USN										

Internal Assessment Test 3 – February 2023

Sub:	Computer Organization and Architecture				Sub Code:	21CS34	Branch:	CSE
Date:	09/02/23	Duration:	90 minutes	Max Marks:	50	Sem / Sec:	III / A, B, C	OBE

	Answer any FIVE FULL Questions	MARKS	CO	RBT
1	(a) Perform Addition using 2’s Compliment (i) 5 and 9 (ii) -5 and 7 Perform Subtraction using 2’s Compliment (iii) 5 and 3 (iv) -6 and -2 (v) -7 and 2	[5]	4	L2
	(b) Explain4 bit Carry Look Ahead Adder with neat diagram	[5]	4	L2
2	(a) Multiply 11 and 6 using Sequential Multiplication	[5]	4	L2
	(b) Discuss with neat diagram the single bus organization of the data path inside a processor	[5]	4	L2
3	(a) Explain in detail Hard Wired Control	[5]	4	L1
	(b) Explain the basic organization of a Micro Programmed Control unit.	[5]	4	L1

CI CCI HoD

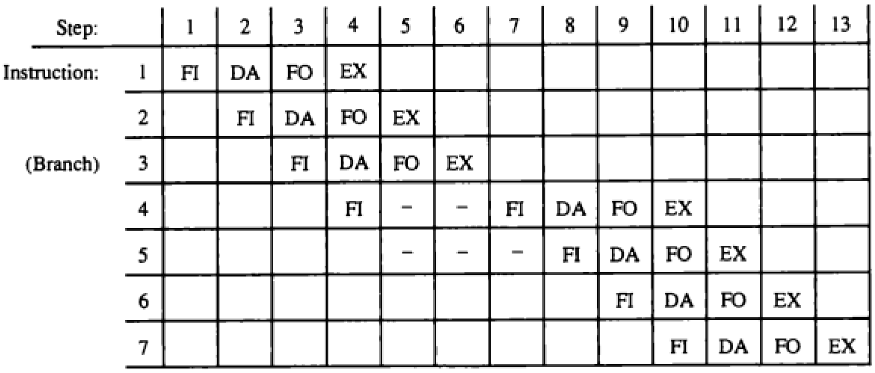
USN										

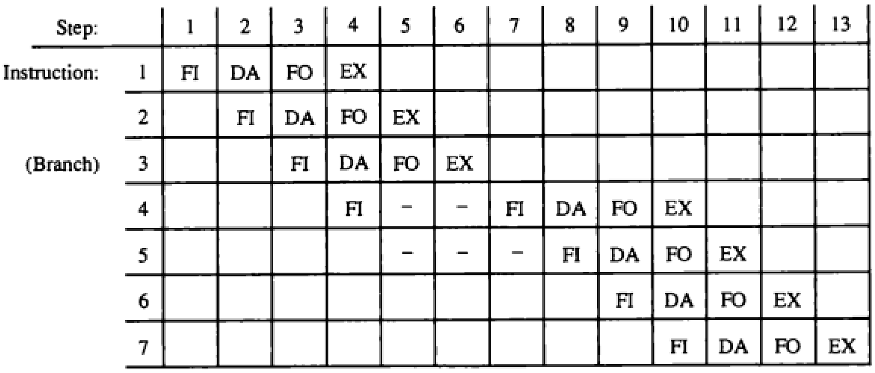
Internal Assessment Test 3 – February 2023

Sub:	Computer Organization and Architecture				Sub Code:	21CS34	Branch:	CSE
Date:	09/02/23	Duration:	90 minutes	Max Marks:	50	Sem / Sec:	III / A, B, C	OBE

	Answer any FIVE FULL Questions	MARKS	CO	RBT
1	(b) Perform Addition using 2’s Compliment (vi) 5 and 9 (vii) -5 and 7 Perform Subtraction using 2’s Compliment (viii) 5 and 3 (ix) -6 and -2 (x) -7 and 2	[5]	4	L2
	(b) Explain4 bit Carry Look Ahead Adder with neat diagram	[5]	4	L2
2	(a) Multiply 11 and 6 using Sequential Multiplication	[5]	4	L2
	(b) Discuss with neat diagram the single bus organization of the data path inside a processor	[5]	4	L2
3	(a) Explain in detail about Hard Wired Control	[5]	4	L1
	(b) Explain the basic organization of a Micro Programmed Control unit.	[5]	4	L1

CI CCI HoD

<table>
<tr><td rowspan="2">4</td><td>(a) Explain in detail about Parallel Processing. Also mention about the Flynn’s Classification.</td><td>[5]</td><td>5</td><td>L1</td></tr>
<tr><td>(b) Consider the following Pipeline and answer the questions below.

