Gasser Ahmed
gasser18@vt.edu
ECE 5484, Homework 4

1. Consider the MARIE program below.

a) List the hexadecimal code for each line of the program (including the symbols).

	hex		
100	1109	Begin	LOAD Base
101	310A		ADD Offs
102	410B	Loop	SUBT Two
103	210C		STORE Addr
104	8800		SKIPCOND 800
105	9108		JUMP Done
106	A000		CLEAR
107	110A		LOAD Offs
108	7000	Done	HALT
109	0200	Base	HEX 200
10A	000B	Offs	DEC 11
10B	0002	Two	HEX 0002
10C	0007	Addr	HEX 007

b) Draw the symbol table.

Symbol	Location
Addr	10C
Base	109
Begin	100
Done	108
Loop	102
Offs	10A
Two	10B

c) What is the value stored in the AC when the program terminates?
000B (Hex)

---

2. Write the assembly language equivalent of the following MARIE machine language instructions:

a) 0100 0001 1000 0101
Subt 185

b) 1000 0100 0000 0000
Skipcond 400

c) 0111 0000 0000 0000
Halt

3.

```
        ORG 100
If,     Load      X       /Load X
        Subt      One     /Subtract 1, store result in AC
        Skipcond  000     /If AC < 0 (X < 1), skip the next instruction
        Jump      Endif   /Jump to Endif if X is not less than 1
Then,   Load      X       /Reload X so it can be subtracted
        Subt      Two     /Subtract 2 from X
        Store     Y       /Y  = X - 2
        Clear             /Move 0 into AC
        Store     X       /Set X to 0
Endif,  Load      Y       /Load Y into AC
        Add       Seven   /Add 7 to Y
        Store     Y       /Y = Y + 7
        Halt              /Terminate program
X,      Dec ?             /X has starting value, not given in problem
Y,      Dec ?             /Y has starting value, not given in problem
One,    Dec 1             /Use as a constant
Two,    Dec 2             /Use as a constant
Seven,  Dec 7             /Use as a constant
```

---

4. Fill in the following table to show how the given integers are represented, assuming 16-bits are used to store values and the machine uses 2's complement notation.

Integer	Binary	Hex	2 Byte Big Endian	2 Byte Little Endian
7	0000000000000111	0007	0007	0700
1329	0000010100110001	0531	0531	3105
-7	1111111111111001	FFF9	FFF9	F9FF
-27109	1001011000011011	961B	961B	1B96
31307	0111101001001011	7A4B	7A4B	4B7A

5.

Mode	Value loaded into AC
Immediate	0x600
Direct	0x500
Indirect	0x100
Indexed	No Answer

6. Since all the stages of the pipeline can work individually on an operation, it can process four instructions at the same time. So, in the first step, only the first stage is occupied with the first part of executing the instruction. Then in the second step, the second stage of the pipeline is processing the second step of the first instruction, while the first stage is staring to process a new one and so on for the rest of the stages. Once the pipeline is in full operation, it is constantly working on 4 operations simultaneously/concurrently, starting a new one and finishing one in each cycle. Accordingly, it ends up four times faster than one without a pipeline.

7. John Cocke: He developed the concept of RISC by reducing the number of instructions required for processing computations faster than the CISC (Complex Instruction Set Computer) and designed chips that performed simple instructions very quickly.

David Patterson: He coined the term RISC, and leaded the Berkeley RISC project that created a resultant chip, known as RISC-1, with 44,420 transistors.

References

1. *Schofield, Jack (2002-07-27). "John Cocke". The Guardian. Guardian Media Group. Retrieved 2011-05-10.*
2. *"People of ACM - David Patterson". www.acm.org.*