Adnar Lozano
CSE 313 Section 03
CSUSB Spring 17
05/05/17

Homework # 2

Chapter 4: 4.5b, 4.7, 4.9, 4.11

4.5 The following table represents a small memory. Refer to this table for the following questions.

b. The binary value within each location can be interpreted in many ways. We have seen that binary values can represent unsigned numbers, 2's and so forth.

Address	Data
0000	0001 1110 0100 0011
0001	1111 0000 0010 0101
0010	0110 1111 0000 0001
0011	0000 0000 0000 0000
0100	0000 0000 0110 0101
0101	0000 0000 0000 0110
0110	1111 1110 1101 0011
0111	0000 0110 1101 1001

(1) Interpret location 0 and location 1 as 2's complement integers.

Ans: Location 0: 0001 1110 0100 0011

2^12 + 2^11 + 2^10 + 2^9 + 2^6 + 2 + 1

4096 + 2048 + 1024 + 512 + 64 + 2 + 1 = **7747**

Location 1: 1111 0000 0010 0101

2's compl: 0000 1111 1101 1011

- (2^11 + 2^10 + 2^9 + 2^8 + 2^7 + 2^6 + 2^4 + 2^3 + 2 + 1)
- (2048 + 1024 + 512 + 256 + 128 + 64 + 16 + 8 + 2 + 1) = **- 4059**

(2) Interpret location 4 as an ASCII value.

Ans: Location 4: 0000 0000 0110 0101 = 64 + 32 + 4 + 1 = 101 in ASCII = **e**

(3) Interpret location 6 and 7 as an IEEE floating point number. Location 6 contains number [15:0]. Location 7 contains number [31:16]

Ans: Locations 6 and 7: 0000 0110 1101 1001 1111 1110 1101 0011

Signed bit: 0 = Positive
Exponent: 0000 1101 = 8 + 4 + 1 = 13
Fraction: 1011 0011 1111 1101 1010 011
Formula: (-1)^(s) x 1.Fraction x 2^(Exponent - 127)
(-1)^0 x 1.Fraction x 2^(13-127)

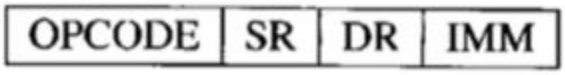
Number represented: **1.10110011111111011010011 x 2^-114**

(4) Interpret location 0 and location 1 as unsigned integers.

Ans: Location 0: 0001 1110 0100 0011
2^12 + 2^11 + 2^10 + 2^9 + 2^6 + 2 + 1
4096 + 2048 + 1024 + 512 + 64 + 2 + 1 = **7747**

Location 1: 1111 0000 0010 0101
2^15 + 2^14 + 2^13 + 2^12 + 2^5 + 2^2 + 1
32768 + 16384 + 8192 + 4096 + 32 + 4 + 1 = **61477**

4.7 Suppose a 32-bit instruction takes the following format:
If there are 60 opcodes and 32 registers, what is the range of values that can be represented by the immediate (IMM)? Assume IMM is a 2's complement value.



Ans: 60 opcodes = $log_2(60)$ = 6 bits
32 registers = $log_2(32)$ = 5 bits
n-bits for IMM: 32 - OPCODE - SR - DR = 32 - 6 - 5 - 5 = 16
| OPCODE 6-bits | SR 5-bits | DR 5-bits | IMM 16-bits |
2's complement: Max int: 2^(n-1) - 1 = 2^(16-1) - 1 = **32767**
Min int: -2^(n-1) = -2^(15) = **- 32768**

4.9 The FETCH phase of the instruction cycle does two important things. One is that it loads the instruction to be processed next into the IR. What is the other important thing?

Ans: The other important thing the FETCH phase does is the loading of the address of the next instruction into the PC (Program Counter).

4.11 State the phases of the instruction cycle and briefly describe what operations occur in each phase.

Ans: The phases for the instruction cycle are as follows:

Fetch

- Gets instructions from memory and loads the address of the next instruction in the PC

Decode

- Analyses what the instruction does

Evaluate Address:

- Calculates the address of the memory location that is needed to process the instruction

Fetch Operands

- Gets the source operands either from memory or Registers

Execute

- Performs the execution of the instruction

Store Result

- Stores the result of the execution to the specified destination

Chapter 5: 5.1, 5.4, 5.8, 5.15

5.1 Given instructions ADD, JMP, LEA, and NOT, identify whether the instructions are operate instructions, data movement instructions, or control instructions. For each instruction, list the addressing modes that can be used with the instruction.

Ans: ADD

- Operates
- Register for Destination Source 1
- Register or IMM for addressing for source 2

JMP

- Controls
- Register addressing

LEA

- Data Movement
- Immediate addressing (IMM)

NOT

- Operates
- Register addressing

5.4 Say we have a memory consisting of 256 locations, and each location contains 16 bits.

a. How many bits are required for the address?

Ans: We need to take the log base 2 of 256

$log_2(256)$ = 8 bits

b. If we use the PC-relative addressing mode, and want to allow control transfer between instructions 20 locations away, how many bits of a branch instruction are needed to specify the PC-relative offset?

Ans: We need at least 5 bits. This would allow the address to be within -32 or + 31 locations since the PC-relative offset are in 2's complement.

C. If a control instruction is in location 3, what is the PC-relative offset of address 10. Assume that the control transfer instructions work the same way as in the LC-3

Ans: If control instruction is in location 3, then PC is at location 4 because it increments by 1. Then, we subtract 10 - 4 to get 6.
This means that the PC-relative offset of address 10 is: 10 - 4 = 6

5.8 We want to increase the number of registers that we can specify in the LC-3 ADD instruction to 32. Do you see any problem with that? Explain.

Ans: Yes, because a 32-bit word or register would require 5-bits to represent each GPRs.
The opcode for the LC-3 ADD requires 4-bits, but we need to be able to express operations using 3 GPRs for the ADD opcode. This can be a problem because it would require at least 19-bits to represent the instruction.
| ADD 4-bits | DR 5-bits | SR1 5-bits | SR2 5-bits |
Register addressability: 4 + 5 + 5 + 5 = 19-bits

5.15 State the contents of R1, R2, R3, and R4 after the program starting at location x3100 halts.

Address	Data
0011 0001 0000 0000	1110 001 000100000
0011 0001 0000 0001	0010 010 000100000
0011 0001 0000 0010	1010 011 000100000
0011 0001 0000 0011	0110 100 010 000001
0011 0001 0000 0100	1111 0000 0010 0101
:	:
:	:
0011 0001 0010 0010	0100 0101 0110 0110
0011 0001 0010 0011	0100 0101 0110 0111
:	:
:	:
0100 0101 0110 0111	1010 1011 1100 1101
0100 0101 0110 1000	1111 1110 1101 0011

Ans:

1110 001 000100000	(LEA R1, 0x20)	R1 <- 0x3121
0010 010 000100000	(LD R2, 0x20)	R2 <- Mem[0x3122] = 0x4566
1010 011 000100001	(LDI R3, 0x20)	R3 <- Mem[Mem[0x3123]] = 0xabcd
0110 100 010 000001	(LDR R4, R2, 0x1)	R4 <- Mem[R2 + 0x1] = 0xabcd
1111 0000 0010 0101	(TRAp 0x25)	Halts the program