Mid-Semester Examination
Computer Organization and Architecture
(Course No: CS31001 and CS31007)

Autumn Semester, 2012
Time: 2 hours
Full Marks: 60

Attempt All Questions
Answer in Brief

1. Consider two different hardware implementations M_1 and M_2 of the same instruction set. There are three classes F, I, and N of instructions in the instruction set. M_1's clock rate is 600 MHz, M_2's clock cycle is 2 ns. The average CPI for the three classes on M_1 and M_2 are as follows:

Class	CPI for M_1	CPI for M_2
F (Floating Point)	5.0	4.0
I (Integer arithmetic)	2.0	3.8
N (Nonarithmetic)	2.4	2.0

   (a) What are the peak performances of M_1 and M_2 in MIPS?

   (b) If 50% of all instructions executed in a certain program are from class N, and the rest are divided equally among F and I, which machine is faster and by what factor?

   (c) Designers of M_1 plan to redesign the machine for better performance. With the assumption of part (b), which of the following redesign options has the greatest performance impact and why?

      i. Using a faster floating point unit with double the speed (class F-CPI = 2.5).

      ii. Adding a second integer ALU to reduce the integer CPI to 1.20.

      iii. Using faster logic that allows a clock rate of 750 MHz with the same CPIs.

(d) The CPIs given include the effect of instruction cache misses at an average rate of 5%. Each cache miss imposes a 10-cycle penalty (ie. adds 10 to the effective CPI of the instruction causing the miss). A fourth redesign option is to use a larger instruction cache that would reduce the miss rate from 5% to 3%. How does this compare to the three options in part (c) for machine M_1?

(e) Characterize application programs that would run faster on M_1 than on M_2, ie. what can you say about the instruction mix of the three instructions for which machine M_1 is faster? Assume that the CPIs of the instructions are as stated in the beginning without any change.

2+2+2+2+2=10 marks

2. Consider the IEEE floating point representation consisting of the exponent e and the significand s. For a 32-bit floating point number (as studied in the class) answer the following:

(a) How many distinct floating point numbers can you specify?

(b) What is the range of the floating point numbers neglecting the existence of denormalized numbers?

(c) What is the range of the floating point numbers considering the existence of denormalized numbers?

(d) Define gap as the difference between two successive numbers in a representation. Evaluate the same for floating point numbers and compare it with that of a fixed point representation where there are 16 bits for the integer part, and 16 bits for the decimal part.

2+3+3+2=10 marks

3. A designer needs to add an instruction for *high speed* exponentiation, $z = x^y$, where x and y are represented as unsigned 32 bit numbers. In order to develop efficient architectures for performing the computation answer the following questions:

(a) Assume $y = \sum_{i=0}^{31} y[i]2^i$, where $y[i] \in \{0, 1\}$. You will get extra credit for provide suitable optimizations. Ensure that the data path and control path of the design are separate.

(b) Can you apply Boothe's encoding to the representation of y? Justify when the encoding is useful.

5+5=10 marks

4. Consider the single cycle execution unit discussed in the class. Suppose we would like to make some changes to the Next address logic. The array of the 30 AND gates with the BrTrue signal, when a branch is to be taken is eliminated and the immediate value, sign-extended to 30 bits, is connected directly to the top adder input. A separate incrementer is introduced to compute $(PC)_{32} + 1$. The BrTrue signal is then used to control a multiplexer that allows the IncrPC output to be taken from the adder or the newly introduced incrementer. Compare this design with the original design.

5 marks

5. The following program is to be executed on the multi-cycle micro-MIPS ISA discussed in the class.

```
if(i <= j) {x=x+1; z=1;}
else {y=y-1; z=2z;}
```

(a) State the assembly code for the program using the micro-MIPS ISA studied in the class and also provided in the appendix. Use the instructions slt, bne, addi, add, j instructions **only**.

(b) State the complete micro-program for the above code snippet.

5+10=15 marks

6. In context to the carry-look ahead adder discussed in the class to add two inputs x and y, answer the following questions:

(a) Define the generate (g_i) and propagate (p_i) signals.

(b) Define an auxiliary variable $t_i = g_i \vee p_i = x_i \vee y_i$. Show that the carry recurrence, $c_{i+1} = g_i \vee p_i c_i$ remains valid if we replace p_i with t_i.

(c) Explain briefly the working of a 4 bit carry look ahead adder using the auxiliary signal and g_i. Can you identify why this circuit will be slightly faster than the previous circuit?

2+4+4=10 marks

Appendix: Micro-MIPS ISA:

Class	Instruction	Usage	Meaning	op	fn
Copy	Load upper immediate	lui rt, imm	$rt \leftarrow (imm, 0x0000)$	15	
Arithmetic	Add	add rd,rs,rt	$rd \leftarrow (rs) + (rt)$	0	32
	Subtract	sub rd,rs,rt	$rd \leftarrow (rs) - (rt)$	0	34
	Set less than	slt rd,rs,rt	$rd \leftarrow$ if $(rs) < (rd)$ then 1 else 0	0	42
	Add Immediate	addi rt,rs,imm	$rt \leftarrow (rs) + imm$	8	
	Set Less than immediate	slti rt,rs,imm	$rt \leftarrow if(rs) < imm$then 1 else 0	10	
Logic	AND	and rd,rs,rt	$rd \leftarrow (rs) \wedge (rt)$	0	36
	OR	or rd,rs,rt	$rd \leftarrow (rs) \vee (rt)$	0	37
	XOR	xor rd,rs,rt	$rd \leftarrow (rs) \oplus (rt)$	0	38
	NOR	nor rd,rs,rt	$rd \leftarrow ((rs) \vee (rt))'$	0	39
	AND immediate	andi rt,rs,imm	$rt \leftarrow (rs) \wedge imm$	12	
	OR immediate	ori rt,rs,imm	$rt \leftarrow (rs) \vee imm$	13	
	XOR immediate	xori rt,rs,imm	$rt \leftarrow (rs) \oplus imm$	14	
Memory Word	Load Word	lw rt,imm(rs)	$rt \leftarrow mem[(rs) + imm]$	35	
	Store Word	sw rt,imm,(rs)	$mem[(rs) + imm] \leftarrow (rt)$	43	
Control transfer	Jump	j L	goto L	2	
	Jump register	jr rs	goto (rs)	0	8
	Branch on less than 0	bltz rs,L	if$(rs) < 0$ then goto L	1	
	Branch on equal	beq rs,rt,L	if $(rs) = (rt)$ then goto L	4	
	Branch on not equal	bne rs,rt,L	if$(rs) \neq (rt)$ then goto L	5	
	Jump and link	jal L	goto L; 31 $\leftarrow$ (PC)+4	3	
	System call	syscall	Associated with an OS system routine	0	12