

* Assignment-1 *

Ques:- Define the following:

(i) Assembler:

An assembler is a program that takes basic computer instructions and converts them into a pattern of bits that the computer's processor can use to perform its basic operations.

The Assembler is a software that converts an assembly language code to machine code. It takes basic computer commands and converts them into Binary Code that Computer's Processor can use to perform its Basic Operations. These instructions are assembler language or assembly language.

(ii) Compiler:

A compiler is a special program that translates a programming language's source code into machine code, bytecode or another programming language. The source code is typically written in a high-level, human-readable language such as Java or C++. A programmer writes the source code in a code editor or an integrated development environment (IDE) that includes an editor, saving the source code to one or more text files.

(iii) Linker:

A Linker is an important utility program that takes the object files, produced by the assembler and compiler, and other code to join them into a single executable file. There are two types of Linkers, dynamic and linkage.

(iv) Loader:

In the world of computer science, a loader is a vital component of an operating system that is accountable for loading programs and libraries. Absolute, Direct Linking, Bootstrap and Relocating are the types of loaders.

- (v) Interpreter: An interpreter is a computer program that is used to directly execute program instructions written using one of the many high-level programming languages.
- The interpreter transforms the high-level program into an intermediate language that it then executes, or it could read the high-level source code and then perform the commands directly, which is done line by line or statement by statement.
- (vi) Assembler Directives: Assembler directives are directions to the assembler to take some action or change a setting. Assembler directives do not represent instructions, and are not translated into machine code.
- (vii) Machine Level Language: Machine Level Language is the language understood by a computer. It is very difficult to understand but it is the only thing that the computer can work with. All programs and programming languages eventually generate or run programs in machine language.
- (viii) Assembly Language: An assembly language is a type of low-level programming language that is intended to communicate directly with a computer's hardware. Assembly language helps in contacting the hardware directly.
- (ix) High Level Language: A high-level language is any programming language that enables development of a program in a much more user-friendly programming context and is generally independent of the computer's architecture.

(x) System Software: System software is a type of computer program that is designed to run a computer's hardware and application programs. System software refers to the low-level software that manages and controls a computer's hardware and provides basic services to higher-level software.

(xi) Application Software: Application software is a type of computer program that performs a specific personal, educational, and business function. Each application is designed to assist end-users in accomplishing a variety of tasks, which may be related to productivity, creativity, or communication.

*Ques-2: What is the difference between Application and System software?

*Ans:-

System Software	Application Software
<ol style="list-style-type: none">1. System software is the type of software which is the interface between application software and system.2. In general, system software are developed using low-level language which is more compatible with the system hardware in order to interact with.3. System software is used for operating computer hardware.4. System software can run independently. It provides platform for running application software.5. Android, Mac operating system, MS Windows etc.	<ol style="list-style-type: none">1. Application software is the type of software runs as per user request. It runs on the platform which is provided by system software.2. In case of Application software high level language is used for their development as they are developed as some specific purpose software.3. Application software is used by users to perform specific task.4. An application software cannot run independently. It can not run without the presence of system software.5. Word Processor, games, media players, etc.

Ques:- Is there any alternative of Von Neumann architecture.

