University of Victoria

Ceng 255 : Introduction to Computer Architecture

Section F01

Midterm Solution

Oct. 18, 2007

To be answered on the paper. Closed book examination.

Students must count the number of pages in this examination paper before beginning to write and report any discrepancy immediately to the invigilator.

This question paper has 8 pages.

Duration: 80 Minutes

Name: Student No.:

Question	1	2	3	4	5	Total
Maximum Mark	10	10	10	10	10	50
Earned Marks						

Hints

Addressing Modes:

Immediate	#value	operand = value
Register	R*i*	EA = R*i*
Absolute (Direct)	LOC	EA = LOC
Indirect	(R*i*) (LOC)	EA = [R*i*] EA = [LOC]
Index	X(R*i*)	EA = [R*i*] + X
Base with index	(R*i*, R*j*)	EA = [R*i*] + [R*j*]
Base with index and offset	X(R*i*, R*j*)	EA = [R*i*] + [R*j*] + X
Autoincrement	(R*i*)+	EA = [R*i*]; Increment R*i*
Autodecrement	-(R*i*)	Decrement R*i*; EA = [R*i*] + X

Binary Number Representation Schemes

- Sign and magnitude where the most significant bit being 0 for a positive number and 1 for a negative number.
- One's complement where a negative number is represented by the complement of its positive representation.
- Two's complement where a negative number is derived by adding 1 to its one's complement.

Basic Performance Equation

$$T = N \times S / R$$

- T = Time required to execute a program
- N = Number of machine language instructions
- S = Number of clock cycles per machine instruction
- R = Clock rate in cycles per second (Hz)

Question 1 [10 Marks]

a) You are told that a particular CISC processor executes 5 million instructions per second (MIPS). You now have to design a RISC processor that executes a program at least as fast as this CISC processor. You know that for a given program the RISC processor will require 1.5 times the number of instructions as compared to the CISC processor. How many instructions per second does the RISC processor need to execute to compete with the CISC processor? [5 Marks]

Let M = instructions per second = R/S

Then T = N/M

$M_{CISC} = 5$

$N_{RISC} = 1.5 * N_{CISC}$

$T_{RISC} = T_{CISC}$

$N_{RISC} / M_{RISC} = N_{CISC} / M_{CISC}$

$M_{RISC} = M_{CISC} * N_{RISC} / N_{CISC}$

$M_{RISC} = 5 * 1.5 = 7.5$ MIPS

b) Given the basic performance equation, a designer's goal is to decrease T. This can be done by decreasing N and S and by increasing R. Discuss how each of these objectives can be achieved. [5 Marks]

N: Better compilation to reduce the number of instructions executed. Code inlining. Loop unrolling.

S: Pipelining. Superscalar.

R: Better circuit technology to increase the clock rate.

Question 2 [10 Marks]

a) What is the range of numbers that can be represented with 8 bits using the following encoding methods: [4 Marks]

i. Unsigned: 0 to 255

ii. Sign and Magnitude: -127 to 127

iii. 1's Complement: -127 to 127

iv. 2's Complement: -128 to 127

b) Assuming 8 bit 2's complement encoding, perform the following arithmetic additions/subtractions. (indicate if there is an overflow): [4 Marks]

```
  01101011
+ 11001100
----------
  00110111  (No Overflow)
```

```
  10001111
- 00010010
----------
```

```
  10001111
+ 11101110
----------
  01111101  (Overflow)
```

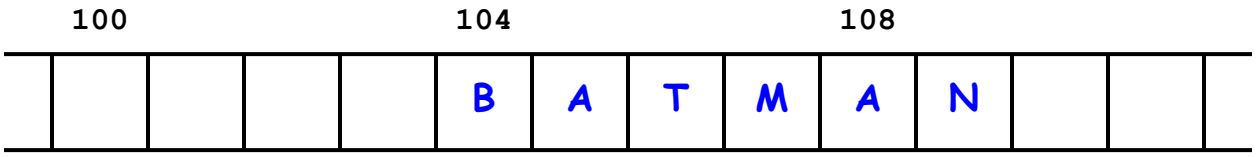
c) Assuming 8 bit (one byte) 2's complement numbers a computer needs to perform the calculation X – Y. If X = -100, what is the range of numbers for Y that would result in an overflow? [2 Marks]

Y = 29 to Y = 127 will result in an overflow

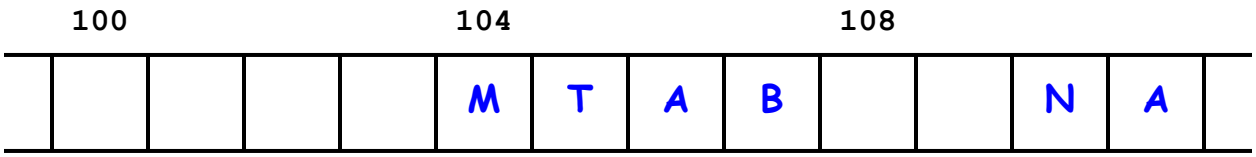
Question 3 [10 Marks]

a) You have a byte-addressable memory architecture with 32 bit words. How would the word "BATMAN" be stored in memory using big-endian and little-endian schemes. The starting location is 104. Also, each character requires one byte of storage space. [6 Marks]

Big-Endian (where more significant bytes are placed into lower memory locations)



Little-Endian (where less significant bytes are placed into lower memory locations)



b) Assume a computer has a byte-addressable memory whose addressable space is 16 Megabytes (16MB). Also, each word is 2 bytes long. How many bits must the memory address register (MAR) be? How many bits must the memory data register (MDR) be? (1MB = 2^{20} Bytes) [4 marks]

