

CS 271 Computer Architecture

Lab1 (Due on Oct. 20, 4:30pm in ET-125E)

The following problems are based on Chapter 4 of Textbook "*Essentials of Computer Organization and Architecture*" by Linda Null. Jones & Bartlett Pub. 4th edition.

Please Download "MARIE and Datapath Simulators" package from the following link:

<http://computerscience.ibpub.com/ecoa/3e/simulators.aspx>

In the package, there are four examples provided.

Use the simulator to write the MARIE assembly code and show the running results. In the submission, please hand in both source code (i.e., MARIE assembly language code) and the running results (i.e., screen shot) in MARIE simulator.

- (1) Find the seventh value of the Fibonacci number sequence. The following formula describes the sequence:

$$\text{Fib}(1) = 1, \text{Fib}(2) = 1, \text{Fib}(n) = \text{Fib}(n-1) + \text{Fib}(n-2).$$

The program calculates $\text{Fib}(7)$.

- (2) Print "Hello CS271" to output.
- (3) In textbook, page 286, problem 29. Consider two cases for the initial value of X: a) $X = 1$; b) $X = 2$. Assume $Y = 0$ initially.
- (4) Create a subroutine to implement the following statement in C language:

`int num = condition ? First number : Second number`

That is, if "condition" is 0, "num" gets the value of "Second number"; otherwise, it gets the value of "First number". Consider two cases: a) condition = 1; b) condition = 0. Assume "First number" is 10, and "Second number" is 20.

Please submit the hard copy of this lab, as well the running results in MARIE simulator by outputting the results to Output window or circling the results shown in register or in memory.

(1)

```

/ Lab 1.1
/ CS 271
/
/ 1
/ 1
/ 1+1 = 2
/ 1+2 = 3
/ 2+3 = 5
/ 3+5 = 8
/ 5+8 = 13 <-- Target
/ 8+13 = 21
/

ORG 100

        LOAD      ONE      / Load 1 into AC.
        Output    / Output 1.

Cond,   LOAD      COUNT    / Load Count into AC
        SUBT      SIX      / Remove 6 from the count
        SKIPCOND  000      / Skip next instruction if AC < 0
        JUMP      End      / Jump to the End Section which ends the loop.

Loop,   LOAD      COUNT    / Load Count into AC.
        ADD       ONE      / Increment Count by 1
        STORE     COUNT    / Store AC in count
        JNS       Fibb     / Jump to Fibb section and store JNS.
        JUMP      Cond     / Jump to Cond section.

Fibb,   HEX       000      / Store value for JNS.
        CLEAR                    / Clear the AC.
        ADD       F1        / Add F1 to AC.
        ADD       F2        / Add F2 to AC.
        STORE     Fi        / Store AC contents into Fi.
        LOAD      F2        / Load F2 into AC.
        STORE     F1        / Store F2 into F1.
        LOAD      Fi        / Load Fi into AC.
        STORE     F2        / Store Fi into F2.
        LOAD      Fi        / Load Fi into AC.
        OUTPUT                    / Output the current Fibonacci.

        JUMPI     Fibb     / Jump back to caller.

End,    HALT                    / Terminate program.

COUNT, DEC      0          / declare variable to hold count for loop
Fi,     DEC      0          / declare variable to hold the result of F1+F2
F1,     DEC      0          / declare variable to hold the Fib(n-1)
F2,     DEC      1          / declare variable to hold the Fib(n-1)
ZERO,   DEC      0          / declare variable to hold 0
ONE,     DEC     1          / declare variable to hold 1
TWO,     DEC     2          / declare variable to hold 2
THREE,   DEC     3          / declare variable to hold 3
FOUR,    DEC     4          / declare variable to hold 4
FIVE,    DEC     5          / declare variable to hold 5
SIX,     DEC     6          / declare variable to hold 6

```

MARIE Simulator

File

Run

Stop

Step

Breakpoints

Symbol Map

Help

	label	opcode	operand	hex
<input type="checkbox"/> 10C		CLEAR		A000
<input type="checkbox"/> 10D		ADD	F1	311A
<input type="checkbox"/> 10E		ADD	F2	311B
<input type="checkbox"/> 10F		STORE	F1	2119
<input type="checkbox"/> 110		LOAD	F2	111B
<input type="checkbox"/> 111		STORE	F1	211A
<input type="checkbox"/> 112		LOAD	F1	1119
<input type="checkbox"/> 113		STORE	F2	211B
<input type="checkbox"/> 114		LOAD	F1	1119
<input type="checkbox"/> 115		OUTPUT		6000
<input type="checkbox"/> 116		JUMPI	Fibb	C10B
<input type="checkbox"/> 117	End	HALT		7000

AC0000(Hex)

IR7000(Hex)

MAR117(Hex)

MBR0006(Hex)

PC118(Hex)

INPUT

ASCII

OUTPUT

1123

DecControl

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
0B0	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
0C0	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
0D0	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
0E0	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
0F0	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
100	111D	6000	1118	4122	8000	9117	1118	311D	2118	010B	9102	010A	A000	311A	311B	2119
110	111B	211A	1119	211B	1119	6000	C10B	7000	0006	000D	0008	000D	0000	0001	0002	0003
120	0004	0005	0006	0007	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
130	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000

Machine halted normally.

