

UVA WELLASSA UNIVERSITY OF SRI LANKA

Department of Computer Science and Technology
 End Semester Examination - Semester -1 - 2008/2009
 CST303-2 Computer Systems Architecture



Time Allowed: Two Hours

Answer 4 Questions Only

For MIPS instruction reference refer to last page of the paper

Question-1

Consider the following integer X, represented in 32-bit 2's complement form:

0000 1111 0011 1111 0000 1111 0000 1111

- What is the representation for $-X$?
- Why is 2's complement representation used for signed integers in almost all computer architectures instead of sign-magnitude representation?

Two 32-bit signed integers, A and B, are added to produce a sum C, it is possible that an overflow will occur.

- Define overflow.
- When adding signed numbers how an overflow will occur?

It is possible to determine whether overflow has occurred by considering only the three high bits, $A[31]$, $B[31]$ and $C[31]$.

- Describe how this can be done.
- Write down a Boolean expression in terms of $A[31]$, $B[31]$ and $C[31]$ for overflow detection.
- Similarly if we consider subtraction ($C=A-B$) how an overflow will occur?
- Can you find a Boolean expression for overflow in this case?

Question-2

A text file is encoded in UTF-16 Unicode characters. (Unicode uses 16bits to represent a single character). This file contains 1000 data records as given below. Assume:

- all data is rounded off to two fixed decimal points.
- Each data row contains three delimiters. (Two tabs and a newline character)
- Range of each data column is limited to (100 -999), (10 - 99) and (1000 - 9999) respectively.

105.34	34.23	3457.66
234.58	12.51	1978.67
..
..
123.23	23.56	1303.45

- (a) What's the size of of the text file in Kilo Bytes?
- (b) If the same text file is stored in ASCII format what's the file size?
- (c) How this file can be stored in a binary format? Precisely describe the data types to be used and how the data can be separated?
- (d) What's the file size in the proposed binary format?

Represent following numbers (in base 10) in IEEE 754 floating point format:

- (e) -2048
- (f) -3.03125
- (g) Write a function in C language to print the content of a given memory address as a 32bit binary number.

Function Prototype: `void printBinary(unsigned int* addr)`

Question-3

Implement following leaf functions (functions which do not call other functions) in MIPS assembly.

- (a) `int max (int x, int y) {`
`.....`
`}`
- (b) `int min (int x, int y) {`
`.....`
`}`
- (c) `void swap(int* x, int* y) {`
`.....`
`}`

Question-4

Implement following c language control statements in MIPS assembly. The body of each loop can be filled with whatever instructions you want.

- (a) `for(i=0;i<100;i++) {`
`....`
`....`
`}`

```
(b) while (i<=100) {
```

```
.....
```

```
.....
```

```
}
```

```
(c) switch (data) {
```

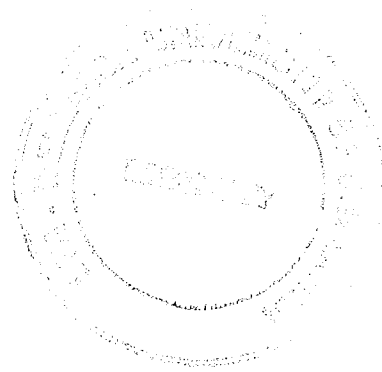
```
case 0 :
```

```
.....
```

```
case 1 :
```

```
.....
```

```
}
```



Question-4

“Roadrunner”, the world fastest supercomputer, having 129600 processor cores has an average performance of 1 450 000 GFlops.

- (a) A satellite image of RGBE format of size 1 000 000 x 1 000 000 pixels is processed by Roadrunner. RGBE format stores colour information of each pixel as a 32bit floating number. If the processing of size W x H image takes 5000 x W x H floating point instructions. How long will Roadrunner take to process the satellite image?
- (b) Microchip PIC10F series 8bit mcicrocontrollers are self clocked with a frequency of 200KHz. They perform each processor instruction (instruction cycle) for 4 clock cycles. Finding the sinusoidal of a given number 'm' up to 'n' decimal places takes $10.5n^3$ floating point operations. PIC10F series microcontrollers take 573 processor instructions to complete a single floating point operation. How long will it take to calculate $\text{Sin}(2.45_{\text{rad}})$ up to 5 decimal places?

Question-5

Describe following topics/concepts briefly but precisely.