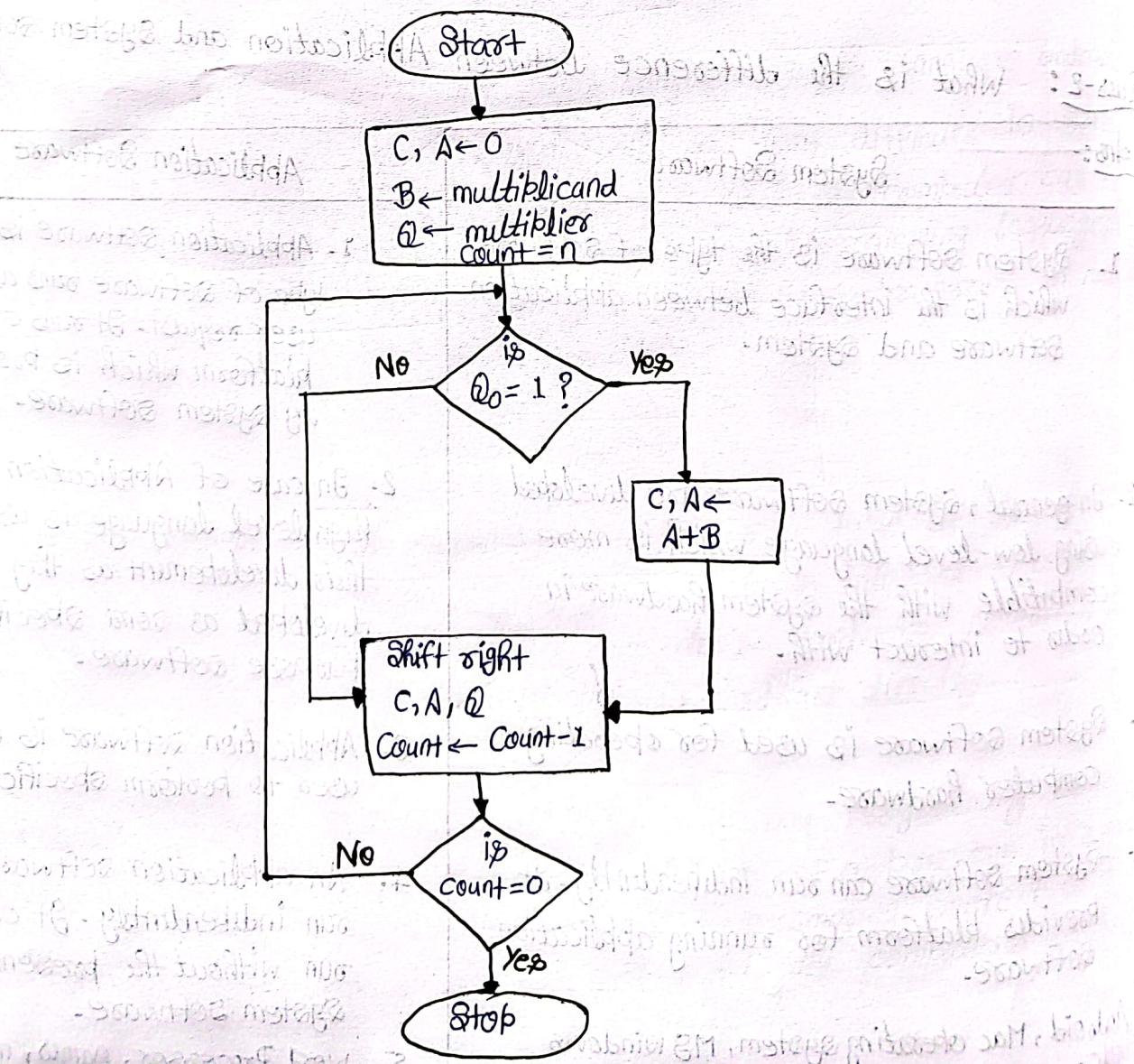
Ans:- Examples of non von neumann machines are the dataflow machines and the reduction machine. In both of these cases there is a high degree of parallelism, and instead of variables there are immutable bindings between names and constant values.

Ques:- Find the unsigned binary multiplication of 12 and 13 and show the step by step process with flowchart.

Sol:- (B) Multiplicand = $12 = 1100$

(D) Multiplier = $13 = 1101$

* Flowchart *



$$12 \times 13 = 156$$

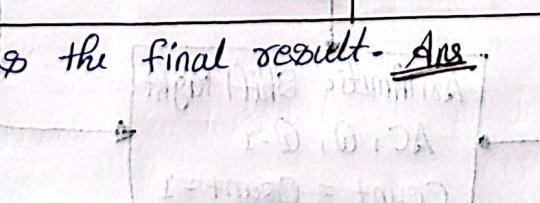
Expected Result

$$156 = 10011100$$

Tracing Table:

C	A	(Q)	Operation	Cycle
0	0000	1101	Initial	
0	1100	1101	Add A $\leftarrow A+B$	1
0	0110	0110	Right Shift	
0	0011	0011	Right Shift	2
0	1111	0011	Add A $\leftarrow A+B$	3
0	0111	1000	Shift Right	
1	0011	1001	Add A $\leftarrow A+B$	4
0	1001	1100	Right Shift	

So, 10011100 is the final result. Ans.



