**CSCI 360-1 Midterm Exam Study Questions Spring 2019**

**Any material contained within this set of study questions not yet covered in class will not be included in your midterm exam.**

**Convert the following UNSIGNED numbers:**

DECIMAL BINARY HEXADECIMAL

2345 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 110010011110 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 235B

**Do the following UNSIGNED integer arithmetic:**

BINARY 1 0 1 0 1 0 1 0 1 1 1 1 0 1 0 0 0 1 1 0

+ 1 0 1 1 1 0 1 1 1 1 - 1 0 0 0 0 1 1 1 1

--------------------- ---------------------

HEXADECIMAL 8 6 7 B 3 F 0 A 7 D C 0

+ E 8 F 9 - 1 0 C F B A 4 D

--------- -----------------

**Do the following FULLWORD (two's complement) arithmetic and determine if any overflow occurred and EXPLAIN why or why not:**

5 8 9 A B 8 7 E C 8 5 2 9 A C D

+ 4 7 B F 8 7 5 2 - C 4 8 8 6 B A 6

----------------- -----------------

**Write the correct number in each blank:**

A fullword contains\_\_\_\_\_bytes. A fullword contains\_\_\_\_\_bits.

RR instruction size is\_\_\_\_\_bytes. RX instruc. size is\_\_\_\_\_bytes.

A PSW contains \_\_\_\_ bits or \_\_\_\_ bytes or \_\_\_\_\_ fullwords.

**What is the ANSI print carriage control character for each of the following: (Note: A blank with nothing written in it is NOT an answer!)**

a. Top of Page\_\_\_\_\_\_ b. Single Space\_\_\_\_\_\_

c. Double Space\_\_\_\_\_\_ d. Triple Space\_\_\_\_\_\_

**Given the following register contents, calculate absolute addresses (in Hexadecimal) for the D(B) or**

**D(X,B) addresses given below.**

R0 = 00000040 R1 = F4F4F4F4 R2 = 00000359 R3 = 80012345

35(0,3) \_\_\_\_\_\_\_\_\_\_\_\_ 28(1) \_\_\_\_\_\_\_\_\_\_\_\_ 16(2,3) \_\_\_\_\_\_\_\_\_\_\_\_

**Answer the following questions using the PSW AT ABEND given below. Give 'address' answers in hexadecimal.**

PSW AT ABEND FFC50001 5F000E34

Contents of memory starting at address (hex) 000E20

Address Contents

------- ------------------------------------------------------------------------

000E20 40125802 5012D503 50301231 5872F01C 1A5500CC 47B0F010 00000000 00000000

1. What is the address of the instruction WOULD HAVE BEEN EXECUTED NEXT had the program

not ABENDed on the current Instruction? \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

2. What is the length (in bytes) of the instruction that caused the ABEND? \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

3. What is the condition code? (Answer with decimal number) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

4. What is the address of the instruction that caused the ABEND? \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

5. What type of program interrupt occurred?

Number \_\_\_\_\_\_\_\_\_\_\_\_ Name \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

6. Write the NEXT instruction that would have been executed as the programmer would have written it

in EXPLICIT assembler language using decimal values. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**(continues on the next page)**

**Fill in the blanks on the assembly listing below, using IMPLICIT addresses where possible:**

LOCATION COUNTER MACHINE SOURCE

VALUE (HEX) LANGUAGE (HEX) LANGUAGE

000000 EXAM1 CSECT

000000 USING EXAM1,15

000000 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ L 2,F33

\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ LA 3,3

\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ MR 2,2

\_\_\_\_\_\_\_\_ 5020 F034 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ AR 3,3

\_\_\_\_\_\_\_\_ 5D20 F02C \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_ 47A0 E000 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ BR 14

\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ F129 DC F'129'

000020 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ F33 DC F'33'

\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ AB DC 2CL3'AB'

\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ BC DC F'-15'

