

**Part I: Assembly and Machine Languages (22 pts)**

1. Assume that assembly code for the following variable definitions has already been generated (and initialization of A and length).

```
int powerOf2;           /* powerOf2 = $t0 */
int decimal;           /* decimal = $t1 */
char A[] = "1011011";  /* A = $s0 */
int length = strlen (A); /* length = $t2 */
char * p;              /* p = $t3 */
```

Write MIPS assembly code for the following code segment, using the registers specified in the comments above (to make it easier for us to go over in class). Include comments. (10 points)

```
powerOf2 = 1;
decimal = 0;
for ( p = A + length - 1; p >= A; p-- )
{
    if ( *p == 1 )
        decimal += powerOf2;
    powerOf2 *= 2;
}
```

## CS 230 Practice Final Exam

2. Assemble the following MIPS assembly language code to machine language. Write the machine language instructions in decimal, not in binary. I've done the first line as an example. (8 pts)

```
swap: add $t1, $a1, $a1      0 5 5 9 0 32
      add $t1, $t1, $t1
      add $t1, $a0, $t1
      lw  $t0, 0($t1)
      lw  $t2, 4($t1)
      sw  $t2, 0($t1)
      sw  $t0, 4($t1)
      jr  $ra
```

3. What is the difference between a *Reduced Instruction Set Computer (RISC)* and a *Complex Instruction Set Computer (CISC)*? What are some benefits and disadvantages of this architectural design? (4 pts)

### Part II: Numbers (12 pts)

4. Explain why the IEEE single precision representation for 1 is 3F800000. (4 pts)
5. What is the IEEE single precision representation for  $-2 \frac{3}{16}$ ? (4 pts)
6. The way a computer represents real numbers is called *floating point* representation. Why is this term used rather than *real number* representation? (4 pts)

### Part III: Logic Gates (10 pts)

7. Show the truth table for a two-input *exclusive-or* function and implement it using AND, OR, and NOT gates. (Reminder: The output of the exclusive-or function is true if and only if exactly one of its inputs is true.) (5 pts)
8. Show the gate diagram for a 1-bit ALU that supports the AND and OR functions. There should be two inputs and one output. Explain how this could be expanded to construct an ALU that acts on a full word. (5 pts)

**Part IV: Datapaths and Pipelining (10 pts)**

9. Using the datapath diagram provided for the homework assignment, illustrate the data and control paths used by the instruction `lw $s0, 8($sp)`. (8 pts)
10. In a single-cycle datapath, instructions are executed one at a time, one instruction in each clock “tick”. In other words, the clock cycle time is set to the time it takes any instruction to fully execute.

In both a multi-cycle datapath and a pipelined datapath, a clock “tick” is the time an instruction spends in any given stage of the datapath. Each stage has a *latency time* associated with it, which is the amount of time required for the stage to do its job. An example set of stages, with latencies, might be:

IF: Instruction fetch (200ps)  
 ID: Instruction decode and register file read (100ps)  
 EX: Execution or address calculation (e.g., time spent in ALU) (200ps)  
 MEM: Data memory access (225ps)  
 WB: Write back to register file (100ps)

In a multi-cycle datapath, instructions are executed one at a time, but only go through the stages that they need to. (E.g., R-format instructions do not have to go through MEM, because they don't read or write to memory.) In a pipelined architecture, as each instruction moves to the next stage the following instruction moves into the stage behind it. (8 pts)

- A stage's latency describes the minimum amount of time an instruction would have to spend in that stage, but, depending on the datapath type, an instruction might spend longer in a stage than its latency requires. Given the latencies described above, what would be the clock cycle time for a single-cycle datapath? For a multi-cycle datapath? For a pipelined datapath?
- How long would a `sw` instruction take in each of these three cases?
- How long would a `beq` instruction take in each of these three cases?
- In a pipelined datapath, why does it make sense to require all instructions to go through all the stages of the datapath, whether they are necessary or not? (E.g., R-format instructions have to go through MEM, even though they don't read or write to memory.)
- If we could split one stage of the pipelined datapath into two new stages, each with half the latency of the original stage, which stage would you split and what would be the new clock cycle time of the processor?

**CS 230      Practice Final Exam**

11. Describe each of the following hazard types: (6 pts)

structural hazards

control hazard

data hazard

12. Describe two methods for handling hazards: one at the hardware level and one at the software level. (4 pts)

**Part V: Memory (24 pts)**

13. Which code fragment is a better example of temporal locality? Which is a better example of spatial locality? Consider data only and not the fetching of instructions. (4 pts)

```
for ( k = 0; k < 100; k++ )    for ( k = 0; k < y; k++ )  
    a[k] = b[k] * 2.5;          sum = sum + z;
```

14. What are a cache "hit" and a cache "miss"? How are they detected? (4 pts)

15. Consider a 2-way set-associative 8-line cache with 1 4-byte word per line. Show the hits and misses and the final cache state for the following sequence of (byte) address references: 3, 16, 7, 6, 3, 5, 15, 16, 17, 20, 19, 22, 48. (8 pts)

16. Describe two replacement algorithms for set-associative memory. (4 pts)

17. What are two reasons for using virtual memory? (4 pts)

**Part VI: Parallel Processing (8 pts)**

18. What is a SIMD parallel processor? (4 pts)

19. How do the processors in a MIMD distributed memory parallel computer communicate? (4 pts)

**20. Take home question: THIS IS REAL – NOT PRACTICE! (10 pts)**

A less technically inclined friend has asked you to explain how computers work. Write a detailed, one-page description for your friend.