Beayna Grigorian Lec. 1

CS 33: COMPUTER ORGANIZATION

94

Computer Science Department
University of California, Los Angeles

John A. Rohr
David A. Smallberg
April 25, 2007

94

EXAMINATION I ANSWER SHEET

1. ALU (Arithmetic Logic Unit), Memory, Accumulator, & Control Unit
2. ALU (Arithmetic Logic Unit), Control Unit, & Register File
3. MIPS is a RISC (Reduced Instruction System Computer) machine
4. register, immediate, and jump
5. big-endian (with 8-bit memory addressing)
6. Assembler generates the machine language code from the assembly code.
7. Linker combines all the object files & machine code to generate the executable file.
8. The symbol table holds all the values for the defined symbols & is generated during the first pass.
9. Pass 1 builds the symbol table using the defined values.
10. Directives or pseudo-operations tell the compiler if what follows it are symbol/value definitions (.data) or instructions (.text).

-6

11. 33_{10}	16. $FACE_{16}$	21. $0001\ 0111\ 0110_2$	26. $1010\ 0101_2$
12. $CAFE_{16}$	17. 6523_8	22. no overflow	27. $0100\ 1000_2$
13. 573.375_{10}	18. no overflow	23. 4653_8	28. 577675_8
14. 10110111.101_2	19. 2439_{16}	24. no overflow	29. $B9F9_{16}$
15. 77754321_8	20. yes, there is overflow	25. $FACADE_{16}$	30. 042_{16}

CS 33: COMPUTER ORGANIZATION

Computer Science Department
University of California, Los Angeles

John A. Rohr
David A. Smallberg
April 25, 2007

EXAMINATION I

CLOSED BOOK

Answer each of the following questions on the answer sheet provided. The point value for each question is given in parentheses before the question number. There are a total of 100 points on the examination. All numbers are decimal integers unless otherwise indicated in a specific problem.

Only the answers on the answer sheet will be graded. Every numerical answer is either completely correct or wrong. No partial credit will be given on numerical problems. If a problems specifies operands with a given bit width, the result must have the same number of bits or it will be incorrect.

(2) 1. What are the four major components of von Neumann computer architecture?

(2) 2. What are the three major sections of a Central Processing Unit (CPU)?

(2) 3. State whether MIPS is a RISC or CISC machine and the meaning of the acronym you select.

(2) 4. What are the three primary instruction formats used in the MIPS processor?

(2) 5. What is the primary addressing mode used in the MIPS processor?

(2) 6. What is the purpose of an assembler?

(2) 7. What is the purpose of a linker?

(2) 8. What is the purpose of the symbol table in an assembler?

(2) 9. What is the purpose of Pass 1 in a two-pass assembler?

(2) 10. What is the purpose of a directive or pseudo-operation?

(4) 11. Convert the unsigned binary number **100001** to decimal.

(4) 12. Convert the unsigned decimal number **51966** to hexadecimal.

(4) 13. Convert the unsigned octal number **1075.3** to decimal.

(4) 14. Convert the unsigned decimal number **183.625** to binary.

(4) 15. Convert the decimal number **-10031** to **24-bit** two's-complement octal.

(4) 16. Generate the negative of the 16-bit two's-complement hexadecimal number **0532**.

(4) 17. Add the unsigned 12-bit octal numbers: **3742** + **2561**.

(4) 18. State whether or not the calculation in the problem above generates an **overflow.**

(4) 19. Add the unsigned 16-bit hexadecimal numbers: **9A7D** + **89BC**.

(4) 20. State whether or not the calculation in the problem above generates an **overflow.**

(4) 21. Add the two's-complement 12-bit binary numbers:
0100 1100 0001 + **1100 1011 0101**.

(4) 22. State whether or not the calculation in the problem above generates an **overflow**.

(4) 23. Add the two's-complement 16-bit octal numbers: **5236** + **7415**.

(4) 24. State whether or not the calculation in the problem above generates an **overflow**.

(4) 25. Subtract the two's-complement 24-bit hexadecimal numbers:
BEA DED – **C3E 30F**.

(4) 26. Subtract the two's-complement 8-bit binary numbers: **1100 0011** – **0001 1110**.

(4) 27. Compute the **AND** of the 8-bit binary numbers: **1100 1001** **AND** **0110 1110**.

(4) 28. Compute the **OR** of the 18-bit octal numbers: **532 674** **OR** **457 231**.

(4) 29. Compute the **XOR** (Exclusive-Or) of the 16-bit hexadecimal numbers:
1234 **XOR** **ABCD**.

(4) 30. Compute the **NOR** of the 12-bit hexadecimal numbers: **5BC** **NOR** **F99**.