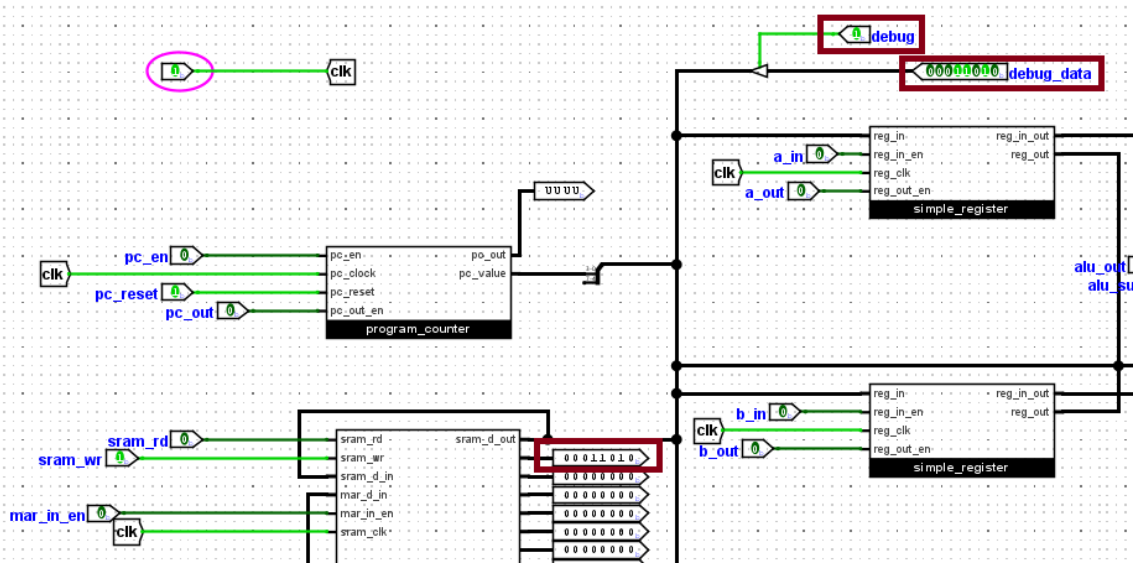


Figure 6: Load data in RAM - 1

- • Set debugdata to 0000 0000.
- • Toggle marinen and give a clock pulse.
- • Turn off marinen.
- • Set debugdata to 0001 1010.
- • Toggle sramwr and give a clock pulse.
- • Turn off sramwr and observe the data saved in address 0000. similarly, different data can be saved in a particular address