(2)

```
|/
/ Matthew Hunt
/ 10/20/2016
/ Lab 1.2
/ CS 271
/
      ORG 100                / Store information in memory starting at address 100.

Getch, LoadI    Chptr      / Load the character found at address chptr.
      Skipcond  400        / If the character is a null, we are done.
      Jump      Outp       / Otherwise, jump to output section.
      Halt                      / End Operation

Outp,  Output                / Output the character.
      Load      Chptr       / Load the character pointer.
      Add       One         / Add one to the character pointer.
      Store     Chptr       / Store the character pointer.
      Jump      Getch       / Jump back to get character section.

One,   Hex      0001        / Declare hex varriable and assign it to 1.
Chptr, Hex      10B        / Declare hex varriable and assign it to 10B.
String, Dec     072        / Declare decimal varriable and assign it to ASCII value of H.
      Dec     101        / Declare decimal varriable and assign it to ASCII value of e.
      Dec     108        / Declare decimal varriable and assign it to ASCII value of l.
      Dec     108        / Declare decimal varriable and assign it to ASCII value of l.
      Dec     111        / Declare decimal varriable and assign it to ASCII value of o.
      Dec     032        / Declare decimal varriable and assign it to ASCII value of [space].
      Dec     067        / Declare decimal varriable and assign it to ASCII value of C.
      Dec     083        / Declare decimal varriable and assign it to ASCII value of S.
      Dec     050        / Declare decimal varriable and assign it to ASCII value of 2.
      Dec     055        / Declare decimal varriable and assign it to ASCII value of 7.
      Dec     049        / Declare decimal varriable and assign it to ASCII value of 1.
      Dec     000        / Declare decimal varriable and assign it to ASCII value of [null].
      END                / End of file.
```

	label	opcode	operand	hex
<input type="checkbox"/> 100	Getch	LOADI	Chptr	D10A
<input type="checkbox"/> 101		SKIPCOND	400	8400
<input type="checkbox"/> 102		JUMP	Outp	9104
<input type="checkbox"/> 103		HALT		7000
<input type="checkbox"/> 104	Outp	OUTPUT		6000
<input type="checkbox"/> 105		LOAD	Chptr	110A
<input type="checkbox"/> 106		ADD	One	3109
<input type="checkbox"/> 107		STORE	Chptr	210A
<input type="checkbox"/> 108		JUMP	Getch	9100
<input type="checkbox"/> 109	One	HEX	0001	0001
<input type="checkbox"/> 10A	Chptr	HEX	10B	010B
<input type="checkbox"/> 10B	String	DEC	072	0048

AC 0000 (Hex)

IR 7000 (Hex)

MAR 103 (Hex)

MBR 0000 (Hex)

PC 104 (Hex)

INPUT ASCII

OUTPUT

H
e
l
l
o

C
S
2

ASCII

Control

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
080	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
090	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
0A0	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
0B0	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
0C0	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
0D0	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
0E0	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
0F0	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
100	D10A	8400	9104	7000	6000	110A	3109	210A	9100	0001	0116	0048	0065	006C	006C	006F

Machine halted normally.

(3)

```
/
/ Matthew Hunt
/ 10/20/2016
/ Lab 1.3
/ CS 271
/
/ if X > 1 then
/     Y = X + X;
/     X = 0;
/ endif;
/ Y = Y + 1;
/
/ Case: X = 0 then Y should be 1
/ Case: X = 1 then Y should be 1
/ Case: X = 2 then Y should be 5
/

                ORG 100

If,      Load      X      / Load X into AC.
        Subt       One    / Subtract one from X.
        Store      Temp   / Store X-1 in Temp.
        Load      Temp   / Load temp into AC
        Skipcond   400    / If AC = 0, skip the next instruction
        Jump      Then   / Jump to Then section.
        Jump      EndIf  / Jump to EndIf section.

Then,    Load      Y      / Load Y into AC.
        Add       X      / Y+X
        Add       X      / Y+X
        Store     Y      / Store AC into Y.
        Load     X      / Load X into AC.
        Subt     X      / X-X
        Store     X      / Set X to zero.
        Jump     EndIf  / Jump to EndIf section.

EndIf,   Load      Y      / Load Y into AC.
        Add       One    / Y+1
        Store     Y      / Store Y+1
        Load     Y      / Load Y into AC.
        Output    / Output AC to console.
        Halt     / Terminate program

X,       Dec       02     / Declare variable to hold X.
Y,       Dec       00     / Declare variable to hold Y.
Temp,    Dec       00     / Declare variable to hold temporary value of X.
One,     Dec       01     / Declare varriable to hold one.
END
```

	label	opcode	operand	hex
<input type="checkbox"/> 109		ADD	X	3115
<input type="checkbox"/> 10A		STORE	Y	2116
<input type="checkbox"/> 10B		LOAD	X	1115
<input type="checkbox"/> 10C		SUBT	X	4115
<input type="checkbox"/> 10D		STORE	X	2115
<input type="checkbox"/> 10E		JUMP	EndIf	910F
<input type="checkbox"/> 10F	EndIf	LOAD	Y	1116
<input type="checkbox"/> 110		ADD	One	3118
<input type="checkbox"/> 111		STORE	Y	2116
<input type="checkbox"/> 112		LOAD	Y	1116
<input type="checkbox"/> 113		OUTPUT		6000
<input type="checkbox"/> 114		HALT		7000

