Ahsanullah University of Science and Technology (AUST)

Department of Computer Science and Engineering

Laboratory Manual

Course No: CSE 3110

Course Title: Digital System Design Lab

For the students of 3rd year 1st semester of

B.Sc. in Computer Science and Engineering Program

TABLE of CONTENTS

COURSE OUTCOMES

➔ Explain the basic concepts of digital system design.

➔ Design combinational, sequential and complex digital systems.

➔ Apply modern tools, open-source hardware and software platforms, design methodologies for developing digital systems.

➔ Practice safety norms, anti-littering behavior, maximizing energy efficiency and minimizing environmental impact during the design and development of the digital systems.

➔ Construct both software and hardware implementations of a digital system incorporating memory modules as a project in the multidisciplinary context for sustainable social and economic development in Bangladesh and around the globe.

➔ Participate actively in all project development phases and communicate effectively as a team member or leader.

➔ Present the project to internal and external project examiners utilizing a multimedia system and produce a comprehensive report.

➔ Estimate initial budget for required equipment, maintain the estimated budget, make final budget after project submission.

PREFERRED TOOL(S)

✔ Proteus Design Suite
✔ DC Power Supply
✔ Breadboard
✔ Pulse Generator
✔ Digital Probe
✔ Different Digital Integrated ICs

TEXT/REFERENCE BOOK(S)

1. Digital Logic and Computer Design (Indian Edition) - M. Morris Mano
2. Digital Computer Electronics (3rd Edition) – Albert Paul Malvino
3. Handbook, Microprocessor Data – BPB Publications, New Delhi, (1992): 39

ADMINISTRATIVE POLICY OF THE LABORATORY

1) You are expected to conduct yourself professionally, and keep your working area neat and clean.
2) You are required to return all the equipment and parts used in the experiment to their proper places before you leave the lab.
3) You are expected to work in a group consisting of at most four people.
4) Viva for each experiment will be taken.
5) Plagiarism is strictly forbidden and will be dealt with punishment.

Session 1 and Session 2: Simulation and Implementation of Arithmetic Logic Unit

Objectives:

The objective of session 1 is to simulate a 4-bit ALU using 4-bit full adder and also implement the "Carry", "Overflow", "Sign", and "Zero" flag. The ALU should perform according to the selector combination given to you. All possible function combinations are given here.

Functions	Explanation
A	Transfer **A**
A + 1	Increment **A**
A - 1	Decrement **A**
A + B	Sum **A** and **B**
A - B	Subtract **B** from **A**
A + B + 1	Sum **A** and **B** with carry
A - B - 1	Subtract **B** from **A** with borrow
A . B	Logical 'AND' of **A** and **B**
A|B	Logical 'OR' of **A** and **B**
A ⊕ B	Logical 'XOR' of **A** and **B**
A '	Logical 'NOT' of **A** and **B**

The objective of session 2 is to implement the circuit designed in Session 1.

Submission Deadline:

1. Design and Simulation – 2nd Class
2. Physical Implementation – 3rd Class

Design Guideline

The possible solution of the given selector combination for the functions is shown here. The process will be explained in detail in the class.

First you have to make the function table as:

S1	S0	Cin	Output	X	Y	Z
0	0	0	A+1	A	0	1
0	0	1	A	A	0	0
0	1	0	A+B+1	A	B	1
0	1	1	A+B	A	B	0
1	0	X	AB	AB	0	X
1	1	X	A'	A'	0	X

Then you have to find the equations of X, Y and Z:

$X = S1'A + S1S0'(AB) + S1S0A'$
$Y = S1'S0B$
$Z = S1'S0'Cin' + S1'S0Cin'$
   $= S1'Cin'$

A block diagram of this circuit is shown in Figure 1.

S1 $\Rightarrow$
S0 $\Rightarrow$
Cin $\Rightarrow$
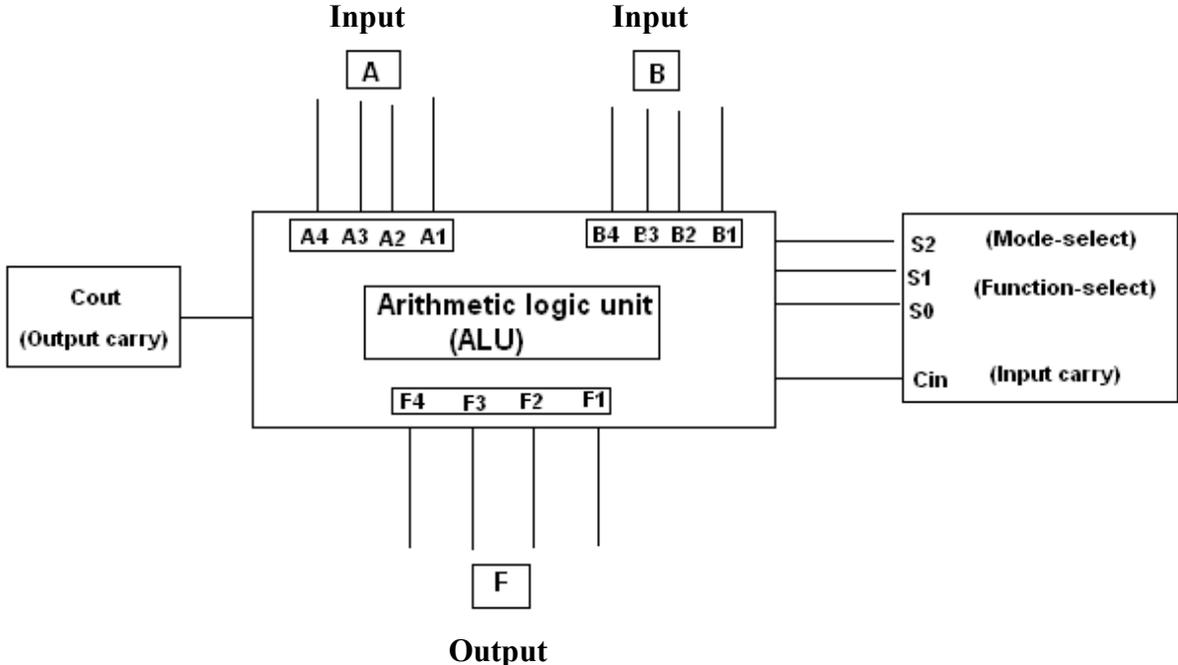**A** $\Rightarrow$
B $\Rightarrow$

Figure 1: Block Diagram of ALU

The next step is to build this circuit in Proteus and after successful simulation you have to implement it in hardware.

