MANAS.P.S
1MS18CS065

Ramaiah Institute of Technology
(Autonomous Institute, Affiliated to VTU)
Department of CSE

Tutorial -II

Programme: B.E
Course: Computer Organization

Term: Jan to May 2020
Course Code: CS45

Name: ~~1MS18CS065~~ MANAS.P.S	**Marks:** 10/10	**Date:** 28/1/2020
USN: 1MS18CS065	**Signature of the Faculty:**	4/2/2020

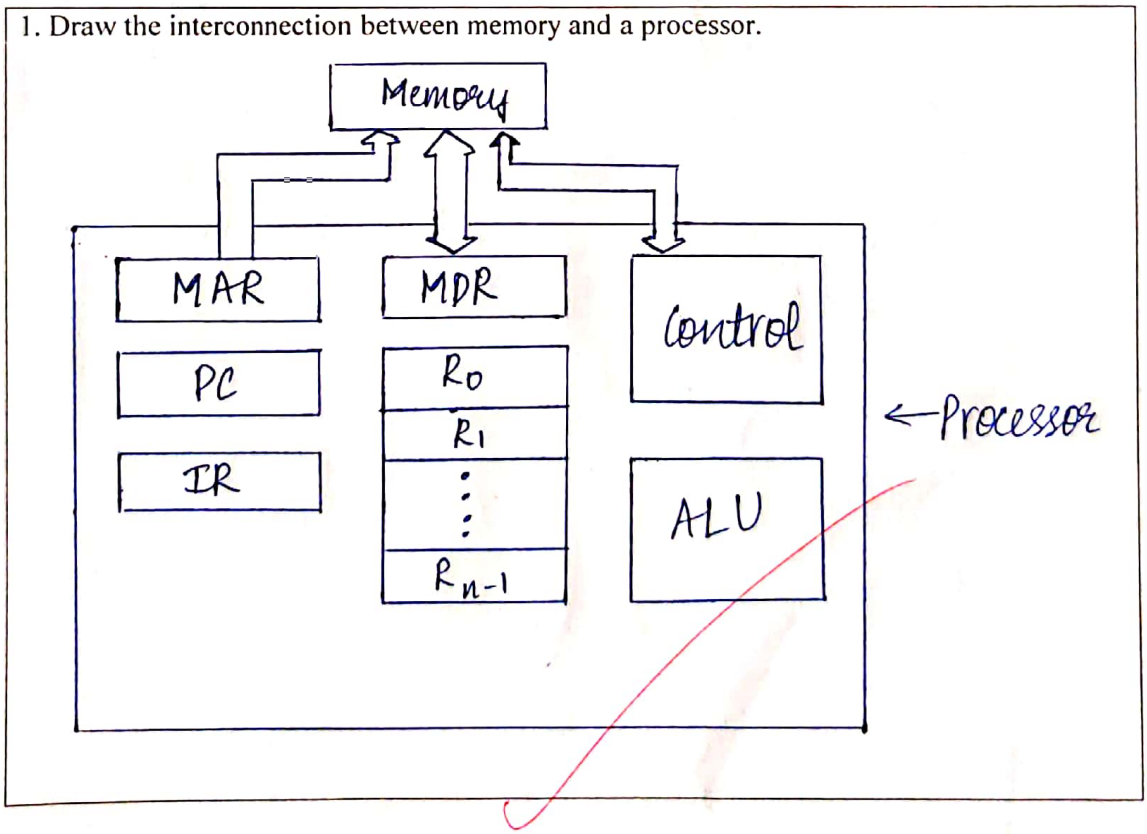
Activity II: Demonstrating Datapath and instruction execution stages using MarieSim Simulator

Objective: To simulate inter communication between CPU and memory.

Simulator Description: MarieSim is a computer architecture simulator based on the MARIE architecture. It provides users with interactive tools and simulations to help them deepen their understanding of the operation of a simple computer. One can observe how assembly language statements affect the registers and memory of a computer system.

