IT5002 Computer Systems and Applications
2025/26 Semester I
Midterm Assessment

Instructions

Questions

Number Systems

1. Convert the following numbers:

a. 0x1FA to base-7.

$$1FA = 1 \times 16^2 + 15 \times 16 + 10$$
$$= 506$$
$$= 1322_7$$

b. 2313.13213_4 to octal.

$$2313.13213_4 = 10\ 11\ 01\ 11\ .01\ 11\ 10\ 01\ 11_2$$
$$= 010\ 110\ 111\ .011\ 110\ 011\ 100_2$$
$$= 010\ 110\ 111\ .011\ 110\ 011\ 100_2$$
$$= 267.3634_8$$

2. What is -13.171875_{10} in 32-bit IEEE-752 floating point format? Express your answer in hexadecimal.

13 = 1101
.171875 = 0.001011
13.171875 = 1101.001011 $\times 2^0$
= 1.101001011 $\times 2^3$
Mantissa = 10100101100000000000000
Exponent = 3 + 127 = 130 = 10000010
Sign = 1
1100 0001 0101 0010 1100 0000 0000 0000
= 0xC152C000

3. Fill in the blanks below:

$$310_4 + 220_4 = W_4 \quad W_4 = \underline{\quad\quad}$$

$$310_4 + 220_4 = \mathbf{1130_4}$$

$$475_9 - 178_9 = X_9 \quad X_9 = \underline{\hspace{2cm}}$$
$$475_9 - 178_9 = \mathbf{286_9}$$

$$345_6 + Y_6 = 1003_6 \quad Y_6 = \underline{\hspace{2cm}}$$
$$345_6 + \mathbf{214_6} = 1003_6$$

MIPS Assembly Language

4. We have two arrays A and B, where $2 contains the number of elements in array A, $3 contains the base address of array A, and $4 contains the base address of array B.

	sll $5, $2, 2	# Address 0xB0FC1000
	add $6, $3, $5	0xB0FC1004
	addi $7, $3, 0	0xB0FC1008
	addi $12, $0, 0	0xB0FC100C
A:	slt $8, $7, $6	0xB0FC1010
	beq $8, $0, C	# i1
	lw $8, 0($7)	
	slti $9, $8, 32	
	bne $9, $0, B	
	sll $8, $8, 1	
	addi $8, $8, 5	
B:	add $13, $4, $12	
	sw $8, 0($13)	
	addi $12, $12, 4	
	addi $7, $7, 4	
	j A	# i2
C:	.. other irrelevant code ..	

Assuming that A = [3, 45, 9, 12, 52, 32, 1, 8, 100, 5] and $2 contains 10:

a. What does array B contain at the end executing this code?
   B = [3, 95, 9, 12, 109, 69, 1, 8, 205, 5]

b. How many instructions are executed in this program, given array A above?
   Outside loop: 4
   Inside loop:
   $6 \times 10 + 4 \times 12 + 2 = 60 + 48 + 2 = 110$
   Total = 114 instructions

c. Choose the statement that best describes this code.

    i.   B[i] = A[i] if A[i] > 32, else B[i] = 4 x A[i] + 5

    ii.   +B[i] = 2 x A[i] + 5 if A[i] >=32, else B[i] = A[i]

    iii.   B[i] = 2 x A[i] + 5 if A[i] <= 32, else B[i] = A[i]

    iv.   B[i] = 4 x A[i] +5 if A[i] > 32, else B[i] = A[i]

    v.   B[i] = A[i] if A[i] < 32, else B[i] = 2 x A[i] + 5

Both ii and v mean the same thing; ii. is automatically accepted. I will be manually grading to accept v. If you selected v you will get 2 points more than what you see on Examplify.

d. What is the instruction encoding for the instruction marked #i1? Express your answer in hexadecimal.

beq $8, $0, C

Offset = 10

Encoding = 000100 01000 00000 0000 0000 0000 1010

0001 0001 0000 0000 0000 0000 0000 1010

= 0x1100000A

e. What is the instruction encoding for the instruction marked #i2? Express your answer in hexadecimal.