<table>
<tr><td>Step:</td><td></td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td></tr>
<tr><td>Instruction:</td><td>1</td><td>FI</td><td>DA</td><td>FO</td><td>EX</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td>2</td><td></td><td>FI</td><td>DA</td><td>FO</td><td>EX</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>(Branch)</td><td>3</td><td></td><td></td><td>FI</td><td>DA</td><td>FO</td><td>EX</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td>4</td><td></td><td></td><td></td><td>FI</td><td>–</td><td>–</td><td>FI</td><td>DA</td><td>FO</td><td>EX</td><td></td><td></td><td></td></tr>
<tr><td></td><td>5</td><td></td><td></td><td></td><td></td><td>–</td><td>–</td><td>–</td><td>FI</td><td>DA</td><td>FO</td><td>EX</td><td></td><td></td></tr>
<tr><td></td><td>6</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>FI</td><td>DA</td><td>FO</td><td>EX</td><td></td></tr>
<tr><td></td><td>7</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>FI</td><td>DA</td><td>FO</td><td>EX</td></tr>
</table>
(i) How many stage pipeline dose the diagram depict? (1)
(ii) How many clock cycles does it take to complete the 7 instructions? (1)
(iii) Redraw the timing diagram after removing the branch instruction. (2)
(iv) How many clock cycles does it take to the instructions after branch is removed? (1)</td><td>[5]</td><td>5</td><td>L3</td></tr>
<tr><td rowspan="2">5</td><td>(a) How many sub operations can an Arithmetic Pipeline be divided. Explain the same in detail.</td><td>[5]</td><td>5</td><td>L1</td></tr>
<tr><td>(b) With neat figure, illustrate the four segment operations of the Instruction Pipeline.</td><td>[5]</td><td>5</td><td>L2</td></tr>
<tr><td rowspan="2">6</td><td>(a) Elucidate Vector Processing. Also explain how Memory Interleaving impacts the performance of vector processing.</td><td>[5]</td><td>5</td><td>L2</td></tr>
<tr><td>(b) Explain the following
(i) Attached Array Processor
(ii) SIMD Array Processor</td><td>[5]</td><td>5</td><td>L1</td></tr>
</table>

<table>
<tr><td rowspan="2">4</td><td>(a) Explain in detail about Parallel Processing. Also mention about the Flynn’s Classification.</td><td>[5]</td><td>5</td><td>L1</td></tr>
<tr><td>(b) Consider the following Pipeline and answer the questions below.

<table>
<tr><td>Step:</td><td></td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td></tr>
<tr><td>Instruction:</td><td>1</td><td>FI</td><td>DA</td><td>FO</td><td>EX</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td>2</td><td></td><td>FI</td><td>DA</td><td>FO</td><td>EX</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>(Branch)</td><td>3</td><td></td><td></td><td>FI</td><td>DA</td><td>FO</td><td>EX</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td>4</td><td></td><td></td><td></td><td>FI</td><td>–</td><td>–</td><td>FI</td><td>DA</td><td>FO</td><td>EX</td><td></td><td></td><td></td></tr>
<tr><td></td><td>5</td><td></td><td></td><td></td><td></td><td>–</td><td>–</td><td>–</td><td>FI</td><td>DA</td><td>FO</td><td>EX</td><td></td><td></td></tr>
<tr><td></td><td>6</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>FI</td><td>DA</td><td>FO</td><td>EX</td><td></td></tr>
<tr><td></td><td>7</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>FI</td><td>DA</td><td>FO</td><td>EX</td></tr>
</table>
(i) How many stage pipeline dose the diagram depict? (1)
(ii) How many clock cycles does it take to complete the 7 instructions? (1)
(iii) Redraw the timing diagram after removing the branch instruction. (2)
(iv) How many clock cycles does it take to the instructions after branch is removed? (1)</td><td>[5]</td><td>5</td><td>L3</td></tr>
<tr><td rowspan="2">5</td><td>(a) How many sub operations can an Arithmetic Pipeline be divided. Explain the same in detail.</td><td>[5]</td><td>5</td><td>L1</td></tr>
<tr><td>(b) With neat figure, illustrate the four segment operations of the Instruction Pipeline.</td><td>[5]</td><td>5</td><td>L2</td></tr>
<tr><td rowspan="2">6</td><td>(b) Elucidate Vector Processing. Also explain how Memory Interleaving impacts the performance of vector processing.</td><td>[5]</td><td>5</td><td>L2</td></tr>
<tr><td>(c) Explain the following
(i) Attached Array Processor
(ii) SIMD Array Processor</td><td>[5]</td><td>5</td><td>L1</td></tr>
</table>

CO PO Mapping

Course Outcomes		Modules covered	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4
CO1	Explain the organization and architecture of computer systems with machine instructions and programs	**1**	3	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
CO2	Analyze the input/output devices communicating with computer system	**2**	3	-	-	-	-	-	-	-	-	-	-	-	2	-	-	-
CO3	Demonstrate the functions of different types of memory devices	**3**	2	2	2	-	-	-	-	-	-	-	-	-	-	2	-	-
CO4	Apply different data types on simple arithmetic and logical unit	**4**	3	-	-	-	-	-	-	-	-	-	-	-	-	2	-	-
CO5	Analyze the functions of basic processing unit, Parallel processing and pipelining	**5**	3	2	3	2	-	-	-	-	-	-	-	-	-	2	-	-

COGNITIVE LEVEL	REVISED BLOOMS TAXONOMY KEYWORDS
L1	List, define, tell, describe, identify, show, label, collect, examine, tabulate, quote, name, who, when, where, etc.
L2	summarize, describe, interpret, contrast, predict, associate, distinguish, estimate, differentiate, discuss, extend
L3	Apply, demonstrate, calculate, complete, illustrate, show, solve, examine, modify, relate, change, classify, experiment, discover.
L4	Analyze, separate, order, explain, connect, classify, arrange, divide, compare, select, explain, infer.
L5	Assess, decide, rank, grade, test, measure, recommend, convince, select, judge, explain, discriminate, support, conclude, compare, summarize.

