ISLAMIC UNIVERSITY OF TECHNOLOGY (IUT)

ORGANIZATION OF ISLAMIC COOPERATION (OIC)

Course Code: EEE 4308

Course Title: Digital Electronics I Lab

Project: SAP-1 Architecture-based 8-bit Computer Design and Implementation.

Group Name: LOGIC BOTS

Section: C

Members:

MAIMUNA BISWAS NOSHIN **200021347**

SAFRINA KABIR **200021341**

MOHAMMAD TASEEN ISRAQUE CHOWDHURY **200021361**

M. SADMAN ASTER **200021344**

FAEDUN ISLAM AUMIO **200021348**

PHASE A

1. Clock Pulse Generator
2.Program Counter
3.Input Unit and MAR

PHASE B

1.Registers (Accumulator, B register, Output register)
2.RAM

PHASE C

1.Instruction Register
2.Controller Sequence

PHASE D

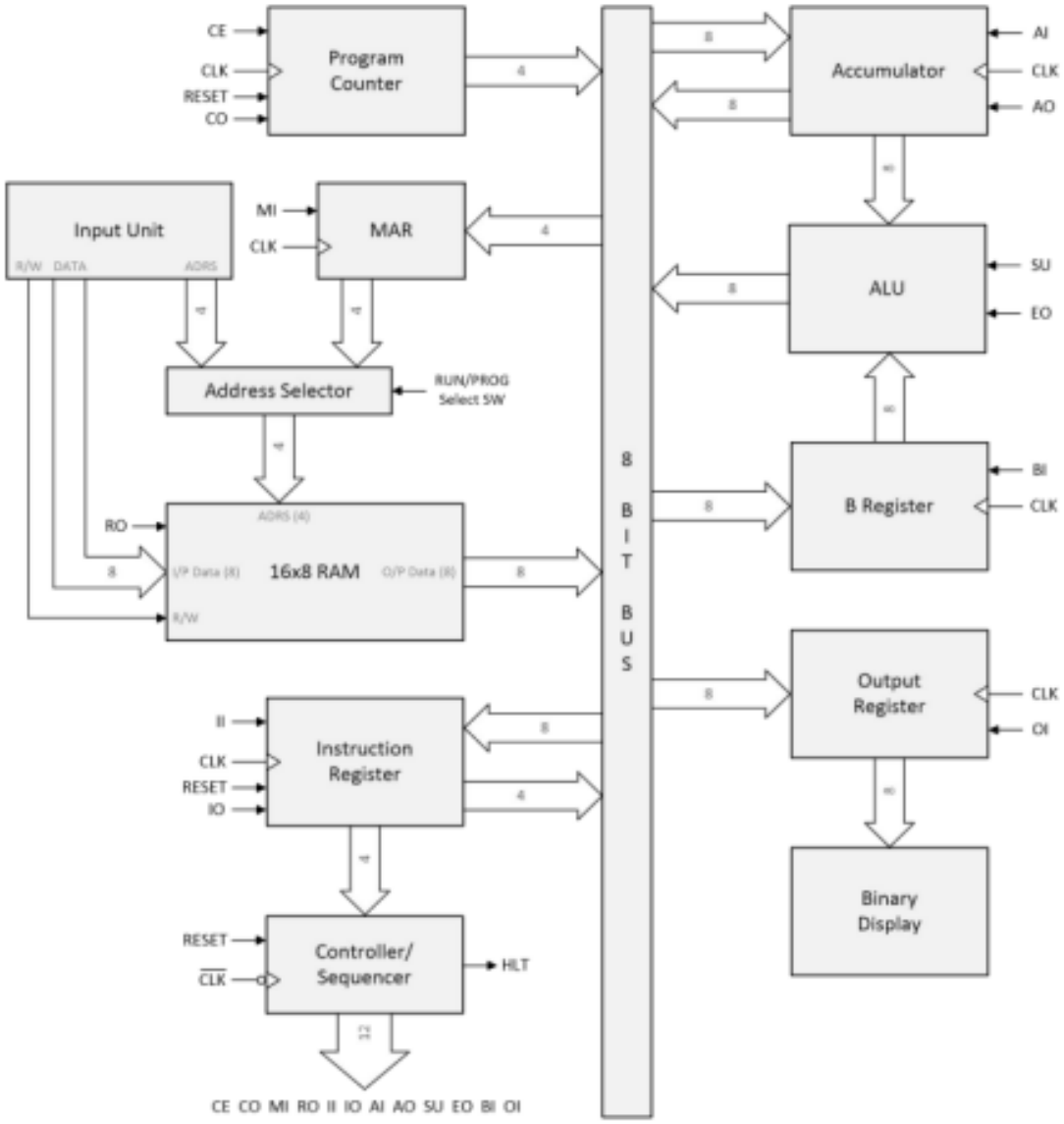
1. Arithmetic Logic Unit
1. 8 bit bus integration
2.Output Unit

