

## CSE378 Review Questions

Snyder

Autumn 04

The following cover the high points of CSE378.

1. What is Moore's law? State its two forms.
2. What does the acronym ISA stand for and what does it mean?
3. Give the Fetch/Execute Cycle.
4. What is the binary equivalent of the hexadecimal: BAD1 ?
5. What are the principal parts of R-type instructions in the MIPS ISA?
6. How does `Mult` differ from other R-type instructions?
7. Explain in words what the computer does with the instruction composed of all zeroes.
8. Give the principal parts of I-type instructions in the MIPS ISA.
9. Explain in words what the computer does with the instruction `2BAD 4ABE` does.
10. Explain how the MIPS instructions `LW` and `SW` specify memory addresses.
11. How does the behavior differ between the MIPS instructions `lb` and `lbu`?
12. What does "aligned" mean for memory references?
13. What is the maximum distance (in bytes) that a PC relative instruction can branch?
14. What does "big endian" mean?
15. What are the principal parts of J-format instructions?
16. What is "pseudo direct addressing"?
17. Explain the operation of the instruction `jal`.
18. Give the symbolic names, e.g. `$zero`, for the 32 MIPS registers and their use.
19. Assume that all of the "user" registers are in use when a procedure is called. Which registers would be saved before the call, which would be saved at the start of the procedure, which would be restored at the end of the procedure, and which would be restored after the return?
20. What does the phrase "Computers just store bits" refer to? Demonstrate the point by interpreting the hexadecimal `BAD1 2DAA` as a signed integer, unsigned integer, sequence of four ASCII characters, an instruction and a single precision floating point number. [For fp, give the mantissa value as a sum of reciprocal powers, e.g.  $1/2^{-1} + 1/2^{-6} + \dots$ ]
21. Describe the relationship between positive and negative values in 2s Complement Arithmetic.
22. In what way is 2s Complement "asymmetric"?
23. The MIPS ISA uses at least four number representations. What are they and when are they used?
24. What does "arithmetic overflow" mean? Why can it not happen when adding numbers of opposite signs?
25. Construct an ALU bit-slice for the four instructions: `and`, `or`, `complement` and `not-and`, where `complement` flips all bits and `not-and` is the complement of a normal `and`.
26. Add a test for "all zeroes" and a test for "all ones" to your ALU bit-slice in #26.
27. In your opinion, what is the best Seattle rock group today?
28. "People are better than computers because they know when to subtract in binary division." But, couldn't a computer be designed to figure out when and when not to subtract? Discuss.
29. Give the *double precision* floating point representation for the decimal values 10, 0 and 0.1.
30. What does NAN mean?
31. Give a MIPS instruction sequence to add two sp fp numbers whose addresses are in `$a0` and `$a1`.
32. Explain informally (as if to another CS major) why CSE378 is totally cool.
33. What does the word Normalize mean?
34. Write a MIPS assembly program to compare two strings to test `=`. Remember about the empty string.
35. Write a MIPS assembly program to evaluate the quadratic formula assuming a square-root procedure.
36. Write a MIPS assembly program to count the number of blanks in a string.
37. Write a MIPS assembly program to find the mode (most frequently occurring) of an array of numbers.
38. In pipelining the MIPS processor, why must instruction and data memory be separated?
39. What are the five stages in the MIPS pipeline?
40. Give the advantages/disadvantages of the single cycle and the multicycle processor designs.
41. Give the advantages/disadvantages of the multicycle and the pipelined processor designs.

42. What does a pipeline register contain and how does it work? (Include mention of what is needed for stalling.)
43. What are hazards, and what kinds are there? Explain how each is avoided/resolved.
44. Why do our processor designs contain several ALUs?
45. Explain what latency and throughput are relative to processor design. How does pipelining affect each?
46. How does a processor determine if by-passing is needed?
47. How does a processor determine if a stall is needed for non-branching instructions?
48. Our original pipelined design required 3 bubbles on a branch, but this was improved to 1 ... what changes made that possible?
49. How are branches predicted? Include discussion of a saturating counter.
50. What does the term precise exceptions mean?
51. What happens when a processor detects an interrupt? Exception?
52. Why is computer memory organized into a hierarchy? Give the levels for a typical processor.
53. What is the principle of locality?
54. Name 5 places where caching is used in computing.
55. What are "cache hit" and "cache miss"?
56. Describe the various types of cache addressing structure.
57. Give the various types of "write" policy? What happens on a "write miss"?
58. What are: tag, index, offset, valid bit, dirty bit?
59. Is "make the cache line as long as possible provided there are at least 64 entries" a good rule? Discuss.
60. What are the advantages/disadvantages of high degrees of associativity.
61. What are typical miss rates for direct mapped caches?
62. What is a compulsory miss?
63. What is a "write buffer"?
64. What does "LRU" mean? Is it a useful technique for L1, L2 and L3 caches?
65. Explain what a victim cache is and how it works.
66. Before virtual addressing and paging, how was memory managed?
67. What are the advantages of pages over segments?
68. What is a PTE? How does it work?
69. Does virtual address translation *have* to take two memory references? Explain.
70. What is the typical miss rate for a paging system?
71. What does the "L" in TLB stand for?
72. What is the role of a TLB? How big are they usually?
73. What is the relationship between the TLB entry and the PTE?
74. Why are precise interrupts important to paging systems?
75. What is the coolest thing you've learned in CSE378?