Computer Organization & Assembly Languages

Final Exam – 2007/1/18

Dept. of Engineering Science, National Cheng Kung University

1. [General Questions]

True/False: Write your response, and provide a reason with each *False*. (24%)

T (A) Reordering code is a possible way to avoid pipeline stalls.

T (B) Forwarding is primarily an attempt to fix data hazards in a pipeline.

F (C) In an instruction set like the 80x86, where instructions are not of the same length, pipelining is considerably easier.

F (D) In a pipelined system, forwarding will eliminate the need of any stalls.

F (E) Pipelining increases throughput and reduces individual instruction execution time.

T (F) Pipeline bubbles cause loss of performance since that a number of pipeline stages are left empty.

2. [Performance - I]

Suppose that a program which executes in 100μs that loads a string into memory and prints it out. You discover a trick to loading the string into memory which will speed up that part of the program by 3 times. If loading the string took 45% of the execution before, what will the execution time be after making this change? (6%)

3. [Performance - II]

Assume that a company exclusively uses three application programs, A, B, and C for 70%, 20%, and 10% of the time, respectively. Based on the execution-time benchmarks in the table below, Dilbert, a chief executive of this company, must choose a computer system based on performance and cost. (For example, if Dilbert, spends 10% more on a computer system, he expects a 10% increase in performance.)

System	Execution Time (sec.)			Cost
	Application A	Application B	Application C	
X	90	10	15	$1,200
Y	80	25	20	$1,200
Z	75	35	30	$1,800

Finally, Dilbert recommends the purchase of computer system X because it's average performance is better than computer system Y and its cost is 33% less than computer system Z. Show whether or not Dilbert made a wise choice. (15%)

4. [Control]

Explain why there is no RegRead control signal for directing if the register file is to be read in the design of both single-cycle and multicycle datapath. (5%)

25%×2 + 75%×1

5. [Single-Cycle/Multicycle Datapath]

(A) How many clock cycles does it take for the following code to execute in a single cycle machine? and in a multicycle machine (according to our design)? (8%)

```
sub $3, $4, $5
lw  $4, 10($6)
add $7, $3, $4
sw  $7, 20($6)
add $7, $2, $1
```

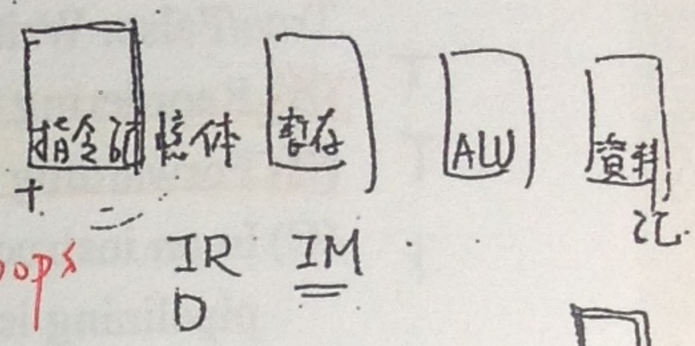
(B) Suppose that the function unit times are given as follows:

150 ps for memory access,

100 ps for ALU operation, and

80 ps for register file read or write.

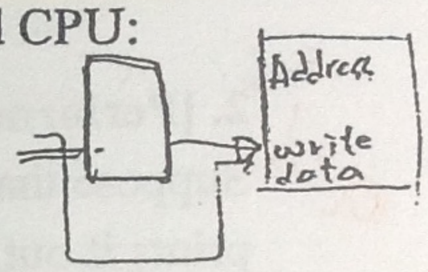
What is the execution time for running the code in (A) on a single-cycle and a multicycle machine, respectively? (8%)



6. [Pipelined Datapath]

Suppose we want to execute the following code segment on the pipelined CPU:

```
add $2, $5, $4
add $4, $2, $5
lw  $5, 100($2)
add $3, $2, $5
```



Suppose there is no hardware supports for the forwarding and the stalling, but the register file can be written at the first half of a cycle and be read at the second half.

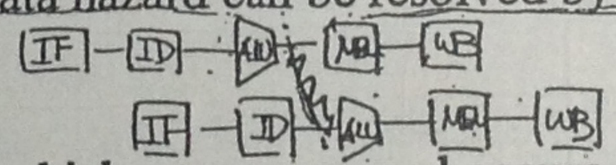
(A) How many NOPs and at what places that you can add to make the code segment execute correctly on our pipeline? (8%)

(B) Suppose the pipeline stalls, but no forwards, when there is data hazard. How many cycles will the above code segment execute? (4%)

7. [Data Hazards]

The following *read-after-write* data hazard can be resolved by forwarding:

```
add $2, $3, $4
sub $5, $2, $6
```



Consider the similar situation in which a memory read occurs after a memory write:

```
sw : $7, 100($2)
lw  $8, 100($2)
```

Write a short paragraph describing how this situation differs from the one involving registers, and describe how the potential *read-after-write* problem is resolved. (10%)

8. [Datapath Improvements]

(A) What is the major advantage of a multicycle datapath over a single-cycle one? (4%)

(B) What are the similarities between a pipelined datapath and a single-cycle datapath? Also, what are the similarities between a pipelined datapath and a multicycle datapath? (8%)