MAR: 16 MB
- ➔ 16 * 2^{20} Bytes
- ➔ 2^4 * 2^{20} Bytes
- ➔ 2^{24} Bytes
- ➔ 24 Bits

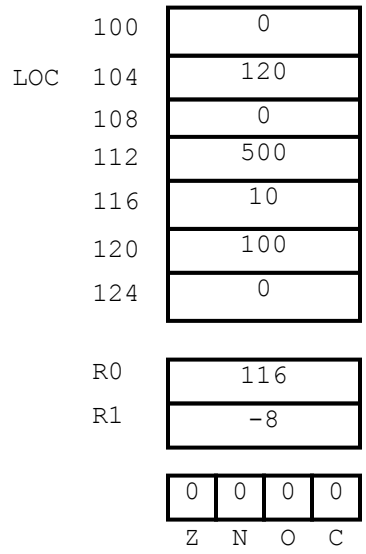
MDR: 2 Bytes
- ➔ 16 Bits

Question 4 [10 Marks]

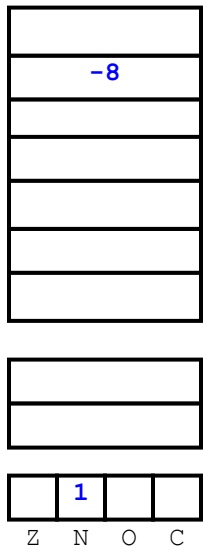
For each instruction provide updated memory values, register values and condition codes. Condition codes: Z (zero), N (negative), V (overflow), C (carry)

NOTE: For each instruction begin with the memory and register contents provided as the "Starting State". Also, you only need to provide the values that change.

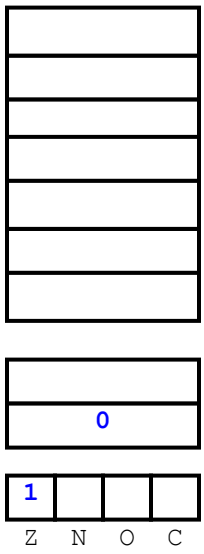
Starting State



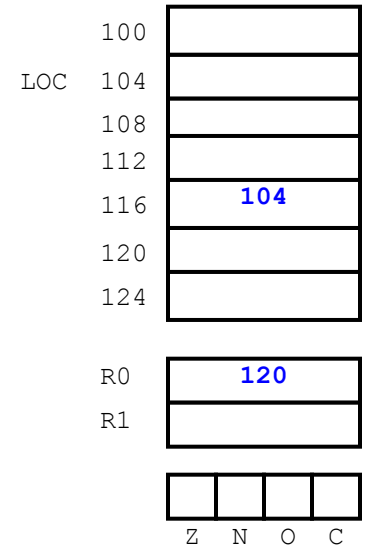
Move R1,LOC



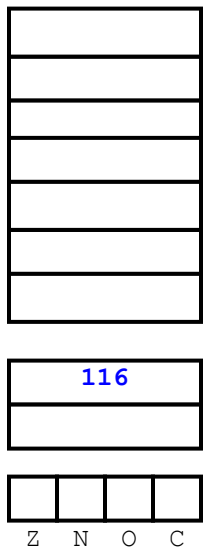
Move 8(R0),R1



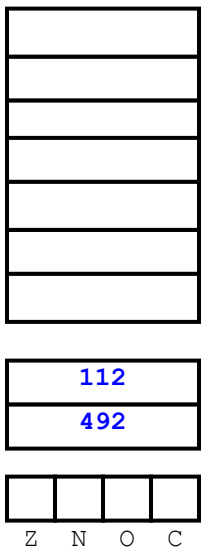
Move #LOC,(R0)+



Add (R0,R1),R0



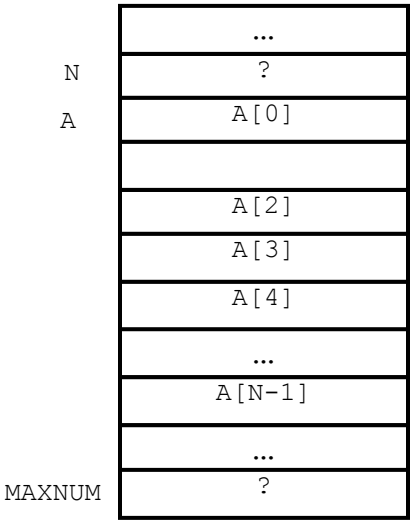
Add -(R0),R1



Question 5 [10 Marks]

You are given an array of numbers starting at memory location A, whose length is N. You need to write an assembly program to determine the largest element of this array. Assume 32 bit words.

At the end, memory location MAXNUM should contain the maximum value.




```
----------------  C Code  ------------------

MAXNUM = A[0];
for (i = 1; i < N; i++) {
      If (A[i] > MAXNUM)
            MAXNUM = A[i];
}
```

```
----------------  Assembly  -----------------

      Move       #A,R1     ; R1 = index for A
      Move       (R1)+,R0  ; R0 = A[0] (MAXNUM)
      Move       N,R2      ; R2 = loop count
      Decr       R2        ; decr loop count
LOOP: Move       (R1)+,R3  ; R3 = A[i]
      Compare    R0,R3     ; A[i] - MAXNUM
      Branch<=0  SKIP      ; R1 = index for A
      Move       R3,R0     ; R0 = A[i] (MAXNUM)
SKIP: Decr       R2
      Branch>0   LOOP
      Move       R0,MAXNUM
```