Activity to be performed by students:

1. Draw the interconnection between memory and a processor.



2. List out the steps required to execute an instruction.

3. Write and execute assembly language program to compute
   i) f=(g+h)*(i+y)
   ii) d=b^2-4ac

4. Describe the factors affecting the performance of a processor

The major factors affecting performance are :

- Design of the compiler : Speed is improved by pipelining, as simultaneous instructions can be executed
- The machine instruction set : The instruction set used to write machine code also affects performance
- Design of hardware : The speed at which memory is recalled by the processor through a bus reduces drastically if a copy of the memory is placed in the cache. It is later read directly from the cache.

Minor factors affecting performance are :

- Elapsed time : Time taken for whole operation
- Processor time : Time during which processor is active (T= NXS/R)
- Processor clock : Higher the clock rate, higher the no. of basic steps per clock cycle.

3. Results and Snapshots:

Attached

RAMAIAH
Institute of Technology

Data Sheet

MARKS :

Name :	MANAS. P. S	Branch:	CSE
USN/Roll No. :	1MS18CS065	Sem/Sec:	IV 'B'
Subject :	Computer Organization & Architecture	Subject Code:	CS45

(2). List out the steps required to execute an instruction

The steps required to execute an instruction are :

- Fetching an instruction : The instruction is fetched from the main memory. The instruct at the current program counter (PC) will be fetched and stored in Instruction register (IR).
- Decode instruction : During this cycle the encoded instruction is interpreted by the decoder int the IR.
- ALU operation : The two operands in the instruction will be operated on given operator in the instructions.
- Access memory : Only 2 kinds of instructions access memory, LOAD copies a value from memory to the Accumulator (AC) and STORE copies a register value to memory.
- Update Register File : Result of the ALU is written back to the register file to update it.
- Update the PC : We need to update the PC to the address of the next instruction, so that we can go back to step1 where the CPU will fetch instruction.

(3) i] Write and execute the assembly language program to compute : $f = (g+h) / (i+y)$

```
        LOAD G
        ADD H
        STORE A
        LOAD I
        ADD Y
        STORE B
LOOP,   LOAD A
        ADD F
        STORE F
        LOAD B
        SUBT ONE
        STORE B
        SKIPCOND 400
        JUMP LOOP
        LOAD F
        OUTPUT
        HALT
G,      DEC 9
H,      DEC 5
Y,      DEC 8
I,      DEC 2
A,      DEC 0
B,      DEC 0
F,      DEC 0
ONE,    DEC 1
```

ii] $d = b^2 - 4ac$

P.T.O.

```
LOAD B
STORE 0
FIRST, LOAD B
       ADD X
       STORE X
       LOAD 0
       SUBT ONE
       STORE 0
       SKIPCOND 400
       JUMP FIRST

SECOND, LOAD A
        ADD Y
        STORE Y
        LOAD C
        SUBT ONE
        STORE C
        SKIPCOND 400
        JUMP SECOND

THIRD, LOAD FOUR
       ADD Z
       STORE Z
       LOAD Y
       SUBT ONE
       STORE Y
       SKIPCOND 400
       JUMP THIRD

LOAD 0
ADD X
SUBT Z
OUTPUT
HALT
A, DEC 6
B, DEC 3
C, DEC 2
0, DEC 0
X, DEC 0
Y, DEC 0
Z, DEC 0
D, DEC 0
ONE, DEC 1
FOUR, DEC 4
```

IMS18CS065
IV-B
MANAS.P.S

MARIE Assembler Code Editor

File Edit Assemble Help

```
LOAD G
ADD H
STORE G
LOAD I
ADD Y
STORE I
LOAD G
LOOP,    LOAD N
         ADD G
         STORE N

         LOAD I
         SUBT ONE
         STORE I

         SKIPCOND 400
         JUMP LOOP

LOAD N
STORE F
HALT
N,       DEC 0
G,       DEC 1
H,       DEC 1
I,       DEC 2
Y,       DEC 2
F,       DEC 0
ONE,     DEC 1
```

C:\Users\Manas Shankar\Desktop\MarieSim\multAdd\multAdd.mas Assembly successful.