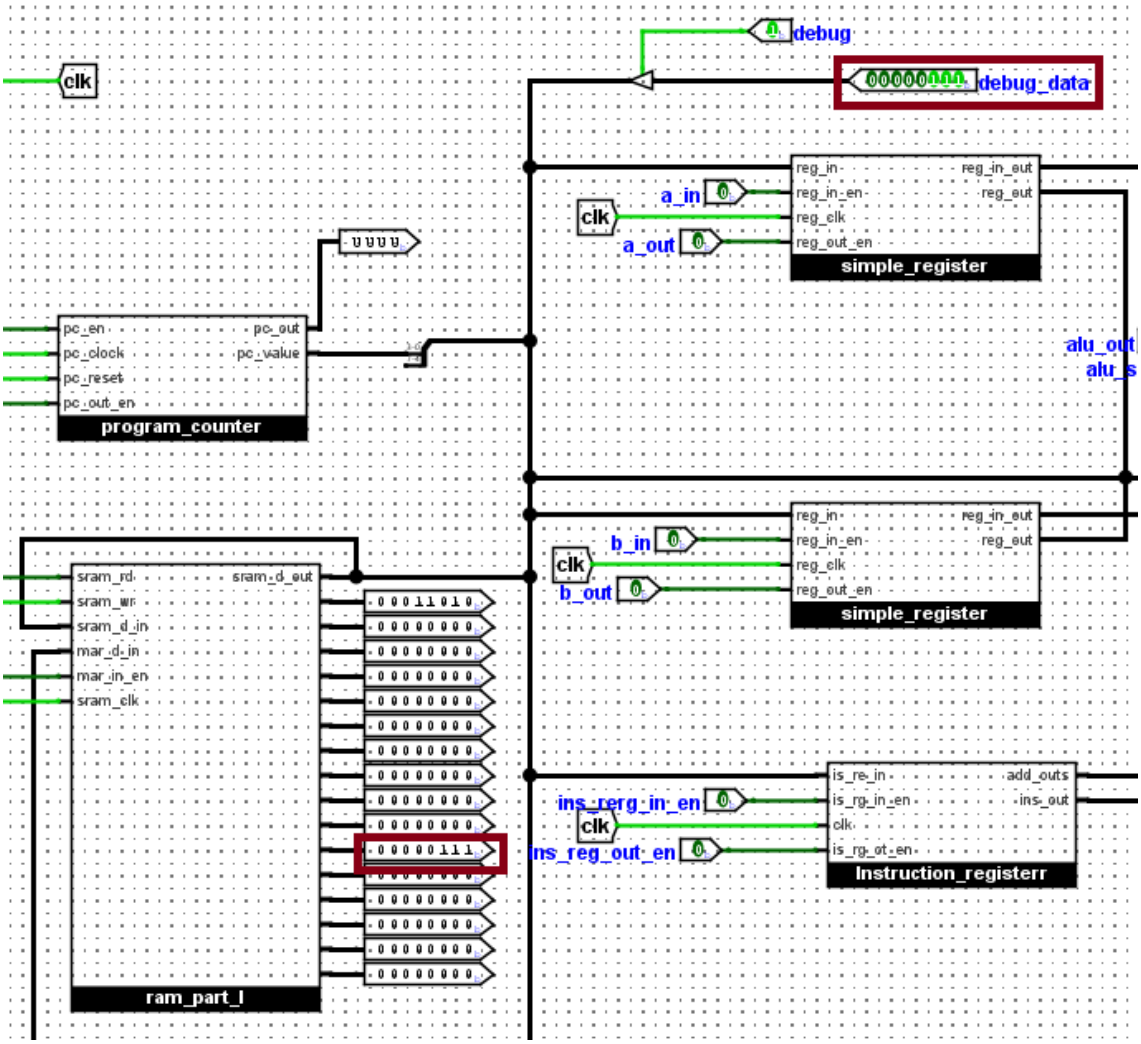


Figure 7: Load data in RAM - 2

Fetch Programing step:

Fetch stage has 3 T-states, all followed by a clock pulse. Before proceeding, reset program counter to 0000.

- T1: Toggle pc out and mar in en and give a clock pulse. Address of next instruction to be executed is sent to MAR from PC. Turn off the control pins.
- T2: Toggle sram rd and ins reg in en and give a clock pulse. Check the IR loaded with the Opcode and operand. Turn off the control pins.
- T3: Toggle pc en and give a clock pulse. This increments the counter value to be 0001 or the location of the next line of code. Turn off the control pin.

Execution Programming Step:

As the loaded instruction is load A or LDA, it also has 3 T-states as follows:

- T1: Toggle ins reg out en and mar in en and give a clock pulse. This saves the address to be fetched into the MAR. Turn off the control pins.
- T2: Toggle SRAM rd and a in and give a clock pulse. This reads the memory contents of address 1010 which is 0111 and saves it in register A. Turn off the control pins.
- T3: This state is unused for LDA