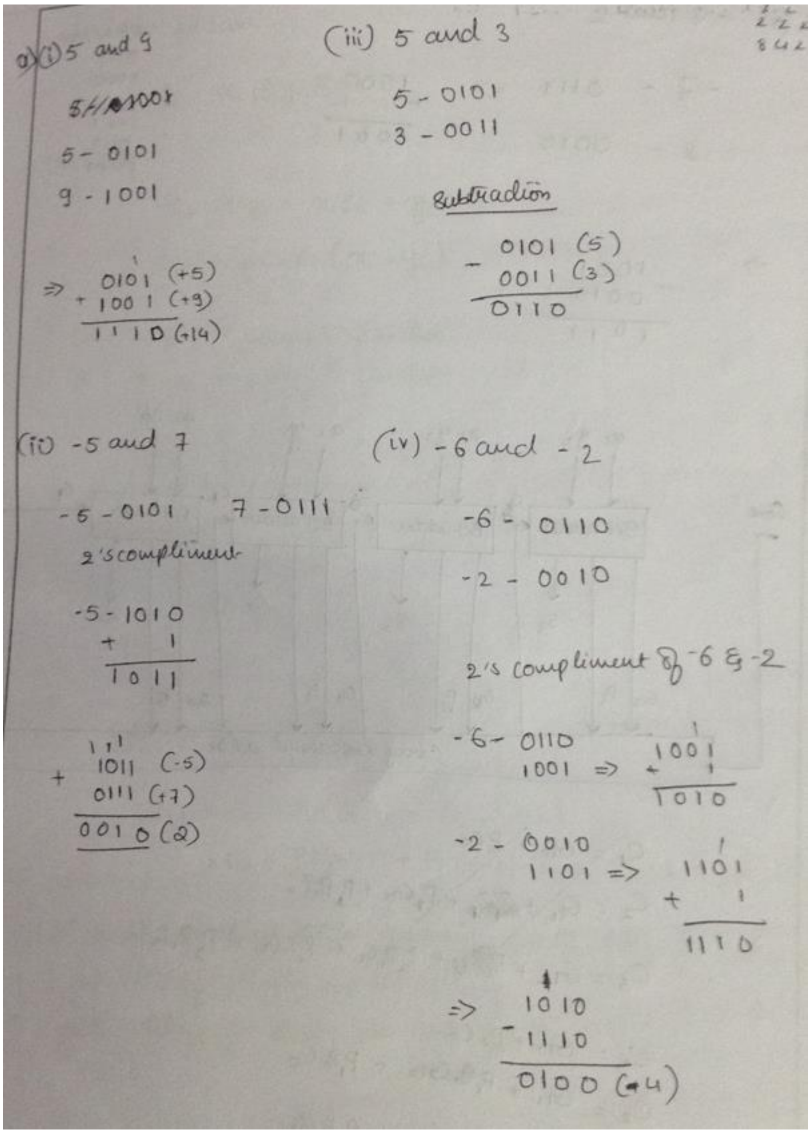
PROGRAM OUTCOMES (PO), PROGRAM SPECIFIC OUTCOMES (PSO)				CORRELATION LEVELS	
PO1	Engineering knowledge	PO7	Environment and sustainability	0	No Correlation
PO2	Problem analysis	PO8	Ethics	1	Slight/Low
PO3	Design/development of solutions	PO9	Individual and team work	2	Moderate/ Medium
PO4	Conduct investigations of complex problems	PO10	Communication	3	Substantial/ High
PO5	Modern tool usage	PO11	Project management and finance		
PO6	The Engineer and society	PO12	Life-long learning		
PSO1	Develop applications using different stacks of web and programming technologies				
PSO2	Design and develop secure, parallel, distributed, networked, and digital systems				
PSO3	Apply software engineering methods to design, develop, test and manage software systems.				
PSO4	Develop intelligent applications for business and industry				

Internal Assessment Test 3 – February 2023

Sub:	Computer Organization and Architecture					Sub Code:	21CS34	Branch:	CSE
Date:	09/02/23	Duration:	90 minutes	Max Marks:	50	Sem / Sec:	III / A, B, C		OBE

Solutions

1a.



a) (i) 5 and 9

~~5+/1001~~

5 - 0101

9 - 1001

```
      1
=>   0101 (+5)
   + 1001 (+9)
   ----------
     1110 (+14)
```

(ii) -5 and 7

-5 - 0101   7 - 0111

2'scompliment

```
-5 - 1010
   +    1
   ------
     1011
```

```
     111
     1011 (-5)
 +   0111 (+7)
    ---------
     0010 (2)
```

(iii) 5 and 3

5 - 0101

3 - 0011

Subtraction

```
   0101 (5)
 - 0011 (3)
 ---------
   0110
```

(iv) -6 and -2

-6 - 0110

-2 - 0010

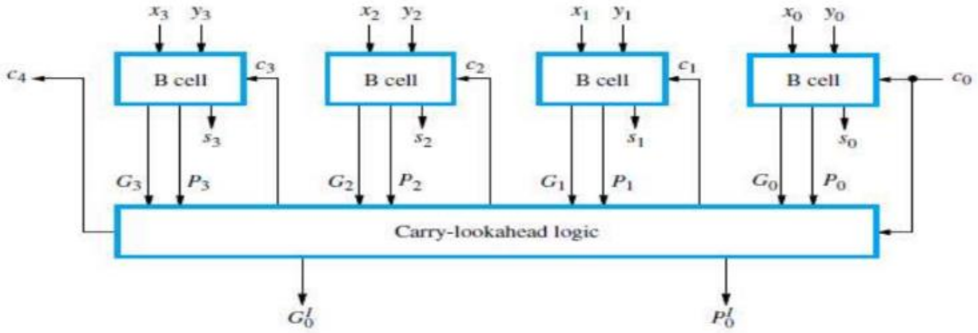
2's compliment of -6 & -2

```
-6 - 0110              1
     1001   =>      1001
                  +    1
                  ------
                    1010

-2 - 0010              1
     1101   =>      1101
                  +    1
                  ------
                    1110

          1
=>     1010
     - 1110
     ------
       0100 (+4)
```

b.