Session 3 and Session 4: Simulation and Implementation of Booth Multiplier

Objectives:

The objective of session 3 is to simulate a 5-bit Booth's Multiplier in Proteus.

The objective of session 4 is to implement the circuit designed in session 3.

Submission Deadline:

1. Design and Simulation – 4th Class
2. Physical Implementation – 5th Class

Booth's Multiplication Algorithm:

Booth's algorithm can be implemented by repeatedly adding one of two predetermined values **A** and **S** to a product **P**, then performing a rightward arithmetic shift on **P**.

Let m and n be the multiplicand and multiplier, respectively;and let x and y represent the number of bits in **m** and **r**.

1. Determine the values of A and S,and the initial value of P. All these numbers should have a length equal to (x+y+1).

    i) A: Fill the most significant (leftmost)bits with the value of m. Fill the remaining(y+1)
    ii) S: Fill the most significant (leftmost) bits with the value of (-m) in two's complement notation. Fill the remaining (y+1) bits with zeros.
    iii) P: Fill the most significant x bits with zeros. To the right of this append the value of r.Fill the least significant (rightmost) bit with a zero.

2. Determine operation according to the two least significant (rightmost) bits of P.
    i) If they are **01**, find the value of P+A. Ignore any overflow.
    ii) If they are **10**, find the value of P+S. Ignore any overflow.
    iii) If they are **00**, do nothing; use P directly in the next step.
    iii) If they are **11**, do nothing; use P directly in the next step.

3. Arithmetically shift the value obtained in the 2nd step by a single place to the right. Let P now equal this new value.

4. Repeat steps 2 and 3 until they have been done y times.

5. Drop the least significant (rightmost) bit from P. This is the product of m and r.

Examples on Booth's Multiplication algorithm will be shown in the class in detail.

5x5 Bit Booth's Multiplier Design

Initialization:

$U \leftarrow 0$ [Partial Product most significant bits]
$V \leftarrow 0$ [Partial Product least significant bits]
$X \leftarrow$ Input [Multiplicand]
$Y \leftarrow$ Input [Multiplier]
$X_{-1} \leftarrow$ [Extra bit for Calculation]
Count $\leftarrow 0$ [Counter]

Algorithm:

Do the following operations depending on the value of last bit (least significant) of X and X_{-1}

      i) If they are **01**, find the value of U-Y.Ignore any overflow.

      ii) If they are **10**, find the value of U+Y. Ignore any overflow.

      iii) If they are **00**, find the value of U+0.

      iii) If they are **11**, find the value of U+0.

Flow Chart:
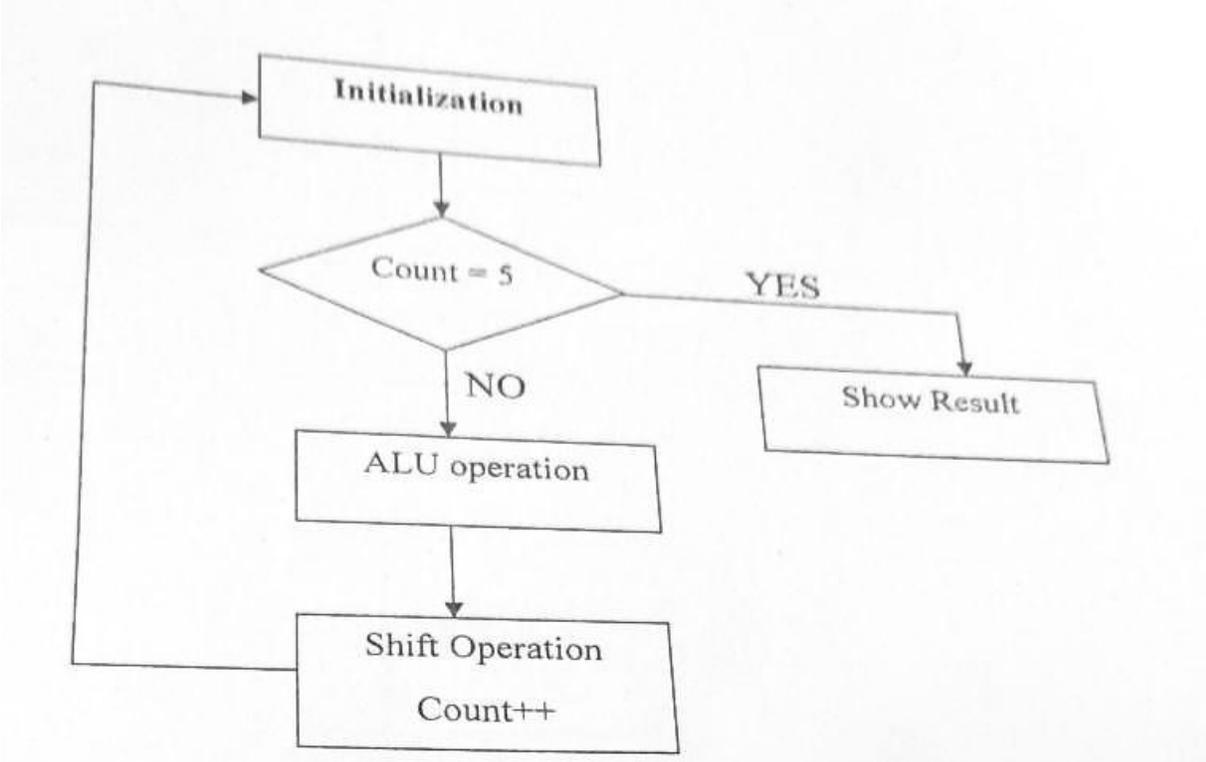
Figure 2: Flow Diagram for Booth's Multiplier Design