IN

Output

531

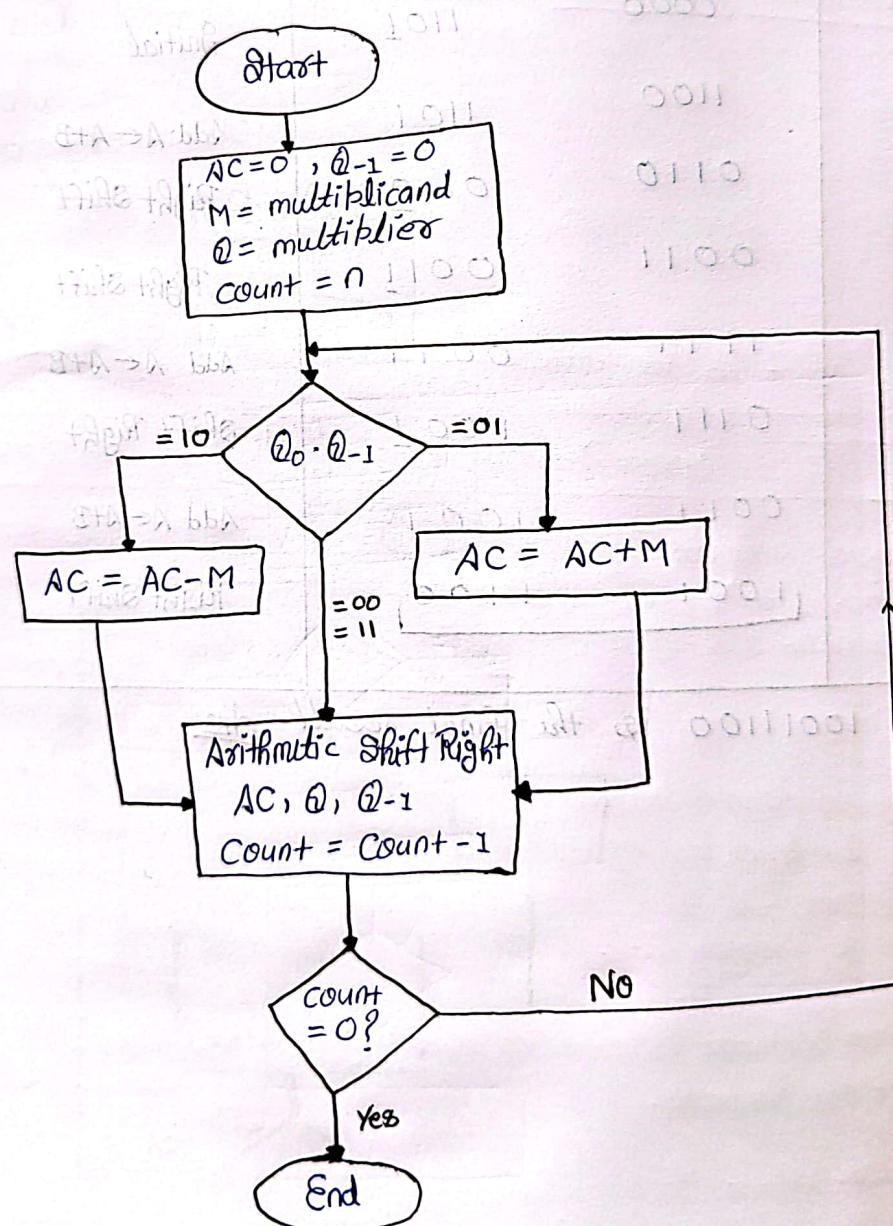
113

Ques:- Find the sign binary multiplication of -7 and 9 using Booth's algorithm and show step by step process with flowchart.

Solⁿ:- $M \rightarrow (-7)_{10} \rightarrow 1001$
 $Q \rightarrow (9)_{10} \rightarrow 1001$
 $(-M) \rightarrow 0111$