Phase E : Complete Integration

Objective:

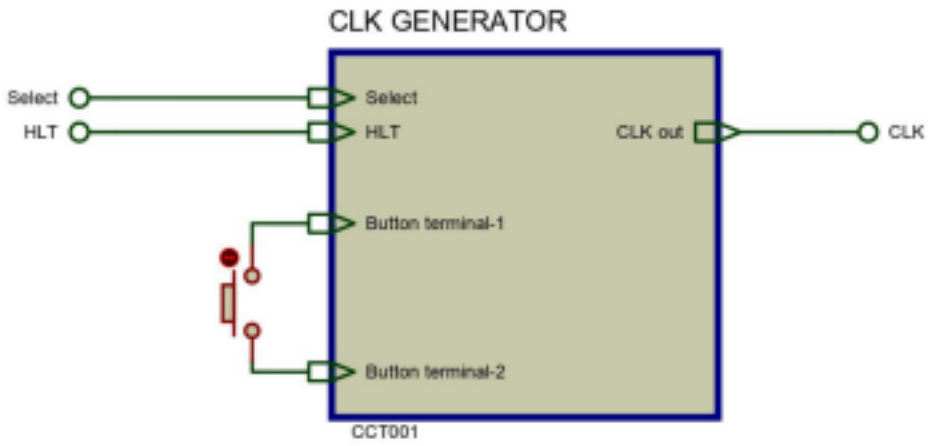
The objective of this project was to implement an 8-bit computer based on the architecture of SAP-1 (Simple-As-Possible) using combinational and sequential circuits.

Project Diagram:



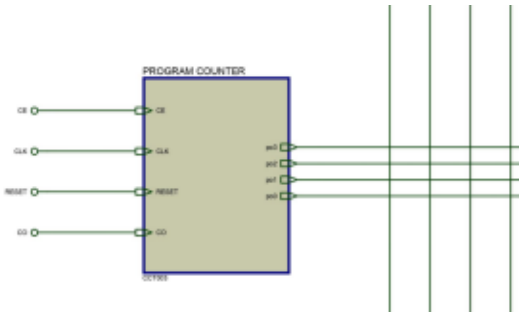
Clock Pulse Generator:

The clock generator was required to have two modes of operation.
(i) Automatic Mode: In this mode, the generated signal would be continuous identical to the output of an astable multivibrator.
(ii) Manual Mode: The generated clock pulse in this mode will have a low output unless the operator manually instructs the clock generator to generate a single pulse, a mode of operation that is similar to a monostable multivibrator.



Program Counter:

Program counter (PC) is a 4-bit counter which can count from 0000 up to 1111. The Program counter's job is to store and send out the memory address of the next instruction to be fetched and executed. It will increment its value after the completion of each instruction.

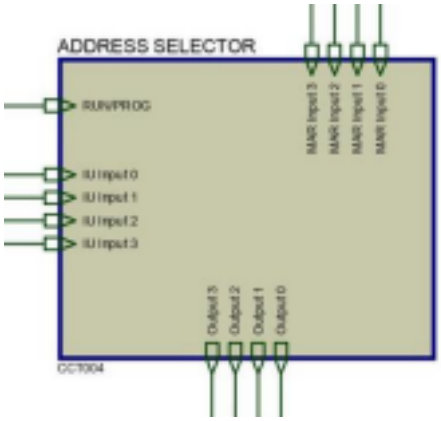


Input Unit:

The Input Unit of the computer is constructed so that the user can manually write the data and program into the addresses of the Random Access Memory (RAM). The data and program are then stored in the RAM and the computer can avail these when it is running. The user also has the option to read the stored data and even change the written data in the RAM. All of these features can be availed using the Input Unit.