The diagram of a carry look ahead adder is as shown above,

In ripple carry adders, the carry propagation time is the major speed limiting factor. Most other arithmetic operations like multiplication and division are implemented using several add/subtract steps, Thus, improving the speed of addition will improve the speed of all other arithmetic operations.

Carry-lookahead adder improves the speed by reducing carry propagation delay. It calculates the carry signals in advance, based on input signals instead of waiting for them to ripple through the adders.

$$s_i = x_i \oplus y_i \oplus c_i$$

$$c_{i+1} = x_i y_i + x_i c_i + y_i c_i$$

$$c_{i+1} = x_i y_i + (x_i + y_i) c_i$$

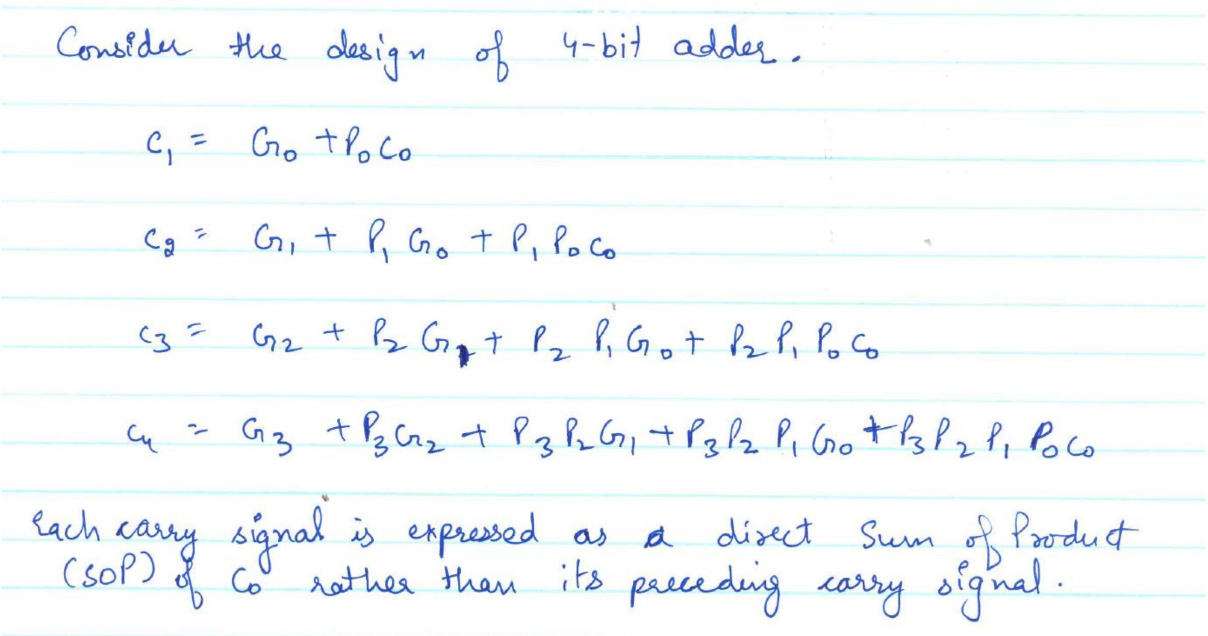
~~$c_{i+1} = x_i y_i$~~

$$c_{i+1} = G_i + P_i c_i$$

where

$G_i = x_i y_i$ → Generate function

$P_i = x_i + y_i$ → Propagate function



Consider the design of 4-bit adder.