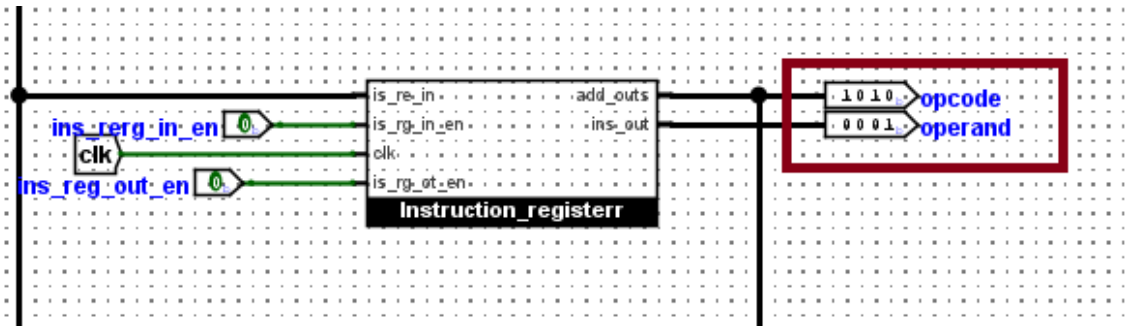


Figure 8: After fetched IR loaded with the Opcode and operand.

