Chittagong University of Engineering and Technology

Department of Electronics and Telecommunication Engineering

ETE 404

VLSI Technology Sessional

Instruction Set of Modified SAP Architecture with Control Sequencer

Submitted by:

Shamanza Chowdhury

ID: 1908008

Submitted to:

Arif Istiaque Rupom

Lecturer

Dept. of ETE,CUET

September 16,2024

1 Instruction Set:

Table 1: Modified SAP Instruction Set with Opcode

Operation	Description	Opcode
LDA	Load data from memory to the accumulator	0001
LDB	Load data from memory to the register B	0010
ADD	Add data from memory to the value in the accumulator	0100
SUB	Subtract data from memory from the value in the accumulator	0101
OUT	Load data from accumulator to the output register	0110
STR	Store data from output register to the memory	0111
Halt	Stop processing	1111

2 Control Sequencer :

The controller-sequencer sends out signals that control the computer and makes sure things happen only when they are supposed to.It is made of Ring counter,Instruction Decoder and AND and OR gates as a control unit. The 15 bit output signals from controller-sequencer is called the control word with 6 T-states which determines how the registers will react to the next positive clock edge. It has the following format: $Cp \sim Ep \sim Lm \sim CE \sim Li \sim Ei \sim La \sim Lb \sim Ea \sim Eu \sim Su \sim Eu \sim RI \sim Lb \sim Lo \sim Eo$

- **Cp (pc_en)**:Controls whether the counting is enabled or not.
- **Ep (pc_out):**Controls the output of the program counter.
- **Lm (mar_in_en):** Enables loading of the Memory Address Register (MAR) with the address from the program counter.
- **CE (sram_rd)**:controls the read operation from the SRAM.
- **Li (ins_reg_in_en):** Enables the instruction register to load the instruction from the bus.
- **Ei (ins_reg_out_en):** Enables the output of the instruction register onto the bus.
- **La (a_in):** Enables data to load into the accumulator from the bus.
- **Lb (b_in):** Enables data into the B-Register from the bus.
- **Ea (a_out):** Sends the content of the accumulator onto the bus.
- **Eu (alu_out):** Enables the output of the adder-subtractor to the bus.
- **Su (alu_sub):** Controls whether the ALU performs addition (when low) or subtraction (when high).
- **RI (sram_wr):**Control the write operation to the SRAM.
- **Lo (o_in):**Load Output Register.
- **Eo (o_out):** Enables the output of the output register to the bus.

2.1 Diagram:

The circuit appearance of Control Sequencer is given below:

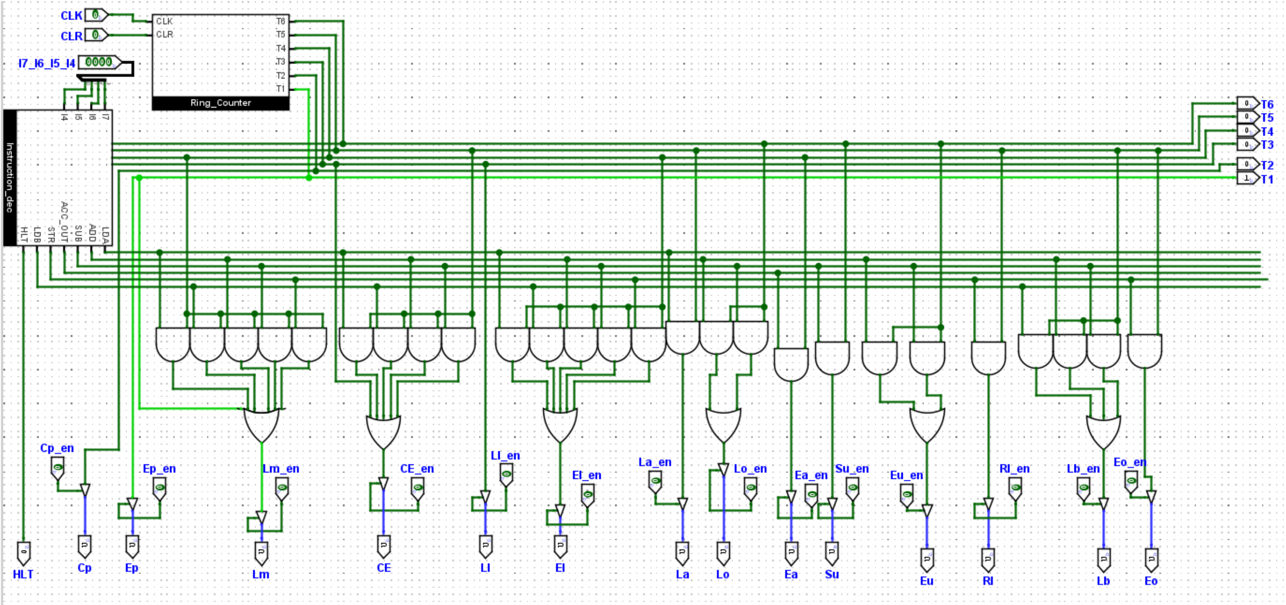

Figure 1: Circuit Appearance of Control Sequencer

3 Final Layout of SAP Architecture:

At this stage, we had assembled all parts of the processor as shown in figure 1. Various parts of the processor had been connected through the central BUS. The control pins of all the subcircuits had been tunneled to a unified location for ease of manipulation. A debug control and debug data pin had also been added to the bus to program the RAM. At this stage, Control Unit had been implemented to automate this manual control switching. This control unit controlled the control BUS to manipulate the various control pins to ensure proper operation.

3.1 Diagram:

The schematic of the SAP Architecture can be seen in the following diagram:

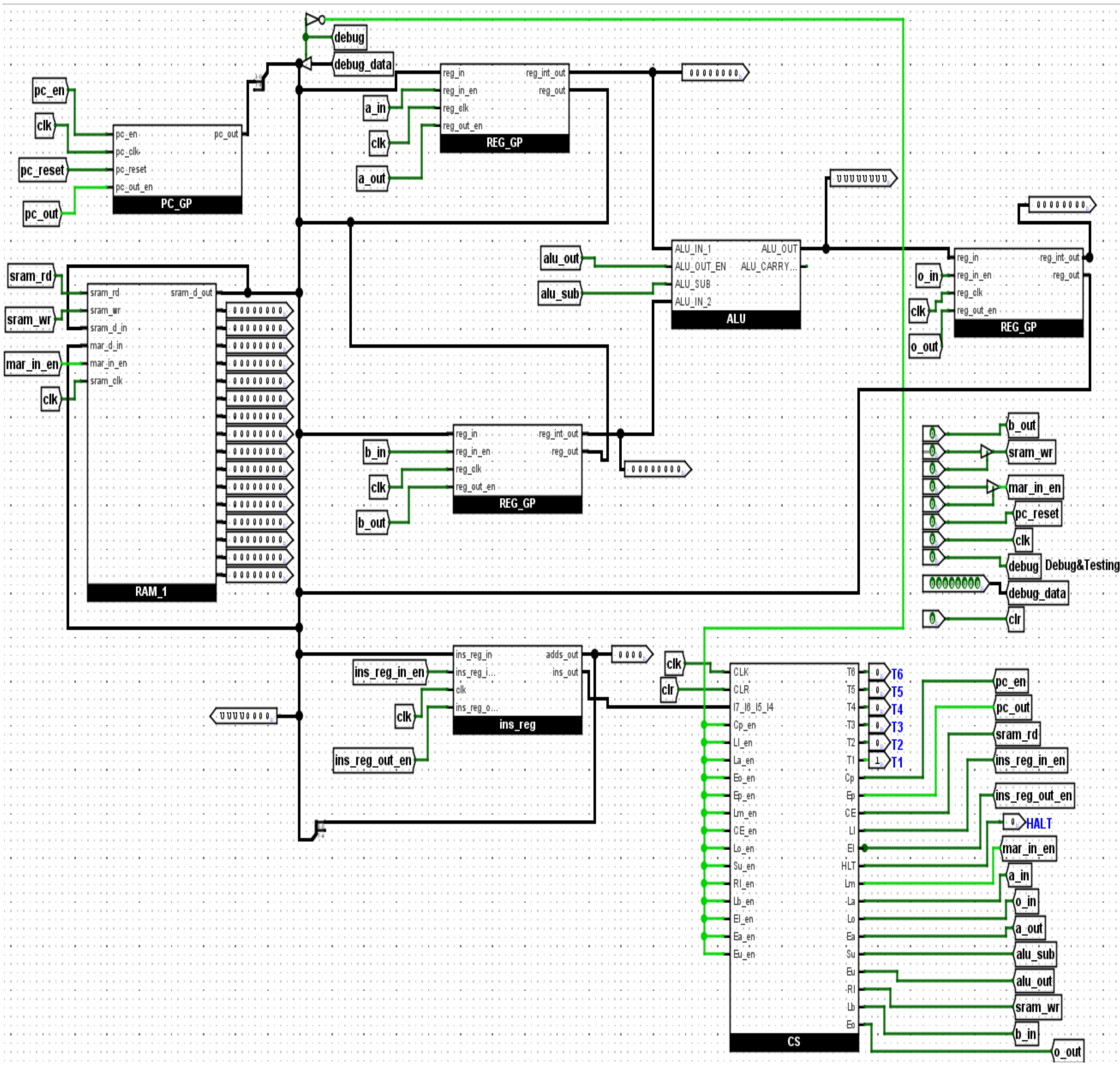

Figure 2: Final Schematic layout of SAP Architecture.

3.2 Behaviour Analysis:

3.2.1 Load Data into Register A: LDA14

To load register A with the memory contents of location 14, use the instruction with Opcode 0001 (load A) and operand 1110 (location 10). This instruction is stored in the first RAM location, and the value to be moved to register A is placed in memory location 14.

Programming Steps:

- The debug pin had been turned on.
- The pc_reset pin had been pulsed.
- The debug_data had been set to 0000 0000.
- The mar_in_en had been toggled, and a clock pulse had been given.
- The mar_in_en had been turned off.
- The debug_data had been set to 0001 1110.
- The sram_wr had been toggled, and a clock pulse had been given.
- The sram_wr had been turned off, and the data saved in address 0000 had been observed.
- The debug_data had been set to 0000 1110.
- The mar_in_en had been toggled, and a clock pulse had been given.
- The mar_in_en had been turned off.
- The debug_data had been set to 0000 0111.
- The sram_wr had been toggled, and a clock pulse had been given.
- The sram_wr had been turned off, and the data saved in address 1110 had been observed.
- The debug pin had been turned off.

Processing Steps of Control Sequencer:

Before proceeding, the program counter was reset to 0000. During this stage, first three T-states were used for fetch cycle and later three T-states used for execute cycle, each followed by a clock pulse with clear=1.

- **T1: pc_out –> mar_in_en** , allowing the address of the next instruction to be executed to be sent from the PC to the MAR.
- **T2: pc_en–>pc_en + 1** , resulting in the incrementation of the counter's value to 0001.
- **T3: sram_rd –>ins_reg_in_en** , ensuring that the instruction register (IR) was loaded with the opcode and operand.

- **T4: ins_reg_out_en–>mar_in_en**,the address to be fetched into the MAR was saved.

- **T5: sram_rd –>a_in**, The memory contents stored at address 1110, which were 0111, were read and saved in register A.

- **T6:**T6 had remained unused for the LDA operation.

