

MCS-284 Intra-term Exam 2

Serial #:

This exam is closed-book and mostly closed-notes. You may, however, use a single 8 1/2 by 11 sheet of paper with *hand-written* notes for reference. (Both sides of the sheet are OK.)

Please write your name only on this page. Do not turn the page until instructed, in order that everyone may have the same time. Then, be sure to look at all problems before deciding which one to do first. Some problems are easier than others, so plan your time accordingly. **You have 50 minutes to work.**

Write the answer to each problem on the page on which that problem appears. You may also request additional paper, which should be labeled with your test number and the problem number.

Printed name: _____

Problem	Page	Possible	Score
1	2	26	
2	4	24	
3	5	26	
4	6	24	
Total		100	

1. [**26 Points**] The designers of the next-generation MIPS architecture are considering adding two instructions to the instruction set, `sws` and `lws`. The names of these two instructions stand for Store Word Stepping and Load Word Stepping. They behave just like the normal `sw` and `lw` instructions, except that each of them also writes a new value into the base address register, found by adding the offset. For example, the following two instructions:

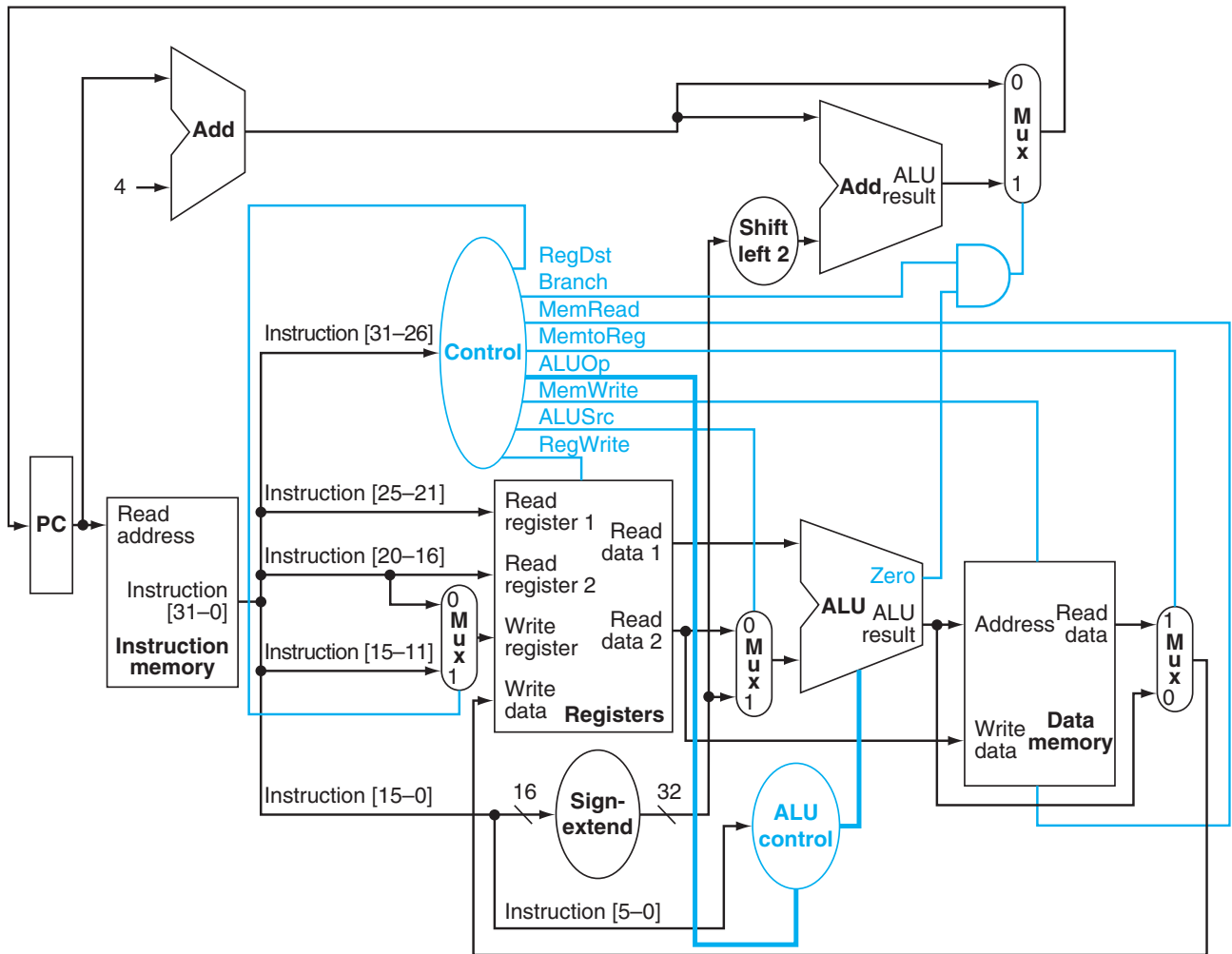
```
sw $t0, 16($t1)
addiu $t1, $t1, 16
```

