CS 250 Spring 2017 Homework 04 SOLUTION

Due 11:58pm Wednesday, February 08, 2017
Submit your typewritten file in PDF format to Blackboard.

1. Complete the table to show how the given binary strings are interpreted in each data representation. For numerical representations write your answer in the form of a decimal number. Use care when writing a decimal number equivalent to show a sign when there is ambiguity if a sign is not shown. If a binary string is not valid for a given representation, write "error" in the table.

Given binary string	Packed BCD	Binary signed digit
00000000	00	0
00010001	11	5
0x81	81	1
0xFF	Error	-15

2. Complete the table to show how the given binary strings are interpreted in IEEE 754 (Standard for Floating Point Arithmetic). Lecture 6-7 v5 Slide 56 will be quite helpful in answering this question and Question 3, next.
Scientific notation in base 2 is the form ±1.b…b x 2^n where b denotes a non-zero bit unless all the bits in the fractional weighted positions are zero in which case a single b bit equal to zero should be shown, and where n denotes a sign-magnitude decimal integer.
Base 10 integers have the form ±d…d where d represents a significant decimal digit.

Given binary string	Value when interpreted per IEEE 754 and written in base 2 scientific form	Value when further converted to base 10 integer; if this conversion is not valid then write "error"
1 10000001 11000000000000000000000	-1.11 x 2^2	-7
0x81000000	-1.0 x 2^{-125}	-2.35…x10^{-38} obtained using a calculator with y^x capability
0xFFFFFFFF	NaN	error

3. Complete the table to show in hexadecimal notation, the IEEE 754 representation of the given numbers and symbols.

Given value	IEEE 754 representation in 0x notation
-2.0 (base 10)	0xC0000000
768 (base 10)	0x44400000
+∞	0x7F800000

4. Using congruence modulo 2^{10}, a circuit maps 32-bit unsigned integer addresses for a main memory onto the addresses of a smaller memory having only 2^{10} locations.
   a. How many addresses are mapped onto each of the locations in the small memory? Express your answer in both exponential form and in weighted positional notation that incorporates the appropriate suffix from the set {kilo-, mega-, giga-, tera-, exa-, and peta-} to yield the simplest result.
   Answer: 2^{22} addresses, or 4 mega-addresses. The number of addresses is 2^{32} and the number of locations is 2^{10}. So the number of addresses per location is $2^{32}/2^{10} = 2^{22}$ or 4 mega-addresses mapped to each location in the smaller memory.

   b. Draw a schematic of a circuit that takes as input a single 32-bit address, performs the congruence mapping computation, and then outputs the intended address for the smaller memory. Label the input address $A_{31}A_{30} \ldots A_1A_0$. Label the output address bits $B_iB_{i-1} \ldots B_1B_0$.
   Answer: Draw 32 parallel wires and label them so that each wire carries one of the 32 input address bits, $A_{31}A_{30} \ldots A_1A_0$. Then extend the wires for the inputs $A_9A_8 \ldots A_1A_0$ so they are clearly longer and then label the outputs using $B_9B_8 \ldots B_1B_0$ so that B_i is paired with A_i. Computation of modulo 2^{10} is carried out simply by not connecting the 20 wires carrying the 20 most significant bits of the input address to the output. This circuit to compute modulo mapping from a large space to a smaller one is cheap and fast.

5. Over time processor circuits have become faster than memory circuits. What aspect of the Harvard Architecture gives it an advantage in comparison with the Von Neumann Architecture for a computer processor designer trying use parallelism to speed up the Fetch-Execute Cycle?
Answer: The separation of memory into a part for instructions and a part for data and each with a path to/from the processor means that the Fetch action could be overlapped with the Execute action for the Harvard Architecture while the Von Neumann Architecture processor has only one path to/from memory, so instruction Fetch cannot overlap with access of operands or writing of results for Execution.

6. Design a simple, single instruction format, in the style shown in Figure 5.1, and machine language for a computer with the following characteristics. Use the table below as a framework for your answer.

   The instructions for the ISA of this computer include only the Boolean operations And, Or, Not, and Exclusive Or. An instruction bit string specifies the location of each operand to be read from memory by holding a pointer to the desired memory location in a designated operand field. The same strategy is used to specify the location for the result to be written. Each operand and the result for each instruction is an 8-bit binary string. The computer memory has 2^{10} storage locations. Each memory location holds a one-byte bit string.

   Using Figure 5.1 as a guide, define this four-instruction ISA by creating a tabular diagram with four rows, one for each of the instructions. Clearly label the fields of your instruction format, and let the leftmost field be the Operation Code, or Opcode, field, followed by all operand fields, and ending with the result field. If a field is not needed, give that field the name "Unused." Next, using the minimum possible number of bits sufficient for each field

to perform its function, specify how long each field in your instruction format is. Finally, for any field in the format for which the bit string is must be known before writing any programs for this computer, use your Designer's Prerogative to choose the bit string that will fill that field and display your choice in your format for each instruction.
Answer:

Operation	**Opcode**	Operand 1	Operand 2	Result
AND	00	10-bit field	10-bit field	10-bit field
OR	01	10-bit field	10-bit field	10-bit field
NOT	10	10-bit field	**Unused,** 10-bit field	10-bit field
XOR	**11**	10-bit field	10-bit field	10-bit field

7. If a computer can add, subtract, multiply, and divide 16-bit integers, 32-bit integers, 32-bit floating-point values, and 64-bit floating-point values, how many unique opcodes will be needed? Hint: an opcode defines not only the operation to be performed but also the data types of its operand(s) and result.
Answer: 16 opcodes are needed, one for each operation and data type combination.