$$c_1 = G_0 + P_0 C_0$$

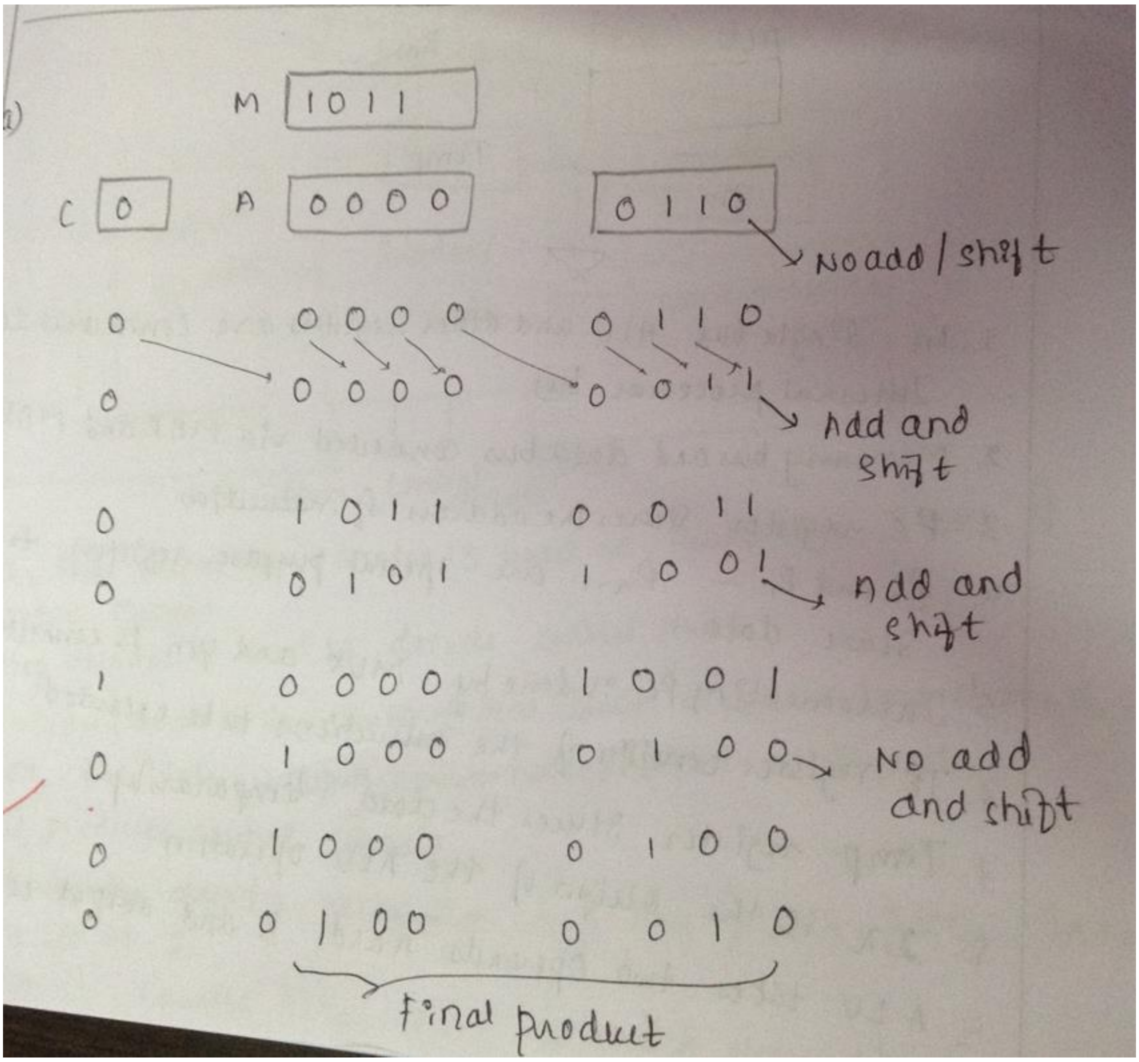
$$c_2 = G_1 + P_1 G_0 + P_1 P_0 C_0$$

$$c_3 = G_2 + P_2 G_1 + P_2 P_1 G_0 + P_2 P_1 P_0 C_0$$

$$c_4 = G_3 + P_3 G_2 + P_3 P_2 G_1 + P_3 P_2 P_1 G_0 + P_3 P_2 P_1 P_0 C_0$$

Each carry signal is expressed as a direct Sum of Product (SOP) of C_0 rather than its preceding carry signal.

2a.

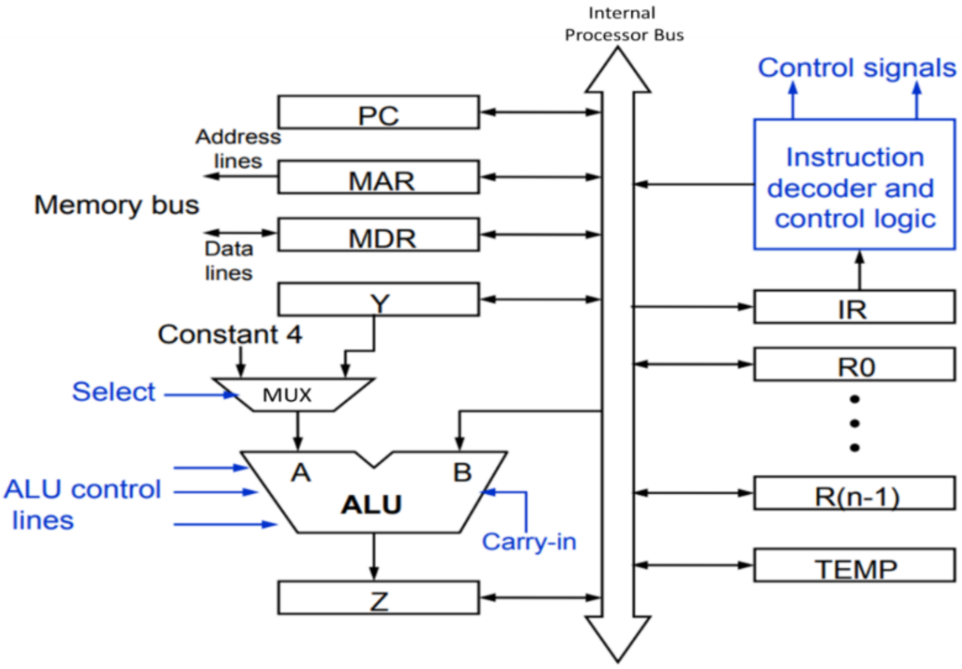


a)

	M 1011		
C 0	A 0000	0110	No add / shift
0	0000	0110	
0	0000	0011	Add and shift
0	1011	0011	
0	0101	1001	Add and shift
1	0000	1001	
0	1000	0100	No add and shift
0	1000	0100	
0	0100	0010	

final product

2b



ALU and all the registers are interconnected via a single common bus.

□ The data and address lines of the external memory bus connected to the internal processor bus via the memory data register, MDR, and the memory addressregister, MAR respectively.

□ Register MDR has two inputs and two outputs.

□ Data may be loaded into MDR either from the memory bus or from the internal processor bus.

□ The data stored in MDR may be placed on either bus.

□ The input of MAR is connected to the internal bus, and its output is connected to the external bus.

□ The control lines of the memory bus are connected to the instruction decoder and control logic.

□ This unit is responsible for issuing the signals that control the operation of all the units inside the processor and for increasing with the memory bus.

□ The MUX selects either the output of register Y or a constant value 4 to be provided as input A of the ALU.

