PRCMP PL02
Storing data in computers

May 2023

1 Storing unsigned integers in computers

We are used to exploring some degree of freedom when representing numbers. Typically, we write only strictly necessary digits; additionally, we have symbols to denote signs (positive or negative) and to separate the integer and fractional parts (the radix character or decimal mark).

However, computers represent values in rigid and strict ways. Although this rigidity does not seem natural to humans, it simplifies computers' practical implementation and operation.

Processors have general-purpose registers, which are memories directly accessed and operated by the Arithmetic Logic Unit (ALU), the processor component where all arithmetic and logic calculations occur. Such registers store a fixed number of bits, which usually match the size of ALU input and output operands. Typical register sizes are 16-bit (for 8-bit and 16-bit architectures), 32-bit and 64-bit (for 32-bit and 64-bit architectures, respectively).

In some instruction sets, the registers can operate in diverse modes, breaking down their storage memory into smaller ones. For example, a 32-bit register can store 32-bit, 16-bit or 8-bit words, depending on the load instruction executed.

However, there are cases in which registers do not have the size required to store some quantities of the problem: e.g. to sum two 64-bit numbers on a 32-bit processor. In such cases, we operate sequentially on partial values. For this example, we would first sum the lower 32-bits and then the higher 32-bits.

The programmer selects the appropriate size to store each quantity in the program. In other words, it is the programmer who chooses the size of the variables. An efficient approach is to select sizes that match the register modes or (if necessary) some of their multiples: 8-, 16-, 32-, 64- and 128-bit.

Questions

1. The relative humidity is a physical quantity that ranges from 0% to 100%.
   (a) How many bits are required to represent the relative humidity?
   (b) Taking into account the common sizes of registers, what would be the minimum size you would choose to store one value of relative humidity?
   (c) Let us assume that the relative humidity of a point of interest is 83%. Write this value with the number of bits you selected in the previous question.
   (d) Write the same number if you have chosen to store it in a word of a 32-bit processor.
2. Determine the range of unsigned integers (minimum and maximum values) you can store for each number of bits.
   (a) 4 bits.
   (b) 8 bits.
   (c) 16 bits.
   (d) 32 bits.

(e) 64 bits.

(f) 128 bits.

3. Determine the minimum number of bits required to store values of the following physical quantities. Which ones fit in a 16-bit word?

   (a) Speed of an aeroplane: [0; 540] knots.

   (b) The altitude of an aeroplane: [0; 41 000] feet.

   (c) Speed of a car: [0; 230] km/h.

   (d) The attendance at a football match: [0; 60 000] people.

   (e) Distance between Earth and Moon: [363 104; 405 696] km

2 Representing signed integers on computers

Questions

4. Represent the following signed (decimal) integers in 8-bit (1) sign–magnitude, (2) one's complement e (3) two's complement.

   (a) 7

   (b) -7

   (c) 0

   (d) -0

   (e) 15

   (f) -21

   (g) 28

   (h) -30

5. Represent the following signed (decimal) integers in 8-bit two's complement.

   (a) 100

   (b) -100

   (c) 71

   (d) -33

   (e) 13

   (f) 1

   (g) -1

   (h) -128

6. What values (in decimal) are represented by the following numbers in 8-bit two's complement?

   (a) $1001\,0010_2$

   (b) $0101\,1010_2$

   (c) $1000\,0000_2$

   (d) $0000\,0100_2$

   (e) $A5_{16}$

   (f) $5A_{16}$

   (g) $0F_{16}$
   (h) FF_{16}
7. Represent 26 in:
   (a) 8-bit two's complement.
   (b) 16-bit two's complement.
   (c) 32-bit two's complement.
   (d) Do you find any relation between the three results? Can you transform the 8-bit number into the 32-bit number?
8. Represent -24 in:
   (a) 8-bit two's complement.
   (b) 16-bit two's complement.
   (c) 32-bit two's complement.
   (d) Do you find any relation between the three results? Can you transform the 8-bit number into the 32-bit number?
9. Compute the following sums in 8-bit two's complement. Check if each result is arithmetically correct.
   (a) $8 + (-14)$
   (b) $11 + (-20)$
   (c) $80 + 50$
10. Compute the following subtractions in 8-bit two's complement. Check if each result is arithmetically correct.
   (a) $8 - 14$
   (b) $11 - 20$
   (c) $(-80) - 50$

3 Representing floating point numbers on computers

Questions

11. What value (in binary) does each number represent in single-precision floating-point format (32 bits, IEEE 754)?
   (a) 1 10001001 10100100000000000000000
   (b) 0 01111010 01110000000000000000000
12. How is each of the following values represented in the single-precision floating-point format (32 bits, IEEE 754)?
   (a) $-0,000101101101$
   (b) $1110,11011$

4 Representing character sequences.

Questions

13. How will the following sentence be encoded in ASCII?

    `PRCMP: the best course in the world!`

14. What sentence is encoded in ASCII by the following sequence of values?

    084 104 101 032 113 117 105 099 107 032 098 114 111 119 110 032 102 111 120 044 032 106 117 109 112 115 032 111 118 101 114 032 116 104 101 032 108 097 122 121 032 100 111 103 033 000