could be replaced with one:

```
sws $t0, 16($t1)
```

The machine-language format of the two new instructions uses the `Rs`, `Rt`, and `Imm` fields in the same way as for `sw` and `lw`.

- (a) Modifying the textbook's processor designs to accommodate one of these instructions would require a change in the register file. (The other instruction could be accommodated using the existing register file.) Which instruction requires the register file change? What would the change be?
- (b) The datapath and control table of the single-cycle processor are reproduced on the next page. Make any modifications necessary to add the `lws` instruction. If you add any new control signals to the datapath, add columns for them to the table and show the values in those new columns in all rows, old as well as new.



Instruction	Reg Dst	ALU Src	Memto Reg	Reg Write	Mem Read	Mem Write	Branch	ALU Op1	ALU Op0
R-format	1	0	0	1	0	0	0	1	0
lw	0	1	1	1	1	0	0	0	0
sw	X	1	X	0	0	1	0	0	0
beq	X	0	X	0	0	0	1	0	1
lws									

2. [**24 Points**] For each of the following eight sequences of instructions, indicate whether the textbook's pipelined processor would use forwarding, stalling, both, or neither. As usual, we are assuming that registers are written in the first half of the clock cycle and read during the second half; this doesn't count as forwarding.

- (a) lw \$1, 0(\$2)
add \$3, \$1, \$4
add \$4, \$5, \$6
add \$5, \$6, \$7
- (b) add \$1, \$2, \$3
add \$2, \$3, \$4
add \$3, \$4, \$5
add \$5, \$4, \$3
- (c) lw \$1, 0(\$2)
add \$3, \$4, \$5
add \$4, \$5, \$6
add \$7, \$1, \$2
- (d) add \$1, \$2, \$3
add \$2, \$3, \$4
sw \$1, 4(\$3)
add \$3, \$4, \$5
- (e) add \$1, \$2, \$3
lw \$4, 0(\$5)
add \$5, \$6, \$7
add \$6, \$4, \$4
- (f) lw \$1, 0(\$2)
add \$3, \$2, \$1
add \$4, \$5, \$6
add \$5, \$1, \$6
- (g) add \$1, \$2, \$3
add \$2, \$3, \$4
sw \$3, 4(\$1)
add \$3, \$4, \$5
- (h) add \$1, \$2, \$3
add \$2, \$3, \$4
add \$3, \$4, \$5
add \$5, \$1, \$2

3. [**26 Points**]

- (a) Consider a direct-mapped cache with a total capacity of 8 words and a block size of one word. Assume the cache is initially empty and that we are using word addresses, so that words have addresses 0, 1, 2, etc. For the following sequence of addresses, indicate which are misses and which are hits. Also show which address (if any) is contained in each location of the cache at the end. To obtain partial credit, you will need to show your work by showing not just the final contents of each cache location, but also all prior contents, crossing them out when they are replaced.

5 10 4 2 10 5

- (b) Repeat the problem with a direct-mapped cache with a total capacity of 8 words and a block size of two words.

5 10 4 2 10 5

4. [**24 Points**] The diagram on the next page shows the pipelined datapath. Twelve lines on it have been marked with circled letters, (a) through (l). Suppose we execute the following sequence of instructions. During the first clock cycle, the first instruction is fetched from address 2000 in instruction memory:

```
lw $5, 8($4)
sub $4, $7, $3
sw $9, 4($8)
add $8, $2, $1
beq $5, $6, 1000
```

Suppose further that the following registers and data memory locations contain the specified values when execution starts:

<u>Registers</u>	<u>Data Memory Locations</u>
1: 10	24: 17
2: 20	28: 42
3: 30	44: 13
4: 40	48: 79
5: 50	84: 22
6: 60	88: 28
7: 70	
8: 80	
9: 90	

During the fifth clock cycle, what value is on each line? If there isn't enough information given for you to know the value, write N/A.

