CS152 Computer Architecture and Engineering

Homework #1

Professor David Patterson

John Lazzaro

Fall 2004

Homework 1 is due Wednesday, 9/15, no later than 5:00pm in the HW boxes in 283 Soda. Please include the TIME and TA NAME of the DISCUSSION section that you attend as well as your NAME, STUDENT ID and LOGIN. Homeworks will be handed back in discussion sections.

Homework Policy:

No late homeworks will be accepted.

You may discuss problems with your friends, but all work must be done individually and you must be able to prove that you understand everything that you hand in.

If we can't read it from 2 feet away, we won't grade it.

---

1. MIPS Instruction Set Architecture and Semantics (10 points)

Consider the following FOR-loop code sequence in a C-like language:

```
i = 0;
for (j = 1; j < 10; j++)
{
        i = i + 2
}
…
```

To make sure you understand the calculation that this loop implements, what is the value of the variables at the point of exit from the look (indicated by …above):

i = _____ j = _____

The (symbolic) MACHINE-level code for this loop is written below. THERE ARE MULTIPLE ERRORS IN THIS CODE SEQUENCE! You may modify an instruction or reorder instructions; but you should NOT rewrite the program from scratch.

Word Address	**MIPS Machine Instruction**	**Comment**	**If instruction is incorrect, indicate the fix in the box below**
0	ADDI $1, $0, #0	r1 ← r0 + 0	
1	ADD $2, $1, $0	r2 ← r1 + r0	
2	ADDI $3, $2, #10	r3 ← r2 + 10	
3	SUB $4, $2, $3	r4 ← r2 – 3	
4	BEQZ $2, #8	IF r2 = 0 THEN GOTO Word 8	
5	ADDI $1, $1, #2	rl ← rl + 2	
6	J #2	GOTO Word 2	
7	ADDI $2, $2, #1	r2 ← r2 + 1	
8	…	Loop exit	
9	…		

Question #2: Instruction Set Architecture and Registers Sets [15 pts]

Imagine a modified MIPS instruction set with a register file consisting of 64 general-purpose registers rather than the usual 32. Assume that we still want to use a uniform instruction length of four bytes and that the opcode size is constant. Also assume that you can expand and contract fields in an instruction, but that you cannot omit them.

a) How would the format of R-type (arithmetic and logical) instructions change? Label all the fields with their name and bit length. What is the consequence of this change? **[4 pts]**

opcode	rs	rt	rd	shamt	func

b) How does this change the I-type instructions? What is the consequence of this change? **[3 pts]**

opcode	rs	rt	immediate

c) How does this change the J-type instructions? What is the consequence of this change **[3 pts]**

Jump	target

d) Imagine we are translating machine code to use the larger register set. Give an example of an instruction that used to fit into the old format, but is impossible to translate directly into a single instruction in the new format. Write a short sequence of instructions that could replace it. **[5 pts]**

Problem 3: (20 points) Complete the skeleton of MIPS assembly language (with delayed branches) below for the following C function. (Underlines may not be exactly right).

```
extern int f (int);

int foo(int *A, int n)
{
        int i = 0;
        int sum = 0;
        for (i = 0; i<n;i++){
                sum = sum + f(A[i+1]);
        }
return(sum);
}
```

```
foo:
        subu    $sp,$sp,40
          sw    $31,32($sp)
          sw    $19,28($sp)             # save A in $19
          sw    $18,24($sp)             # save n in $18
          sw    $17,20($sp)             # i in $17
          sw    $16,16($sp)             # sum in $16

$L7:                                    # top of loop

$L3:                                    #fall-through

          j     $31                     # delayed return jump
          addu    $sp,$sp,40
          .end    foo
```

Question #4:

Go to the resource page and download “From ASIC to ASIP: The Next Design Discontinuity” by Kurt Keutzer, Sharad Malik, and Richard Newton. Summarize in two paragraphs the arguments the authors make as to why chip design costs are rising and will continue to rise.