□ The constant 4 is used to increment the contents of the program counter.

3a.

Hardwired Control

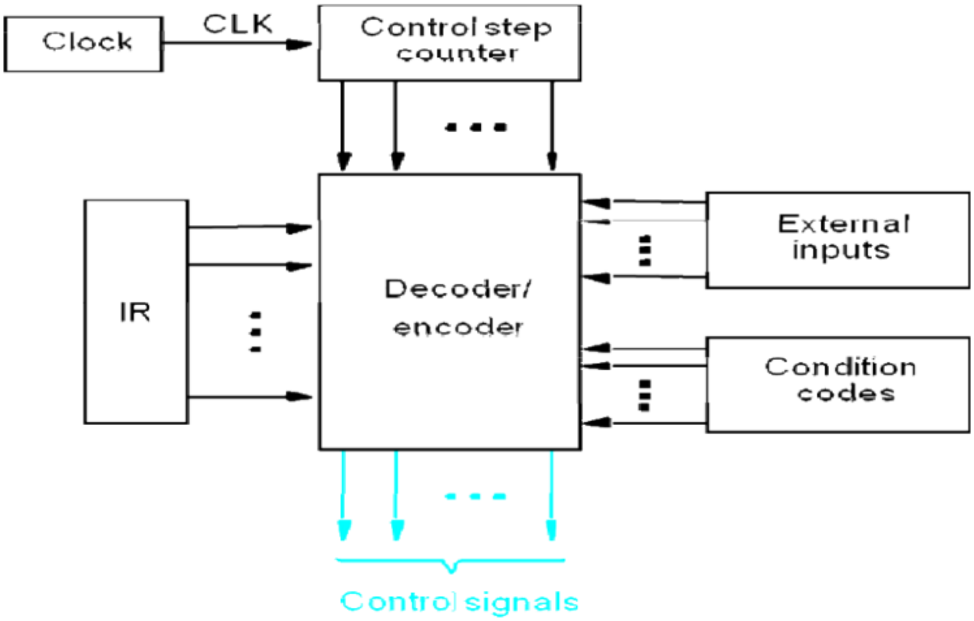
□ To execute instructions, the processor must have some means of generating the control signals needed in the proper sequence.

□ Two categories: hardwired control and micro programmed control

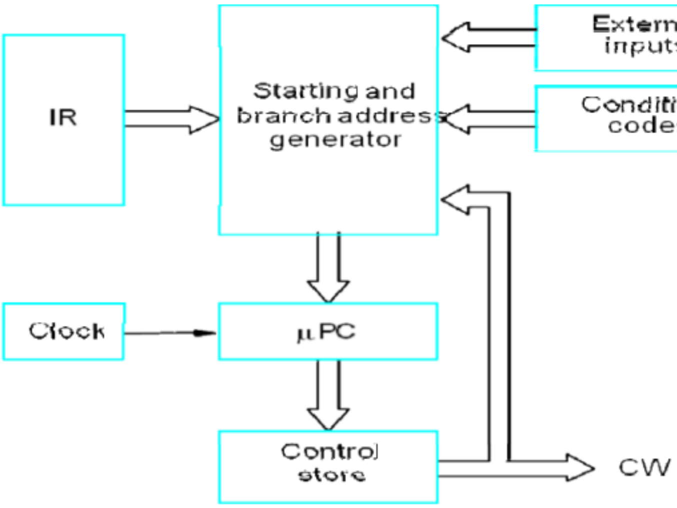
□ Hardwired system can operate at high speed; but with little flexibility.

3b

Control Unit Organization



3b



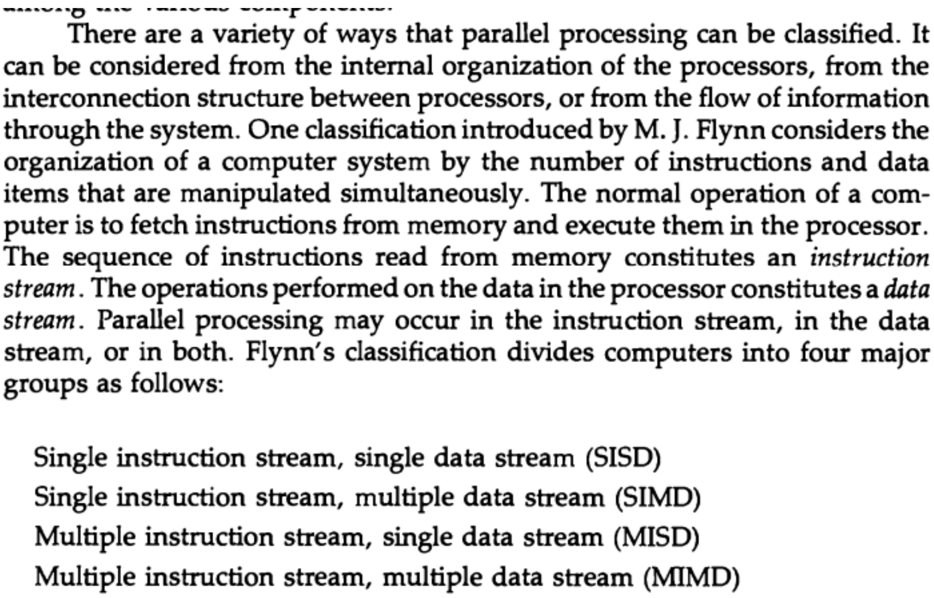
Microinstructions

 A straightforward way to structure microinstructions is to assign one bit position to each control signal.

 However, this is very inefficient.

 The length can be reduced: most signals are not needed simultaneously, and many signals are mutually exclusive.