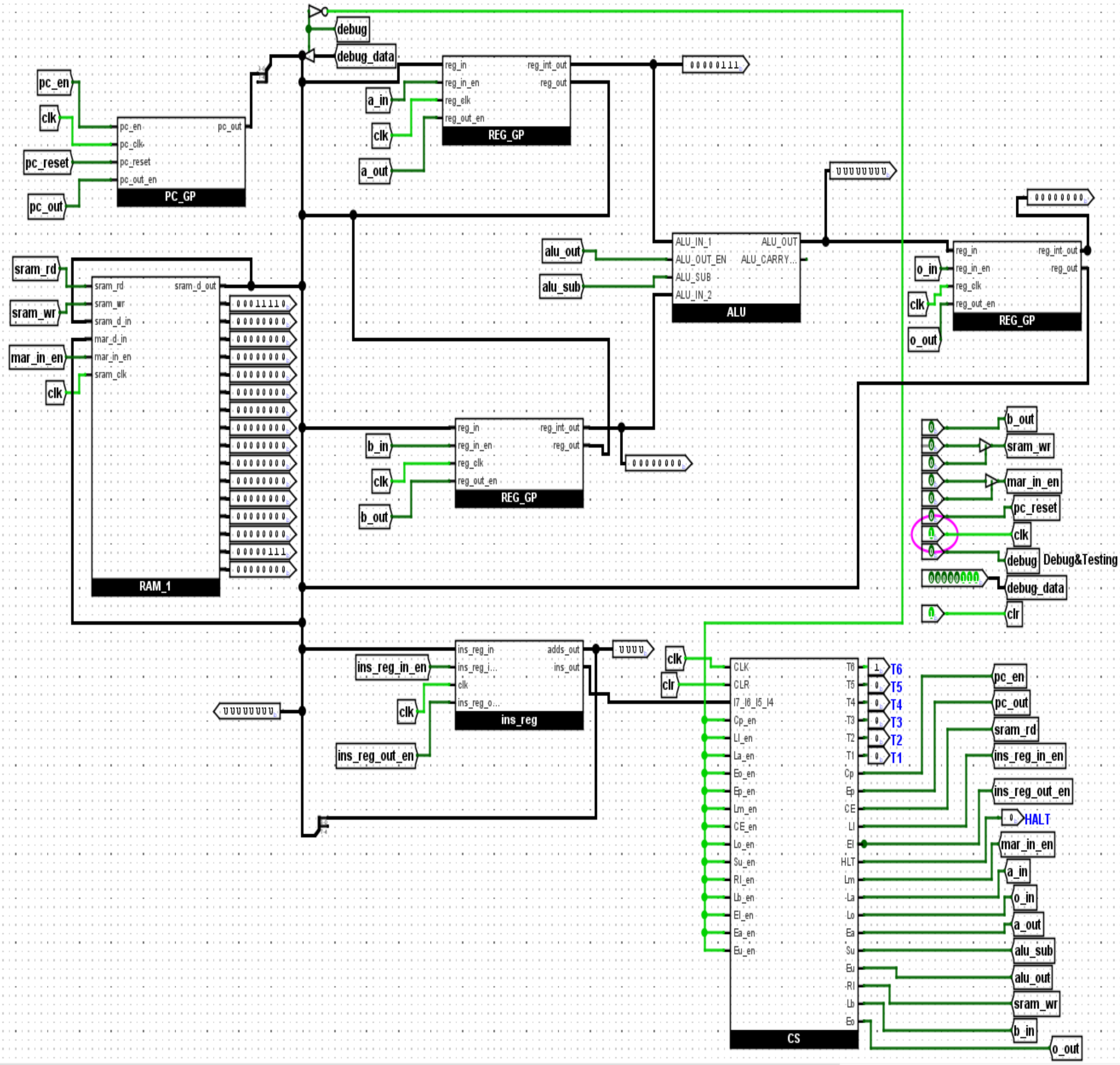

Figure 3: Testing the SAP Architecture during Load Data into Register A.

3.2.2 Add data from Memory to the Value in the Accumulator: LDA15

To add the content of memory location 15 to the accumulator, use the instruction with Opcode 0100 (add) and operand 1111 (location 15). This instruction is stored in the second RAM location, and the value to be added in the in the register B is placed in memory location 15.