\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ CD DC 3C''''

\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ TOTAL DS F'33'

END EXAM1

**Write assembly instructions to perform the following tasks. (Define any extra storage you reference.) Inefficient code would not receive full credit. Use literals only where necessary.**

1. Quadruple the value in register 2.

2. Increment the value in register 7 by 5182.

3. Initialize register 0 to zero.

4. Multiply the contents of register 11 by -99.

5. Divide the contents of register 8 (which contains 32) by the contents of register 6.

6. Test the contents of registers 2 and 3. Branch to the label FOUND if the contents of register 2 are

not equal to the contents of register 3.

7. Write the equate statement necessary to equate the value 3 with the symbol SAVE.

8. Save the contents of register 10 into a fullword labelled TRUMP.

**(continues on next page)**

**Given the starting values listed for registers and a portion of storage, show the complete contents (in hex) after execution of the instruction. Starting values apply to each instruction. Results are NOT cumulative.**

**STARTING VALUES (For EACH INSTRUCTION):**

R0 = 00000200 R1 = FF000204 R3 = 0000036C R4 = FFFFFFFF

R5 = FFFFFFFF R6 = 00000004 R7 = 00000064 R8 = 00000004

ADDRESS CONTENTS

000200 F5F5F5F5 0000003C 00000123 00000008

LR 5,6 R5 = \_\_\_\_\_\_\_\_\_\_\_\_\_

LA 1,2(0,3) R1 = \_\_\_\_\_\_\_\_\_\_\_\_\_

L 6,4(6,1) R6 = \_\_\_\_\_\_\_\_\_\_\_\_\_

DR 4,8 R4 = \_\_\_\_\_\_\_\_\_\_\_\_\_ R5 = \_\_\_\_\_\_\_\_\_\_\_\_\_

AR 0,6 R0 = \_\_\_\_\_\_\_\_\_\_\_\_\_

MR 4,4 R4 = \_\_\_\_\_\_\_\_\_\_\_\_\_ R5 = \_\_\_\_\_\_\_\_\_\_\_\_\_

LTR 3,4 R3 = \_\_\_\_\_\_\_\_\_\_\_\_\_

**Write the assembler source code to implement the expression below. Assume that all fields have already been correctly defined as fullwords in your storage area. Provide all necessary labels. Use literals if appropriate. Rounding is NOT necessary.**

VALUE = ((NUM4 - NUM3)/2) \* NUM2) + NUM4

**(continues on the next page)**

**The following complete assembly language program reads an unknown number of 80 byte records and calculates the average of the first number found on all records. Some parts of the program have been left out, and you are to fill in each blank with the appropriate value. What follows is the complete program: no additional fields are available, nor may they be written in.**

CALCUL8 CSECT , Begin program

USING \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Define base register

SR 3,3 Initialize sum of card values

SR 4,4 Initialize count of valid values

XREAD CARD,80 Read first record

\*

LOOP1 BC \_\_\_\_\_\_\_\_\_\_\_\_,ENDLOOP1 Finished if no more records

XDECI \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Get first number from record

BC \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ B if none found

AR 3,0 Else add to total

A 4,\_\_\_\_\_\_\_\_\_\_\_\_\_\_ And increment count of numbers

NEXTCARD XREAD CARD,80 Get next input record

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Continue

\*

ENDLOOP1 LTR 4,\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Any numbers found?

BC B'1000',SAVE No, save zero

M 2,\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Prepare to divide

DR 2,\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Get average in R3

AR 2,2 Double the remainder

CR 2,\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Compare with divisor

BC \_\_\_\_\_\_\_\_\_\_\_,SAVE Branch if no rounding needed

A 3,\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Else round up

\*

SAVE ST 3,AVG Save for XDUMP

XDUMP \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Display value saved above

BCR \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Return to caller

\*

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (This belongs in every program!)

CARD DS \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

DC \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

AVG DS \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

END CALCUL8