- (a) Green computing and performance per watt
- (b) Processor Virtualization
- (c) SIMD Instructions
- (d) Flash Memory

Category	Instruction	Example	Meaning	Comments
Arithmetic	add	add \$s1,\$s2,\$s3	$\$s1 = \$s2 + \$s3$	Three operands; overflow detected
	subtract	sub \$s1,\$s2,\$s3	$\$s1 = \$s2 - \$s3$	Three operands; overflow detected
	add immediate	addi \$s1,\$s2,100	$\$s1 = \$s2 + 100$	+ constant; overflow detected
	add unsigned	addu \$s1,\$s2,\$s3	$\$s1 = \$s2 + \$s3$	Three operands; overflow undetected
	subtract unsigned	subu \$s1,\$s2,\$s3	$\$s1 = \$s2 - \$s3$	Three operands; overflow undetected
	add immediate unsigned	addiu \$s1,\$s2,100	$\$s1 = \$s2 + 100$	+ constant; overflow undetected
	move from coprocessor register	mfc0 \$s1,\$epc	$\$s1 = \epc	Copy Exception PC + special regs
	multiply	mult \$s2,\$s3	Hi, Lo = $\$s2 \times \$s3$	64-bit signed product in Hi, Lo
	multiply unsigned	multu \$s2,\$s3	Hi, Lo = $\$s2 \times \$s3$	64-bit unsigned product in Hi, Lo
	divide	div \$s2,\$s3	Lo = $\$s2 / \$s3$, Hi = $\$s2 \bmod \$s3$	Lo = quotient, Hi = remainder
	divide unsigned	divu \$s2,\$s3	Lo = $\$s2 / \$s3$, Hi = $\$s2 \bmod \$s3$	Unsigned quotient and remainder
move from Hi	mfhi \$s1	$\$s1 = \text{Hi}$	Used to get copy of Hi	
move from Lo	mflo \$s1	$\$s1 = \text{Lo}$	Used to get copy of Lo	
Data transfer	load word	lw \$s1,100(\$s2)	$\$s1 = \text{Memory}[\$s2 + 100]$	Word from memory to register
	store word	sw \$s1,100(\$s2)	$\text{Memory}[\$s2 + 100] = \$s1$	Word from register to memory
	load half unsigned	lhu \$s1,100(\$s2)	$\$s1 = \text{Memory}[\$s2 + 100]$	Halfword memory to register
	store half	sh \$s1,100(\$s2)	$\text{Memory}[\$s2 + 100] = \$s1$	Halfword register to memory
	load byte unsigned	lbu \$s1,100(\$s2)	$\$s1 = \text{Memory}[\$s2 + 100]$	Byte from memory to register
	store byte	sb \$s1,100(\$s2)	$\text{Memory}[\$s2 + 100] = \$s1$	Byte from register to memory
	load upper immediate	lui \$s1,100	$\$s1 = 100 \times 2^{16}$	Loads constant in upper 16 bits
Logical	and	and \$s1,\$s2,\$s3	$\$s1 = \$s2 \& \$s3$	Three reg. operands; bit-by-bit AND
	or	or \$s1,\$s2,\$s3	$\$s1 = \$s2 \$s3$	Three reg. operands; bit-by-bit OR
	nor	nor \$s1,\$s2,\$s3	$\$s1 = \sim (\$s2 \$s3)$	Three reg. operands; bit-by-bit NOR
	and immediate	andi \$s1,\$s2,100	$\$s1 = \$s2 \& 100$	Bit-by-bit AND with constant
	or immediate	ori \$s1,\$s2,100	$\$s1 = \$s2 100$	Bit-by-bit OR with constant
	shift left logical	sll \$s1,\$s2,10	$\$s1 = \$s2 \ll 10$	Shift left by constant
	shift right logical	srl \$s1,\$s2,10	$\$s1 = \$s2 \gg 10$	Shift right by constant
Conditional branch	branch on equal	beq \$s1,\$s2,25	if ($\$s1 == \$s2$) go to PC + 4 + 100	Equal test; PC-relative branch
	branch on not equal	bne \$s1,\$s2,25	if ($\$s1 != \$s2$) go to PC + 4 + 100	Not equal test; PC-relative
	set on less than	slt \$s1,\$s2,\$s3	if ($\$s2 < \$s3$) $\$s1 = 1$; else $\$s1 = 0$	Compare less than; two's complement
	set less than immediate	slti \$s1,\$s2,100	if ($\$s2 < 100$) $\$s1 = 1$; else $\$s1 = 0$	Compare < constant; two's complement
	set less than unsigned	sltu \$s1,\$s2,\$s3	if ($\$s2 < \$s3$) $\$s1 = 1$; else $\$s1 = 0$	Compare less than; natural numbers
	set less than immediate unsigned	sltiu \$s1,\$s2,100	if ($\$s2 < 100$) $\$s1 = 1$; else $\$s1 = 0$	Compare < constant; natural numbers
Unconditional jump	jump	j 2500	go to 10000	Jump to target address
	jump register	jr \$ra	go to \$ra	For switch, procedure return
	jump and link	jal 2500	$\$ra = \text{PC} + 4$; go to 10000	For procedure call