Address Selector:

Address Selector has the ability to select the address from the Input Unit or the Memory Address Register (MAR). The RUN/PROG switch is connected to the Address Selector. The following subcircuit of the address selector was drawn in Proteus:



Memory Address Register (MAR):

Memory address register (MAR) is a 4-bit parallel in parallel out register. MAR stores the 4-bit address of data or instruction which are placed in RAM. When the computer is in the RUN state, the 4-bit address is obtained via the bus from the Program Counter and then stored. This stored address is sent to the RAM where data or instructions are read from.

MI In (Stores the current values of the bus into the Memory Address Register (MAR))

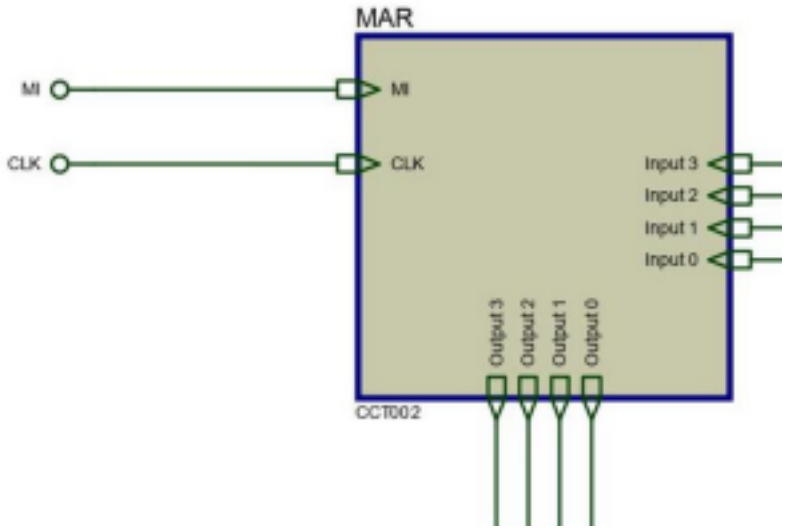
CLK In (Clock pulse input.)

OUT

(Output 0, Output 1, Output 2, Output 3)

Out

(4-bit address data.)



Registers:

All of these registers are of 8-bit size. Accumulator (also called as A register) has the ability to output intermediate results after each ALU operation. Accumulator and B register are used to store 8-bit data as input to ALU input. The purpose of the output register is to store result to display when requested. The Controller-Sequencer unit sends control signals to these registers for controlling input/output enable.

Accumulator In

Stores the current values of the bus into Accumulator.

Accumulator Out

Accumulator sends its stored content to the bus. Output of accumulator is disconnected when AO is low.

B Register In

Stores the current values of the bus into B Register.

Output Register In

Stores the current values of the bus into Output Register.

Random Access Memory (RAM):

The RAM for this computer is 16 x 8 bit (16 memory locations each storing 8 bits of data). The RAM can be programmed by means of the address switches and data switches of input unit. During a computer run, the RAM will receive its 4-bit address from the MAR.
In order to construct the Random Access Memory (RAM), two 74LS89 64-bit RAM ICs were used.