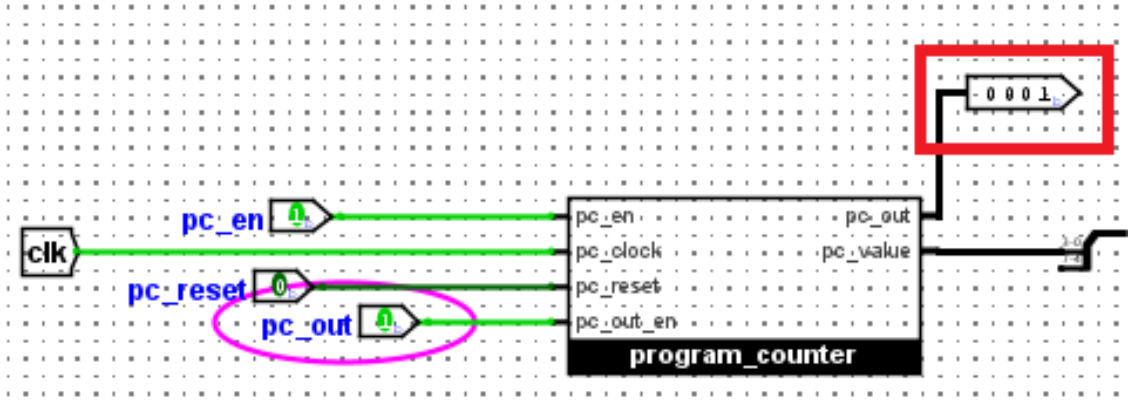


Figure 9: counter is updated after fetching

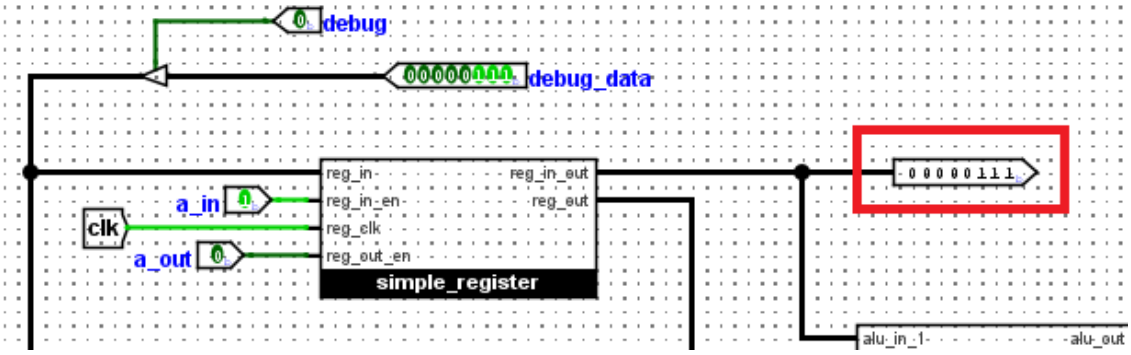


Figure 10: Register A is updated

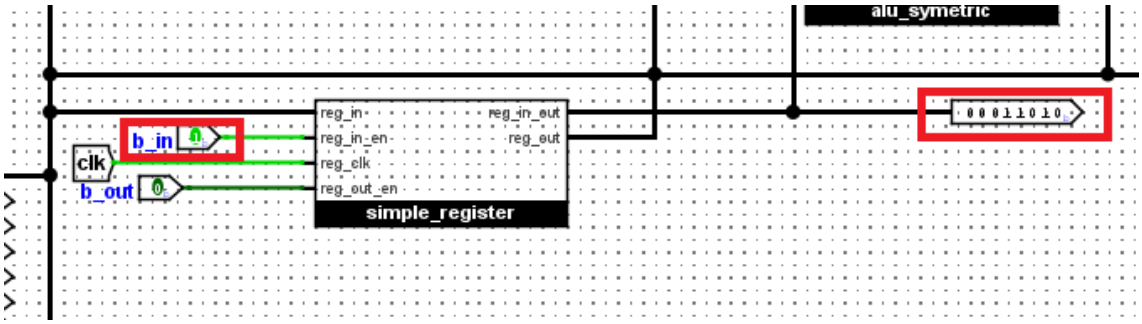