State Diagram:
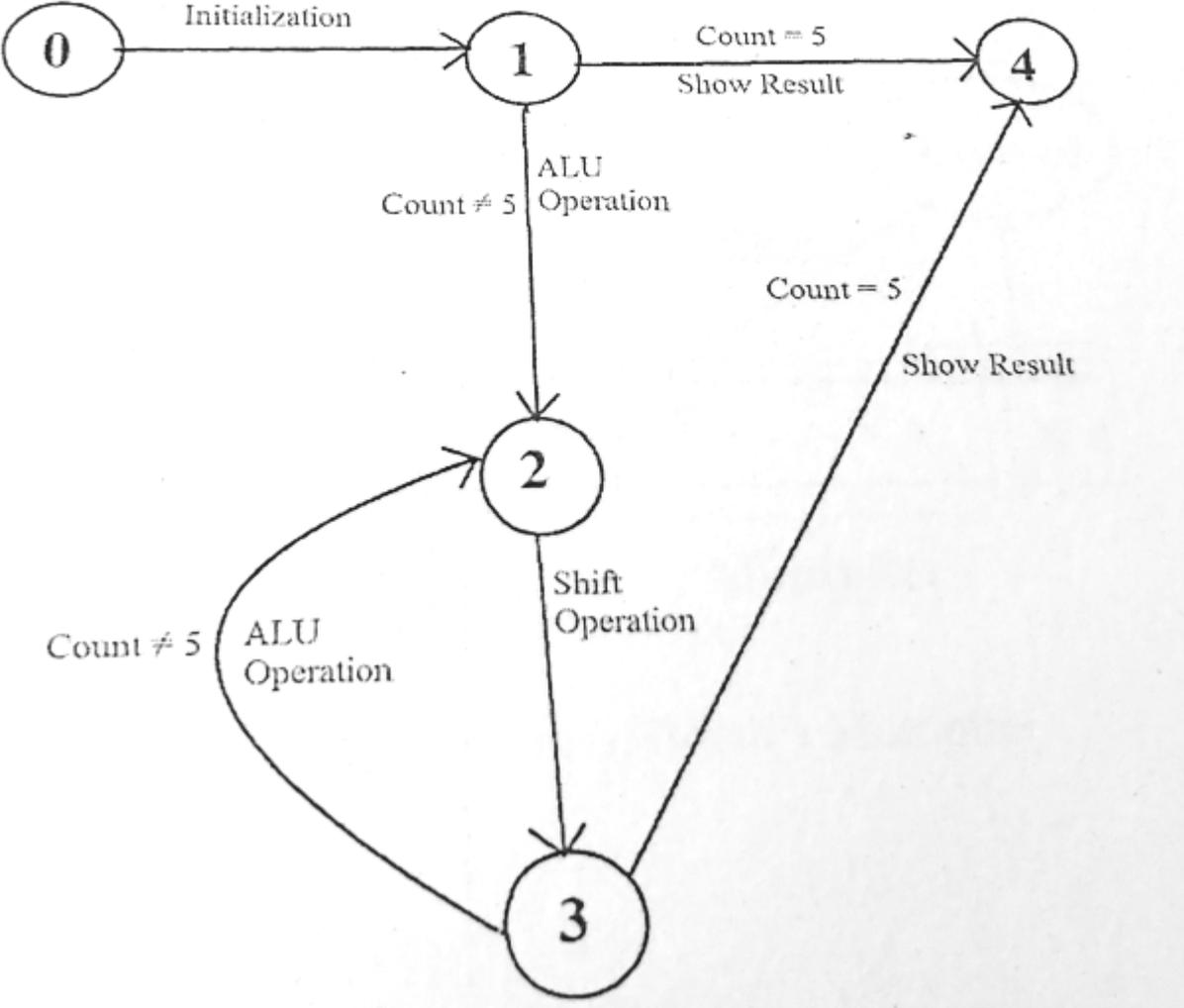
Figure 3: State Diagram for Booth's Multiplier Design

ALU Operations:

S1	S0	Operation
0	0	U + 0
0	1	U + Y
1	0	U - Y
1	1	U + 0

Architecture:
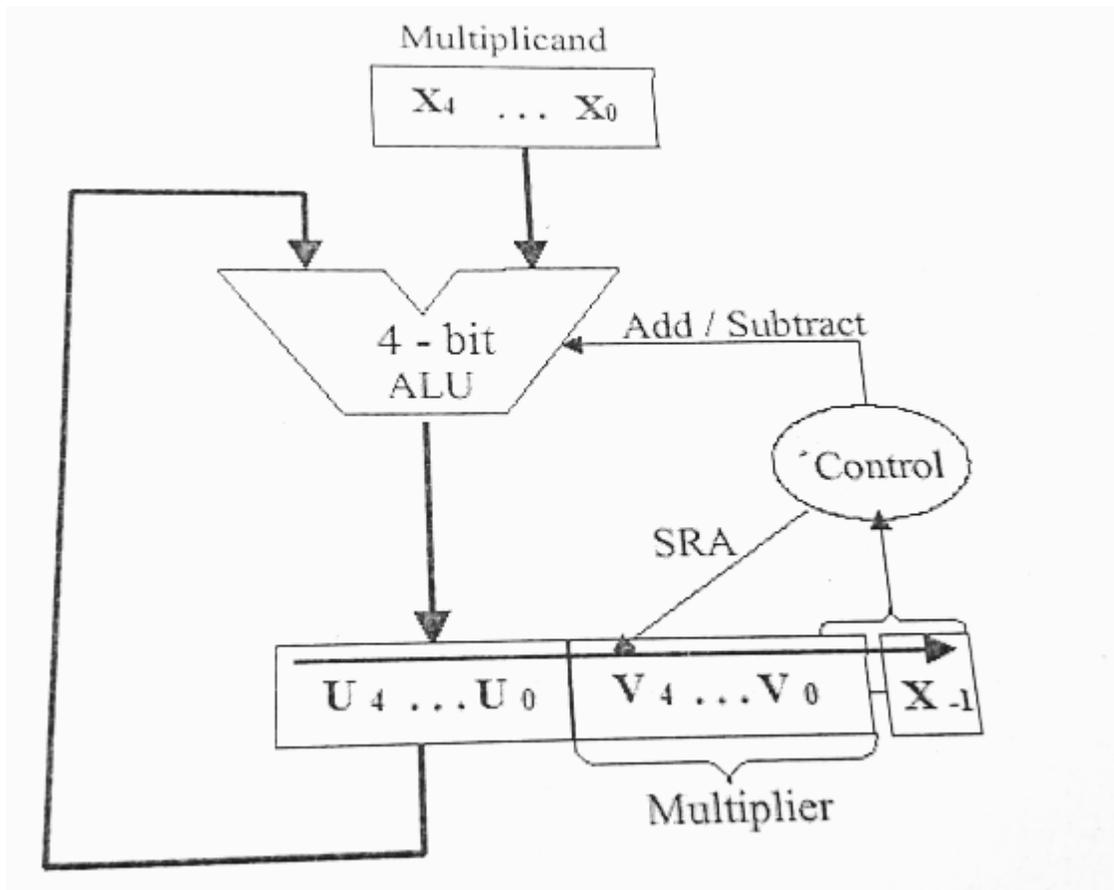
Figure 4: Architecture of Booth's Multiplier