Programming Steps:

- The debug pin had been turned on.
- The pc_reset pin had been pulsed.
- The debug_data had been set to 0000 0001.
- The mar_in_en had been toggled, and a clock pulse had been given.
- The mar_in_en had been turned off.
- The debug_data had been set to 0100 1111.
- The sram_wr had been toggled, and a clock pulse had been given.
- The sram_wr had been turned off, and the data saved in address 0001 had been observed.
- The debug_data had been set to 0000 1111.
- The mar_in_en had been toggled, and a clock pulse had been given.
- The mar_in_en had been turned off.
- The debug_data had been set to 0000 0001.
- The sram_wr had been toggled, and a clock pulse had been given.
- The sram_wr had been turned off, and the data saved in address 1111 had been observed.
- The debug pin had been turned off.

Processing Steps of Control Sequencer:

Before proceeding, the program counter was reset to 0000. During this stage, first three T-states were used for fetch cycle and later three T-states used for execute cycle, each followed by a clock pulse with clear=1.

- **T1: pc_out –> mar_in_en** , allowing the address of the next instruction to be executed to be sent from the PC to the MAR.
- **T2: pc_en–>pc_en + 1** , resulting in the incrementation of the counter's value to 0010.
- **T3: sram_rd –>ins_reg_in_en** , ensuring that the instruction register (IR) was loaded with the opcode and operand.
- **T4: ins_reg_out_en–>mar_in_en**,the address to be fetched into the MAR was saved.

- **T5: sram_rd –>b_in**, the memory contents stored at address 1111, which were 0001, were read and saved in register B.
- **T6:alu_out –>o_in**,the output of the ALU saved into output register.

