PHASE E REPORT

CourseCode :EEE4308
Course Name : Digital Electronics Lab

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

Group : Team YOROZUYA

Members:

NAME	ID
Khandaker Adeba Tabassum	200021102
Sadat Al Rashad	200021106
Nahadia Tabassum Oyshi	200021112
Md. Nazmul Aman	200021132
Farhan Sharif Rishad	200021136

Supervisor :
Mr. Nadim Ahmed
Lecturer, Electrical & Electronic Engineering Department
Islamic University of Technology

Introduction:

In this project we implemented an 8-bit computer based on the architecture of SAP-1 (Simple-As-Possible) using combinational and sequential circuits. We divided this project into some submodules .

In phase A we implemented submodules of Clock Pulse Generator, Program Counter, Input Unit and MAR .

1. **Clock Pulse Generator**

Clock signal should be generated using 555 timer IC. The clock module needs to have two modes of operation available- (i) Auto and (ii) Manual. In manual mode, pressing a switch/push
button will advance each clock cycle. In auto mode, the click pulse will be generated after a certain time depending upon the value of resistance and Capacitor.

2. **Program Counter:**

Program counter (PC) is a 4-bit counter which can count from 0000 up to 1111. 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 each instruction cycle (completion of an instruction)

Pin	Type	Function
CE	In	Program Counter Enable: Increments the program counter on the next clock-cycle.
CO	In	Program Counter Out: Puts the current program counter value on the bus. Output is disconnected when CO is low.
Reset	In	Resets the program counter to 0000
Clock	In	Clock pulse input.

Out	Out	4-bit counter output.

3. **Input Unit and MAR:**

Memory address register (MAR) is a 4-bit register. MAR stores the 4-bit address of data or instructions 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. Functions of its I/O pins are as follows-

Pin	Type	Function
MI	IN	Memory Address In: Stores the current values of the bus into the Memory address register (MAR).
CLK	IN	Clock pulse input.
OUT	OUT	4-bit address data.

Input unit is used when a computer is needed to be programmed by writing data into the RAM (PROG state). There should be a selector to choose between MAR output (4-bit) and Input address output (4-bit) depending on the state (RUN/PROG). Output of the selector should be connected to address pins of RAM. The following is a description of switches that input unit should contain:

Switches	Function
Input Data Switches	8 switches (D7-D0) for writing 8-bit data to a specific address.
Address switches	4 switches (A3-A0) for accessing a particular address of RAM for writing.
R/W Switches	For switching between read and write mode.

RUN/PROG Switches	PROG state is used for writing data into the RAM before running the computer. RUN switch will be active when a certain program is being executed on the computer.

In phase B,we implemented submodules of RAM and 3 registers namely Accumulator, B Register and Output Register.

1.**Random Access Memory (RAM):**

Our RAM size should be 16 x 8 bit (16 memory locations each storing 8 bits of data). The RAM will be programmed by means of the address switches and data switches of the input unit. During computer run, the RAM will receive its 4-bit address from the MAR.

Pin	Type	Function
ADRS	In	4-bit address input for accessing RAM locations.
I/P Data	In	In PROG state, used for writing 8-bit data to address location.
R/W	In	Read or write mode
CS	In	Chip Select pin.
RO	In	Put the currently selected RAM byte onto the bus Output is disconnected when RO is low.
O/P Data	Out	8-bit output data

2. **Accumulator, B Register and Output Register**:

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

Pin	Name	Function
AI	Accumulator In	Stores the current values of the bus into Accumulator.
AO	Accumulator Out	Accumulator sends its stored content to the bus. Output the accumulator is disconnected when AO is low.
BI	B Register In	Stores the current values of the bus into B Register.
OI	Output Register In	Stores the current values of the bus into Output Register

In Phase C we implemented submodules of Controller Sequencer and Instruction Register.

1.**Controller Sequencer:**

Controller-sequencer generates necessary control signals for each block so that actions 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 CLK edge. So, controller-

sequencer must be designed in such a way so that correct control bits are already available to all the blocks before the next positive CLK edge. 12 control bits are – **CE, CO, MI, RO, II, IO, AI, AO, SU, EO, BI, OI.**

These 12 control bits are used to control the other submodules. Besides these 12 signals, controller-sequencer also generates a HLT (halt) signal, which can halt the process of computer by stopping the main clock.

2.**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 goes into the controller-sequencer while the lower nibble should be sent to the bus. A reset pin is required for resetting the instruction register, when computer starts. Control bits of instruction register are-

Pin	Name	Function
II	Instruction Register In	Writes the current values of the bus into the instruction register.
IO	Instruction Register Out	Sends the stored data (only lower nibble) onto the bus. Output pins are disconnected when IO is low.

In phase D we implemented submodules of Arithmetic and Logic Unit (ALU), Integration with 8-bit Bus, Output Unit.

1.Arithmetic and Logic Unit (ALU):

ALU is used to add or subtract 8 bit number. 2's complement is used for the addition and subtraction. Two control bits of ALU :

Pin	Name	Function
SU	Addition/Subtraction Select	When SU=0, addition operation, when SU=1, subtraction operation will be executed.
EO	ALU Out	Puts the ALU output onto the bus. Output of ALU should be disconnected when RO is low.

Addition:

The ALU uses 2 74LS283 *4-bit binary adders with carry*. The **carry output** is connected to the **carry input** of U4 thus a carry after bit 4 is propagated to the second nibble of the result. The inputs 1 and 2 of the ICs are connected directly to the outputs of the 74LS173 registers of RA and (through XOR gates) RB instead of the DB. For addition, the **subtract line** is low, thus the XOR gates output the same value as the input from RB. To only drive the DB lines when the result of the computation is needed, another 74LS245 is used as tristate buffer, enabling the display the result.

Subtraction:

For subtraction of RA and RB, the output of the XOR gates is always the *logical inverse* of the input from RB. Thus the second byte into the 8-bit adder is the 1's *complement* of RB. To calculate a 2's complement, result drives the input effectively adding 1 to the result.

2.Integration with 8-bit Bus :

8-bit is 8 wires used for data transfer between modules. At a single time instance only one module will send data to the bus and only one module will receive the data. Two modules can never send the data to the bus at the same time. When a module is sending the data to the bus all other modules outputs must be disconnected.

3.Output Unit :

Output unit is a row of eight LEDs to show the contents of output register. Because each LED connects to one flip-flop of the output port, the binary display shows us the content of the output port. Therefore, after we transferred an answer from the accumulator to the output port, we can see the answer in binary form. Binary display unit is the output device of the SAP-1 microprocessor.

Designing and Implementation:

Phase A:

Clock Pulse Generator

Clock Pulse is a signal used to synchronize the operations of an electronic system. Clock pulses are continuous, precisely spaced changes in voltage. We have implemented two types of clock pulse in our design. Namely: Astable and Monostable. Astable Pulses are continuous auto pulses while the Monostable pulses are manually generated with the help of the push switch.
Both type of pulses are generated using 555 timer ICs.

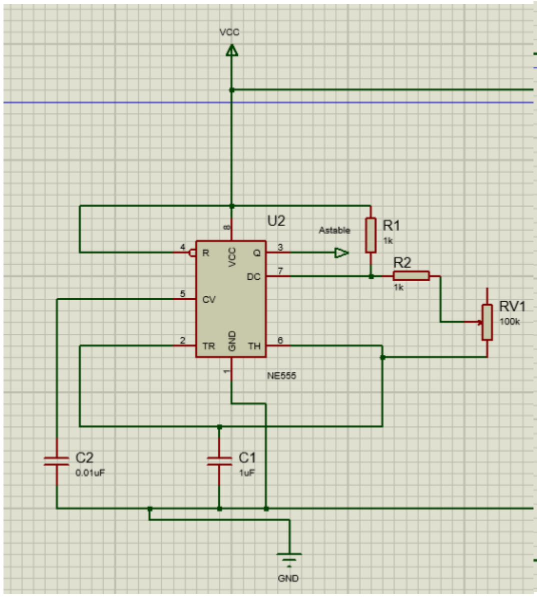


Fig: Astable Mode

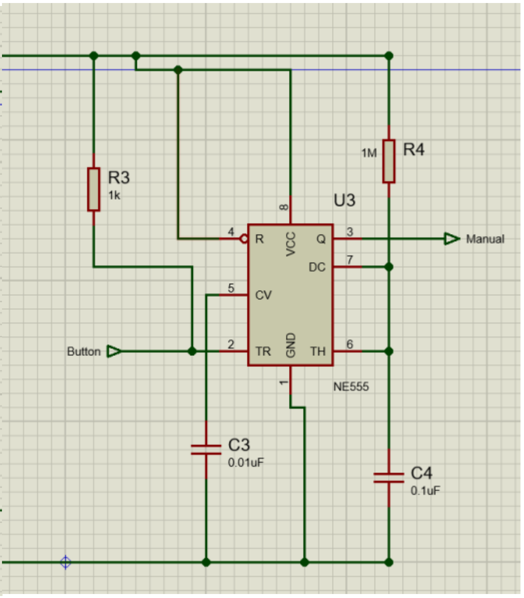


Fig: Monostable Mode

The selection between these two modes were done by using a spdt switch along with a debouncing circuit, and a combinational logic design implemented by using only NAND gates.

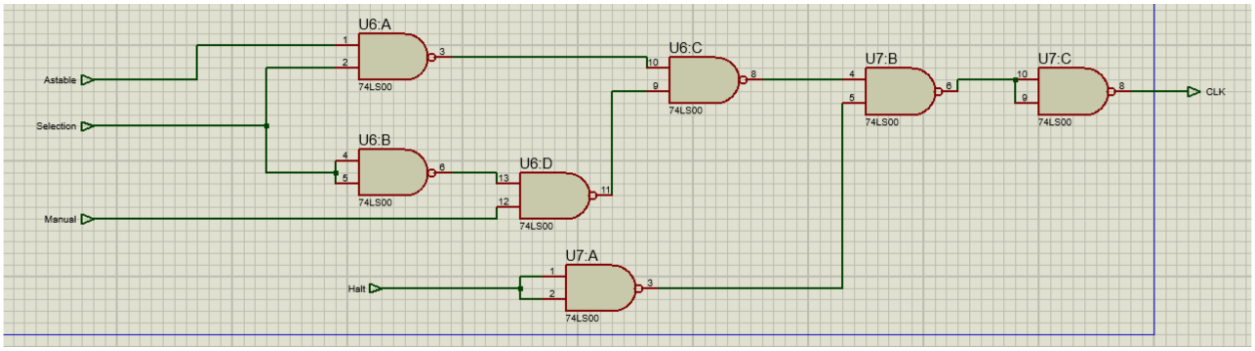


Fig: Clock Switching Logic

There’s also a halt input which can stop the output of the CP generator.

Program Counter:

A program counter contains the address (location) of the instruction being executed at the current time. Program counter is an asynchronous counter that can count from 0000 to 1111 in binary. In program counter, J-K Flip-Flops are clocked by the output of the previous flip-flops while the first Flip-flop is clocked from the output of our CP generator. As we are counting 4 bit binaries, we need 4 parallel JK FlipFlops(74HC107) for this.

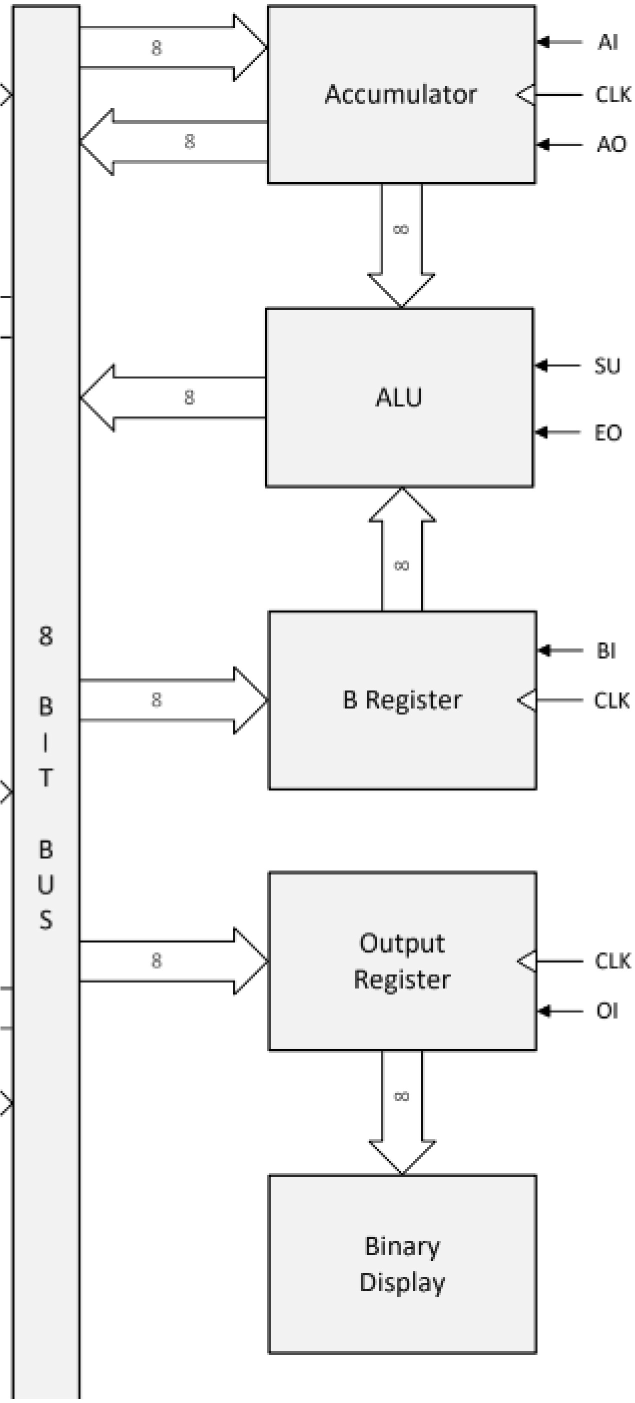
Input Unit and MAR:

Input Unit is a simple arrangement of 14 different inputs which will act as inputs in different modules of our SAP-1. 8 inputs will act as our 8-bit binary input, 4 inputs will act as our memory address location,1 input will act as a switch of Read/Write, 1 input will act as a switch between Run/Prog. 2 8-pin DIP Switch will act as our input unit.

Memory Address Register(MAR) will act as a memory unit of our SAP-1 computer. It will hold the value of the counter in the previous clock pulse cycle. We have used 74LS173 IC as our MAR and 4 LEDs act as our output.

Phase B:

Accumulator, B Register and Output Register:



C

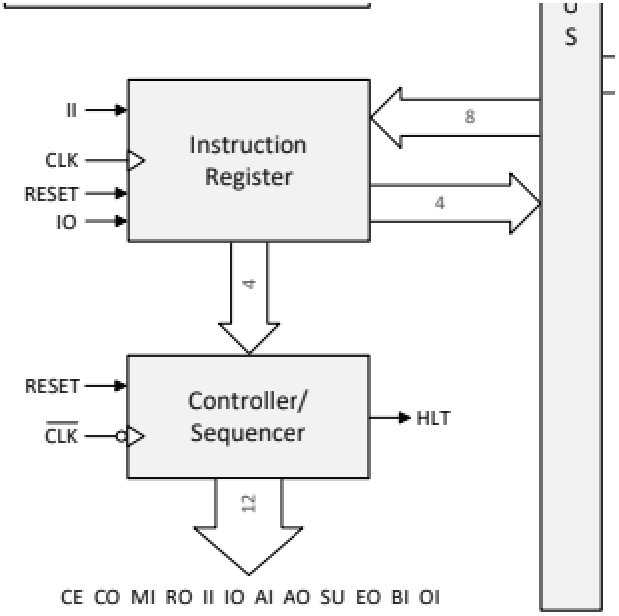
All these 3 types of registers are 8 bit registers. B register and output register even works the same way. Accumulator has a little bit of difference in its working principle. These registers are used for different purposes. During each clock pulse, the input data from the bus will be stored in the Accumulator, and based on the sequence of instructions provided by the user, it will transfer the data to the ALU or back to the bus within the particular state. The B-register receives the 8-bit data from the bus during each clock pulse and transfers the data to the ALU during the negative edge triggering of the clock pulse. Similarly the output register receives and stores the data from the 8-bit bus (transferred from the accumulator) and sends it to the output unit.