Session 5 and Session 6: Simulation of Microprocessor (SAP-1)

Objective:

The objective of session 5 is to design SAP-1 (a simple PC) partially; which means you have to design different blocks (i.e. PC, MAR, IR etc.) of SAP-l architecture individually (i.e.without any address bus and data bus connection) in Proteus.

The objective of session 6 is to show the complete simulation of SAP-1 circuit in Proteus.

Submission Deadline:
1. Update of the Proteus Simulation-6th Class
2. Final Submission of Proteus Simulation-7th Class

Design Guideline

Basic design is provided here. It will be explained elaborately in the class.

SAP-1 Architecture

$$C_P \cdot E_P \cdot \overline{L}_M \cdot \overline{CE} \cdot \overline{L}_I \cdot \overline{E}_I \cdot \overline{L}_A \cdot E_A \cdot S_U \cdot E_U \cdot \overline{L}_B \cdot \overline{L}_O$$
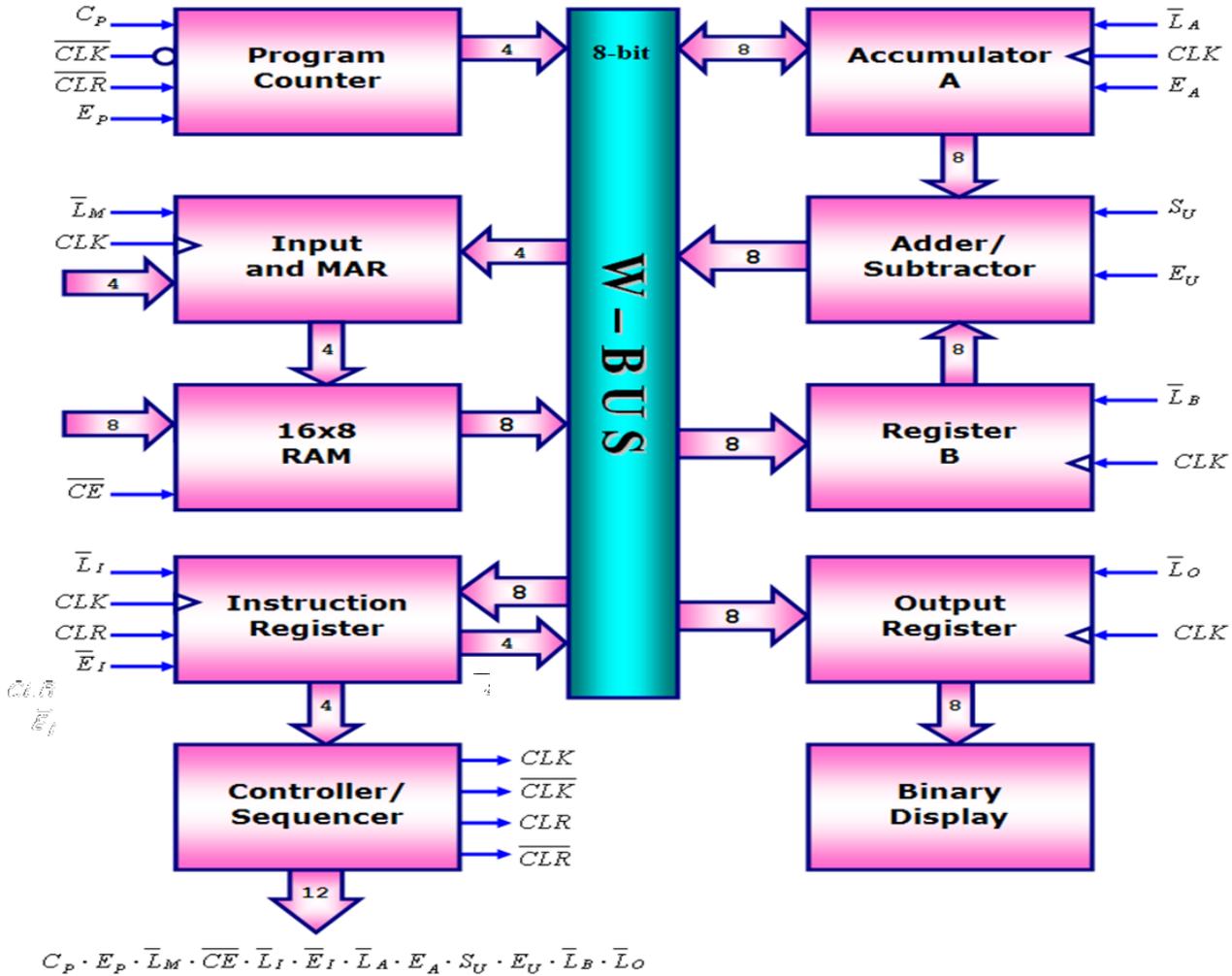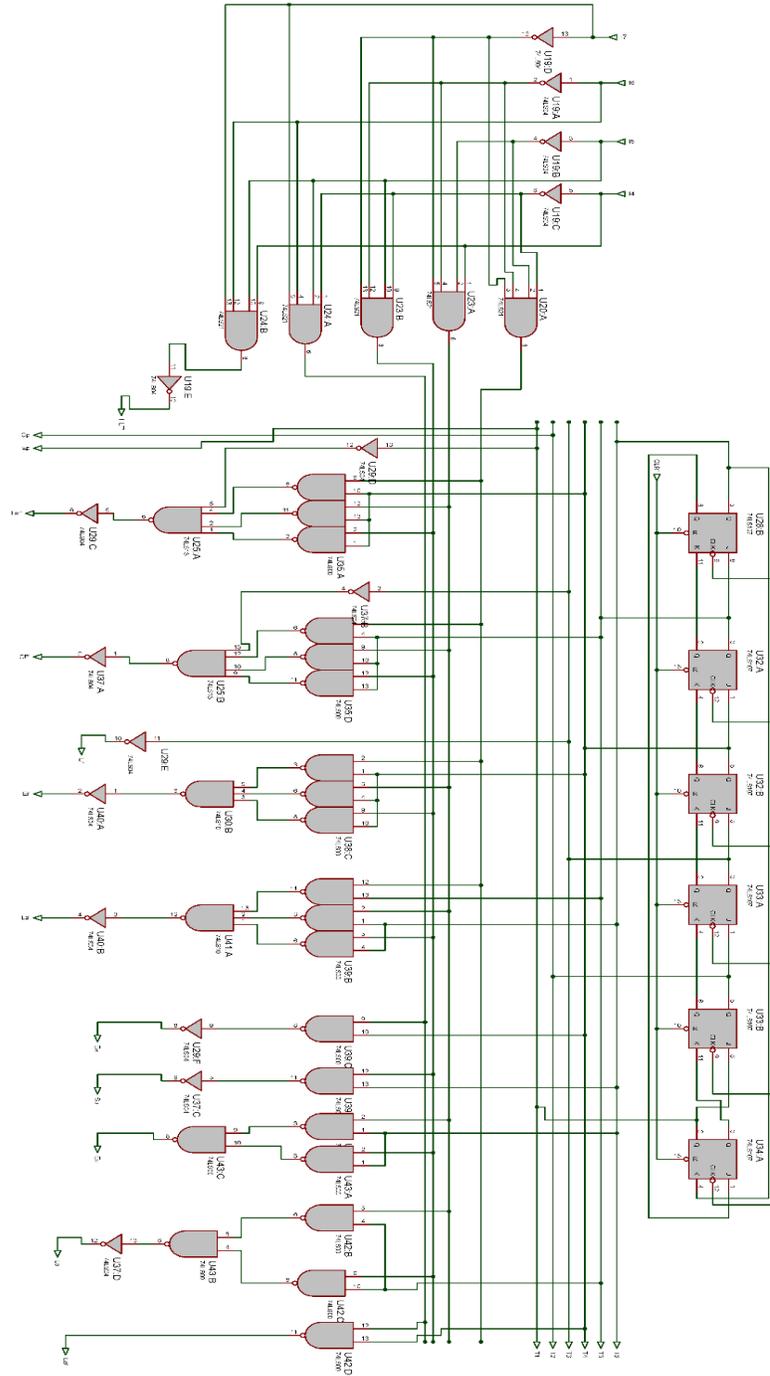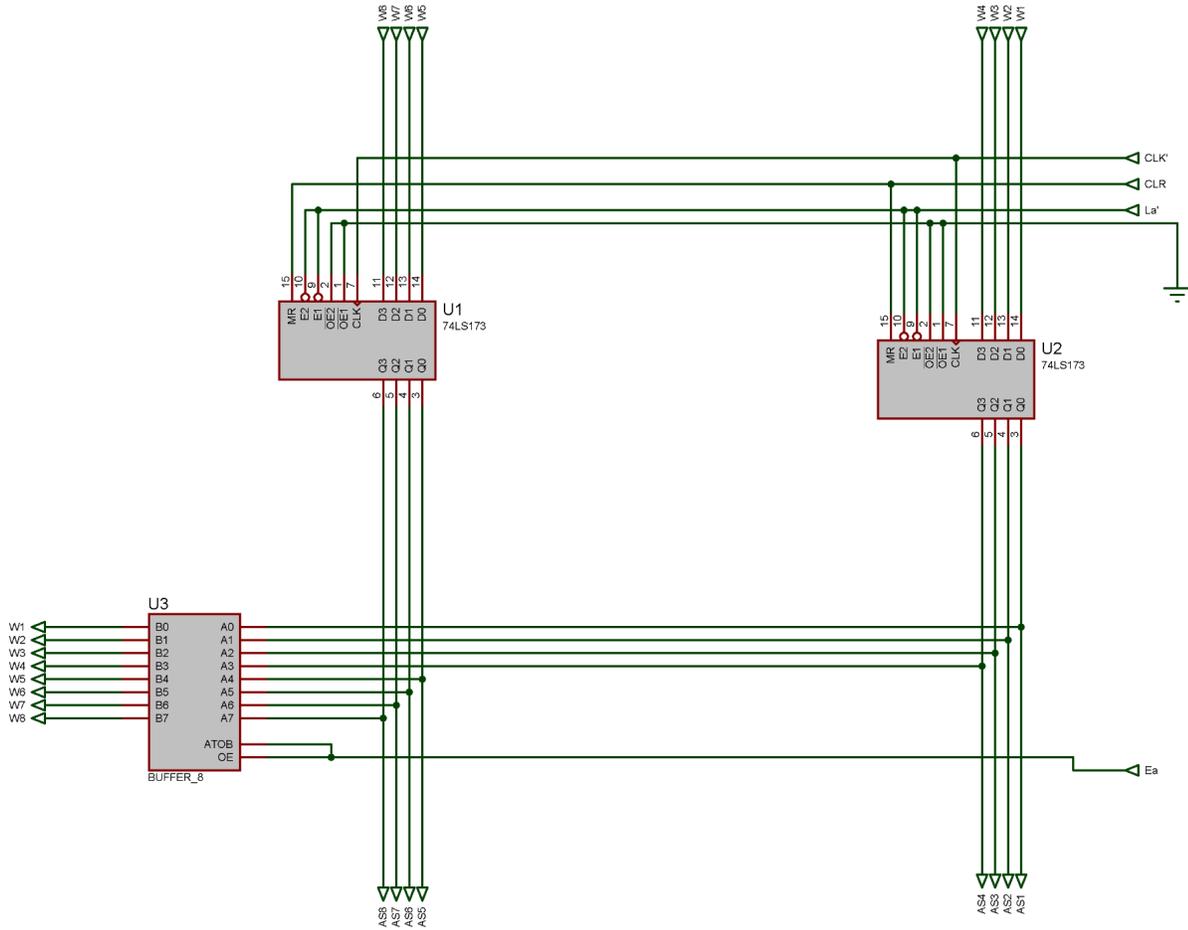
