CS203A Fall 2004

Instructor: Jun Yang
Quiz 1
10/12/2004

Short questions (8pts)

What are the major distinctions between RISC and CISC? (**4pts**)

	CISC	RISC
1.	variable length instruction	single word instruction
2.	variable format	fixed-field decoding
3.	memory operands	load/store architecture
4.	complex operations	simple operations

What are the two typical software-implemented calling conventions? Give brief descriptions. (**4pts**)

Caller saving: the caller saves the registers in use
Callee saving: the callee saves registers it will use

Understanding MIPS (3pts)

Let R2 = 0x1234abcd, what is R4 in hexadecimal after executing the following MIPS code (**2pts**)? What does the following MIPS code do (**1pt**)?

```
srl   R1, R2, 24      # R1 = 0x12, R2 = 0x1234abcd
and   R4, R1, 0xff    # R4 = 0x12,

srl   R1, R2, 16      # R1 = 0x1234
and   R1, R1, 0xff    # R1 = 0x34
sll   R1, R1, 8       # R1 = 0x3400
or    R4, R4, R1      # R4 = 0x3412

srl   R1, R2, 8       # R1 = 0x1234ab
and   R1, R1, 0xff    # R1 = 0xab
sll   R1, R1, 16      # R1 = 0xab0000
or    R4, R4, R1      # R4 = 0xab3412

and   R1, R2, 0xff    # R1 = 0xcd
sll   R1, R1, 24      # R1 = 0xcd000000
or    R4, R4, R1      # R4 = 0xcdab3412
```

Ans: R4 = 0xcdab3412. Change endian-ness.

Performance evaluation (4pts)

We would like to improve the basic 5-stage pipeline to speed up the execution of a program P. Suppose that originally the branch targets are all computed in the EX stage. Fetching for the target only begins when the branch enters the MEM stage. The mis-fetched instructions are all canceled (nullified) from the pipeline. Thus, for every branch, there is effectively 2-cycle loss (pipeline stalls) in the target instruction.

The improvement will move the branch target calculation into the ID stage, i.e, one stage earlier. Assume 1) this change does not affect the clock cycle time; 2) originally 20% of the total cycles are due to the branch stalls; 3) there are no other stalls in this pipeline. What is the speedup of the improvement?

Ans: Execution time = Total Cycles x CC

Total $\text{Cycles}_{\text{after}}$ = Total $\text{Cycles}_{\text{before}}$ x (1 – 20%x50%) = 0.9 Total $\text{Cycles}_{\text{before}}$

$$\text{Speedup} = \frac{ExecutionTime_{before}}{ExecutionTime_{after}} = \frac{Total\_Cycles_{before} \times CC}{Total\_Cycles_{after} \times CC} = \frac{1}{0.9}$$