V1

March 1, 1999

cs470 - Computer Architecture 1
Spring 1999

Midterm Exam
open books, open notes

Starts: **6:25 pm** Ends**: 8:00 pm**

Name:____________________________________(please print)

ID:________________________

Problem	Max points	Your mark	Comments
1	25		10+10+5
2	25		10+10+5
3	10		10
4	15		5+10
5	15		8+7
	90		

1. Your first job as a new employee with ACME Computing is to upgrade your boss' computer with a new disk and and a new disk controller. These changes will make every disk access two times faster. **With these enhancements in place**, disk accesses account for 25% of the running time. The overall cost of the system increases by 20%.

a) what is the overall speedup?

b) what is the overall speedup if you also improve the graphics system with a new graphics card? The new card will make all graphics 25 times faster. Graphics represent 30% of the workload of the original machine (before any improvment is done).

c) Assuming just the disk improvement, decide whether the new system will be more cost effective than the current one. We say that one system is more *cost effective* than another if the ratio of performance divided by cost is higher.

2. Consider the following C statement:

```
a = a*b + b;
```

You have an 8-bit acumulator machine with 16 bit addresses. `a` and `b` are of type `very short int` (i.e. 8-bit) and are stored in memory starting with the address `0x0DDD` (where DDD stands for the first three digits of your SSN). Their initial values are `0x2a` for `a` and `0x02` for `b`. The code starts in memory at address `0x5ffc`. Below is a list of opcodes (each opcode is one byte wide):

Instruction	Opcode
`add`	`0x20`
`sub`	`0x21`
`mul`	`0x22`
`div`	`0x23`
`push`	`0x24`
`pop`	`0x25`

a) Compile the C code for this accumulator machine; for each instruction in the program show the instruction format on the right hand side; clearly mark the boundaries of each instruction.

Instruction	Instruction Format						

b) Show the sequence of addresses issued by the CPU to execute this code; for each address indicate whether it’s a memory read or write and what’s on the data bus for that particular memory access.

Address	R/W	Data		

Address	R/W	Data		

c) What is the average number of memory accesses per instruction?

3. Write MIPS assembly code for the following piece of a C program:

```
x[5] = x[6] + a;
```

You shall assume that `a` corresponds to register `$t1` and the array of words `x` begins at address `0x10010010`. You must use the **native** MIPS instruction set.

4. You have a register-register architecture that has two addressing modes, base-displacement and memory indirect, besides register and immediate. You want to improve this architecture by eliminating the memory indirect addressing mode: this will decrease the clock cycle by 10% but will increase somehow the instruction count because you will have to replace instructions like:

```
lw    R1, @(R2)
```

with a sequence of instructions. Assume that the frequency of memory indirect addressing is 5%, and that the overall CPI does not change.

a) show the sequence of instructions that will replace every memory indirect addressing

b) how does the performance of the new architecture compare with the performance of the original?

5. You know the following about the *time distribution* of instructions in your favorite application (which may be a word processor, a spreadsheet or maybe a database):

	t_i	CPI_i
ALU	50%	4
Load/store	40%	6
Branches	10%	5

a) compute the average CPI for your application.

b) compute the MIPS for your machine using the above table; the clock rate for your machine is 200 MHz. You get full credit for this (even if you did not solve part a, if you make reasonable assumptions about the data you need.