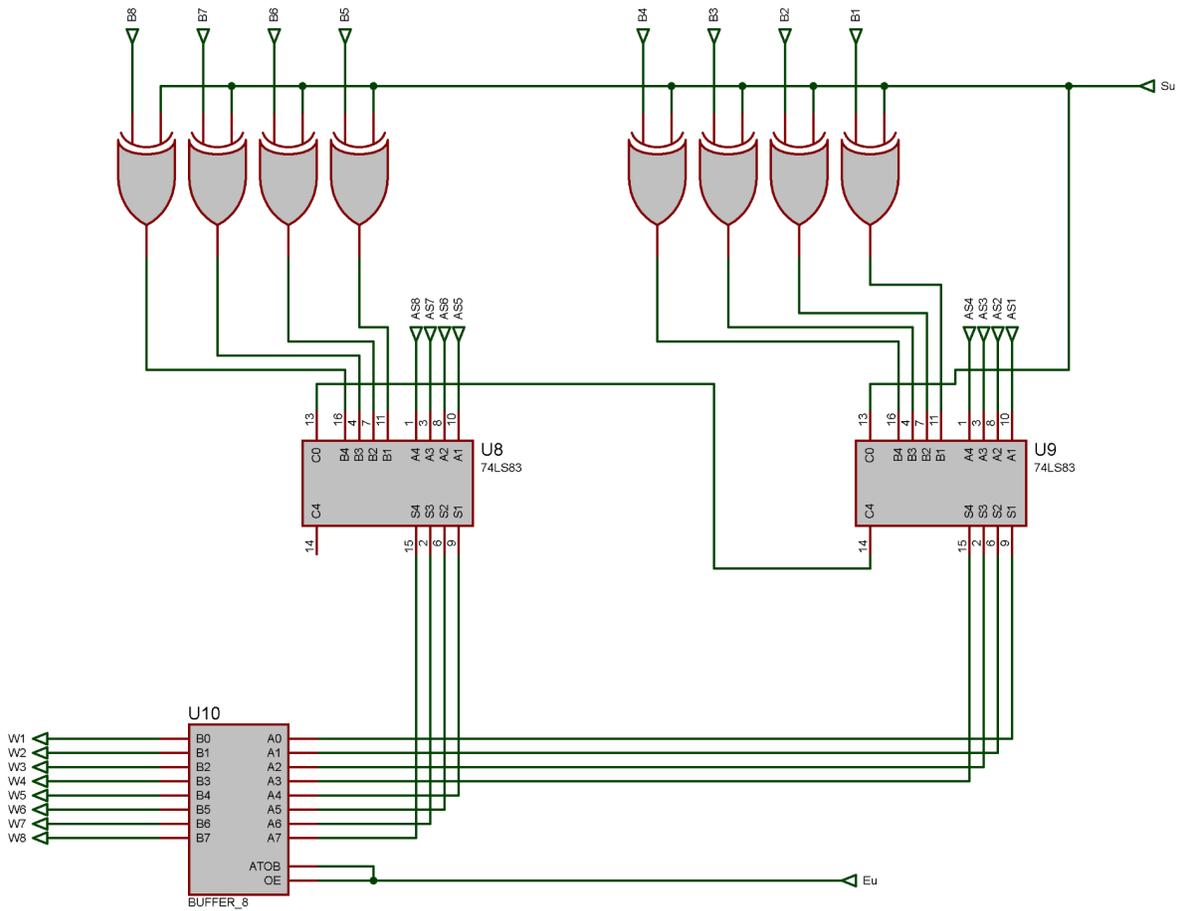# Program Counter:
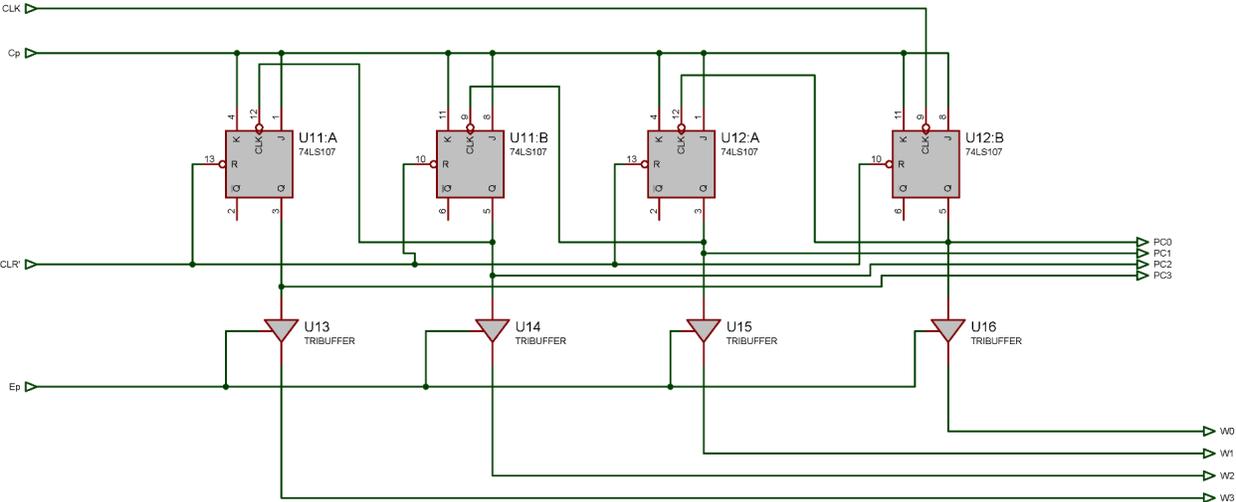
MAR and RAM:
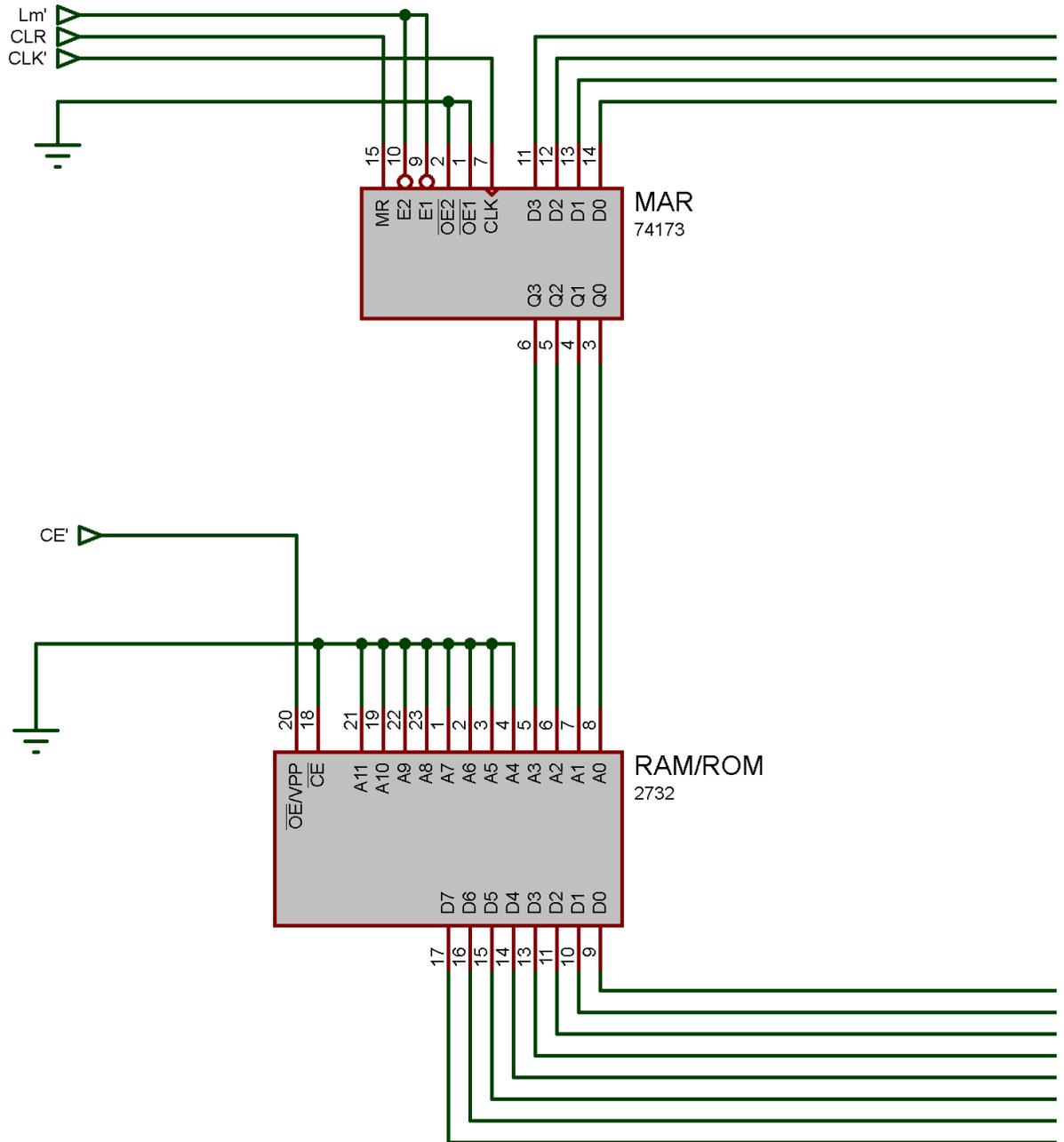
Instruction Register:
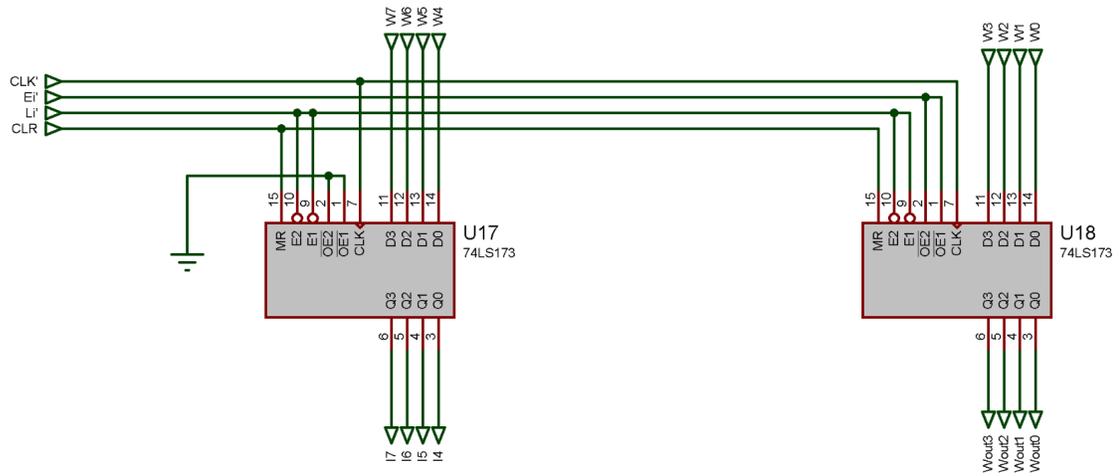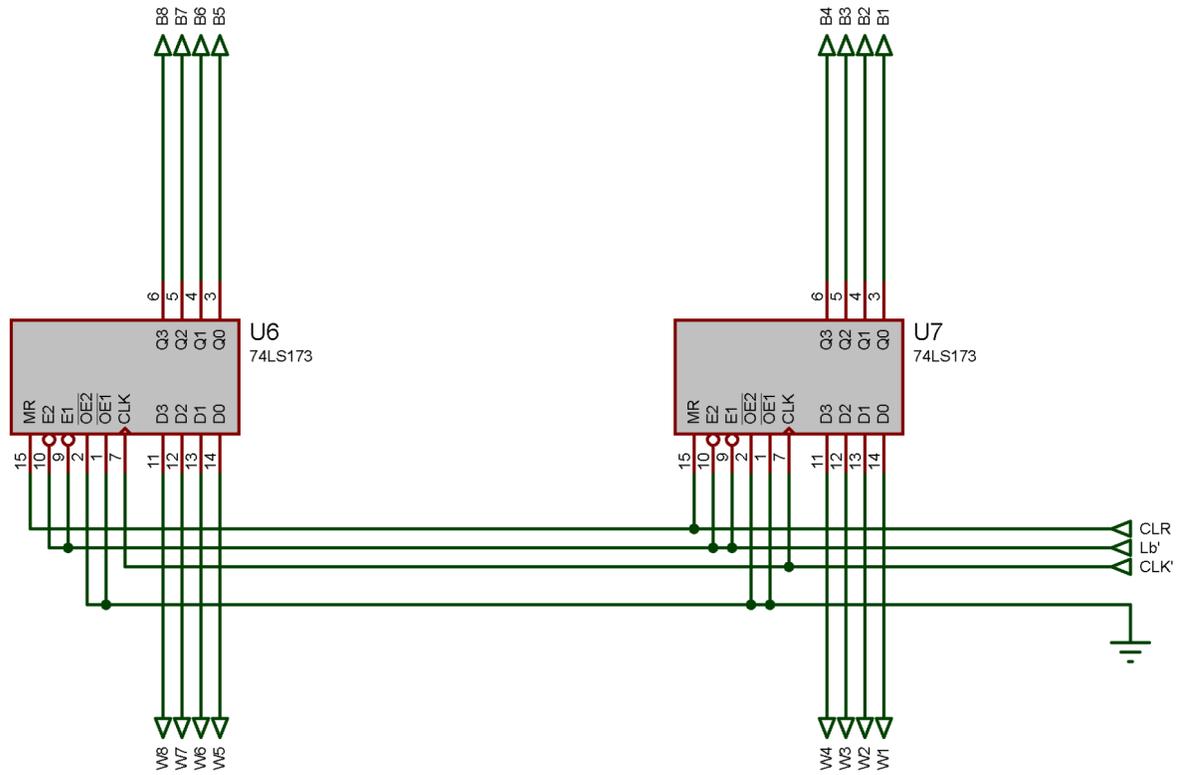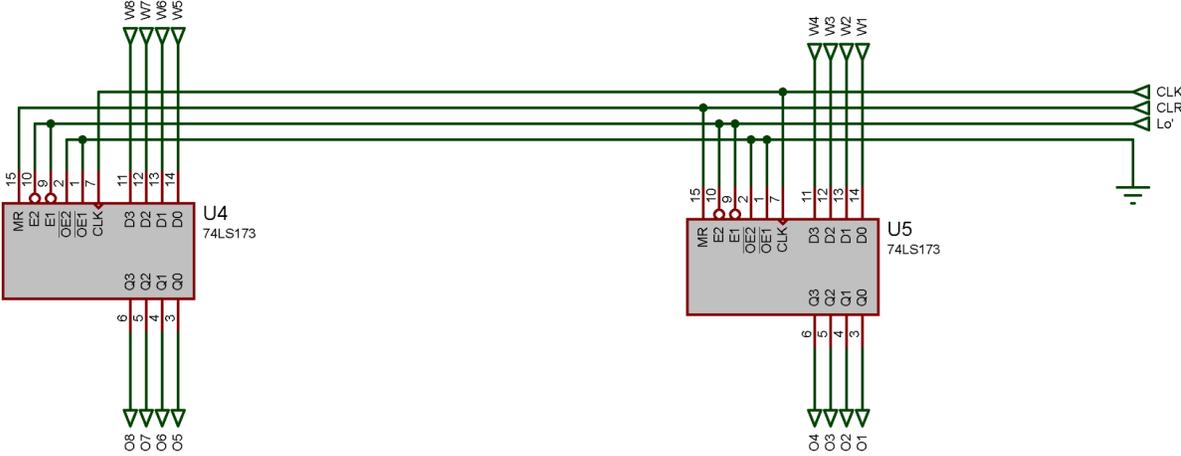
Controller/Sequencer:

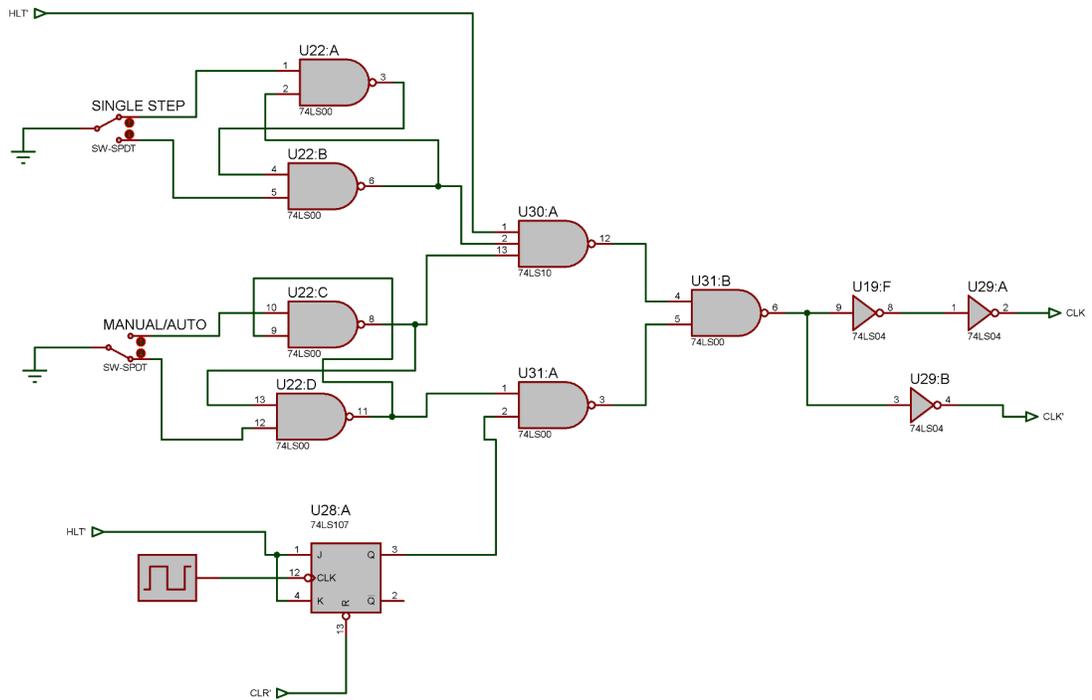# Accumulator:

Adder/Subtractor:

B Register:

Output Register:

Clock Circuit:

Lab Final Examination

There will be a 40 minute written examination of 20 marks. There will be 6 questions on the basic topics of Arithmetic Logic Unit, Booth's Multiplier and SAP-1.