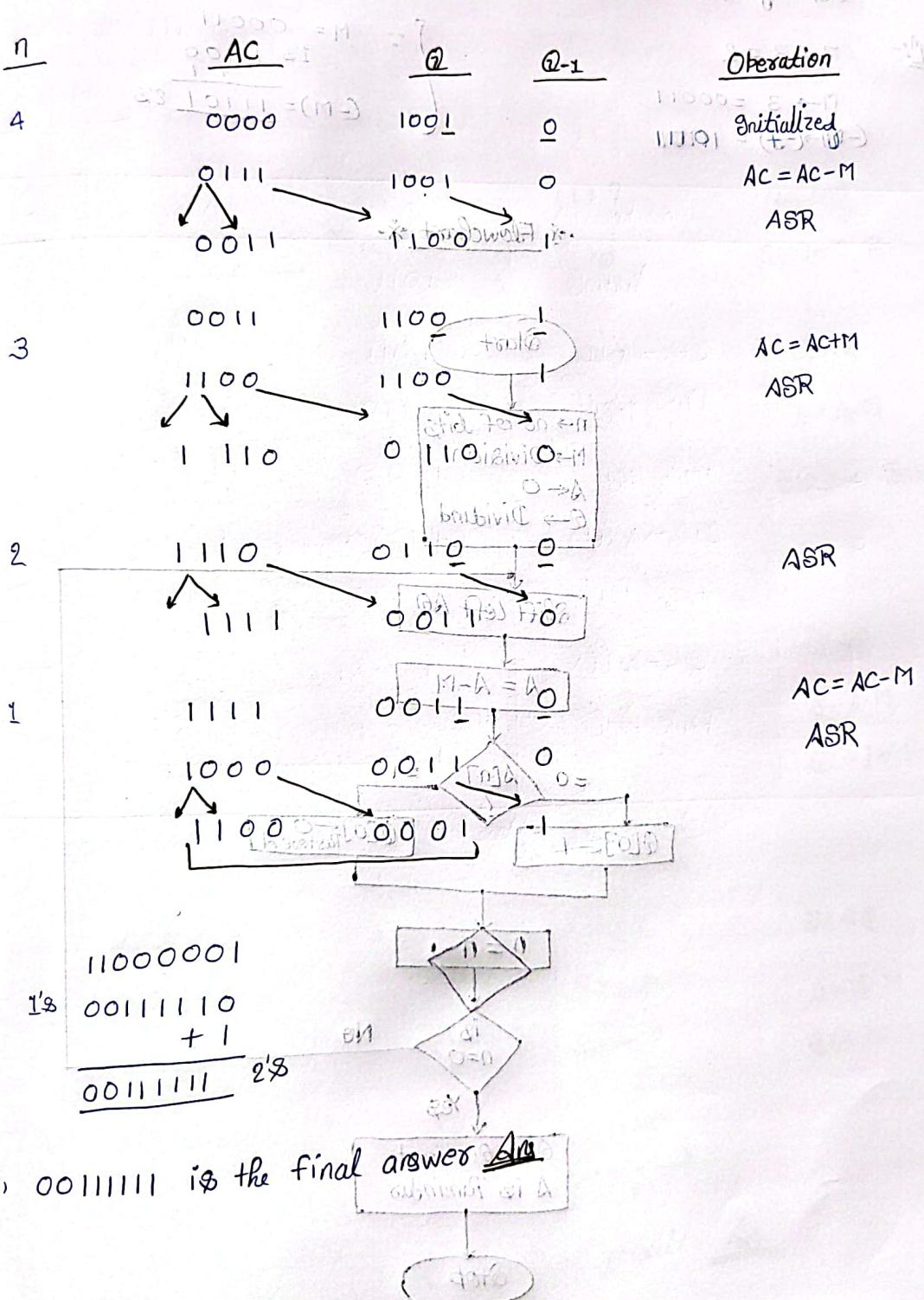
$$\left\{ \begin{array}{l} \because (-7)_{10} = 0111 \\ \text{1's complement} \\ 1000 \\ +1 \\ \hline (-7)_{10} = 1001 \end{array} \right.$$

Booth's Flowchart



Tracing Table:

निम्नलिखित ग्राफ से F का विकल्प चुनिए जो अंत में दिये गए डेटा पर कार्यवालत की असर देता है।



Ques:- Find the unsigned binary division of $7 \div 3$ using division algorithm and show step by step process with flowchart.