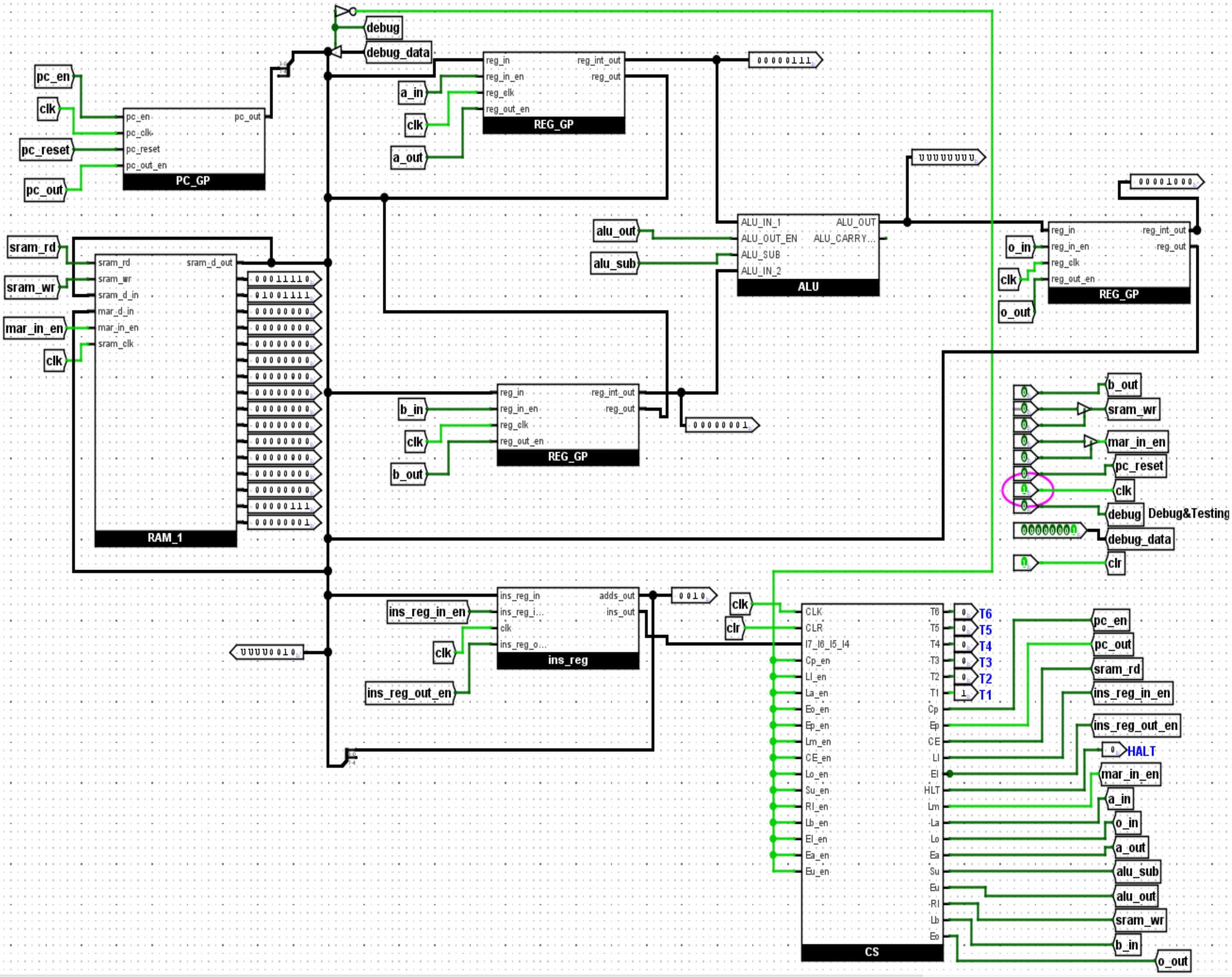

Figure 4: Testing the SAP Architecture during Add Data and‘save it to output register.

3.2.3 Store data from output register to the memory: STR13

To store the content of output register to memory location 13, use the instruction with Opcode 0111 (store) and operand 1101 (location 13). This instruction is stored in the third RAM location, and the value would be stored in the memory location 13.

Programming Steps:

- The debug pin had been turned on.
- The pc_reset pin had been pulsed.
- The debug_data had been set to 0000 0010.
- The mar_in_en had been toggled, and a clock pulse had been given.
- The mar_in_en had been turned off.
- The debug_data had been set to 0111 1101.
- The sram_wr had been toggled, and a clock pulse had been given.
- The sram_wr had been turned off, and the data saved in address 0010 had been observed.
- The debug pin had been turned off.

Processing Steps of Control Sequencer:

Before proceeding, the program counter was reset to 0000. During this stage, first three T-states were used for fetch cycle and later three T-states used for execute cycle, each followed by a clock pulse with clear=1.

- **T1: pc_out –> mar_in_en** , allowing the address of the next instruction to be executed to be sent from the PC to the MAR.
- **T2: pc_en–>pc_en + 1** , resulting in the incrementation of the counter's value to 0011.
- **T3: sram_rd –>ins_reg_in_en** , ensuring that the instruction register (IR) was loaded with the opcode and operand.
- **T4: ins_reg_out_en–>mar_in_en**,the address to be fetched into the MAR was saved.
- **T5: o_out –>sram_wr**, The contents of the output register stored at memory address 1101, which were 1000, the sum of the value saved in the both register .
- **T6:**T6 had remained unused for the STR operation.