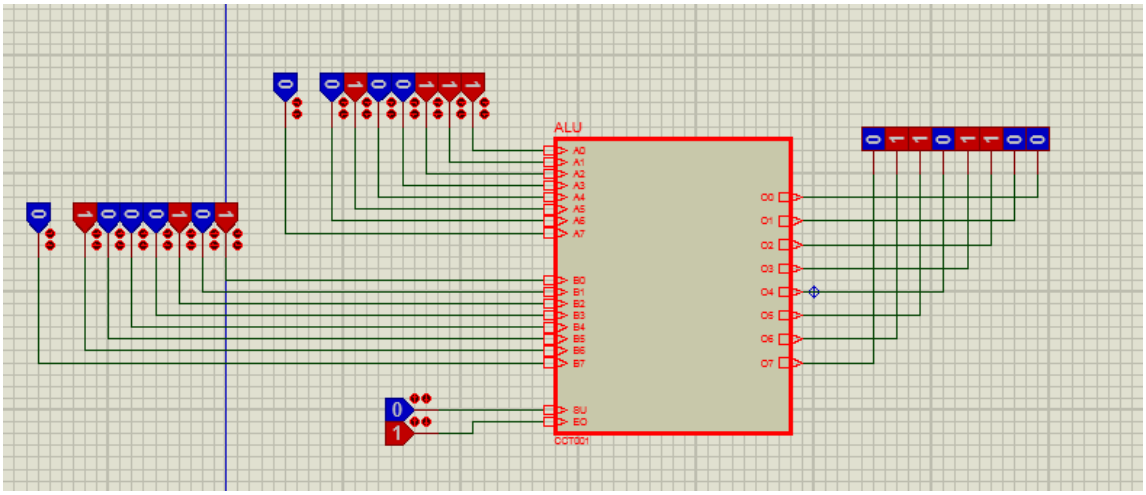
Operation of the RAM:

The RAM that was required for this project was a 16x8 bit RAM. A memory circuit of such capacity was built using individual RAM cells, decoders and logic gates. Although, the circuit design had no flaws, Proteus was unfortunately unable to simulate the functions of that circuit and generate an input. That is why another RAM was built using two 74LS89 64-bit RAM ICs.
The catch with this IC is the fact that it is a 64-bit RAM. This means there are 16 address location for 4- bit data inputs. But the RAM for this computer is required to have 8-bit data inputs. So, two solve this problem 2 ICs were used and their address inputs were shorted. This would increase the data inputs to 8 bits and the both of the ICs would have the same address inputs from the address pins. The address pins of the RAM were connected to the address inputs of both of the ICs in the same sequence. Meaning the LSB of the address pins of the RAM was connected to the LSB of the address inputs of both of the ICs. Whenever an address data is sent through the address inputs, the same address is selected in both of the ICs.
The data input pins are connected in such a way that the lower nibble of the 8-bit data coming from the bus is saved in the right IC and the higher nibble of that 8-bit data is saved in the left IC. The data input pins of the RAM are directly connected to the data pins of the IC. LSB of the lower nibble input is connected to LSB of the right IC and MSB of the lower nibble input is connected to the MSB of the right IC. Similarly,

LSB of the higher nibble input is connected to the LSB of the left IC and MSB of the higher nibble input is connected to the MSB of the left IC.

Arithmetic and Logic Unit (ALU):

ALU has the capability of 8-bit addition and subtraction. For addition and subtraction, 2's complement approach should be used.

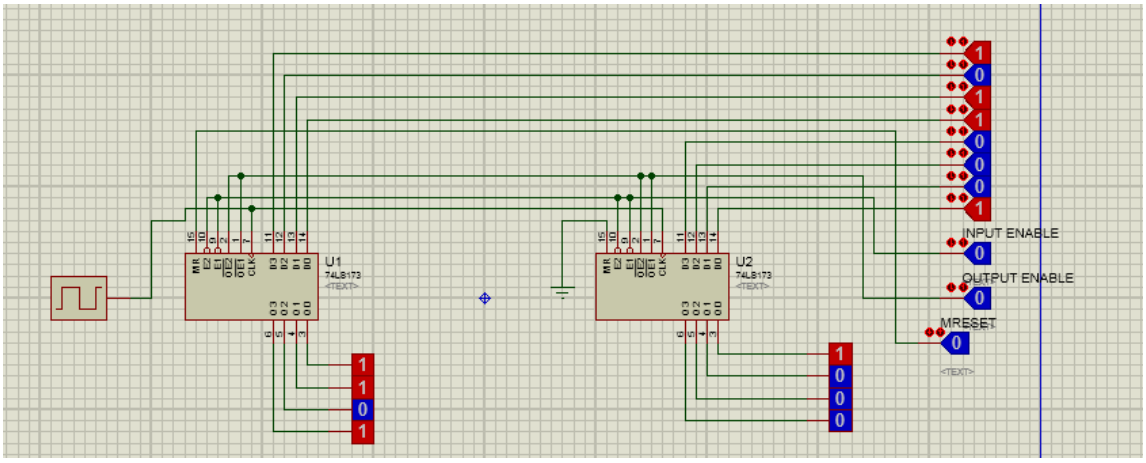


Operation of the ALU:

The IC on the right, gets the lower nibble input from both the Accumulator and the B register and performs its operations on the acquired data while the IC on the left gets the higher nibble input from both the Accumulator and the B register and performs its operation on the higher nibble of the data. The corresponding bits of the Accumulator and the B Register are grouped together. Meaning, the LSB of the Accumulator and the LSB of the B register are sent to the inputs of the adder that will produce the LSB of the result.
The carry input of the right IC is connected to the SU pin. The carry output of that right IC is connected to the carry input of the left IC. The carry output of the left IC is not utilized.
The data from the Accumulator is sent directly to the inputs of both of the ICs. But the data from the B Register is sent through an X-OR gate and the output of that X-OR gate is connected to the inputs of the IC. The other inputs of these X-OR gates are connected to the SU data pin.

Instruction Register:

Function of instruction register is to receive and store the 8-bit instruction placed on the bus from the RAM. The contents of the instruction register are then split into two nibbles (nibble means 4-bit). The upper nibble which contains the instruction to be performed goes into the controller-sequencer while the lower nibble containing the address of the instruction is sent to the bus. A reset pin is required for resetting the instruction register, when the computer starts.



Operation of the Instruction Register:

Each D flip-flop gets its input from the output of a 2 input OR gate. The inputs of these OR gates are taken from two 2 input AND gates. For each group of AND gates the left AND gate has two inputs, one connected to the data input pins and the other input is connected directly to the II pin. The right AND gate has one of its inputs connected to the output of the flip-flop and the other input is connected to the II pin inverted using a NOT gate. Considering the two different states of the II pin, the Instruction Register has the following method of operation

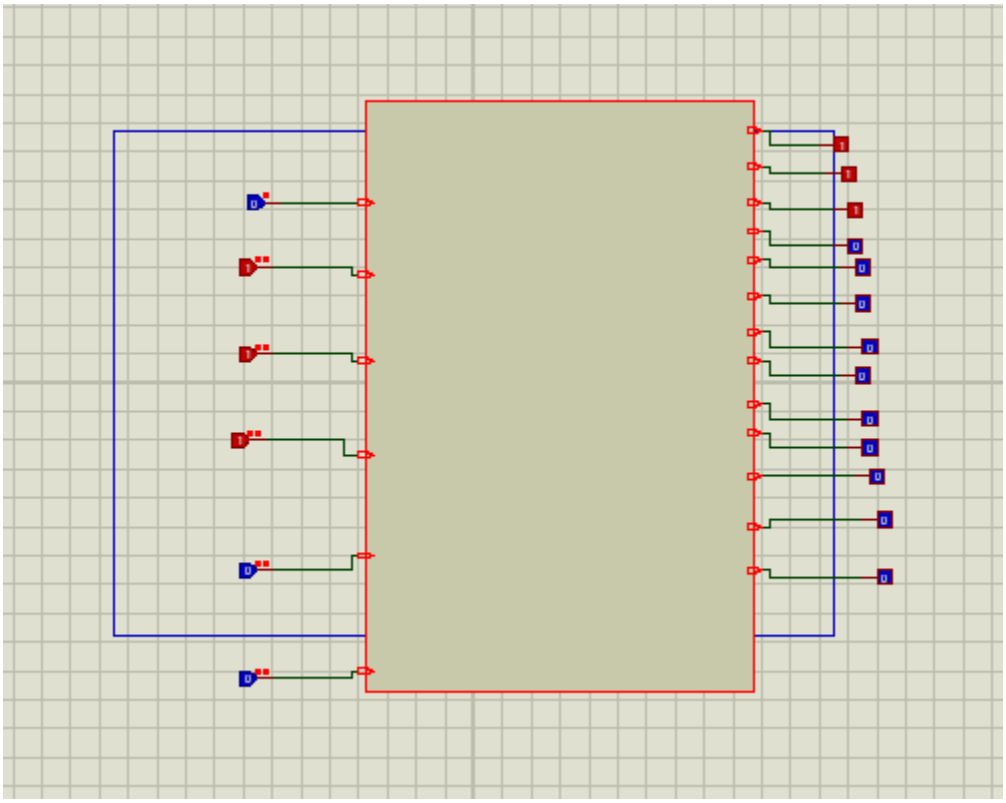
When II = 1, the right AND gates get a 0 input from the NOT gate. That is why the outputs of these AND gates will be 0 regardless of what is present at the other input of the AND gate. On the other hand, the left AND gates get a 1 input. So, whatever data that is fed to the left AND gates from the bus input pins are available at the output of the AND gates. Now, the OR gates have the inputs from the output of the AND gates. So, the output of the left AND gates will be available at the outputs of these OR gates because the other inputs of the OR gates are 0. Consequently, the D flip-flops will have the data given by the bus input pins and store it.