Figure 11: Register B is updated

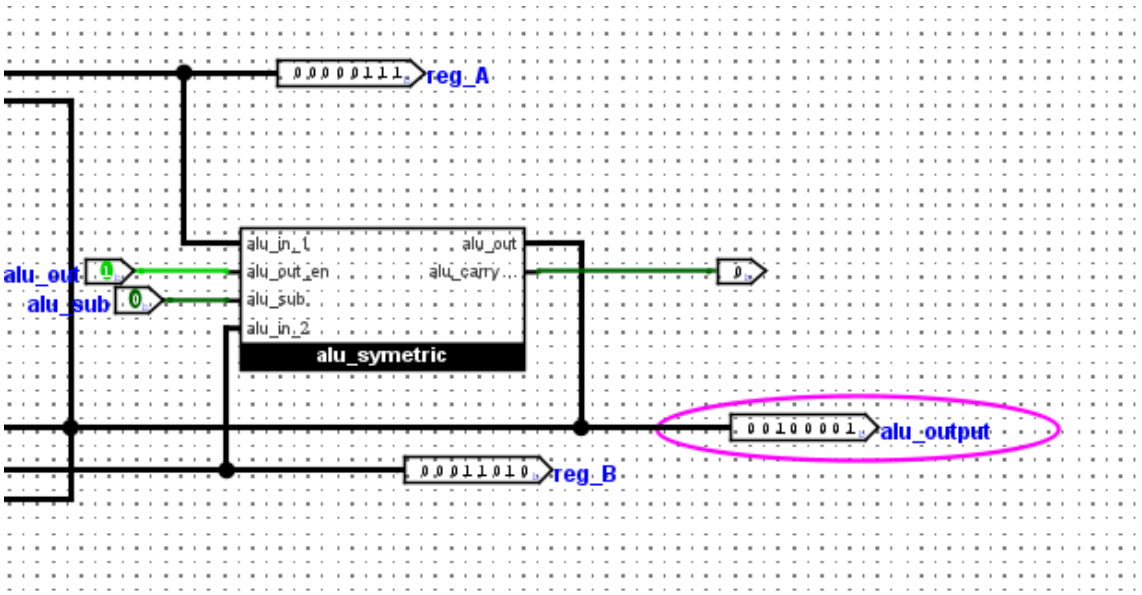


Figure 12: ALU output of summing Reg A and Reg B

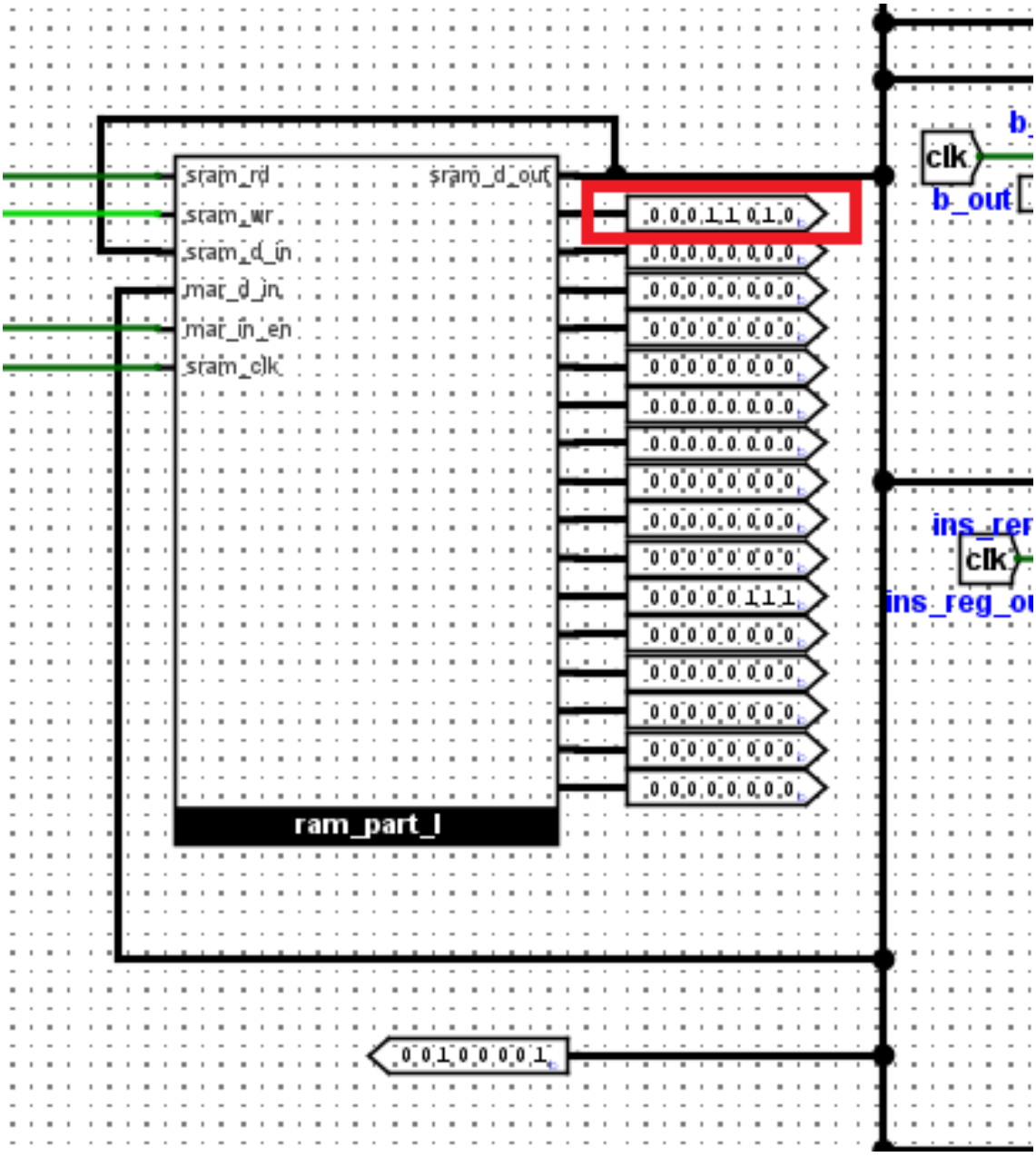


Figure 13: Output of ALU is stored in RAM