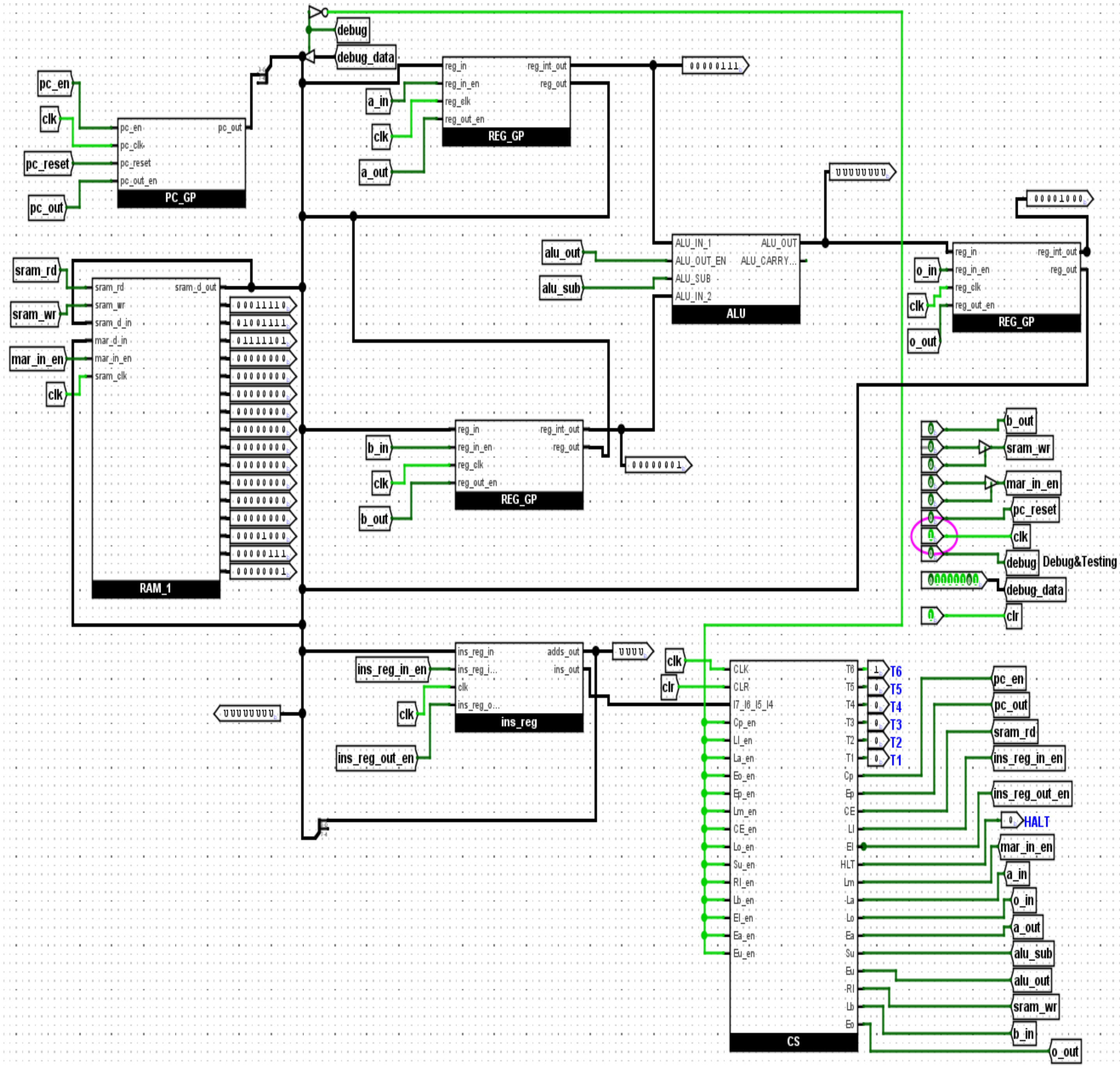

Figure 5: Testing the SAP Architecture during Store ALU result into RAM.

References

[1] wikipedia, ”Instruction Cycle,[Online].

Available: https://en.wikipedia.org/wiki/Instruction_cycle.

[2] FutureLearn, ”Fetch-Decode-Execute Cycle,” [Online].

Available: https://www.futurelearn.com/info/courses/how-computers-work/0/steps/49284.

[3] B.Eater,”Building an 8-bit breadboard computer!,” [Online].

Available: https://www.youtube.com/playlistlist=PLowKtXNTBypGqImE405J2565dvjafglHU