Toac

Sol:-

$$7 \div 3 = 2$$

$$M \rightarrow 3 = 00011$$

$$(-M) \rightarrow (-3) = 11101$$

$$M - CA = CA$$

RCA

$$MCA = CA$$

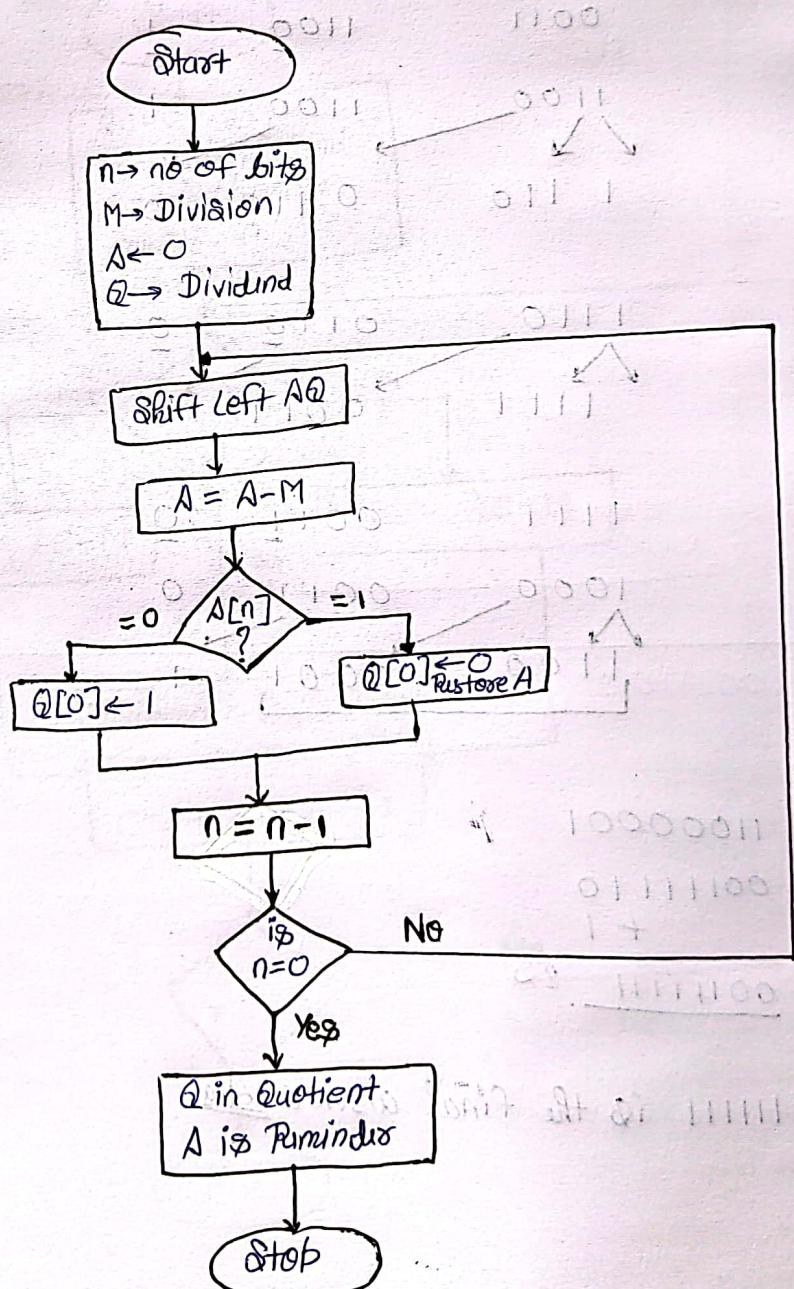
RCA

$$M - CA = CA$$

RCA

$$\left\{ \begin{array}{l} M = 00011 \\ 1's \quad \underline{\quad 11100} \\ + 1 \\ \hline (-M) = \underline{\quad 11101} \end{array} \right. \quad 2's$$

* Flowchart *



* Tracing Table:

<u>n</u>	<u>M(n+1)</u>	<u>A(n+1)</u>	<u>Q(n)</u>	<u>Operation</u>
4	00011	00000	00111	Initialization
	00000	111?		SLAQ
	11101	111?		$A = A - M$
	00000	1110		$Q[0] \leftarrow 0$ Restore A
3	00000	1110		SLAQ
	00001	110?		$A = A - M$
	11110	110?		$Q[0] \leftarrow 0$ Restore A
	00001	1100		
2	00001	1100		SLAQ
	00011	100?		$A = A - M$
	00000	100?		$Q[0] \leftarrow 1$
	00000	1001		
1	00000	1001		SLAQ
	00001	001?		$A = A - M$
	11110	001?		$Q[0] \leftarrow 0$ Restore A
	00001	0010		
		(1) (2)		

So, 00001 and 0010 is the final result. Ans