Ahsanullah University of Science and Technology (AUST)
Department of Computer Science and Engineering

LABORATORY MANUAL

Course No. : CSE3110
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 OBJECTIVES

The main objectives of Digital System Design lab are control logic design, micro-program control design and microprocessor design. This lab is designed to teach students how to design Arithmetic Logic unit (ALU), Booth's Multiplier and a Microprocessor (SAP-1). Each design is done in two phases. In the first phase the students have to simulate the circuit in Proteus and in the second phase they have to implement it with hardware.

PREFFERED 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" written by "M. Morris Mano". Publisher: Pearson Education India, 2014.

2. "Digital Computer Electronics" written by "Albert Paul Malvino and Jerald A. Brown". Publisher: Gregg Division, McGraw-Hill, 1977.

3. Handbook, Microprocessor Data. "BPB Publications." New Delhi, (1992): 39.

ADMINISTRATIVE POLICY OF THE LABORATORY

1) You are expected to conduct yourself professionally, and to keep your working area neat and clean.

2) You are required to return all equipments and parts used in the experiment to its proper places before you leave the lab.

3) You are expected to work in a group consists of at most three 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 combination is 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
AB	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**

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 showed here. The process will be explained in details in the class.

First you have to make the function table as:

S_1	S_0	C_{in}	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:

