

MANAS.P.S
1M818CS065

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: 1M818CS065 MANAS.P.S	Marks: 10/10	Date: 28/1/2020
USN: 1M818CS065	Signature of the Faculty: <i>[Signature]</i>	

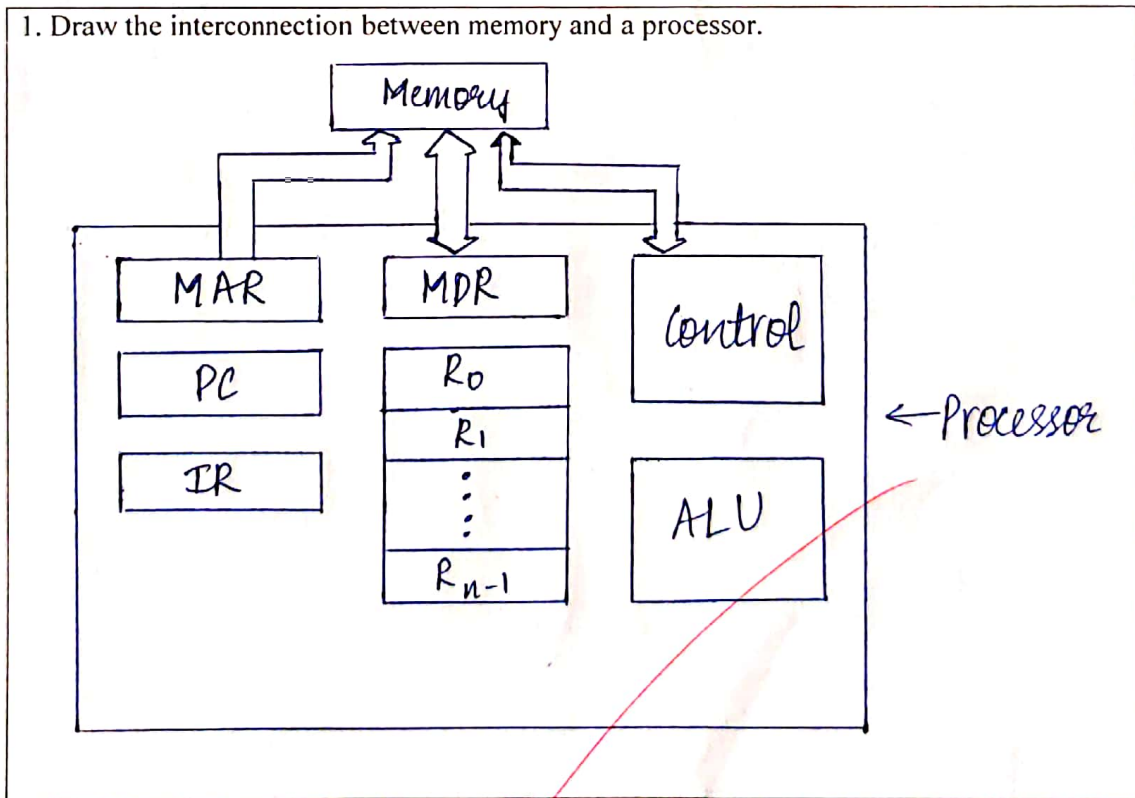
Activity II: Demonstrating Datapath and instruction execution stages using MarieSim Simulator

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



Name :	MANAS. P. S	Branch:	CSE
USN/Roll No. :	1M818CS065	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 instruction 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 into 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 step 1 where the CPU will fetch instruction.

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

```
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
      SKIPLOND 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 O

FIRST, LOAD B
ADD X
STORE X

LOAD O
SUBT ONE
STORE O
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 O
ADD X
SUBT Z
OUTPUT
HALT

A, DEC 6
B, DEC 3
C, DEC 2

O, DEC 0
X, DEC 0
Y, DEC 0
Z, DEC 0
D, DEC 0
ONE, DEC 1
FOUR, DEC 4

1M818C8065
IV-B
MANAS.P.S

```

MARIE Assembler Code Editor
File Edit Assemble Help

LOAD G
ADD H
STORE G
LOAD I
ADD H
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.

```

$$f = (g + h)^x / (i + y)$$

```

Assembly Listing for multAdd.mas

000 0013 | LOAD G
001 0014 | ADD H
002 0015 | STORE G
003 0016 | LOAD I
004 0017 | ADD H
005 0018 | STORE I
006 0019 | LOAD G
007 0020 | LOOP LOAD N
008 0021 | ADD G
009 0022 | STORE N
010 0023 |
011 0024 | LOAD I
012 0025 | SUBT ONE
013 0026 | STORE I
014 0027 |
015 0028 | SKIPCOND 400
016 0029 | JUMP LOOP
017 0030 |
018 0031 | LOAD N
019 0032 | STORE F
020 0033 | HALT
021 0034 |
022 0035 | N, DEC 0
023 0036 | G, DEC 1
024 0037 | H, DEC 1
025 0038 | I, DEC 2
026 0039 | Y, DEC 2
027 0040 | F, DEC 0
028 0041 | ONE, DEC 1

Assembly successful.