When II = 0, the left AND gates get a 0 input. That is why the outputs of these AND gates will be 0 regardless of what is present at the other input of the AND gate. On the other hand, the right AND gates get a 1 input from the NOT gates. So, whatever data that is fed back to the right AND gates from the Q outputs of the D flip-flops are available at the output of the AND gates. So, the output of the right AND gates will be available at the outputs of the OR gates because the other inputs of the OR gates are 0. Consequently, the D flip-flops will keep feeding back the stored data to its inputs.

Controller-Sequencer:

Controller-sequencer generates the necessary control signals for each block so that the actions of the computer occur in a desired sequence. 12 control bits come out of the Controller-Sequencer block. These control bits determine how all other blocks will react to the next positive clock edge. So, Controller Sequencer was designed in such a way so that the correct control bits are already available to all the blocks before the next positive CLK edge. 12 control bits are –

CON = CE CO MI RO II IO AI AO SU EO BI OI



CS Input Pins:

There are 4 CS Input pins that are used to take the instruction input from the Instruction Register. These four pins are: CS Input 3, CS Input 2, CS Input 1 and CS Input 0. They are connected to IR Output 7, IR Output 6, IR Output 5 and IR Output 4 respectively.

CLK pin:

The J-K flip-flops used to build the ring counter in this circuit are negative edge triggered. The Controller Sequencer is required to be positive edge triggered. That is why, the CLK pin is directly connected to the J-K flip-flops.

RESET pin:

The RESET inputs of the J-K flip-flops are active LOW. However, the entire 8-bit computer was designed to have an active HIGH reset. That is why, the RESET pin is sent through a NOT gate before connecting to the inputs of the J-K flip-flops. So that, RESET = 1 will result in logic '0' being present in the RESET pins of the J-K flip-flops and that will reset the ring counter.

Output Pins:

There are a total of 13 output pins. These are CE, CO, MI, RO, II, IO, AI, AO, SU, EO, BI, OI and HLT. These output pins are used to provide the control bits to each of the submodules/blocks.