$X = S_1'A + S_1S_0'(AB) + S_1S_0A'$

$Y = S_1'S_0B$

$Z = S_1'S_0'C_{in}' + S_1'S_0C_{in}'$

$= S_1'C_{in}'$

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

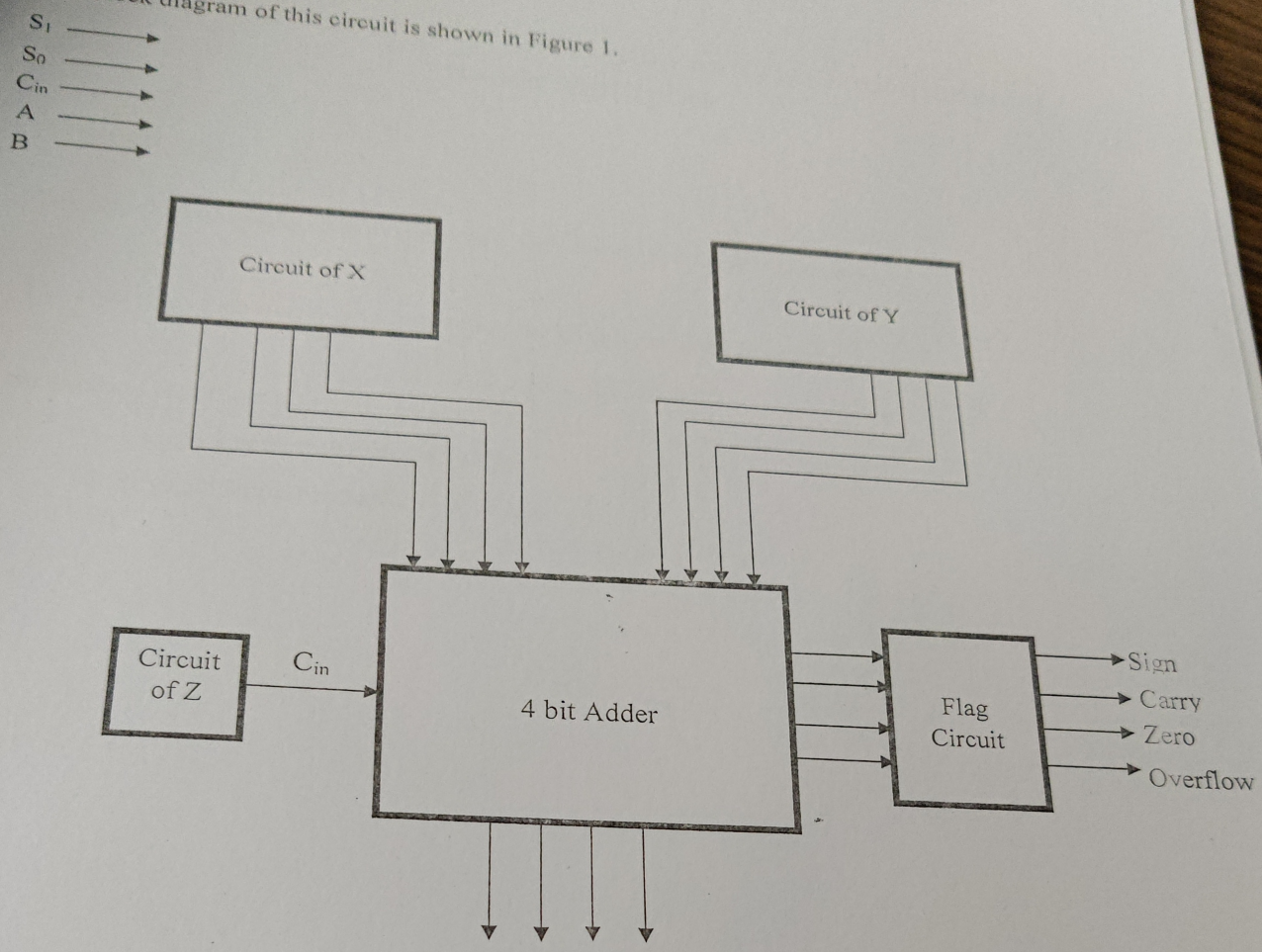


Figure 1: Block Diagram of ALU

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

Design Guideline

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) bits with zeros.

   ii) S: Fill the most significant (leftmost) bits with the value of (-m) in tow's cpmplement 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.

Example:

Find 13 × (-6) with m = 13 and r = -6, and x = 5 and y = 5.

m = 01101 -m = 10011 r = 11010

A: 01101 00000 0

S: 10011 00000 0

P: 00000 11010 0

Perform the loop five times:

1. P = 00000 1101**0 0**

   P = 00000 01101 0

2. P = 00000 0110**1 0**

   P = P+S = 10011 01101 0

   P = 11001 10110 1

3. P = 11001 1011**0 1**

   P = P+A = 00110 10110 1

   P = 00011 01011 0

4. P = 00011 0101**1 0**

   P = P+S = 10110 01011 0

   P = 11011 00101 1

5. P = 11011 0010**1 1**

   P = 11101 10010 1

Answer: 11101 10010.

5×5 Bit Booth's Multiplier Design

Initialization:

U ⟵ 0 [Partial Product most significant bits]

V ⟵ 0 [Partial Product least significant bits]

X ⟵ Input [Multiplicand]

Y ⟵ Input [Multiplier]

X_{-1} ⟵ 0 [Extra bit for calculation]

Count ⟵ 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:

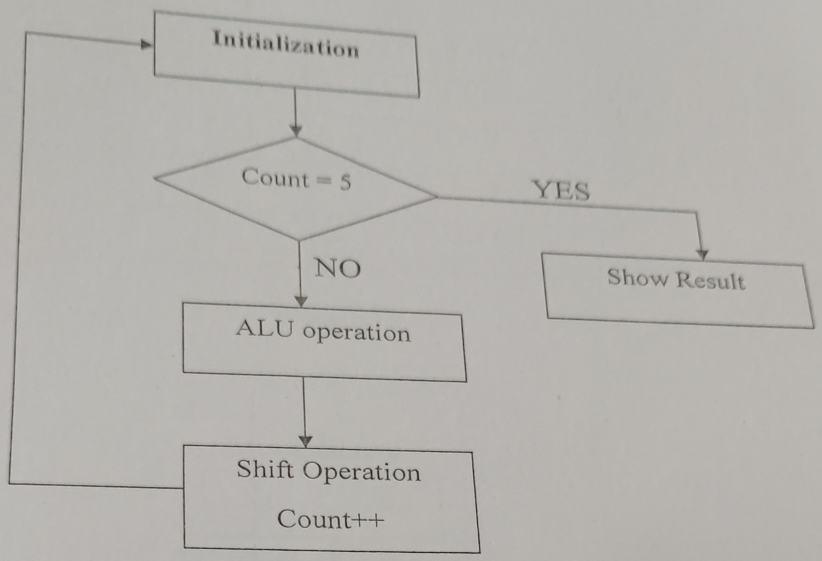


Figure 2: Flow Diagram for Booth's Multiplier Design

State Diagram:

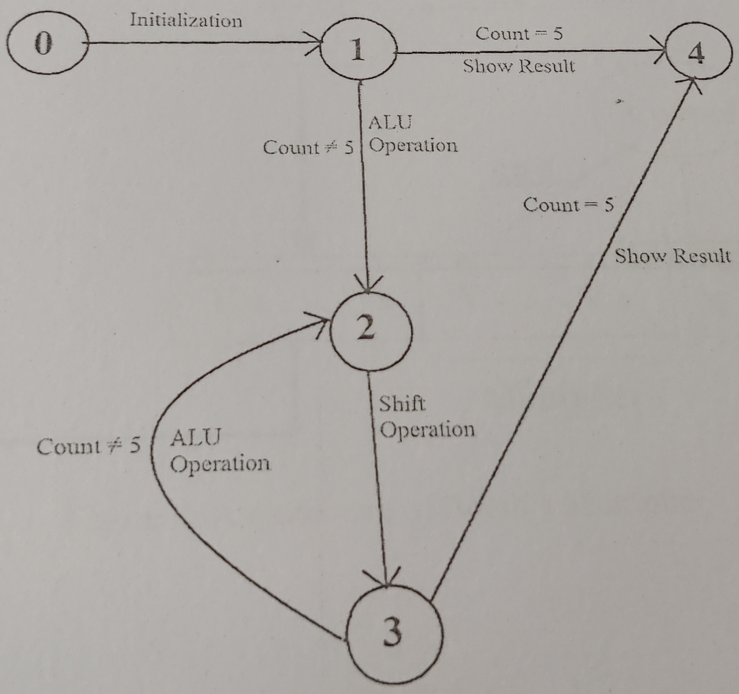


Figure 3: State Diagram for Booth's Multiplier Design

ALU Operations:

S_1	S_0	Operation
0	0	U + 0
0	1	U + Y
1	0	U – Y
1	1	U + 0

Architecture:

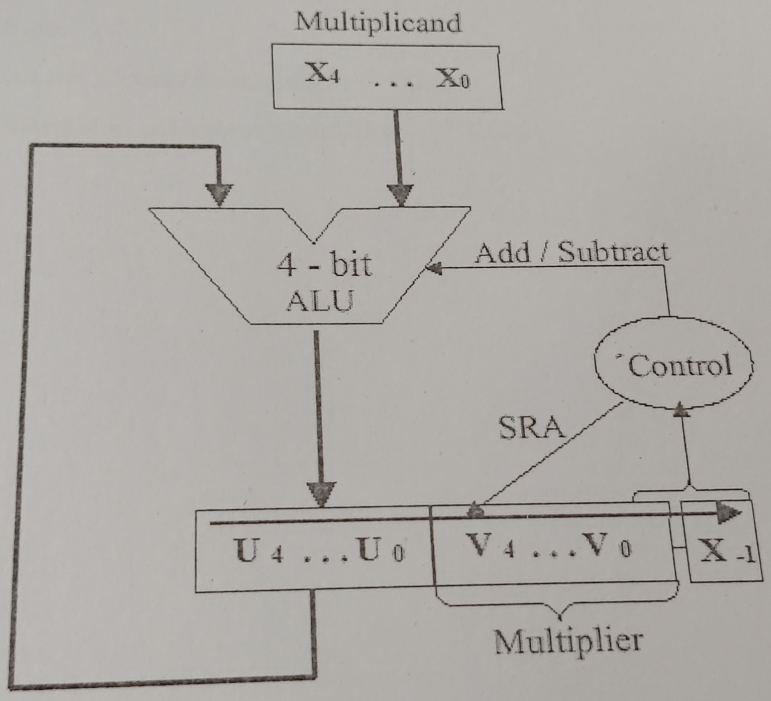


Figure 4: Architecture of Booth's Multiplier

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-1architecture 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-1circuit in Proteus.