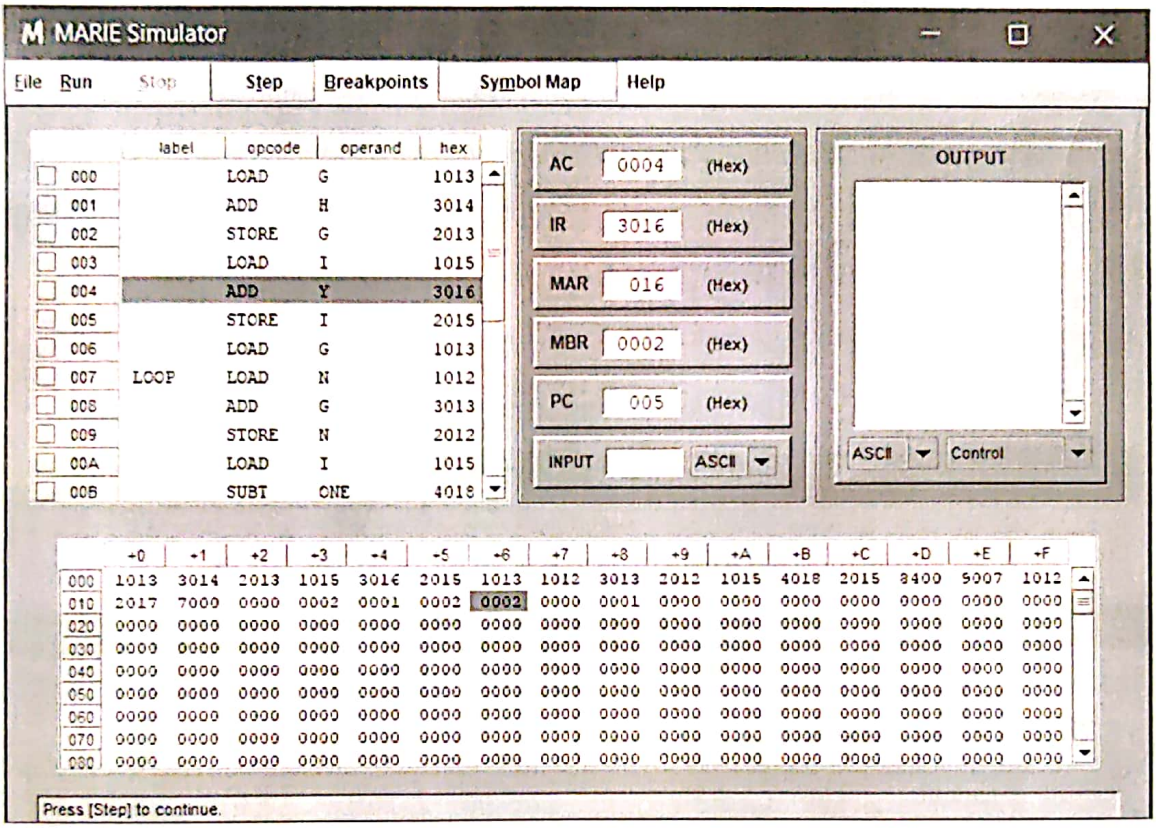
$f = (g+h) * (i+y)$

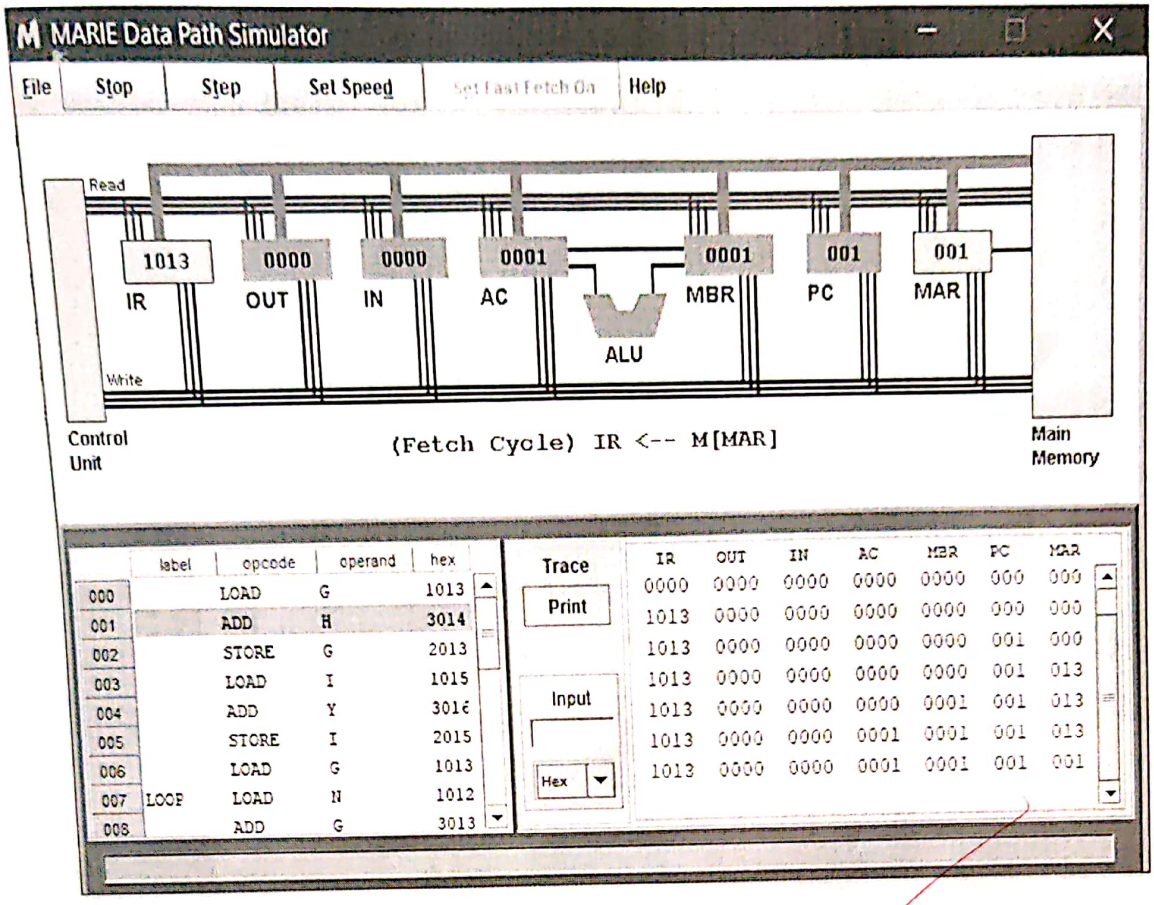
Assembly Listing for multAdd.mas

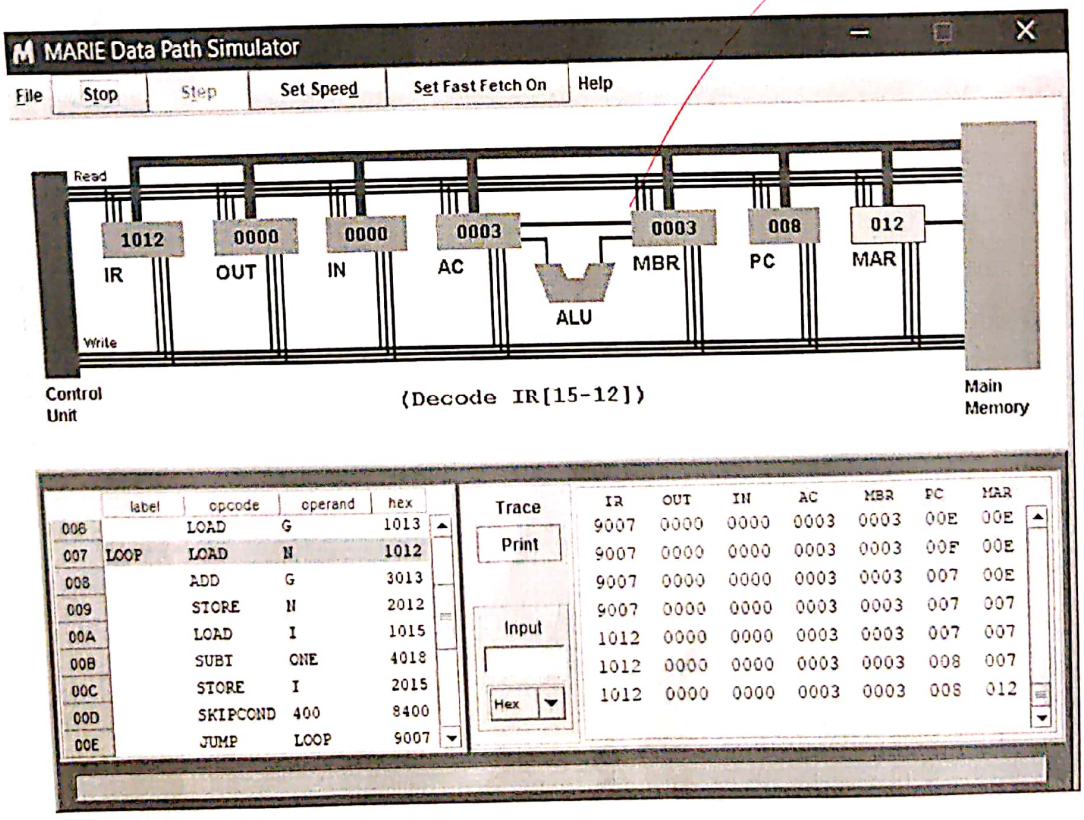
```
000 1013 |        LOAD G
001 2014 |        ADD H
002 2010 |        STORE G
003 1015 |        LOAD I
004 2016 |        ADD Y
005 2015 |        STORE I
006 1013 |        LOAD G
007 1012 | LOOP   LOAD N
008 2013 |        ADD G
009 2012 |        STORE N
         |
00A 1015 |        LOAD I
00B 4018 |        SUBT ONE
00C 2015 |        STORE I
         |
00D 8400 |        SKIPCOND 400
00E 9007 |        JUMP LOOP
         |
00F 1012 |        LOAD N
010 2017 |        STORE F
011 7000 |        HALT
012 0000 | N      DEC 0
013 0001 | G      DEC 1
014 0001 | H      DEC 1
015 0002 | I      DEC 2
016 0002 | Y      DEC 2
017 0000 | F      DEC 0
018 0001 | ONE    DEC 1

Assembly successful.

        SYMBOL TABLE
-------------------------------------------------
 Symbol | Defined | References
-------------------------------------------------
 F      |  017    | 010
 G      |  013    | 000, 002, 00A, 013
 H      |  014    | 001
 I      |  015    | 003, 005, 00B, 00C
 LOOP   |  007    | 00E
 N      |  012    | 007, 009, 00F
 ONE    |  018    | 00B
 Y      |  016    | 004
-------------------------------------------------
```