AC	0005	(Hex)
IR	7000	(Hex)
MAR	114	(Hex)
MBR	0005	(Hex)
PC	115	(Hex)
INPUT		ASCII ▼

OUTPUT	
5	
Dec ▼	Control ▼

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
0B0	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
0C0	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
0D0	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
0E0	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
0F0	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
100	1115	4118	2117	1117	8400	9107	910F	1116	3115	3115	2116	1115	4115	2115	910F	1116
110	3118	2116	1116	6000	7000	0000	0005	0001	0001	0000	0000	0000	0000	0000	0000	0000
120	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
130	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000

Machine halted normally.

(4)

```
1  #include <stdio.h>
2
3  /*Create a subroutine to implement the following statement in C language:
4  int num = condition ? First number : Second number
5  That is, if "condition" is 0, "num" gets the value of "Second number"; otherwise, it gets
6  the value of "First number". Consider two cases: a) condition = 1; b) condition = 0.
7  Assume "First number" is 10, and "Second number" is 20.
8  */
9
10 int x = 10;
11 int y = 20;
12
13 int check_condition(int condition)
14 {
15     if(condition == 0) {
16         return x;
17     }else{
18         return y;
19     }
20 }
21
22 int main()
23 {
24     int condition;
25     condition = 0;
26     //condition = 1;
27
28     printf("%d\n", check_condition(condition));
29 }
30
```

bash - "masterprogr: × Immediate × application.c - Stop × +

Run Run Config Name Command: application.c

Running "/home/ubuntu/workspace/application.c"

10


```

/
/ Matthew Hunt
/ 10/20/2016
/ Lab 1.4
/ CS 271
/
/ Case: cond = 0 then Y should be output
/ Case: cond = 1 then X should be output
/

    ORG     100

    Load   X       / Load X into AC
    Store   Temp    / Store X in Temp.
    Load   Cond     / Load Cond into AC.
    Skipcond 400    / Skip next instruction if X is zero.
    JnS     Subr     / Store the return address, and jump to the procedure.

    Load   Y       / Load Y into Ac.
    Store   Temp    / Store Y in Temp.
    Load   Cond     / Load Cond into AC
    Skipcond 800    / Skip if Cond is 1.
    JnS     Subr     / Store the return address and jump to the procedure.

    Load   Num      / Load Num into AC.
    Output          / Output Num to console.
    Halt           / End program.

X,    DEC     10    / Declare varriable to hold X.
Y,    DEC     20    / Declare varriable to hold Y.
Num,  DEC     00    / Declare varriable to hold the number to output
Cond, DEC     01    / Declare varriable to hold the condition of 1 or 0.
Temp, DEC     0     / Declare varriable to to temporarily hold X and Y.
Subr, HEX     0     / Declare varriable to hold return address.

    Load   Temp    / Load Number into AC.
    Store   Num     / Store Temp in Num.
    JumpI   Subr     / Return to caller.
END

```

MARIE Simulator

File

Run

Stop

Step

Breakpoints

Symbol Map

Help

	label	opcode	operand	hex
<input type="checkbox"/>	105	LOAD	Y	110E
<input type="checkbox"/>	106	STORE	Temp	2111
<input type="checkbox"/>	107	LOAD	Cond	1110
<input type="checkbox"/>	108	SKIPCOND	800	8800
<input type="checkbox"/>	109	JNS	Subr	0112
<input type="checkbox"/>	10A	LOAD	Num	110F
<input type="checkbox"/>	10B	OUTPUT		6000
<input type="checkbox"/>	10C	HALT		7000
<input type="checkbox"/>	10D	X	DEC	10
<input type="checkbox"/>	10E	Y	DEC	20
<input type="checkbox"/>	10F	Num	DEC	00
<input type="checkbox"/>	110	Cond	DEC	01

AC000A(Hex)

IR7000(Hex)

MAR10C(Hex)

MBR000A(Hex)

PC10D(Hex)

INPUT

ASCII

OUTPUT

10

DecControl

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
0B0	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
0C0	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
0D0	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
0E0	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
0F0	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
100	110D	2111	1110	8400	0112	110E	2111	1110	8800	0112	110F	6000	7000	000A	0014	0000
110	0001	0014	0105	1111	210F	C112	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
120	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
130	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000

Machine halted normally.