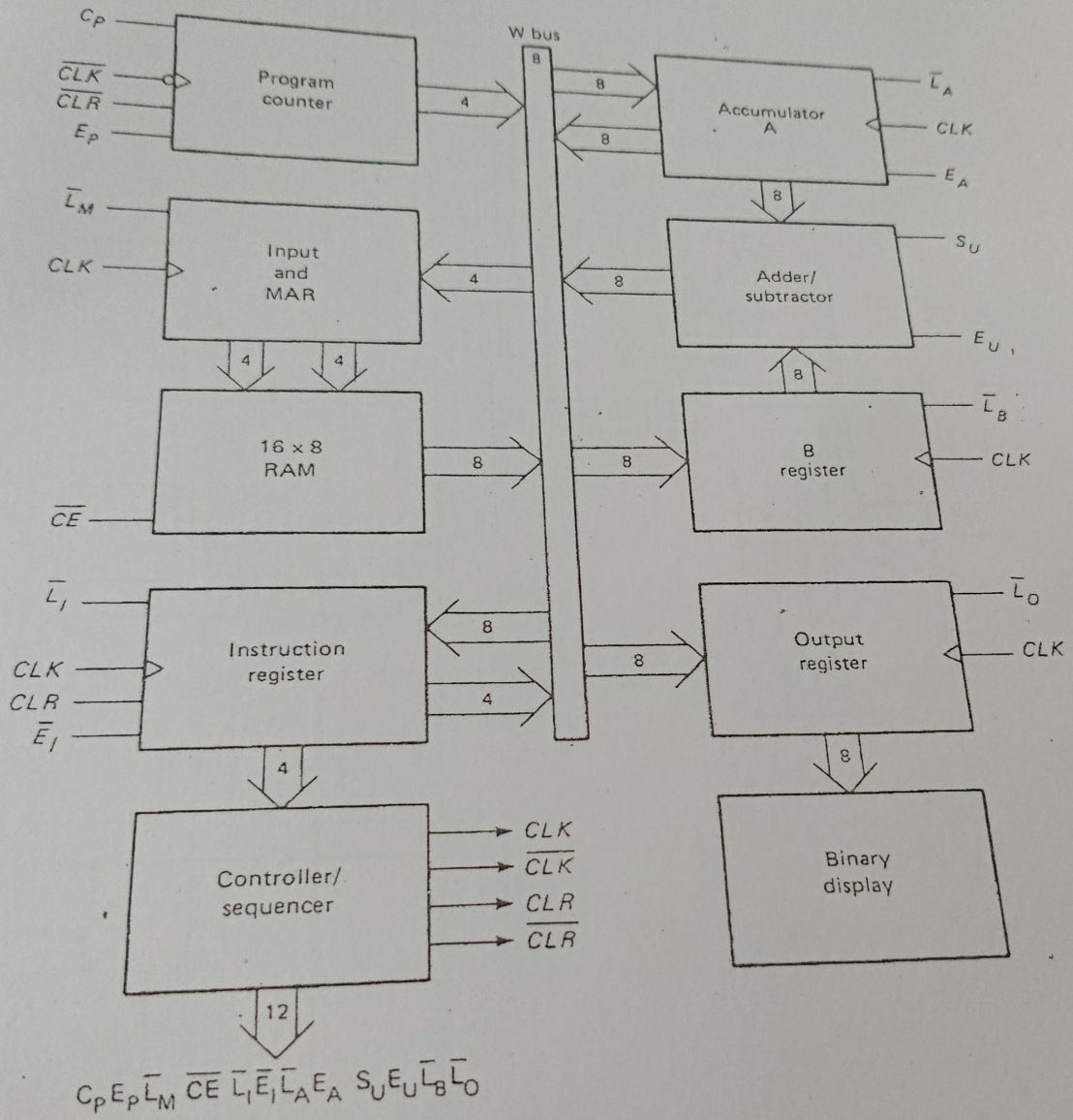
Submission Deadline:

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 in details in the class.