 All mutually exclusive signals are placed in the same group in binary coding.



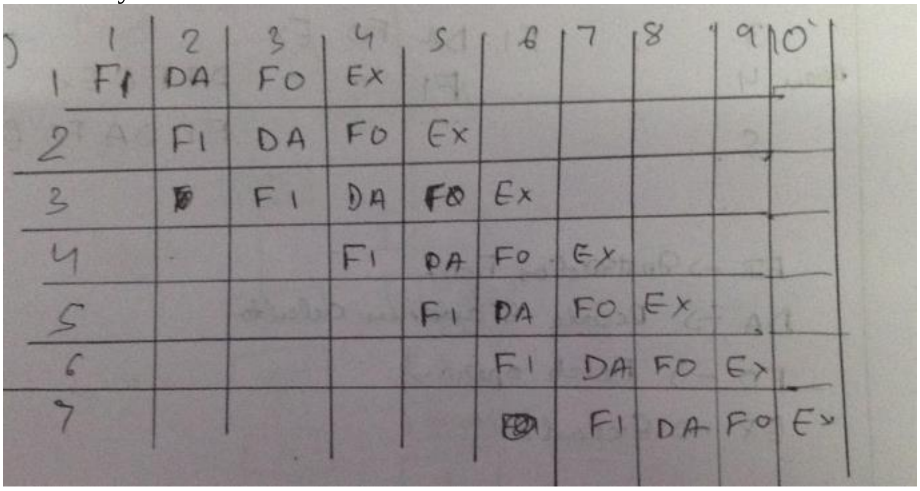
among the various components.

There are a variety of ways that parallel processing can be classified. It can be considered from the internal organization of the processors, from the interconnection structure between processors, or from the flow of information through the system. One classification introduced by M. J. Flynn considers the organization of a computer system by the number of instructions and data items that are manipulated simultaneously. The normal operation of a computer is to fetch instructions from memory and execute them in the processor. The sequence of instructions read from memory constitutes an *instruction stream*. The operations performed on the data in the processor constitutes a *data stream*. Parallel processing may occur in the instruction stream, in the data stream, or in both. Flynn's classification divides computers into four major groups as follows:

Single instruction stream, single data stream (SISD)
Single instruction stream, multiple data stream (SIMD)
Multiple instruction stream, single data stream (MISD)
Multiple instruction stream, multiple data stream (MIMD)

4b.

i. 4 stage
ii. 13 Clock cycles
iii. Diagram shown below
iv. 10 clocl cycles
v.

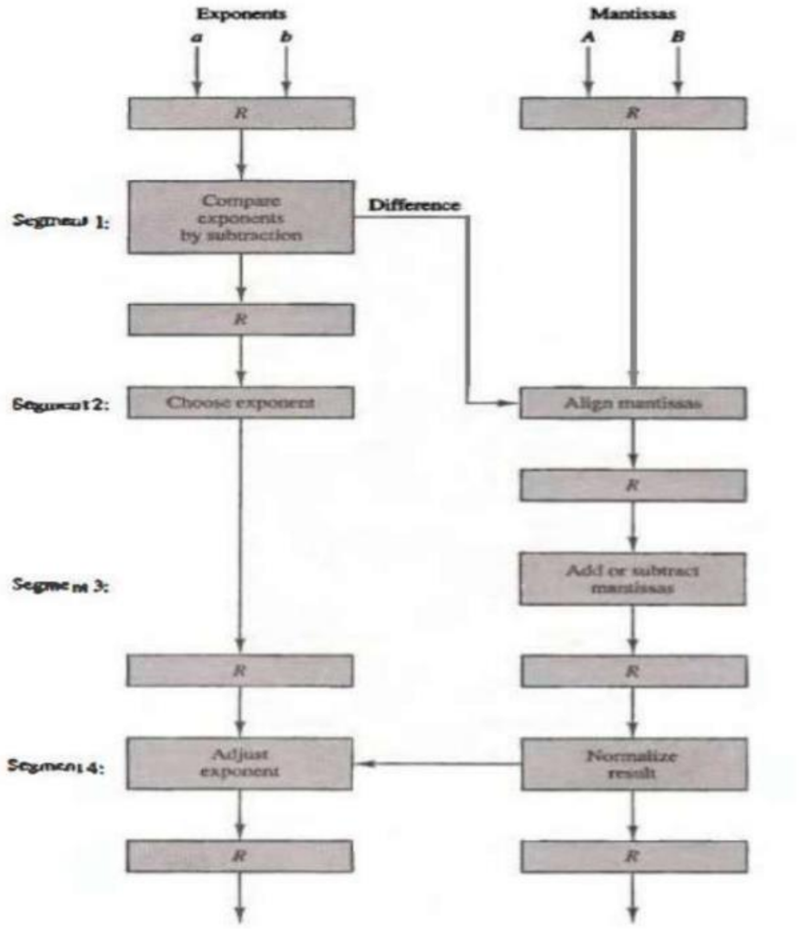


	1	2	3	4	5	6	7	8	9	10
1	FI	DA	FO	EX						
2		FI	DA	FO	Ex					
3			FI	DA	FO	Ex				
4				FI	DA	FO	EX			
5					FI	DA	FO	EX		
6						FI	DA	FO	Ex	
7							FI	DA	FO	Ex

5a

An arithmetic pipeline divides an arithmetic operation into suboperations for execution in the pipeline segment

- Pipeline arithmetic units are usually found in very high speed computers.
- They are used to implement floating-point operations, multiplication of fixed-point
- numbers, and similar computations encountered in scientific problems.
- A pipeline multiplier is essentially an array multiplier as described in Fig. (Next slide)
- With special adders designed to minimize the carry propagation time through
- the partial products.
- Floating-point operations are easily decomposed into suboperations



5b.