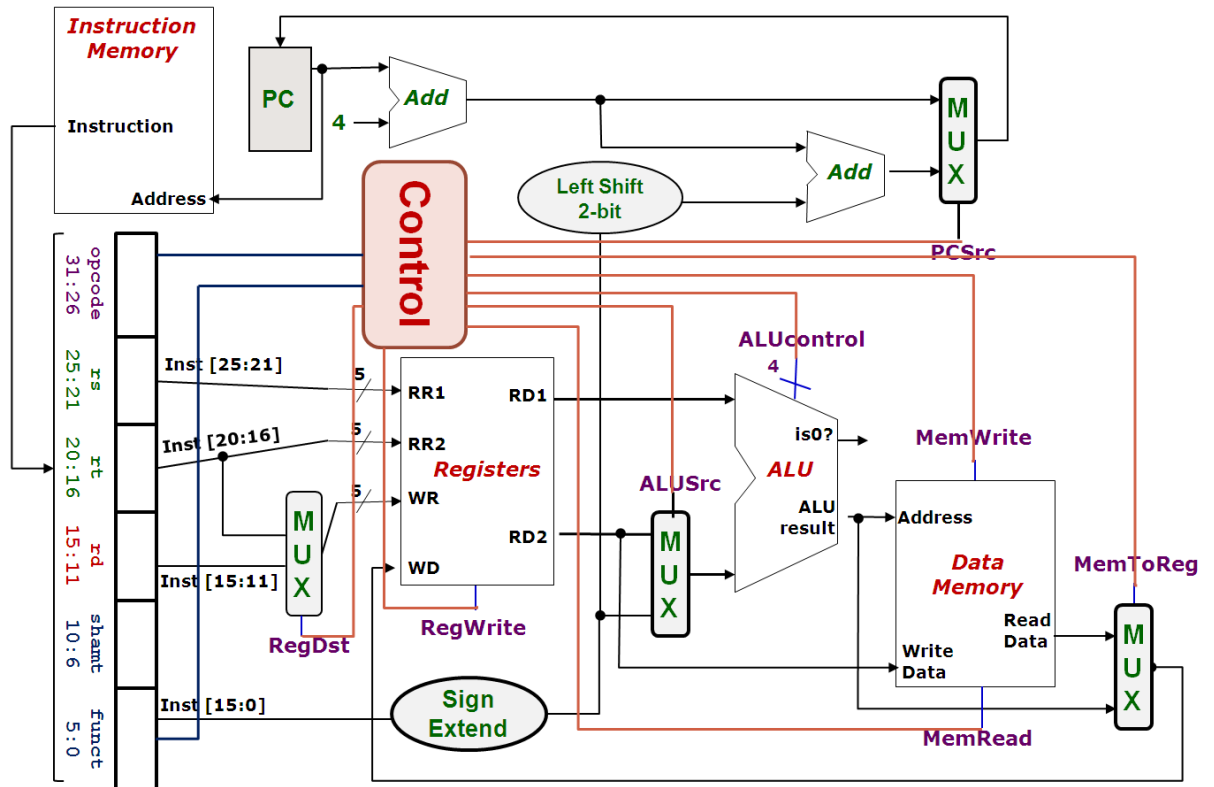A = 0xB0FC1010

= 1011 0000 1111 1100 0001 0000 0001 0000

= 0000 1111 1100 0001 0000 0001 00

Jump = 00 0010

0000 1000 0011 1111 0000 0100 0000 0100

= 0x083F0404

5. The diagram below shows the MIPS 32-bit datapath:

Write down the values for all the various control signals and values for each of the following instructions. If a signal should be "X" ("Don't care"), writing 0 or 1 will be marked as incorrect. **Reminder:** The top input of the MemToReg multiplexor is input 1, the bottom input is input 0.

Each part of this question is independent of the other parts. Write all answers in hexadecimal. The values stored in $0 to $9 are shown below in hexadecimal.

Reg	Val	Reg	Val	Reg	Val	Reg	Val	Reg	Val
$0	0	$1	0C	$2	1A	$3	4B	$4	30
$5	7A	8	F3	$7	2F	$8	10	$9	C1

These are the values stored at memory locations 30 to 40, in hexadecimal. Assume that the most significant byte of a 32-bit word is at the smallest address.

Address	30	31	32	33	34	35	36	37	38	39	4A	4B
Value	1F	2A	30	2A	FC	44	2B	17	30	11	23	7F

Address	30	31	32	33	34	35	36	37	38	39	4A	4B
Value	1F	2A	30	2A	FC	44	2B	17	30	11	23	7F

ALUControl Signals:

ALUcontrol	Function
0000	AND
0001	OR
0010	add
0110	subtract
0111	slt
1100	NOR

Many students mistook "Write Data" to be the WD input on the Register File. This is incorrect; it is clear that Write Data is the data sent to memory. Whether or not it is written is irrelevant; it is sent simply because it is connected to RD2 which always produces a value.

a. addi $3, $4, -5

   -5 = -(0000 0000 0000 0101$_2$) = (1111 1111 1111 1011$_{2s}$)

RR1	4	1
RR2	3	2
RD1	30	3
RD2	4B	4
SignExtendedImmediate	0xFFFFFFFB	5
RegDst	0	6
RegWrite	1	7
ALUSrc	1	8
PCSrc	0	9
ALUControl	2	10
MemRead	0	11
MemWrite	0	12
MemToReg	0	13
Read Data (Value is unknown. Either -, X or some indication that value is unknown is accepted)	-	14
Write Data	4B	15

b. The beq in the following sequence:

There is an error in this question that makes it unclear whether we want beq or bne. The difference is in the PCSrc bit; hence both 0 and 1 are accepted.

L: addi $3, $2, $2
    sll $4, $2, 1
    sub $3, $3, $4
    bne $3, $zero, L   # This beq instruction
    .. Other instructions ..

Offset = -4 = -(0000 0000 0000 0000 0000 0000 0000 0100$_2$) = (1111 1111 1111 1111 1111 1111 1111 1100$_{2s}$)
= 0xFFFFFFFC

RR1	3	1
RR2	0	2
RD1	0	3
RD2	0	4
SignExtendedImmediate	0xFFFFFFFC	5
RegDst	X	6
RegWrite	0	7
ALUSrc	0	8
PCSrc (Question wasn't ambiguous on whether it should be bne or beq, so both 0 and 1 are accepted)	0|1	9
ALUControl	6	10
MemRead	0	11
MemWrite	0	12
MemToReg	X	13
Read Data (Value is unknown. Either -, X or some indication that value is unknown is accepted)	-	15
Write Data	0	16

c. lw $3, 4($4)

RR1	4	1
RR2	3	2
RD1	30	3
RD2	4B	4
SignExtendedImmediate	0x4	5
RegDst	0	6
RegWrite	1	7
ALUSrc	1	8
PCSrc	0	9
ALUControl	2	10
MemRead	1	11
MemWrite	0	12
Address	34	13
Read Data	FC442B17	14
Write Data	4B	15