SAP-1 Architecture



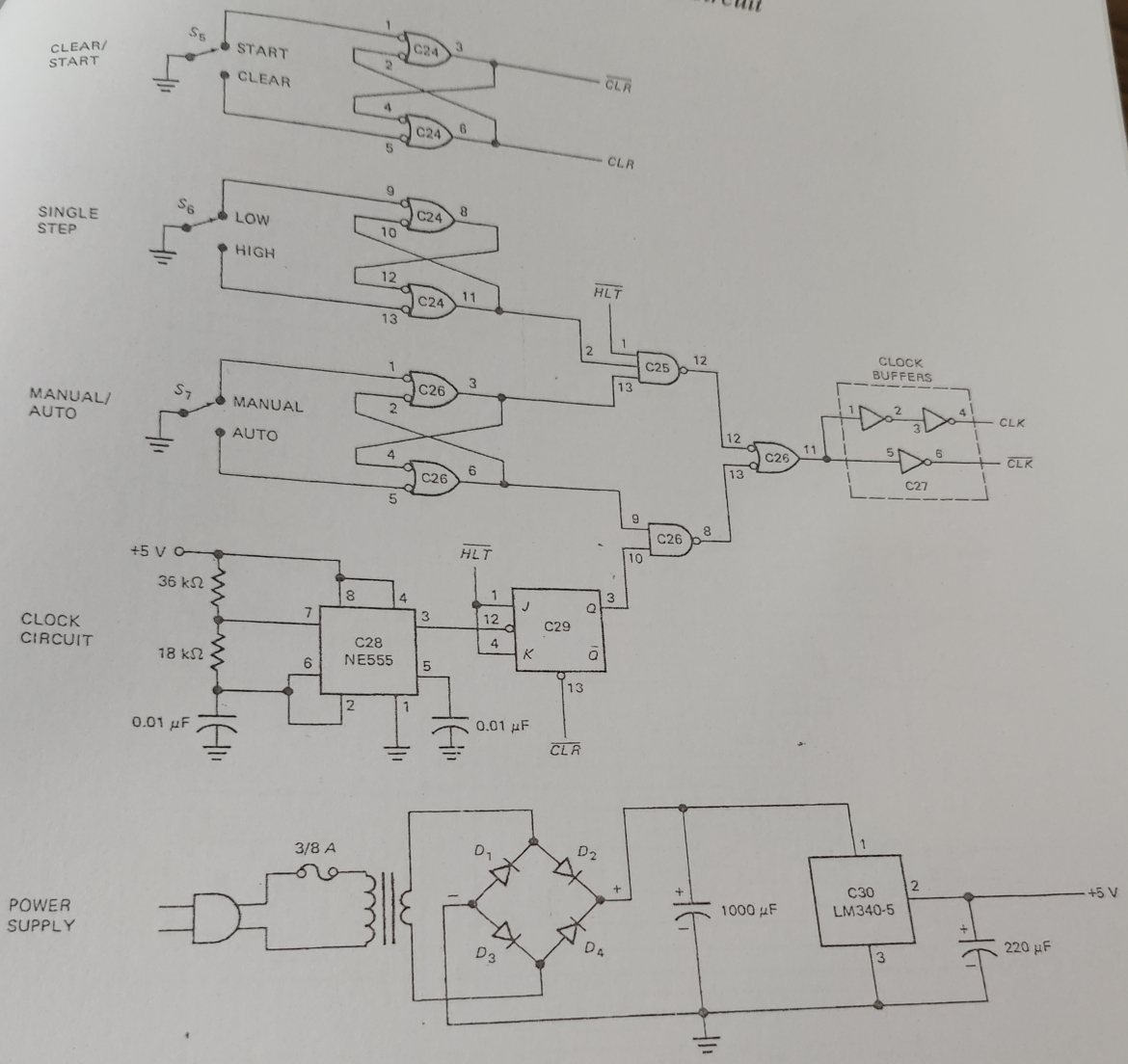
Program Counter and MAR

RAM and Instruction Register

Accumulator and Adder/Subtractor

B Register and Output Register

Power Supply, Clock and Clear Circuit



Instruction Decoder, Ring Counter and Control Matrix

Mid Term Examination

There will be a 15 minute MCQ examination of 20 marks. There will be 20 questions on the basic topics of Arithmetic Logic Unit and Booth's Multiplier. There will be negative marking for the wrong answers.

Final Term Examination

There will be a 15 minute MCQ examination of 20 marks. There will be 20 questions on the basic topics of Arithmetic Logic Unit, Booth's Multiplier and SAP-1. There will be negative marking for the wrong answers.