Instruction Cycle 1) Fetch the instruction from memory 2) Decode the instruction 3) Calculate the effective address 4) Fetch the operands from memory 5) Execute the instruction 6) Store the result in the proper place

Example : Four-segment Instruction Pipeline Four-segment CPU pipeline :

» 1) FI : Instruction Fetch »
2) DA : Decode Instruction & calculate EA
» 3) FO : Operand Fetch
» 4) EX : Execution

6a. Many scientific problems require arithmetic operations on large arrays of numbers. These numbers are usually formulated as vectors and matrices of floating-point numbers. ν A vector is an ordered set of a one-dimensional array of data items. A vector V of length n is represented as a row vector by V = [V1 V2 V, · · · V.] . It may be represented as a column vector if the data items are listed in a column. ν A conventional sequential computer is capable of processing operands one at a time. Consequently, operations on vectors must be broken down into single computations with subscripted variables ν consider thebfollowing Fortran DO loop:

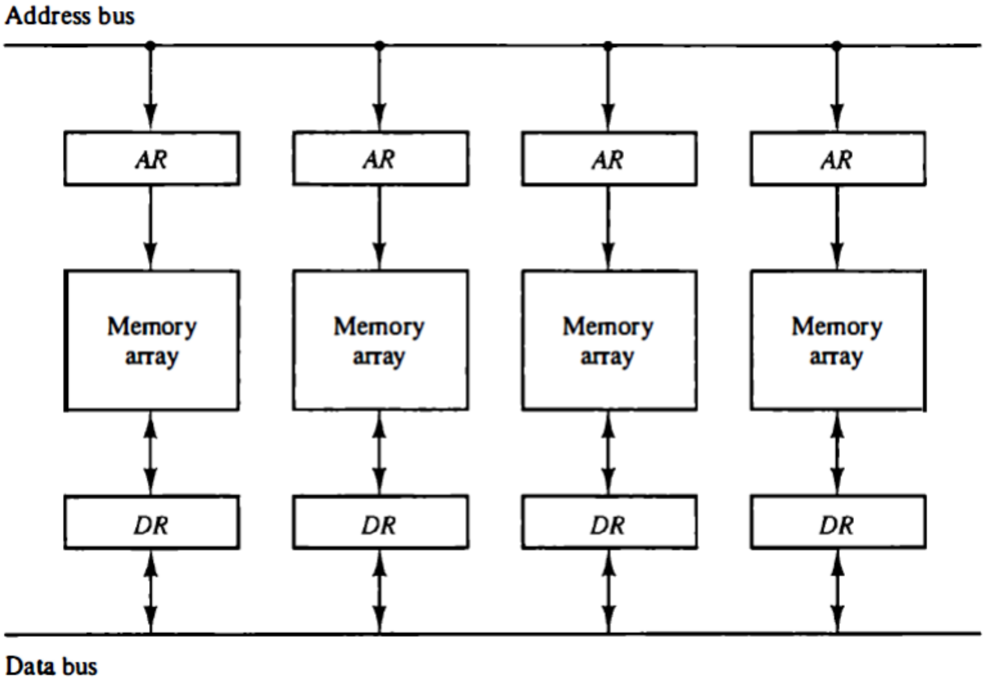
```
DO 20 I = 1 , 100
 20 C ( I ) = B ( I ) + A ( I )
```

This is a program for adding two vectors A and B of length 100 to produce a ν vector C
This is implemented in machine language by the following sequence of operations.

```
 Initialize I = 0
20 Read A ( I )
Read B ( I )
Store C ( I ) = A ( I ) + B ( I )
Increment I = I + 1
If I< 100 Goto 20
```

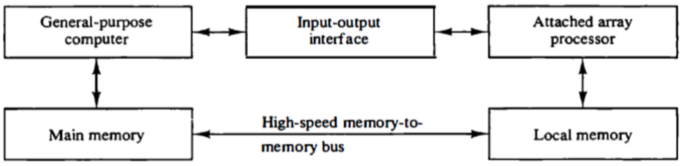
The loop control variable is thenvupdated and the steps repeat 100 times. ν A computer capable of vector processing eliminates the overhead associatedvwith the time it takes to fetch and execute the instructions in the program loop. ν It allows operations to be specified with a single vector instruction of the form C(1 : 100) = A(1 : 100) + B(1 : 100)

Pipeline and vector processors often require simultaneous access to memory from two or more sources ν the memory can be partitioned into a number of modules connected to a common memory address and data buses. A memory module is a memory array together with its own address and data registers. Figure below



- 6b. An array processor is a processor that performs computations on large arrays of data. The term is used to refer to two different types of processors.
- An attached array processor is an auxiliary processor attached to a generalpurpose computer.
- It is intended to improve the performance of the host computer in specific numerical computation tasks.
- An SIMD array processor is a processor that has a single-instruction multiple-data organization Attached Array Processor
- An attached array processor is designed as a peripheral for a conventional host computer, and its purpose is to enhance the performance of the computer by providing vector processing for complex scientific applications
- It achieves high performance by means of parallel processing with multiple functional units.
- It includes an arithmetic unit containing one or more pipelined floating point adders and multipliers. The array processor can be programmed by the user to accommodate a variety of complex arithmetic problems

- Figure shows the interconnection of an attached array processor to a host computer.


General-purpose computer
Input-output interface
Attached array processor
Main memory
High-speed memory-to-memory bus
Local memory