SYMBOL TABLE
-----
Symbol | Defined | References
-----
F | 019 | 019
G | 013 | 000, 002, 006, 008
H | 014 | 001
I | 015 | 003, 005, 009, 009
LOOP | 007 | 007
N | 012 | 007, 009, 009
ONE | 019 | 019
Y | 016 | 016

```

MARIE Simulator

File Run Stop Step Breakpoints Symbol Map Help

	label	opcode	operand	hex
<input type="checkbox"/> 000		LOAD	G	1013
<input type="checkbox"/> 001		ADD	H	3014
<input type="checkbox"/> 002		STORE	G	2013
<input type="checkbox"/> 003		LOAD	I	1015
<input checked="" type="checkbox"/> 004		ADD	Y	3016
<input type="checkbox"/> 005		STORE	I	2015
<input type="checkbox"/> 006		LOAD	G	1013
<input type="checkbox"/> 007	LOOP	LOAD	N	1012
<input type="checkbox"/> 008		ADD	G	3013
<input type="checkbox"/> 009		STORE	N	2012
<input type="checkbox"/> 00A		LOAD	I	1015
<input type="checkbox"/> 00B		SUBT	ONE	4018

AC 0004 (Hex)

IR 3016 (Hex)

MAR 016 (Hex)

MBR 0002 (Hex)

PC 005 (Hex)

INPUT ASCII

OUTPUT

ASCII Control

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
000	1013	3014	2013	1015	3016	2015	1013	1012	3013	2012	1015	4018	2015	3400	5007	1012
010	2017	7000	0000	0002	0001	0002	0002	0000	0001	0000	0000	0000	0000	0000	0000	0000
020	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
030	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
040	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
050	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
060	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
070	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
080	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000

Press [Step] to continue.

1M818CS065
IV - B
MANAS. P.S

The screenshot shows the MARIE Assembler Code Editor with the following assembly code:

```

1000  LOAD X
1001  ADD Y
1002  STORE R1
1003  LOAD R1
1004  ADD Z
1005  STORE R2
1006  LOAD R2
1007  MULT R2
1008  STORE R3
1009  HALT

```

The code performs the following steps:

- Load the value of X into the accumulator.
- Add the value of Y to the accumulator.
- Store the result in register R1.
- Load the value of R1 into the accumulator.
- Add the value of Z to the accumulator.
- Store the result in register R2.
- Load the value of R2 into the accumulator.
- Multiply the accumulator by itself (square the sum of X and Y).
- Store the result in register R3.
- Execute the HALT instruction.

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

Assembly Listing for discriminant.mas

```

000 400A I  SUBT AC1
001 2037 I  STORE A00
002 103F I  LOAD A00
003 400C I  OUTPUT
004 7000 I  PAUSE
005 0000 I  R  DEC 0
006 0001 I  C  DEC 1
007 0004 I  B  DEC 4
008 0004 I  B1 DEC 4
009 0000 I  B2 DEC 0
010 0001 I  A  DEC 1
011 0001 I  C  DEC 1
012 0000 I  A0 DEC 0
013 0000 I  A01 DEC 0
014 0000 I  M  DEC 0
015 0000 I  N1 DEC 0
016 0001 I  C0 DEC 1
017 0004 I  FOUR DEC 4
018 0000 I  A02 DEC 0

```

Assembly successful.

SYMBOL TABLE

Symbol	Defined	References
A	007	000, 010, 014, 017
AC	006	010, 017, 020
AC1	00A	000, 000
A00	00F	002, 002
B	004	005
B1	008	001, 001, 008
B2	009	007, 00C
C	005	015, 018
ENDOF	008	003
FOUR	00E	021, 021
IF	001	
LOOP	000	005
M	00B	00F, 012, 01C, 020
N1	002	000, 004, 004, 006
N1	00C	012, 00C, 02A
ONE	000	00A, 014, 01A, 022
Q	003	
THEN	004	

MARIE Simulator

File Run Stop Step Breakpoints Symbol Map Help

label	opcode	operand	hex
016	SUBT	ONE	403D
017	STORE	A	2037
018	LOAD	C	103E
019	SKIPCOND	400	8400
01A	SUBT	ONE	403D
01B	STORE	C	203E
01C	LOAD	M	103B
01D	STORE	AC	2039
01E	LOAD	N1	103C
01F	ADD	AC	3039
020	STORE	N1	203C
021	LOAD	FOUR	103E

AC 0003 (Hex)

IR 103E (Hex)

MAR 03E (Hex)

MBR 0003 (Hex)

PC 022 (Hex)

INPUT ASCII

OUTPUT

4
3
1
8
2
0

Dec Control

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
000	1032	1035	8000	5008	1032	3034	2032	6000	1035	8400	403D	2035	6000	1037	8400	103B
010	3027	8400	203B	6000	1037	8400	403D	2037	1038	8400	403D	2038	103B	2035	103C	3039
020	203C	103E	403D	203E	8400	5000	1032	2036	103B	1035	103C	203A	1036	403A	203F	103F
030	6000	7000	0008	0001	0004	0002	0000	0000	0000	0001	0000	0001	0002	0001	0003	0000
040	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
050	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
060	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
070	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
080	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000

Press [Step] to continue