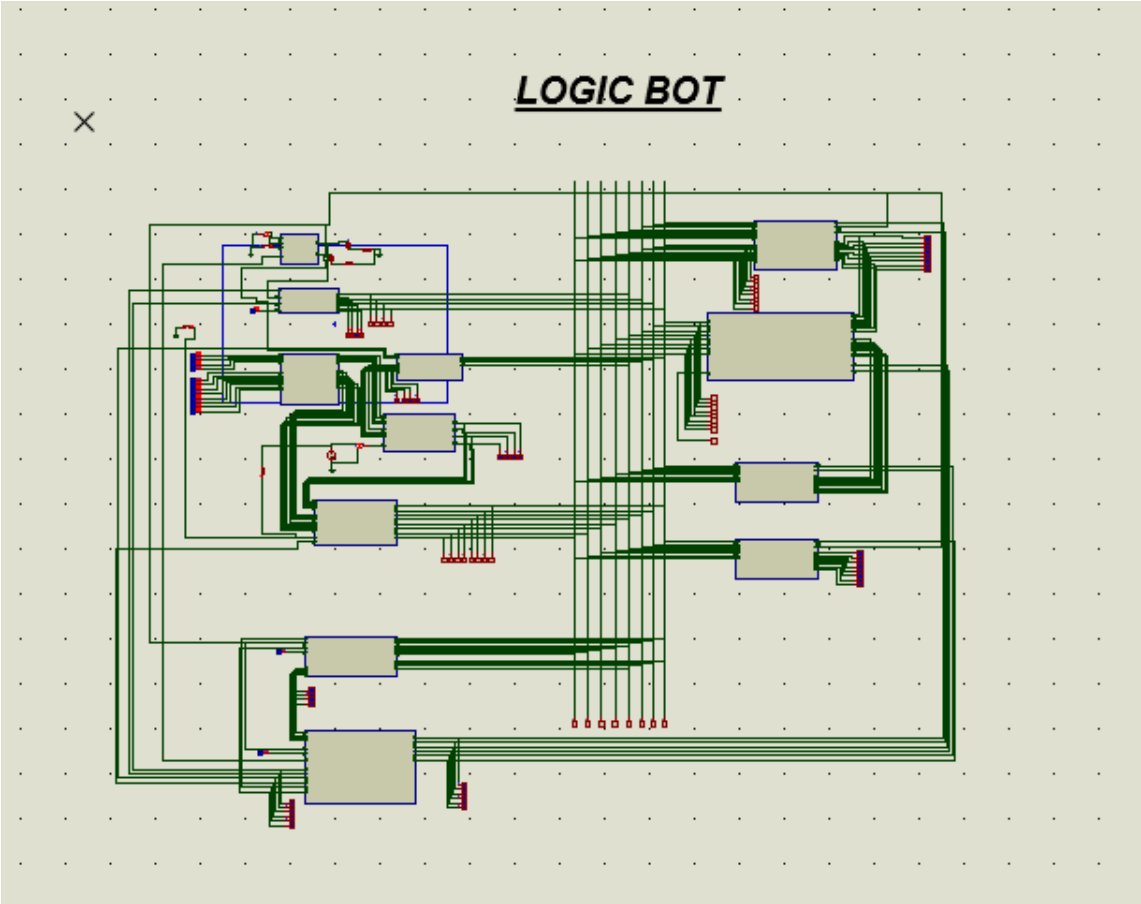
8 bit Bus:

The 8-bit bus has the following characteristics:
1. Bus has 8 lines for address and data transfer (BUS 0, BUS 1, BUS 2, BUS 3, BUS 4, BUS 5, BUS 6 and BUS 7)
2. It is designed in such a way, so that at a single time instance only one module will send data to the bus, and only one module will receive that data.
3. Two modules can never send data to the bus at the same time (bus contention). When a module is sending data to the bus, all other modules' outputs are disconnected.
The four lines of the bus that are to the right side (BUS 0, BUS 1, BUS 2 and BUS 3) are used to transfer the address data. All 8 bus lines are used to transfer any 8-bit data that will be processed using the computer.

Output Display Unit:

The Output Display Unit can show the final output of the computer. In addition to this, this unit has the ability to how the data present in the bus, the data stored in the Accumulator and the B Register. The data stored in the selected address of the RAM can also be viewed using the Output Display Unit.

The Final Project



Problems Faced:

It was later found out that Proteus failed to define some of the logic gates due to the large number of gates required to construct the circuit. As a result, the simulation result showed unknown states at some of the outputs of the RAM in spite of there being no flaw in the design.

After integration we faced problems in running the software. Separate debugging was needed. Similar problems were faced when we integrated the hardware parts too.

Discussion:

The project was successfully completed and the objective that is implementing an 8-bit computer based on the architecture of SAP-1 (Simple-As-Possible) using combinational and sequential circuits was achieved. All the submodules of the computer works coherently and the computer is able to perform addition and subtraction operation on 8-bit data defined by the user. Although the computer has a few limitations, it achieves its intended goals.

The hardware part of the project was incomplete due to unavoidable circumstances. All the part were not giving proper results either. Partial correctness was achieved.

References:

The following references were used in building the computer for this project:
1. Digital Computer Electronics (Third Edition) by Albert Paul Malvino, Ph.D, Jerald A. Brown
2. Building an 8-bit breadboard computer! by Ben Eater