RAM:

RAM module serves as our main memory unit. Our RAM module is a 16x8 unit which indicates that there are 16 memory address locations and it can take 8 bit Inputs. Basically there are two modes of operations in RAM, namely RUN and PROG. In RUN mode, MAR inputs work as primary address inputs. In PROG mode, the user-given input of address bits will be selected. There's also a R/W pin in the RAM which can READ or WRITE input datas. In WRITE mode, the module stores the given 8 bit input in its memory. In READ mode, it shows the value stored at the corresponding memory address as outputs.

Phase C:

Instruction Register:



This is a 8-bit register like B register and output register but has a little bit of difference in its working principle. The input data from the bus (The instruction sent by RAM) will be stored in the register when II is HIGH and then the 8 bit data is split into two 4 bit nibbles. Upper one is sent to the Controller and the lower is sent back to the bus if IO is high.

Controller-sequencer:

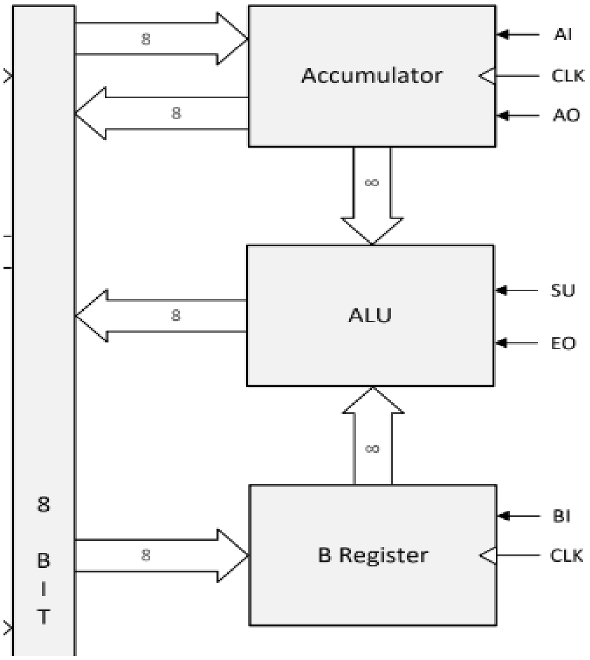
The controller-sequencer receives the instruction from the Instruction register and sends out signals to execute our desired process that can control our SAP and makes sure things happen only when they are supposed to. The 12 bit output signals from controller-sequencer is called the control bit. There is also HLT output which can halt the clock module and hence halt the whole operation of our SAP. In each of our instruction cycles, there are 5 T-states. We reduced one T-state. We combined the T2 and T3 states mentioned in the project manual. T-state means the time required to complete each microinstruction. For each instruction, different control bits are active in different t-state. We have implemented a ring counter to generate our T-states. And we used basic logic gates along with a XOR gate for the selection of our control bits during different T-states.

T-State	LDA(0000)	ADD(0001)	SUB(0010)	OUT(0011)
T1	CO,MI	CO,MI	CO,MI	CO,MI
T2	CE,RO,II	CE,RO,II	CE,RO,II	CE,RO,II
T3	IO,MI	IO,MI	IO,MI	AO,OI
T4	RO,AI	RO,BI	RO,BI	-
T5	-	EO,AI	EO,AI,SU	-

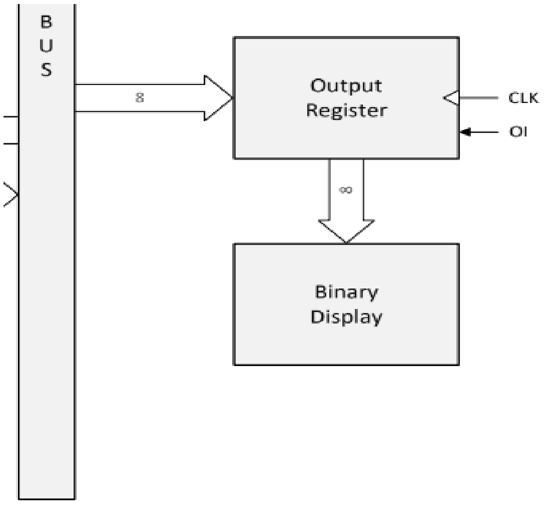
Control Bit Eqns-
CO = T1
MI = CO + IO
CE = T2
RO = CE + (I0’ + I1’)T4
IO = (I0’ + I1’)T3
AI = (I0’ + I1’)T5
BI = (I0 XOR I1)T4
EO = (I0 XOR I1)T5

SU = I1I0’T5
AO = I1I0T3
HLT = I3I2

Phase D:
ALU:



Output Unit:



Software Simulation:

Phase A:

Clock Pulse Generator

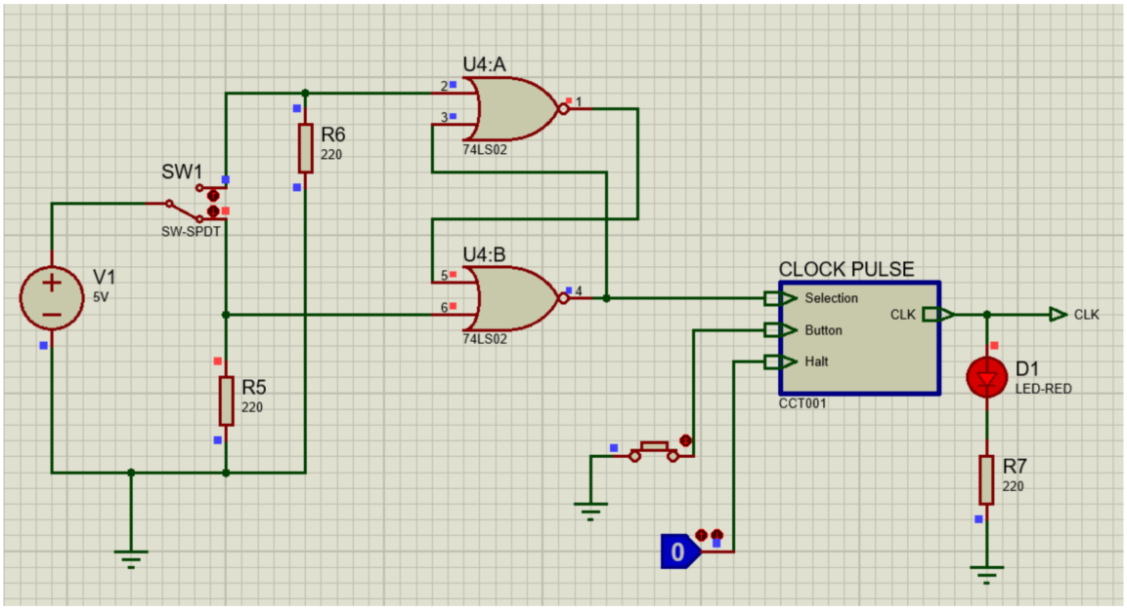


Fig: Parent Sheet

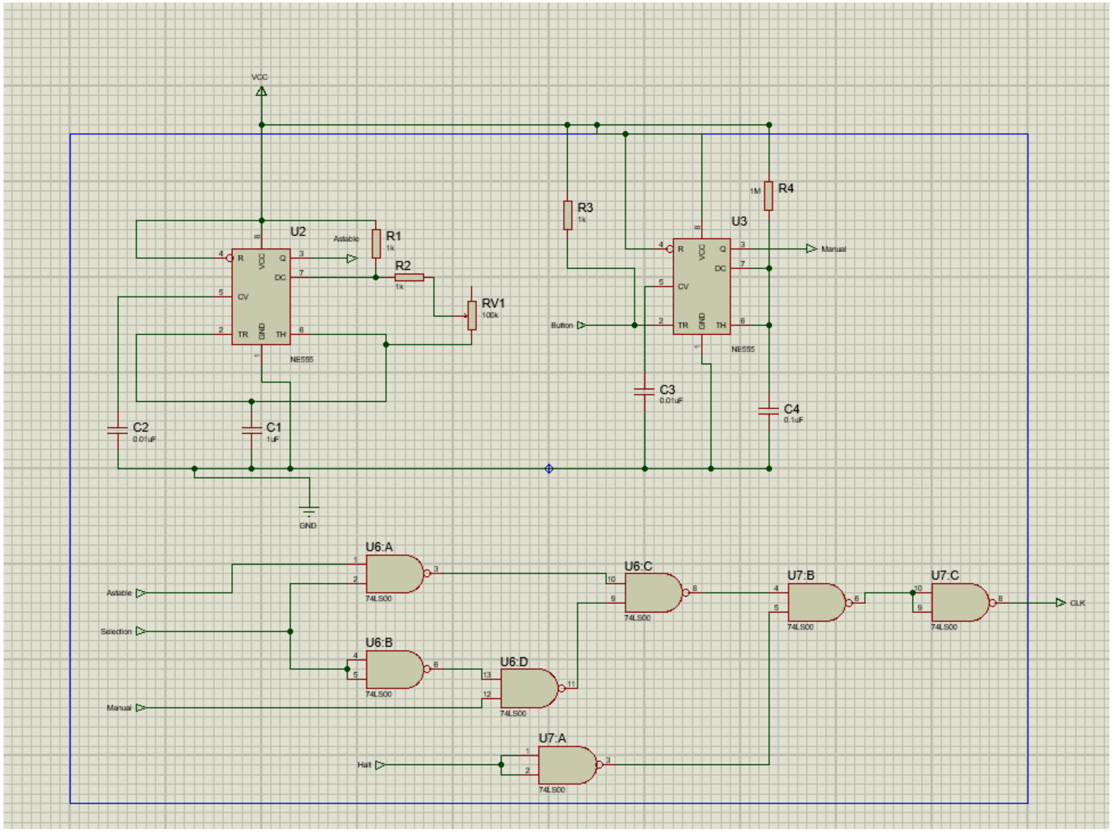


Fig: Child Sheet

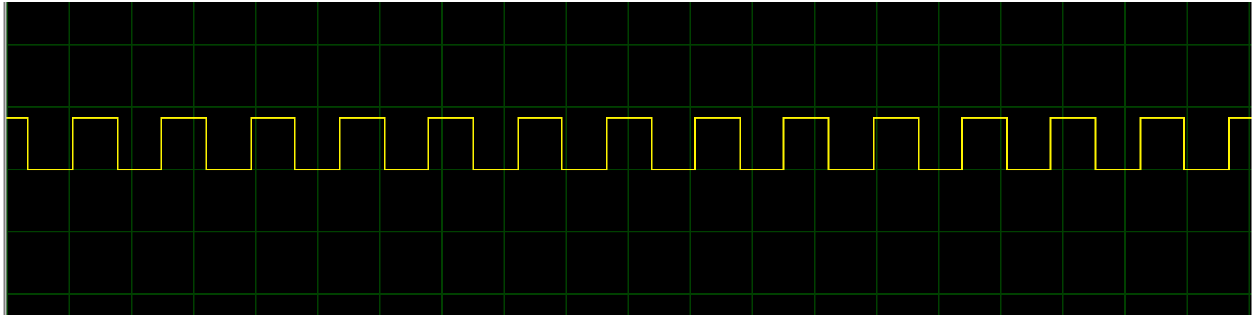


Fig: Output

Program Counter:

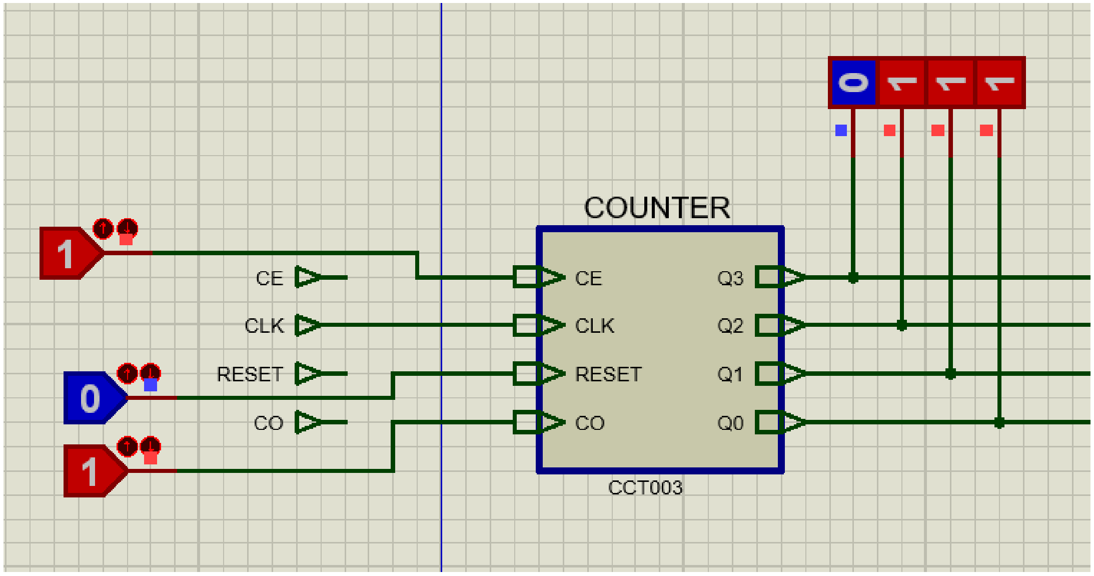


Fig: Parent Sheet and Output

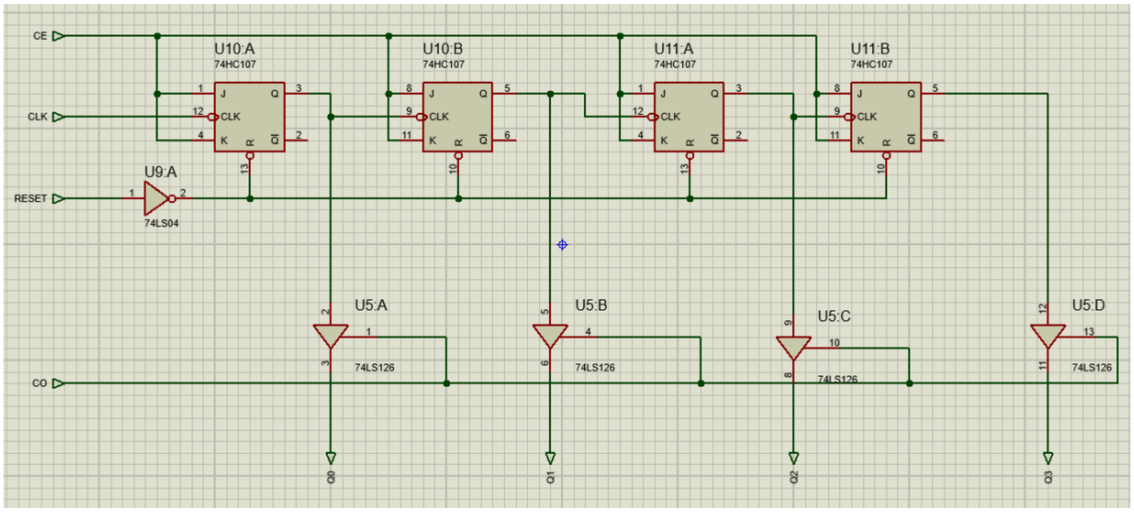


Fig: Child Sheet

Input Unit:

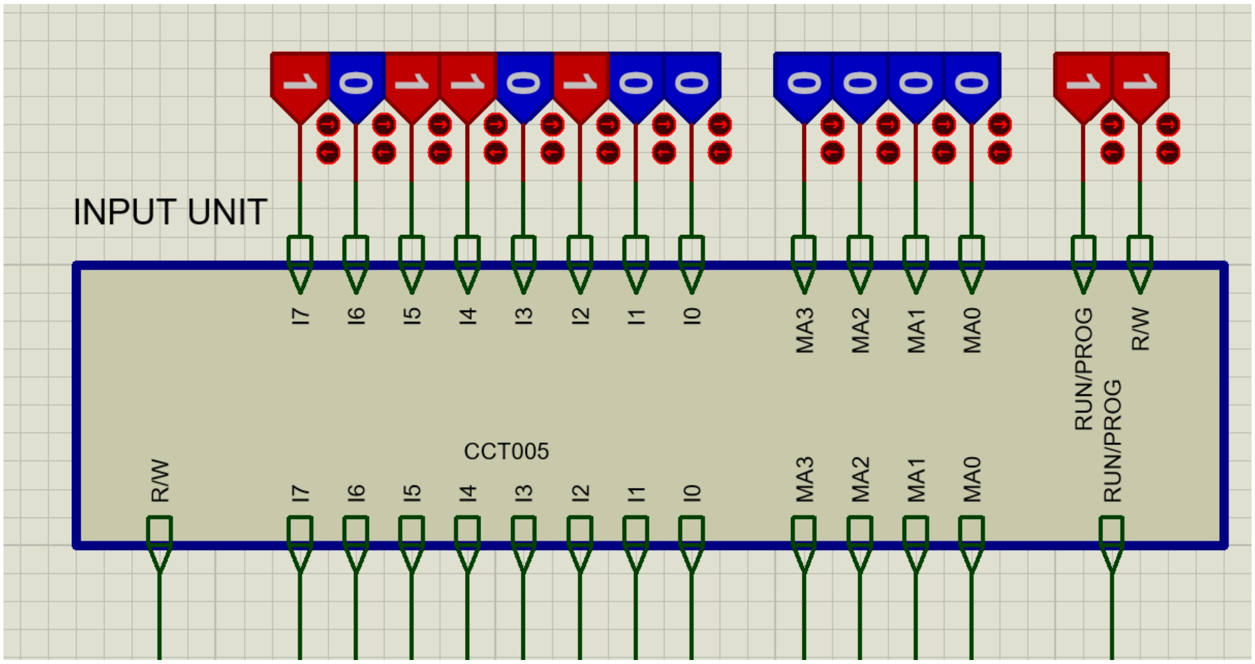


Fig: Parent Sheet

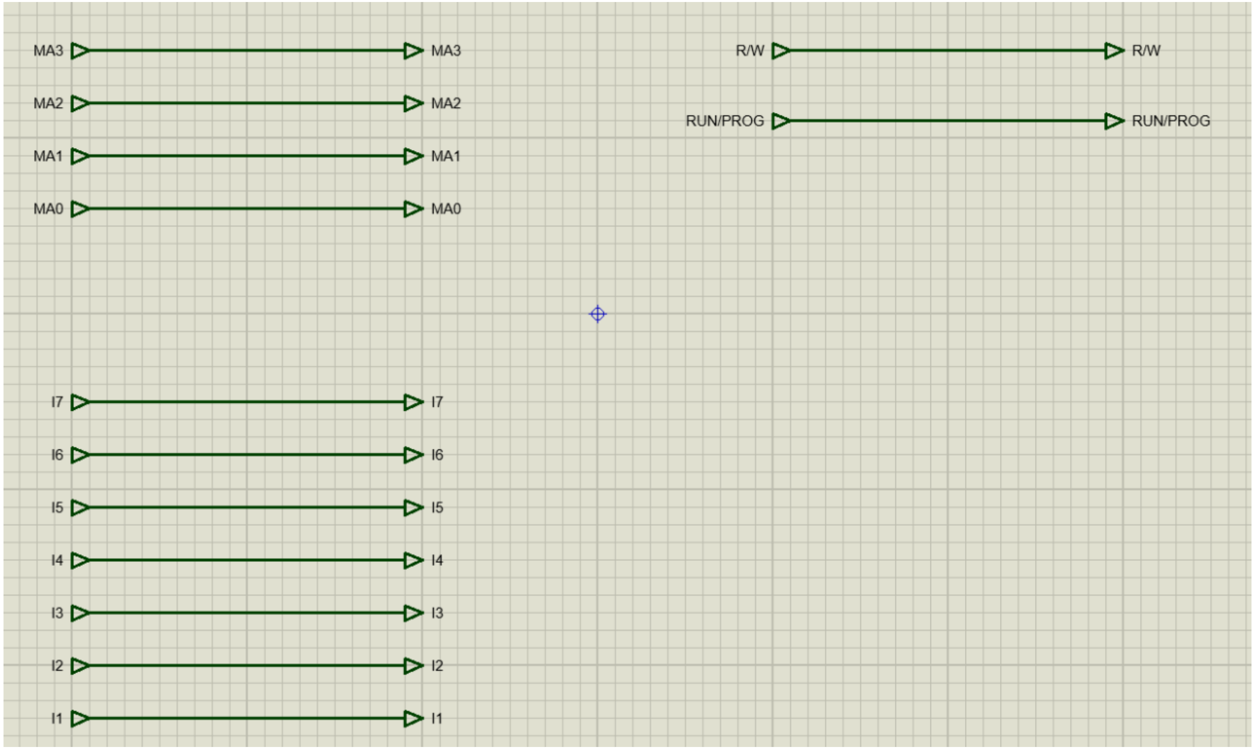


Fig: Child Sheet

MAR:

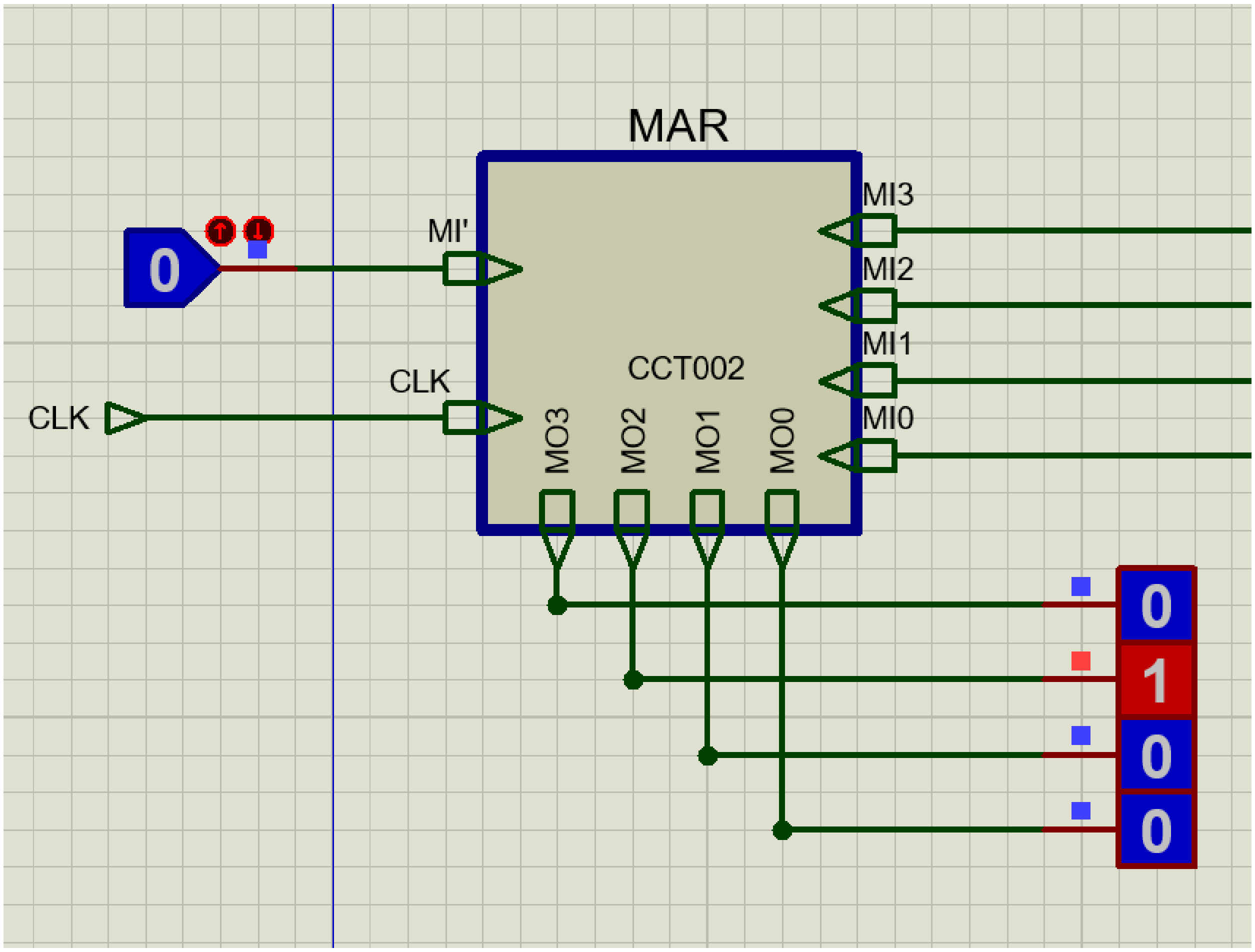


Fig: Parent Sheet and Output

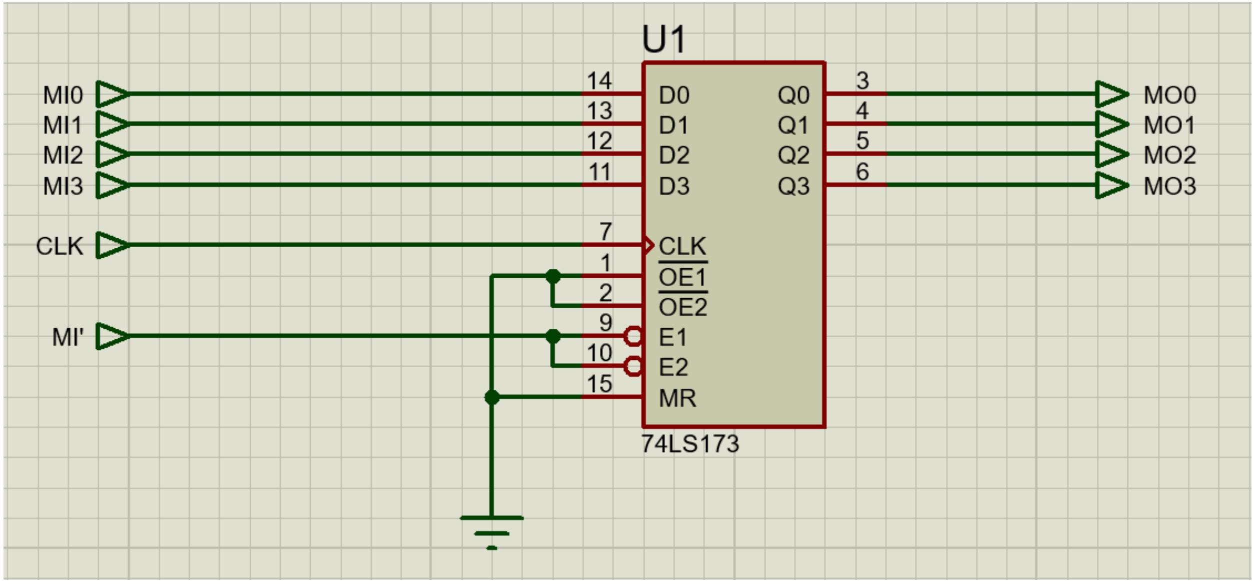


Fig: Child Sheet

Phase B:

RAM:

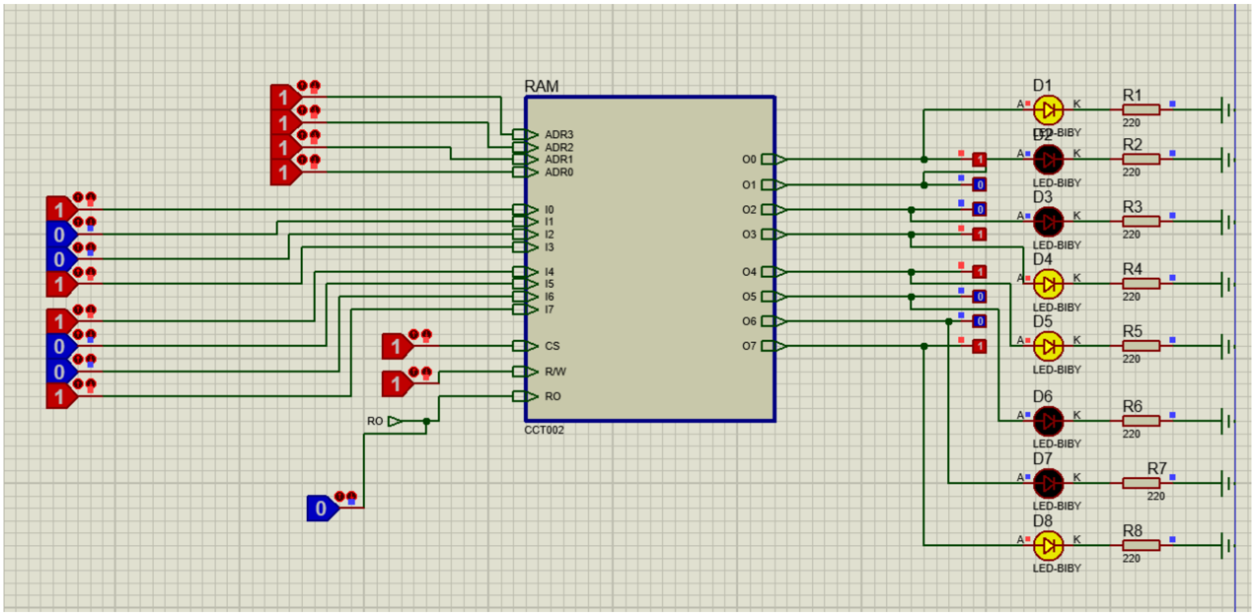


Fig: Parent Sheet

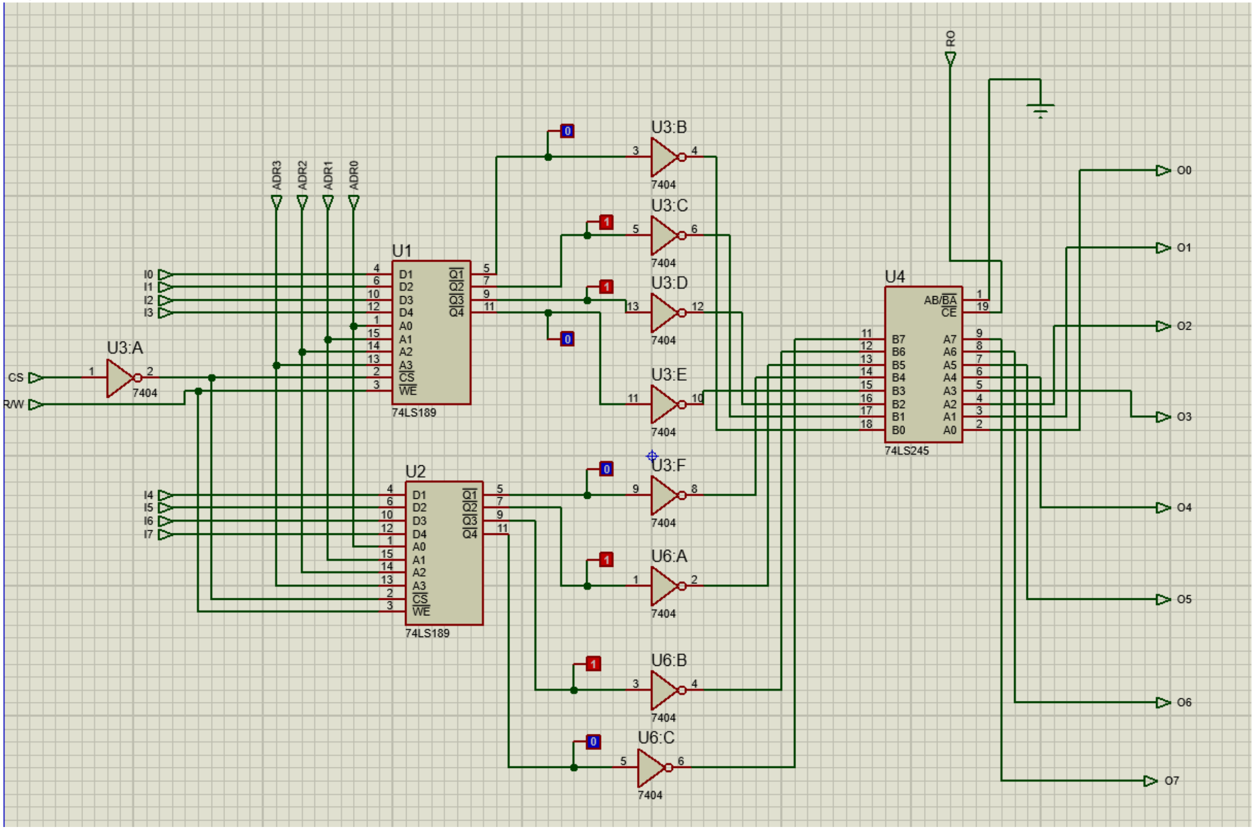


Fig: Child Sheet

ACCUMULATOR:

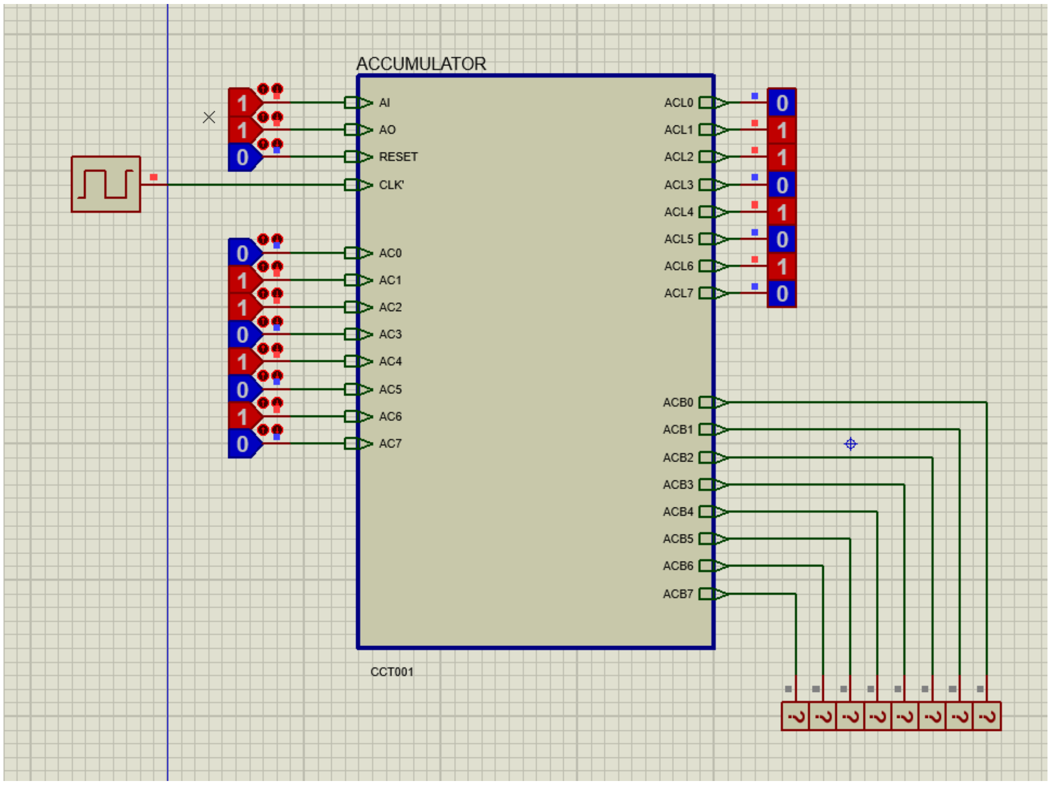


Fig: Parent Sheet and Output

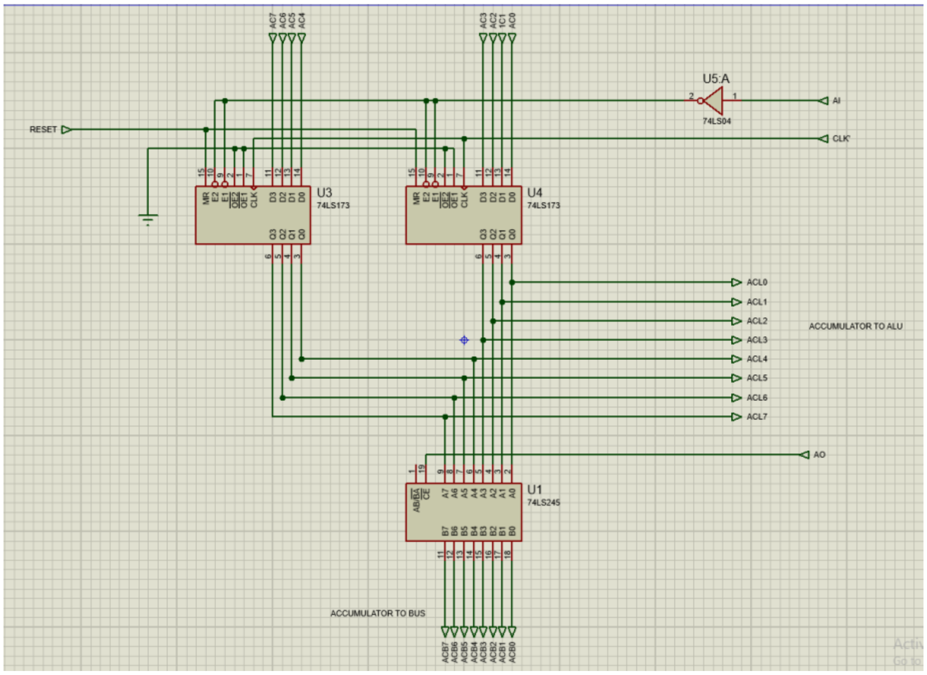


Fig: Child Sheet

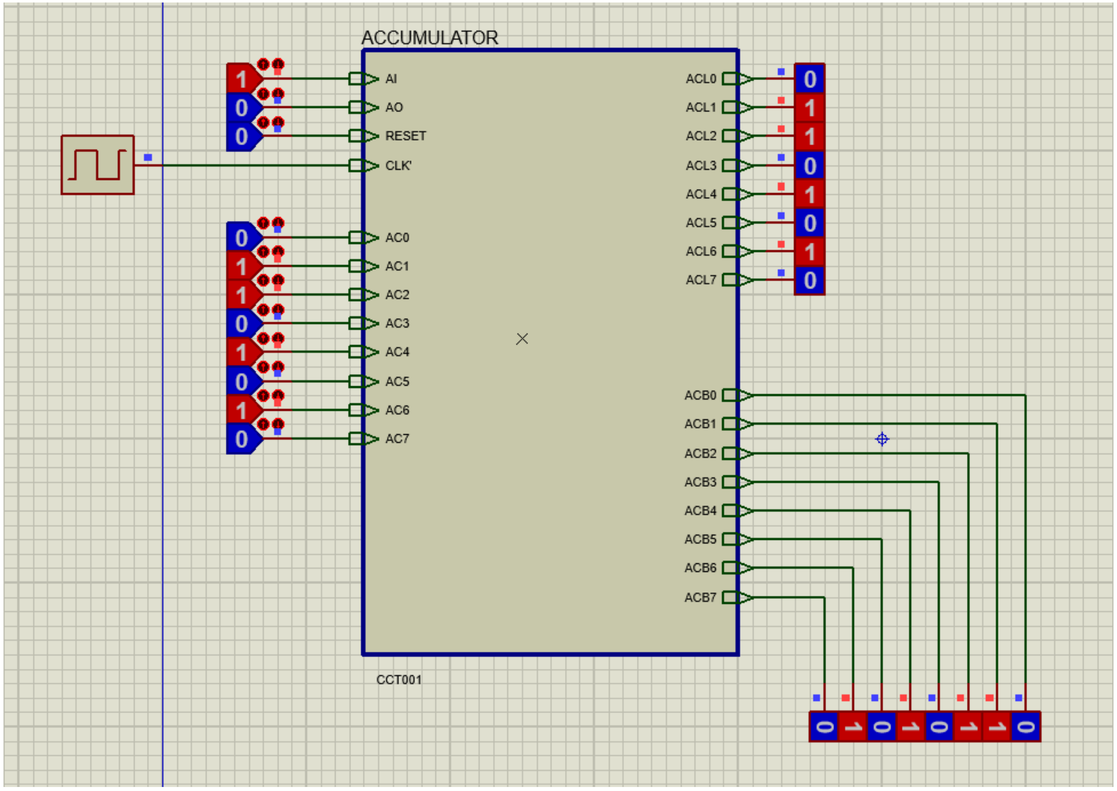


Fig: BUS Output

Output Register:

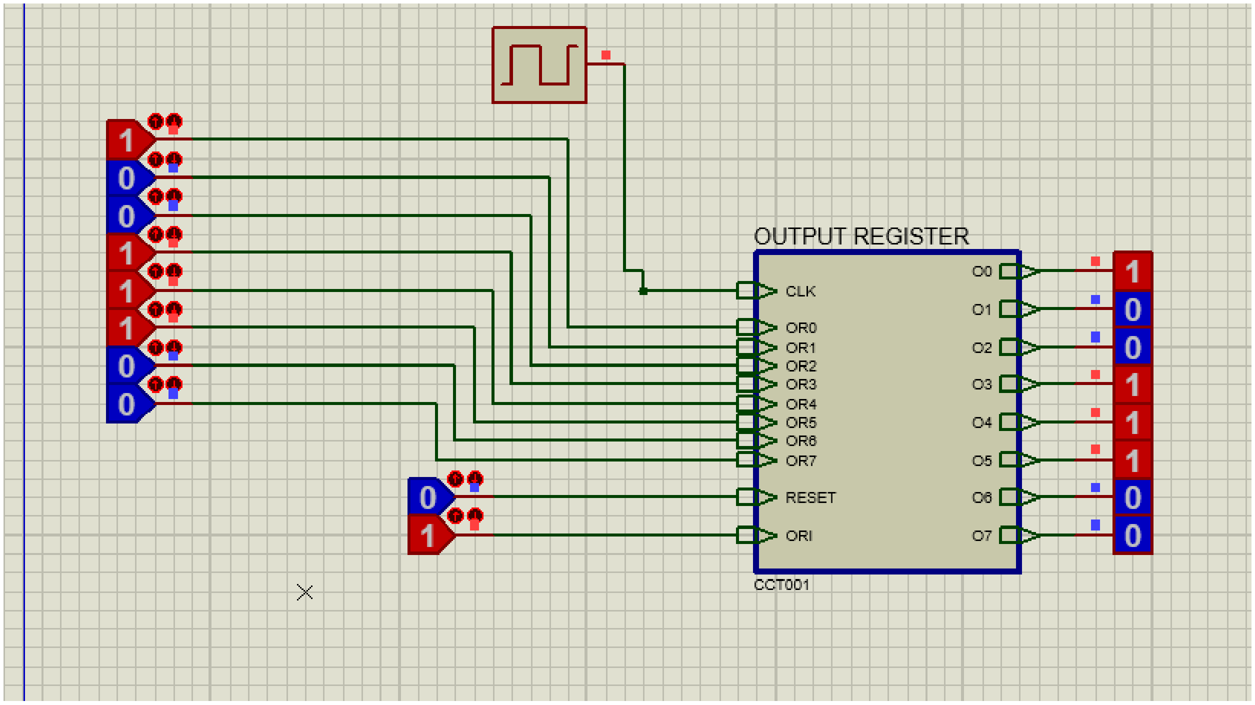


Fig: Parent Sheet and Output

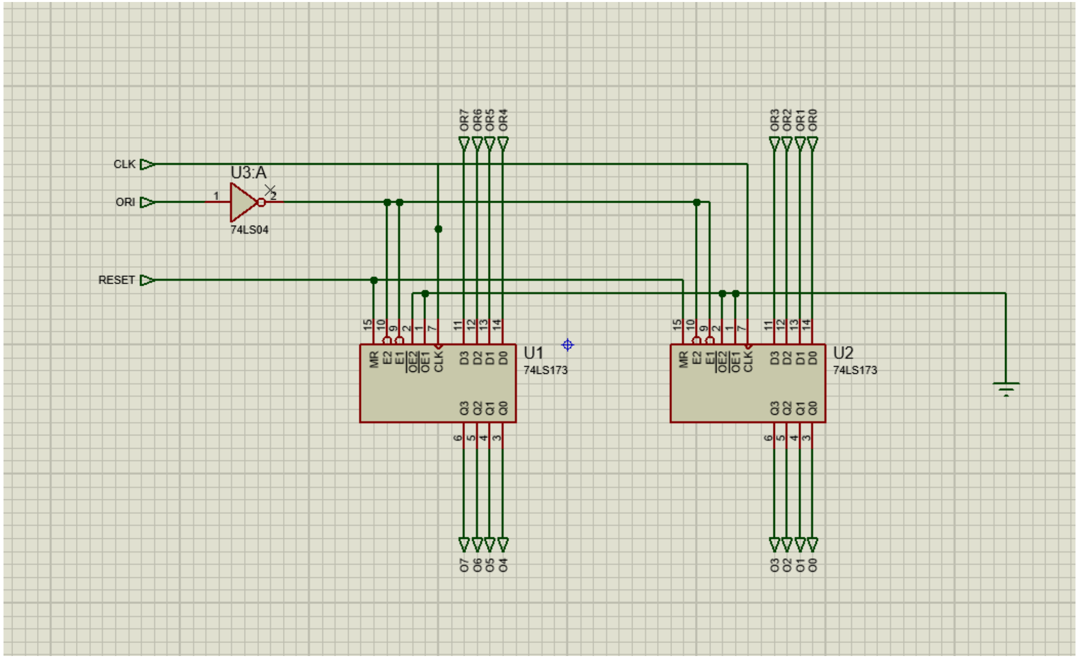


Fig: Child Sheet

B Register:

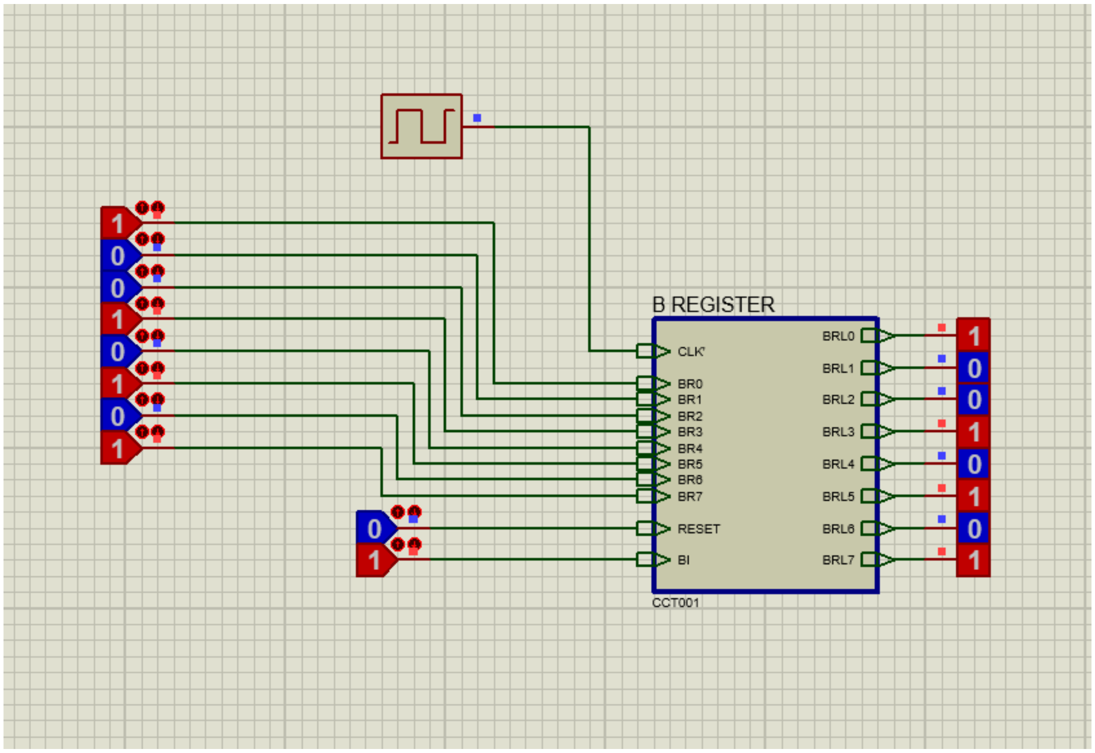


Fig: Parent Sheet and Output

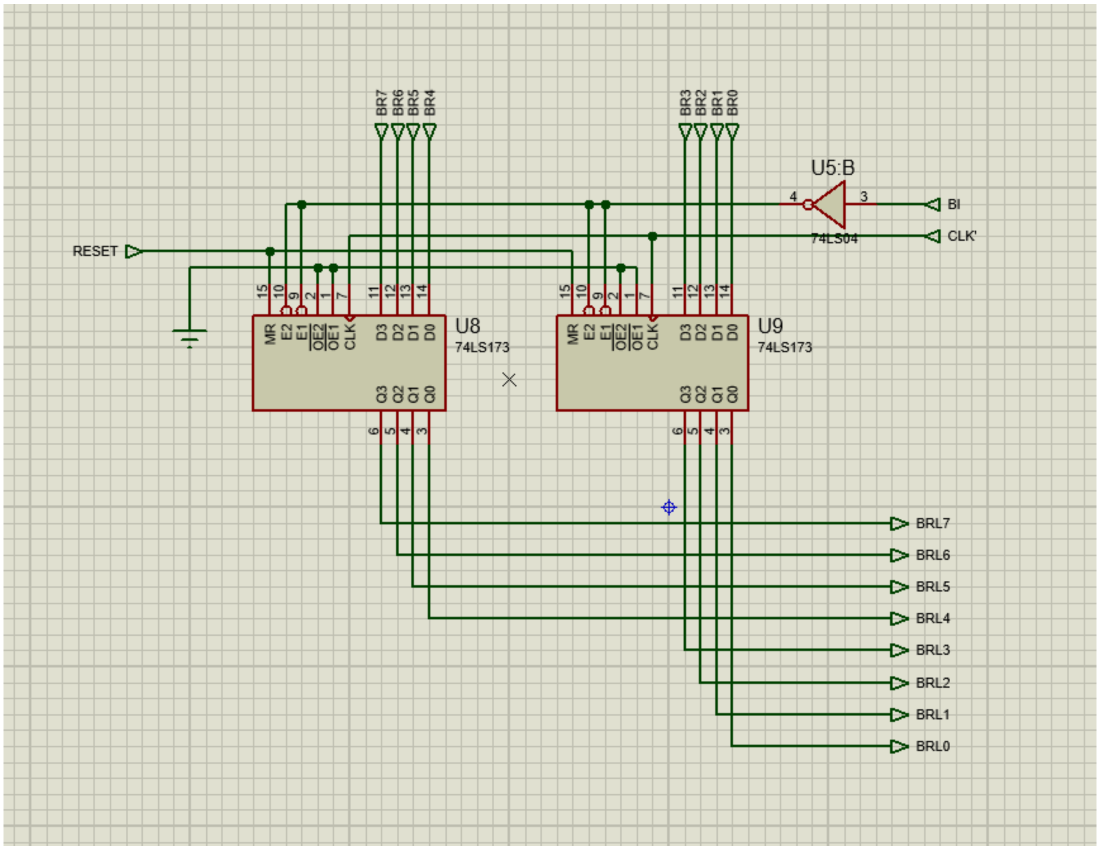


Fig: Child Sheet

Phase C:

Instruction Register:

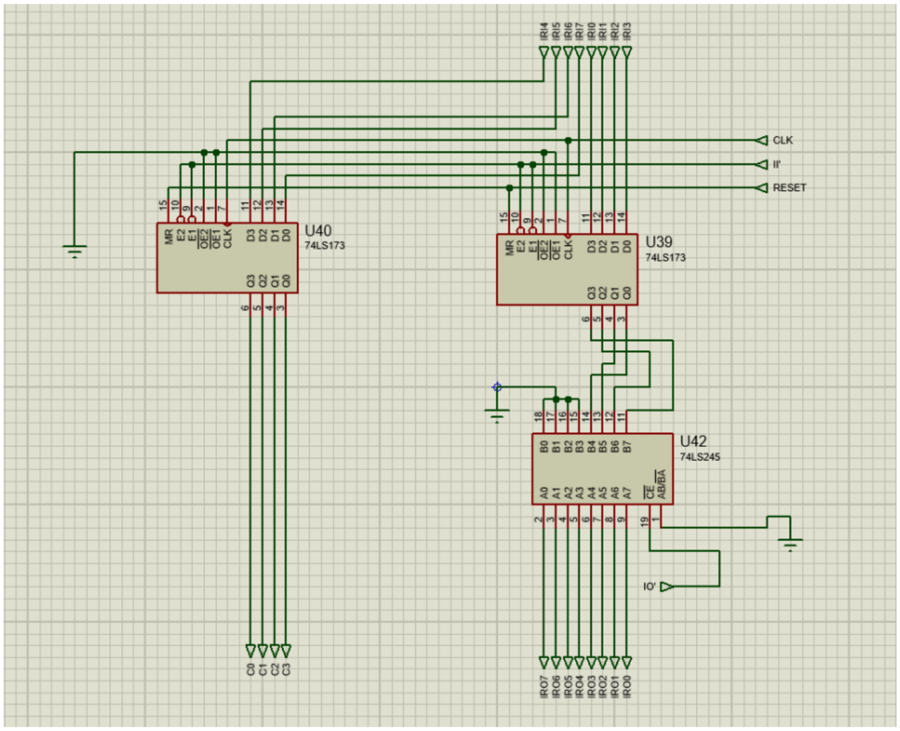


Fig: Parent Sheet

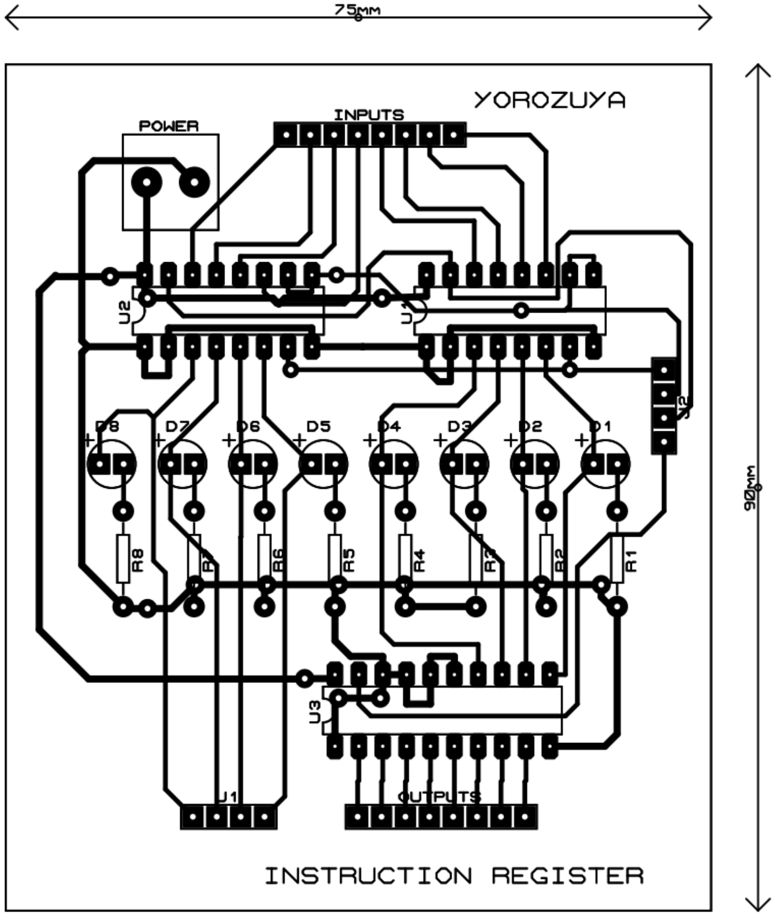


Fig: PCB

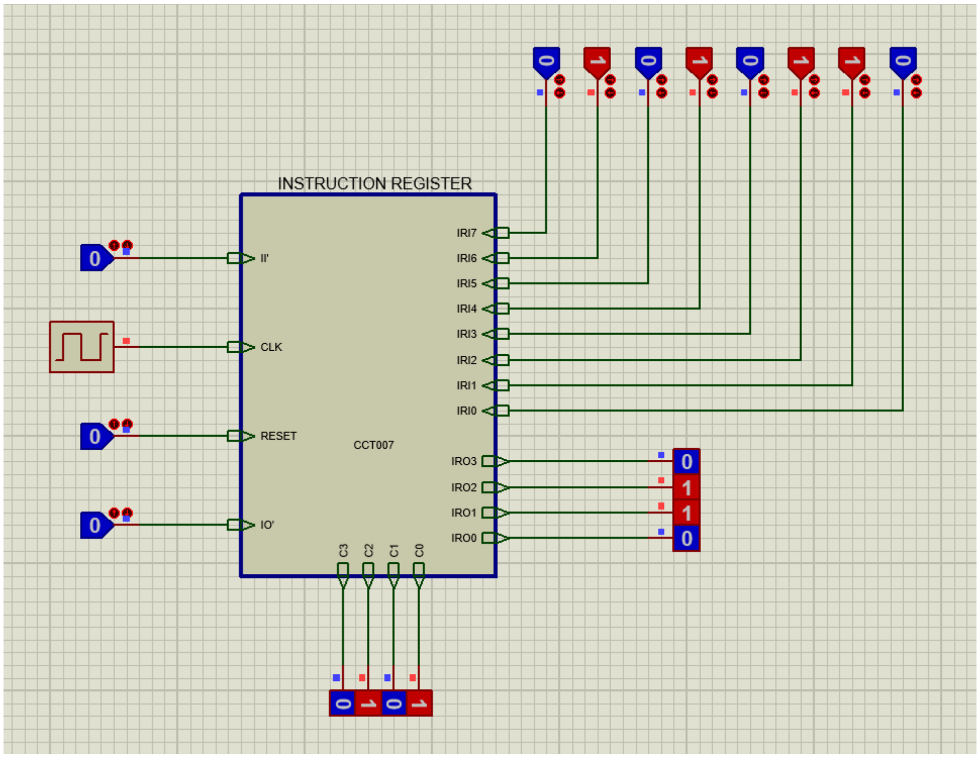


Fig: Child Sheet and Output with IO enabled

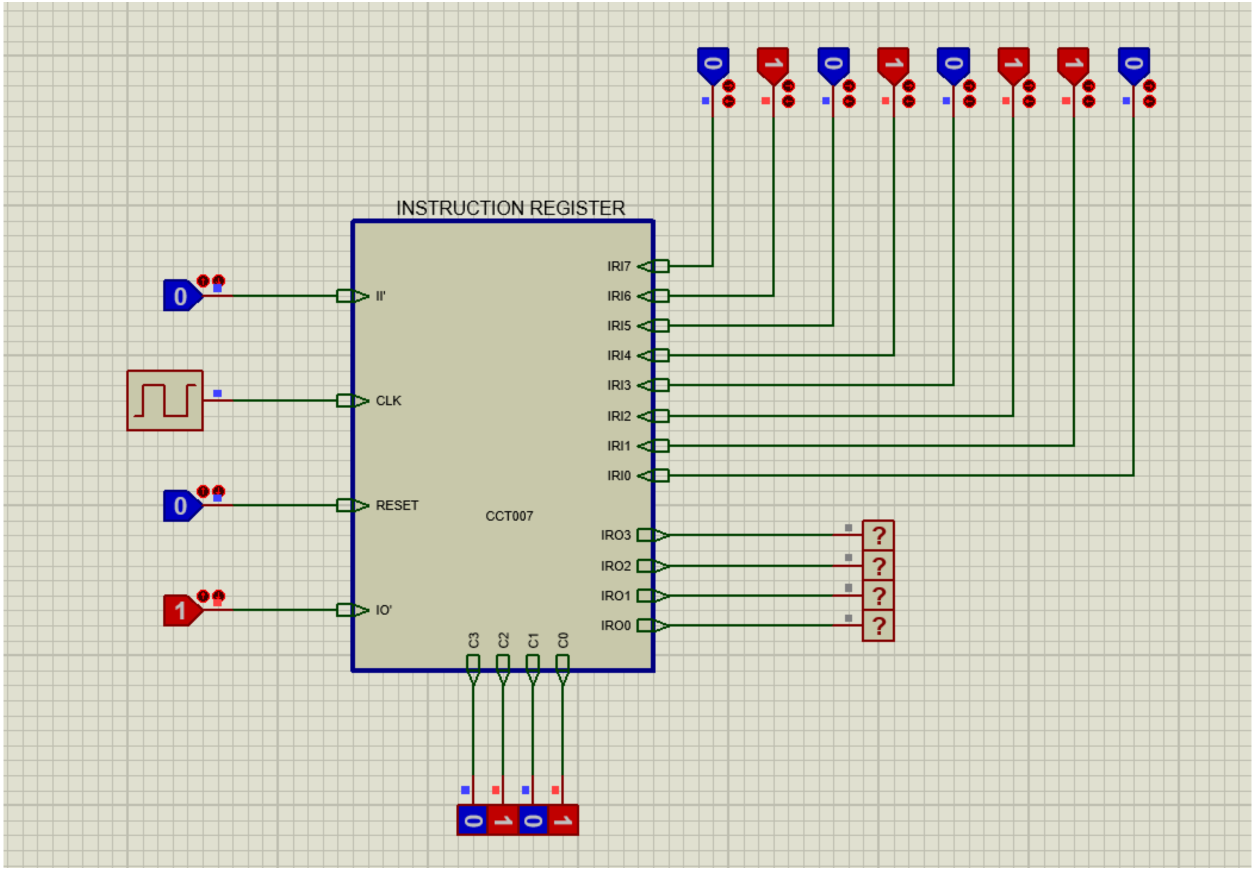


Fig: Child Sheet and Output with IO disabled

Controller-Sequencer:

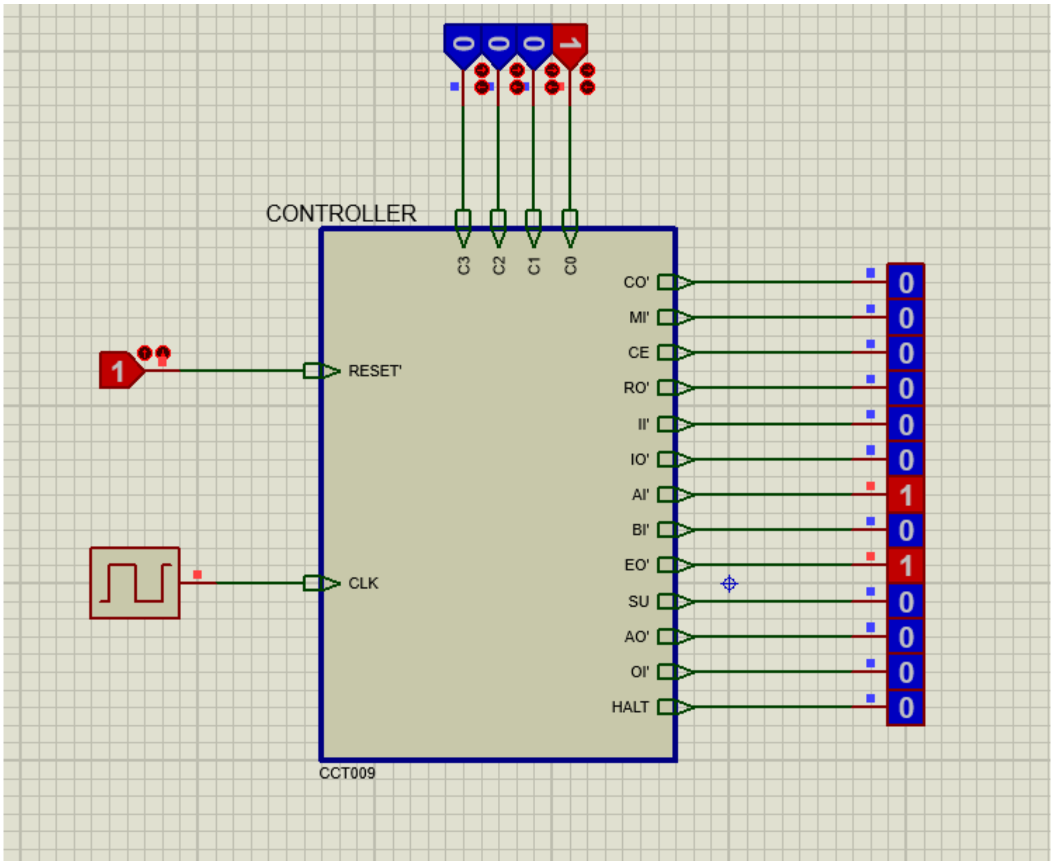


Fig: Parent Sheet and Output

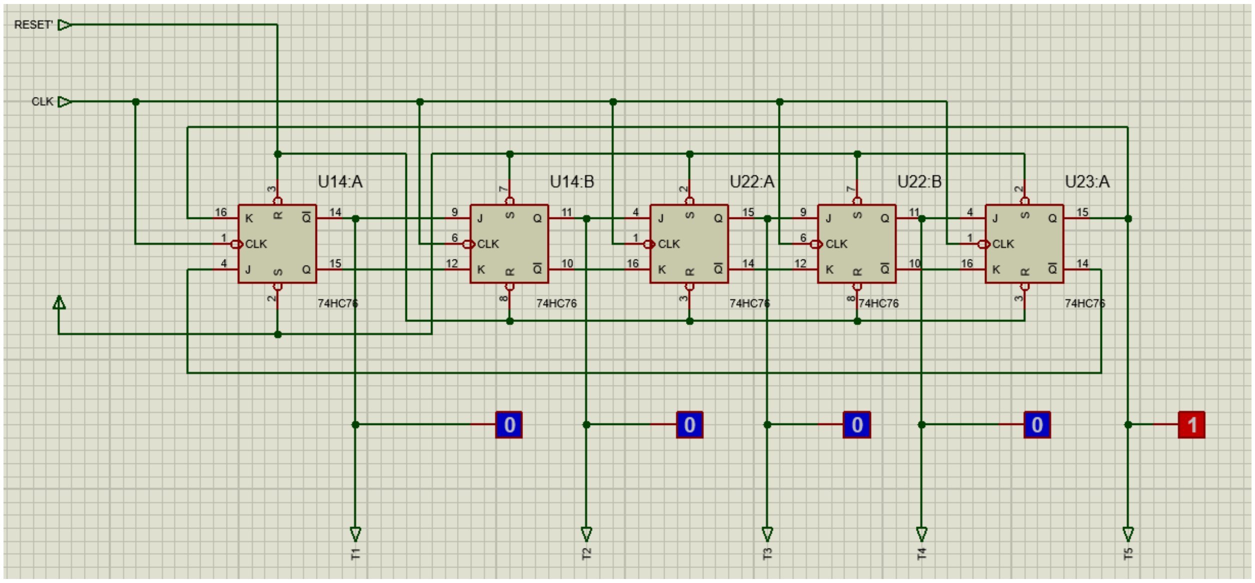


Fig: Child Sheet (T-state Generator)

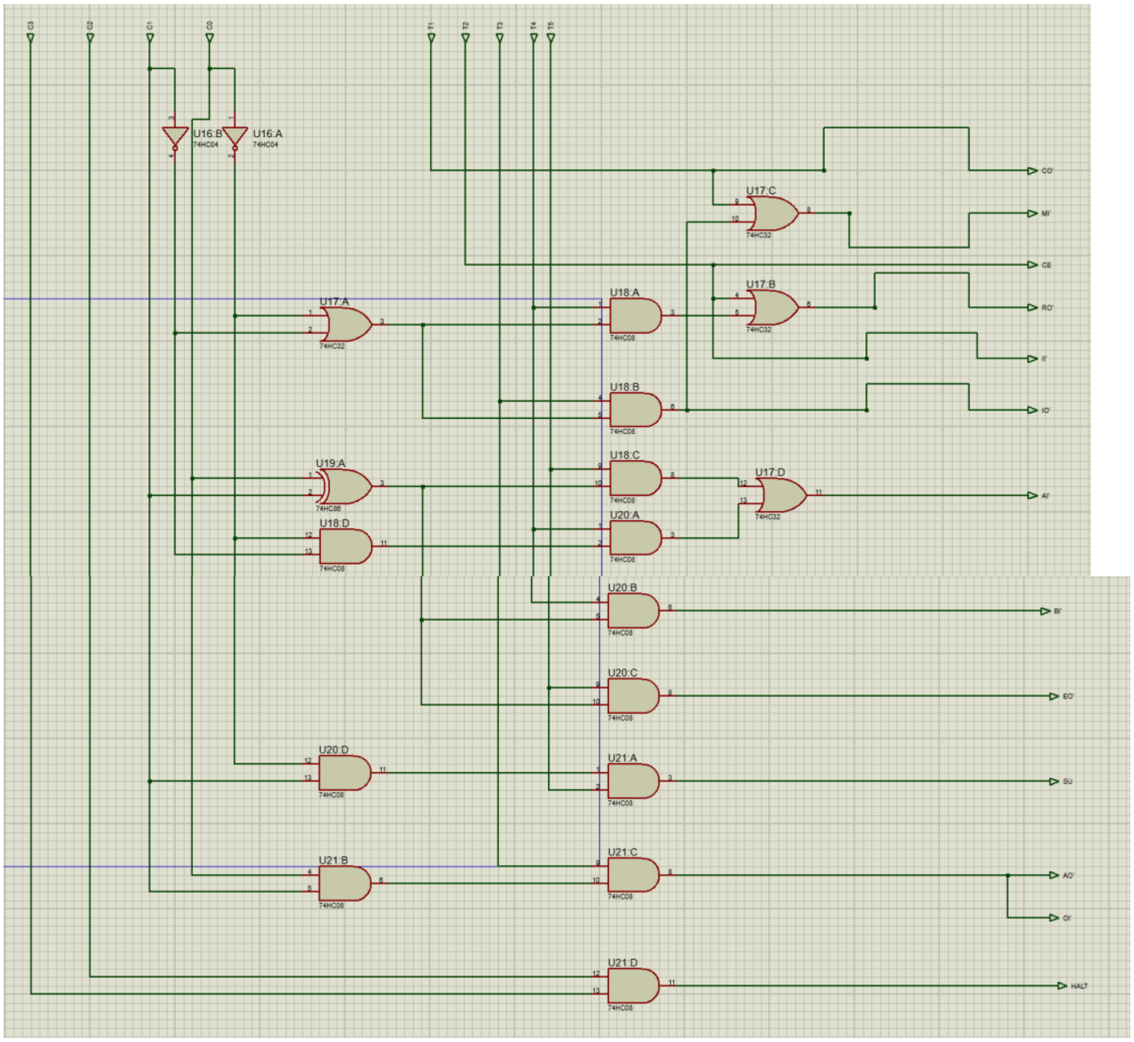


Fig: Child Sheet

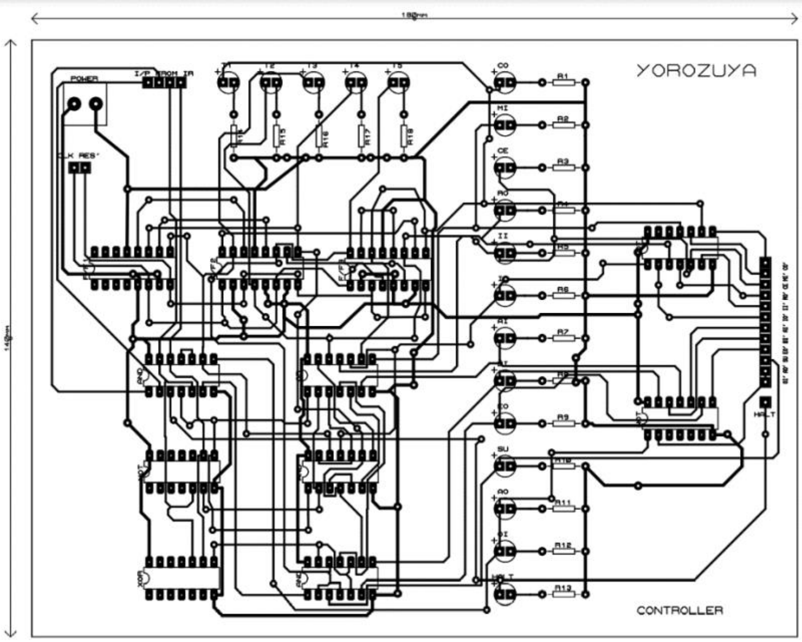
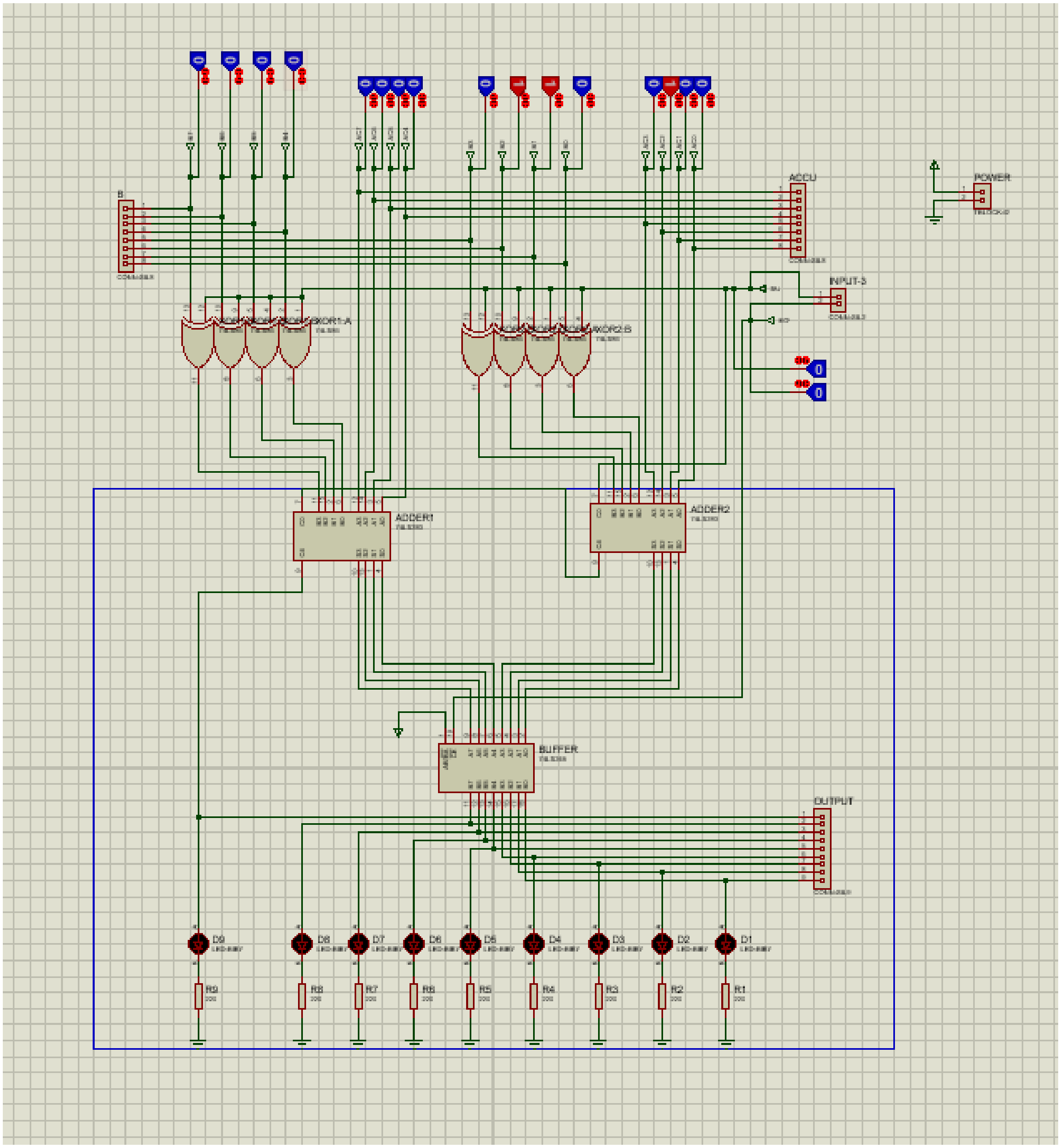


Fig: PCB

Phase D:

ALU:



Output Unit:

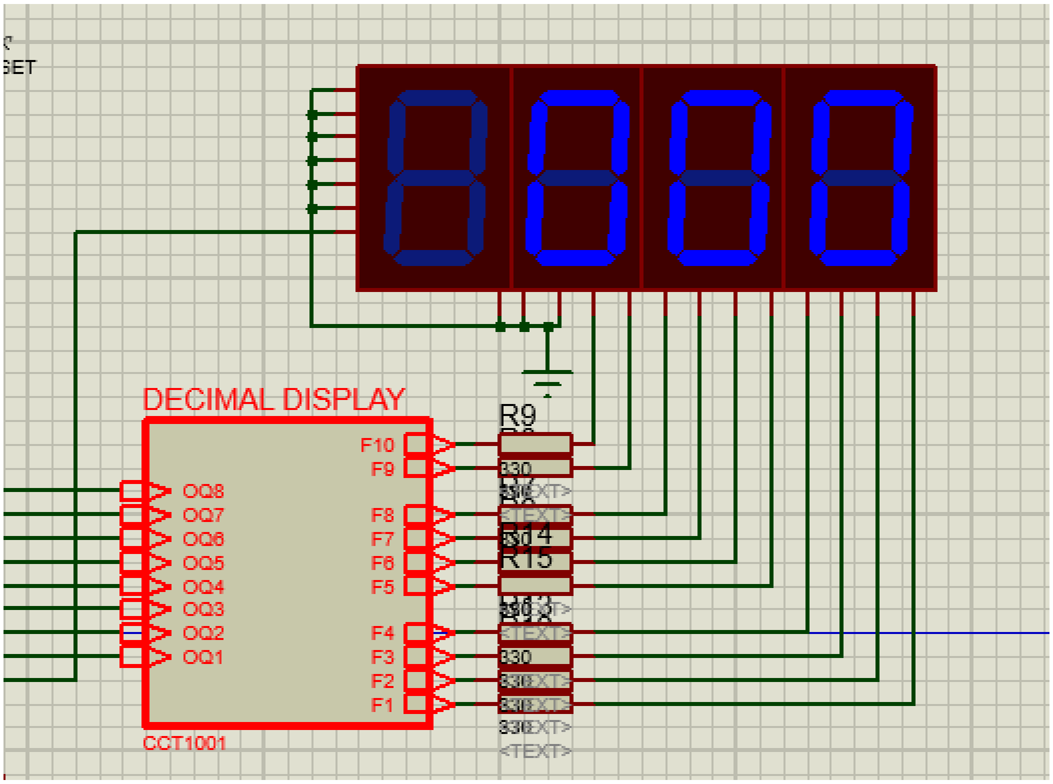


Fig- parent sheet

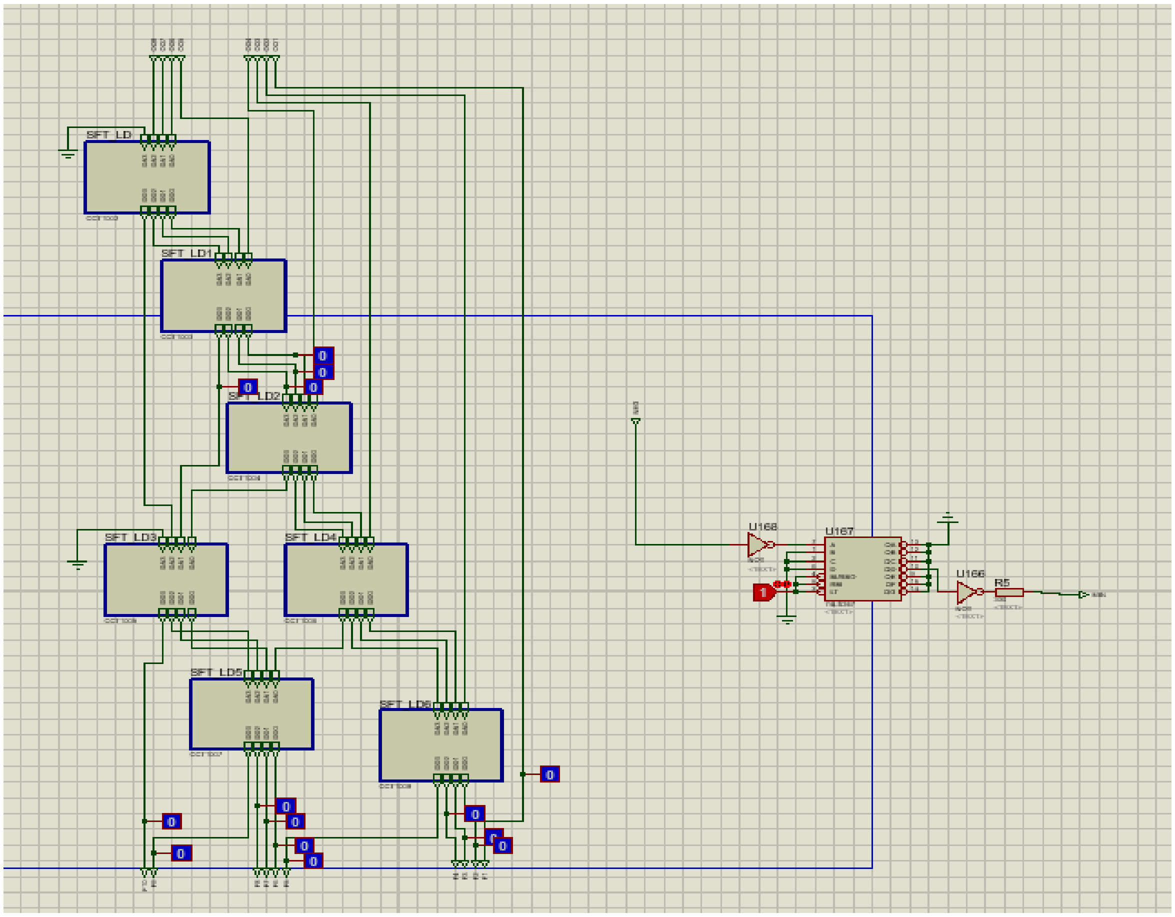


Fig- child sheet

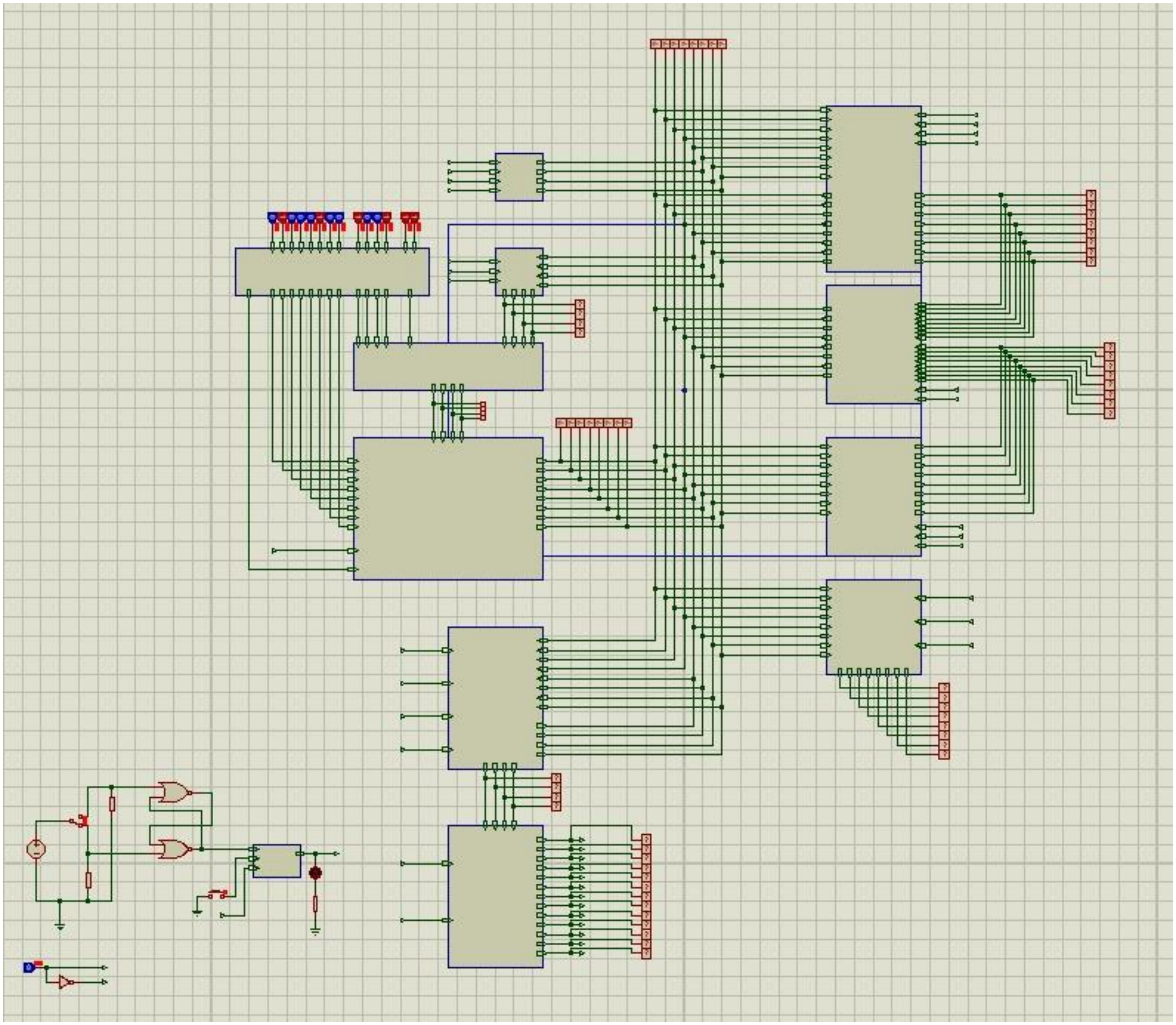
2.Integration with 8-bit Bus :

8-bit is 8 wires used for data transfer between modules. At a single time instance only one module will send data to the bus and only one module will receive the data. Two modules can never send the data to the bus at the same time. When a module is sending the data to the bus all other modules outputs must be disconnected.

2.Integration with 8-bit Bus :

8-bit is 8 wires used for data transfer between modules. At a single time instance only one module will send data to the bus and only one module will receive the data. Two modules can never send the data to the bus at the same time. When a module is sending the data to the bus all other modules outputs must be disconnected.

Project as a whole in software:

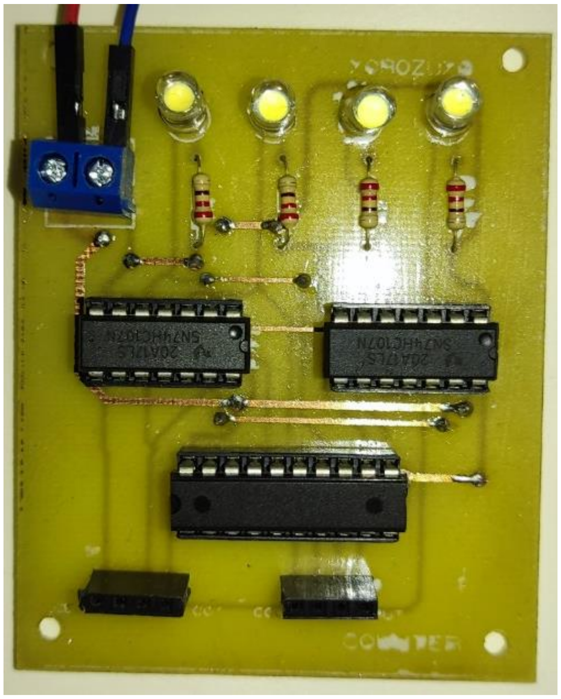


Hardware Implementation:

Phase A:

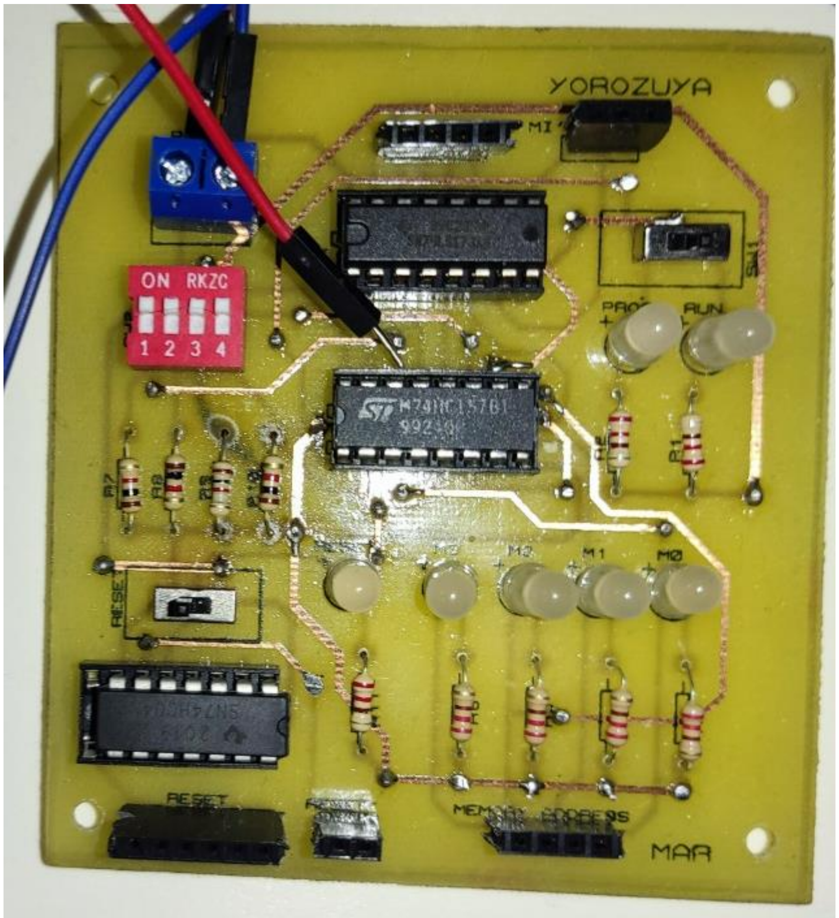
Clock pulse generator:

Counter:



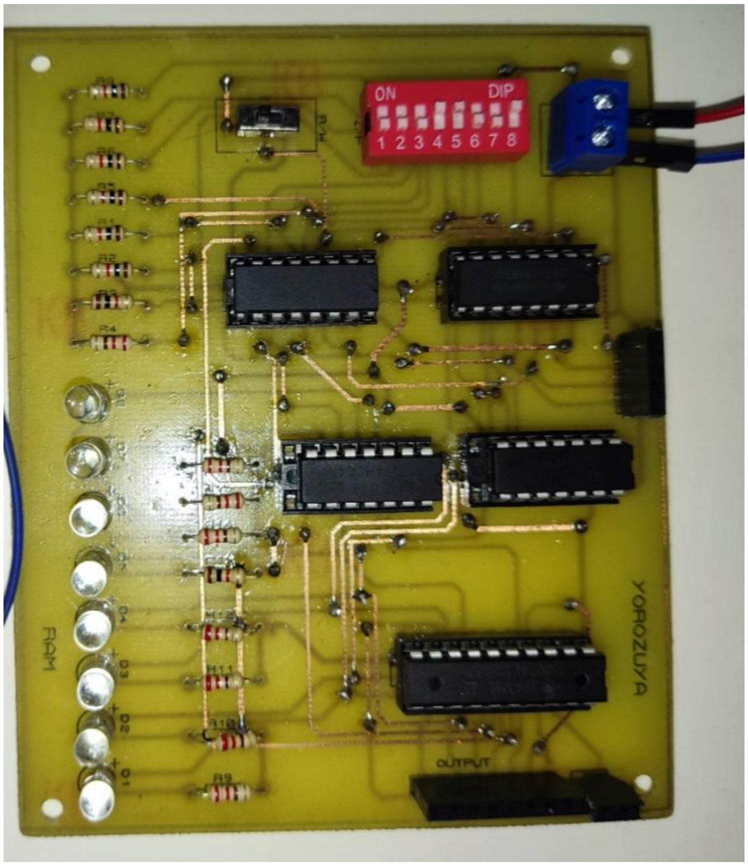
Input unit:

MAR:

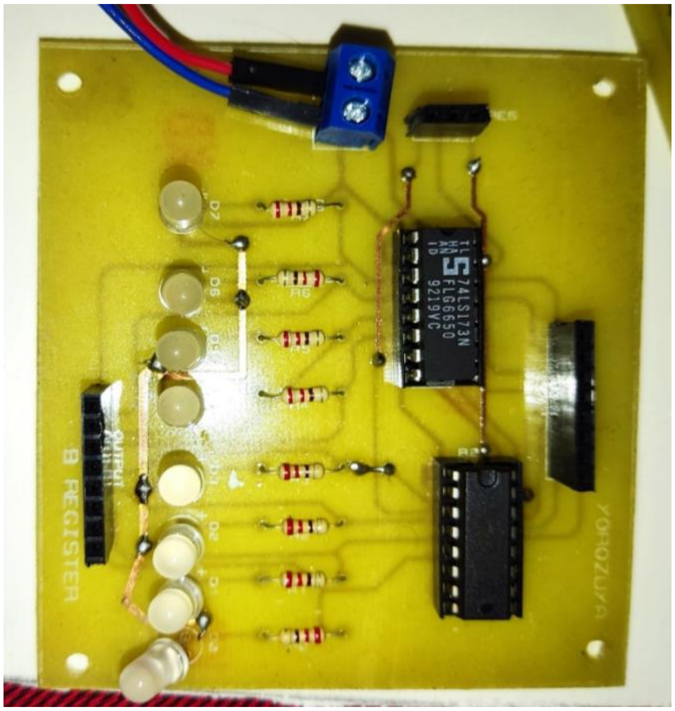

YOROZUYA
MI
SW1
ON RKZC
1 2 3 4
PRO
RUN
R1
R2
M74HC157B1
R7
R8
R9
R10
RESET
M2
M1
M0
RESET
MEMORY ADDRESS
MAR

Phase B:

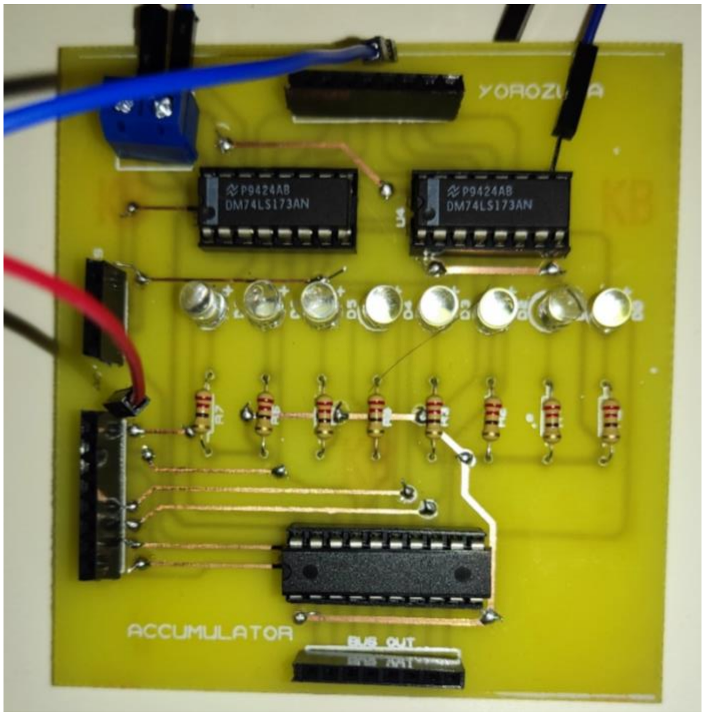
RAM:


ON
DIP
1 2 3 4 5 6 7 8
RAM
YOROZUYA
OUTPUT

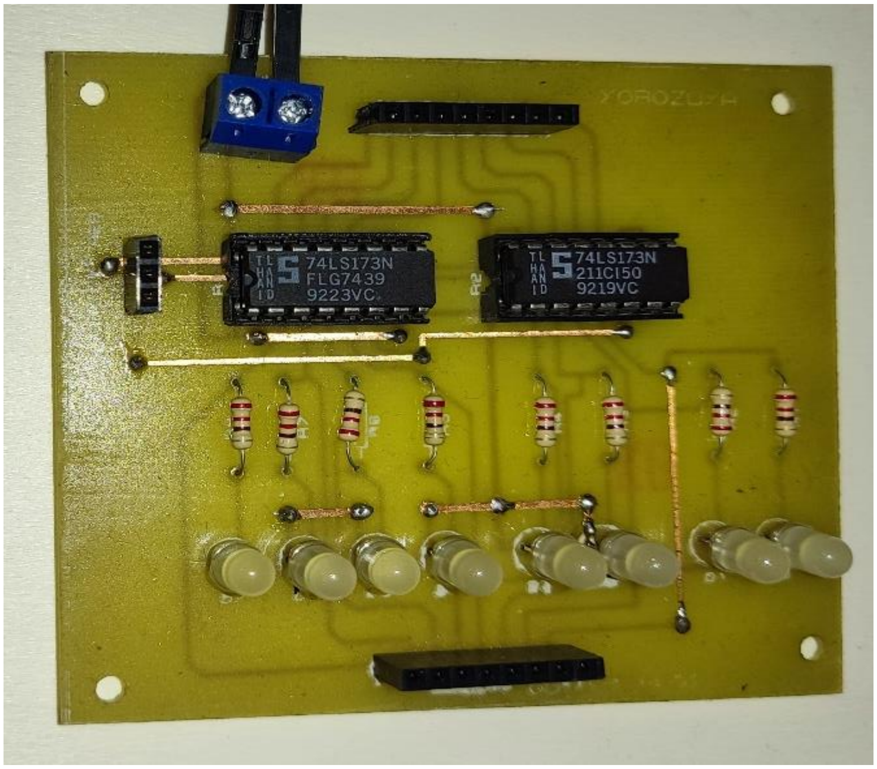
B Register:


74LS173N
B REGISTER
YOROZUYA

Accumulator:

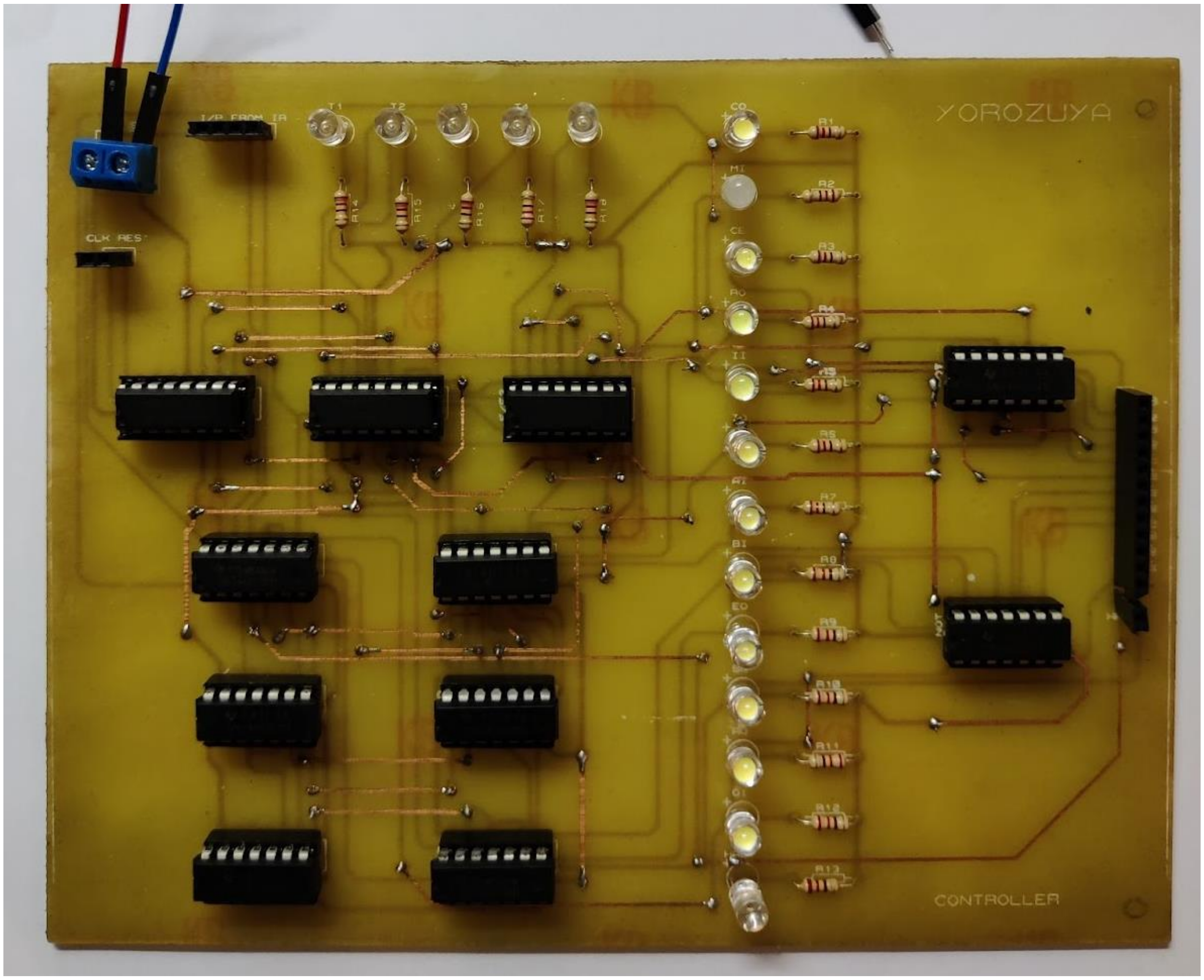


Output Register:

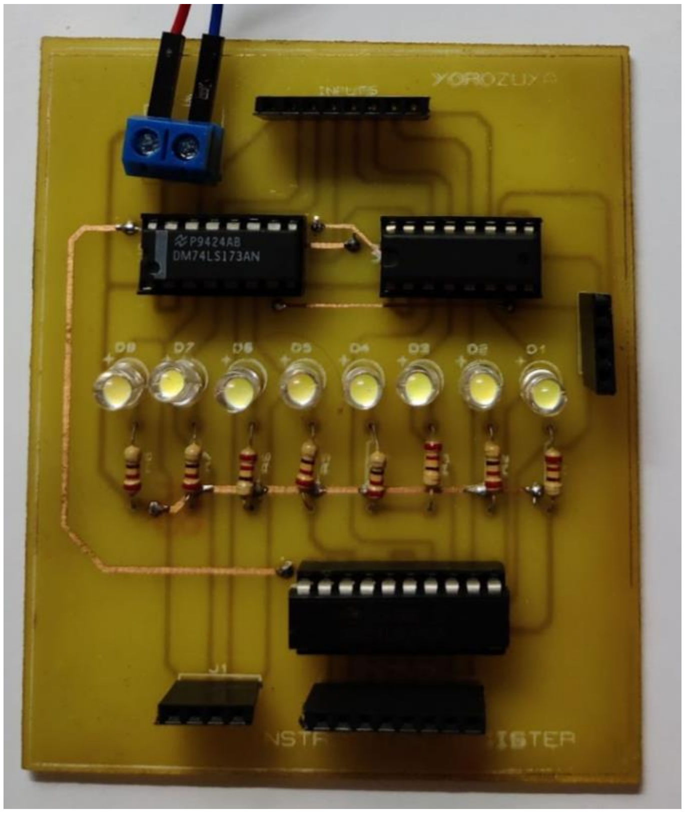


Phase C:

Controller Sequencer:

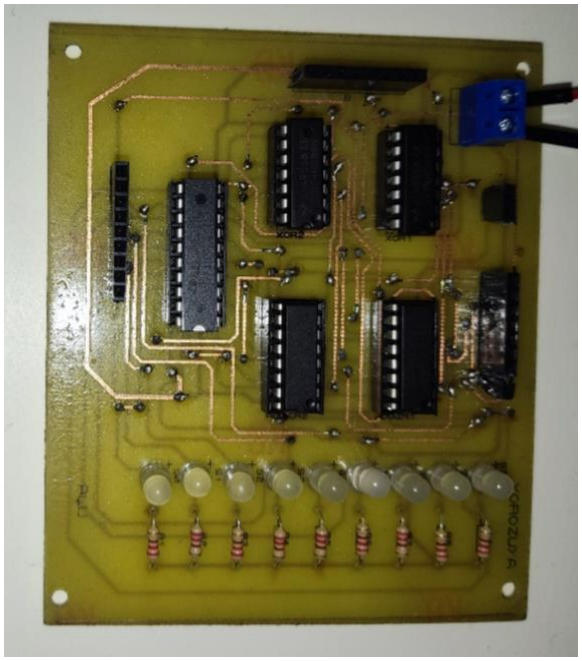


Instruction Register:



Phase D:

ALU:



Bus:

Logistics:

<u>Phase A:</u>

Used Components and Prices:

Breadboard - 155tk/pc

555 timer - 9tk/pc
Breadboard Power Supply - 100tk

12V Adapter - 120tk

74LS00 - 25tk/pc
74LS02 - 26tk/pc
74HC107 - 60tk/pc

74LS173 - 50tk/pc

LED - 5tk/pc

Resistors - 1tk/pc

Capacitors - 2tk/pc

8 pin DIP Switch- 30tk/pc

Push Button - 3tk/pc

SPDT Switch - 5tk/pc

Multimeter - 2000tk

Jumper Wires - 2tk/pc

As we had to travel to Uttara and Puran Dhaka to buy specific ICs some money were also spent for transportation. In total we had to spend around 3500-4000tk for phase-1.

<u>Phase B:</u>

Used Components and Prices:

Breadboard - 155tk/pc

74LS245 - 50tk/pc

74LS173 - 50tk/pc

74LS189 - 70tk/pc

74LS00 - 25tk/pc

LED - 5tk/pc

Resistors - 1tk/pc

8 pin DIP Switch- 30tk/pc

4 pin DIP Switch - 20tk/pc

Jumper Wires - 2tk/pc

As we had to travel to Uttara and Puran Dhaka to buy specific ICs some money were also spent for transportation. In total we had to spend around 1200tk for phase-b.

Phase C:

Used Components and Prices:

74LS245 - 50tk/pc

74LS173 - 50tk/pc

74HC04- 15 TK/pc

74HC08- 15 TK/pc

74HC32- 15 TK/pc

74HC76-40 tk/pc

LED - 5tk/pc

Resistors - 1tk/pc

Female header- 8 tk/pc

T-block- 4 tk/pc

Jumper Wires - 2tk/pc

PCB:

Controller- 535 tk

Instruction Register- 255 tk

As we had to travel to Uttara and Puran Dhaka to buy specific ICs some money were also spent for transportation.

Phase D:

Discussion:

In this project we implemented 8 bit computer based on the architecture of SAP 1. We had to face many challenges to complete the whole project. At the beginning theory related to this project wasn't taught in class and so we faced difficulties to understand the topic. Also We faced a lot of problems finding the correct ICs for our project. we had to travel to Puran Dhaka and other places in search of these components. But even after traveling this much we were unable to find certain ICs so we had to mix LS and HC series ICs. Some of the ICs we bought were also faulty and we had no way of testing these beforehand. So our workload increased.

When we ordered pcb , it took much time to deliver and that was problematic too as we had to finish before the deadline. At first we ordered 4 pcbs and took 4 days to deliver but next time when we ordered it took 1 week. Our two pcbs of clock didn't work properly one for our designing problem other for unknown reason that we couldn't debug. All other pcbs work properly after soldering.

After completing two phases, semester final examination was knocking at the door and we worked for this project amidst huge academic pressure. This hampered our mental health.

This project is the reason of many things. Things I can't describe. So we hate this project and didn't enjoy while doing it because of the huge academic pressure. We are human not a machine.

After passing all these days, all these ups and downs, finally it comes to and end. Surely things before this project and after this project aren't same.