Print

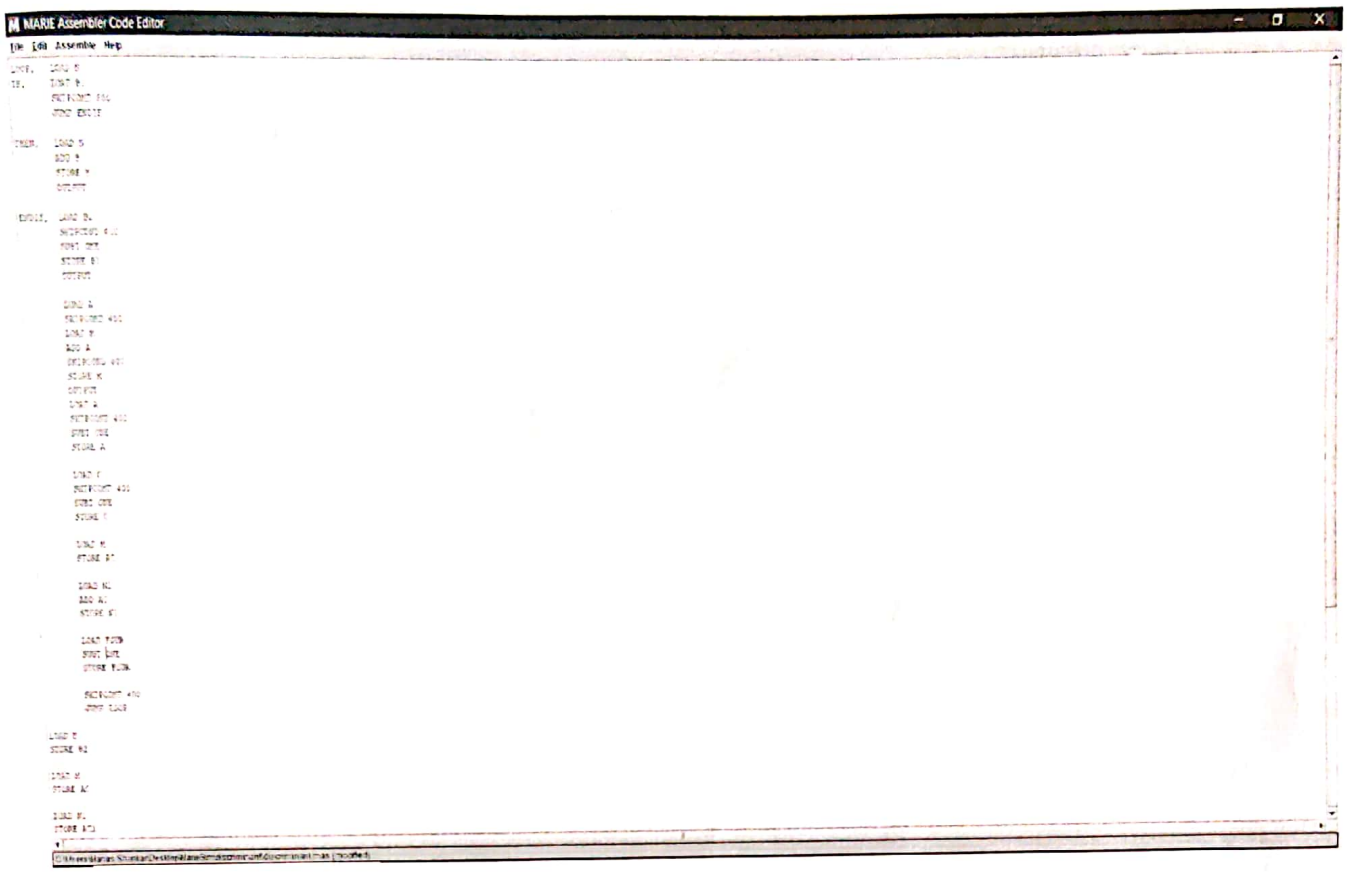
Close







1MS18CS065
IV - B
MANAS. P.S



$d = b^2 - 4 * a * c$