5 Solutions

1. (a) $\log_2(100) = 6.644 \Rightarrow$ 7 bits, allow to represent integers from 0 to 127.
   (b) 8 bits: contemporary architectures use 8, 16, 32 or 64 bits words.
   (c) 0101 0011
   (d) 0000 0000 0000 0000 0000 0000 0101 0011
2. (a) $\{0, \ldots, 15\}$
   (b) $\{0, \ldots, 255\}$
   (c) $\{0, \ldots, 65\,535\}$
   (d) $\{0, \ldots, 4\,294\,967\,295\}$
   (e) $\{0, \ldots, 18\,446\,744\,073\,709\,551\,615\}$
   (f) $\left\{0, \ldots, 3,4028236692 \times 10^{38}\right\}$
3. (a) $\log_2(540) = 9.077 \Rightarrow$ 10 bits $(2^9 \leq 540 < 2^{10})$
   (b) $\log_2(230) = 7.845 \Rightarrow$ 8 bits: $(2^7 \leq 230 < 2^8)$
   (c) $\log_2(60\,000) = 15.873 \Rightarrow$ 16 bits: $(2^{15} \leq 60\,000 < 2^{16})$
   (d) $\log_2(405\,696) = 18.630 \Rightarrow$ 19 bits: $(2^{18} \leq 405\,696 < 2^{19})$
4. (a) S+M: 0000 0111 ; C/1: 0000 0111 ; C/2: 0000 0111
   (b) S+M: 1000 0111 ; C/1: 1111 1000 ; C/2: 1111 1001
   (c) S+M: 0000 0000 ; C/1: 0000 0000 ; C/2: 0000 0000
   (d) S+M: 1000 0000 ; C/1: 1111 1111 ; C/2: Not represented.
   (e) S+M: 0000 1111 ; C/1: 0000 1111 ; C/2: 0000 1111
   (f) S+M: 1001 0101 ; C/1: 1110 1010 ; C/2: 1110 1011
   (g) S+M: 0001 1100 ; C/1: 0001 1100 ; C/2: 0001 1100
   (h) S+M: 1001 1110 ; C/1: 1110 0001 ; C/2: 1110 0010
5. (a) 0110 0100
   (b) 1001 1100
   (c) 0100 0111
   (d) 1101 1111
   (e) 0000 1101
   (f) 0000 0001
   (g) 1111 1111
   (h) 1000 0000
6. (a) -110
   (b) 90
   (c) -128
   (d) 4
   (e) -91
   (f) 90
   (g) 15

   (h) -1

7. (a) 0001 1010
   (b) 0000 0000 0001 1010
   (c) 0000 0000 0000 0000 0000 0000 0001 1010
   (d) There is a relation.
8. (a) 1110 1000
   (b) 1111 1111 1110 1000
   (c) 1111 1111 1111 1111 1111 1111 1110 1000
   (d) There is a relation.
9. (a) $0000\,1000 + 1111\,0010 = 1111\,1010 \Rightarrow$ Correct.
   (b) $0000\,1011 + 1110\,1100 = 1111\,0111 \Rightarrow$ Correct.
   (c) $0101\,0000 + 0011\,0010 = 1000\,0010 \Rightarrow$ Incorrect.
10. (a) $0000\,1000 - 0000\,1110 = 1111\,1010 \Rightarrow$ Correct.
    (b) $0000\,1011 + 0001\,0100 = 1111\,0111 \Rightarrow$ Correct.
    (c) $1011\,0000 - 0011\,0010 = 0111\,1110 \Rightarrow$ Incorrect.
11. (a) -1680

    Sign: negative
    Exponent: 2^{10} (i.e. $137 - 127$)
    Mantissa: 1.10100100000000000000000

    (b) 0.044921875

    Sign: positive
    Exponent: 2^{-5} (i.e. $122 - 127$)
    Mantissa: 1.01110000000000000000000
12. (a) $-0,000101101101 = -1.01101101 \times 2^{-4}$

    Sign (1 bit): negative
    Exponent (8 bits): $2^{-4} \Rightarrow -4 + 127 = 123 = 0111\,1011_2$
    Mantissa (23 bits): 01101101000000000000000
    Result: 1 01111011 01101101000000000000000

    (b) $1110.11011 = 1.11011011 \times 2^{3}$

    Sign (1 bit): positive
    Exponent (8 bits): $2^{3} \Rightarrow 3 + 127 = 130 = 1000\,0010_2$
    Mantissa (23 bits): 11011011000000000000000
    Result: 0 10000010 11011011000000000000000
13. 080 082 067 077 080 058 032 116 104 101 032 098 101 115 116 032 099 111 117 114 115 101 032 105 110 032 116 104 101 032 119 111 114 108 100 033
14. The quick brown fox, jumps over the lazy dog!