COMPS266F : Computer Architecture

Tutorial 1 : Programmable Computers

*Section 1 : Revision of Concepts*

|  |
| --- |
| Question 1 : A \_\_\_\_\_\_\_\_\_\_\_ is a computer that can be re-purposed. |
| Programmable computer |

|  |
| --- |
| Question 2 : \_\_\_\_\_\_\_\_\_\_\_\_ is a major component of a programmable computer, which is responsible for the execution of instructions. |
| Arithmetic Processing Unit |

|  |
| --- |
| Question 3 : Memory is a component that supports \_\_\_\_\_\_\_\_\_\_\_ a previously stored data. |
| Overwrite |

|  |
| --- |
| Question 4 : \_\_\_\_\_\_\_\_\_ are two components that connect the programmable computer and the outside world. |
| I / O |

|  |
| --- |
| Question 5 : The ALU and the programmable computer use\_\_\_\_\_\_\_\_\_\_ to code data in electrical signals. |
| Binary (bits) |

|  |
| --- |
| Question 6 : The ALU uses \_\_\_\_\_\_\_\_ to code numbers for arithmetic and logic operations. |
|  |

|  |
| --- |
| Question 7 : The ALU uses \_\_\_\_\_\_\_\_\_ for addition and subtraction of positive and negative numbers. |
|  |

*Section 2 : Numbering Systems*

A number is a representation of its value in a meaningful form. The value of a number expressed in base b.

**Method 1 : Convert decimal number to base b**

1. Check whether the number is greater than zero (0).
2. If the number is less than the base b, the number is the representation of its value in the base b. Otherwise, go to step 3 and the following steps.
3. Divide the number by the base b to get the quotient and remainder. (That is short division.)
4. Keep dividing the number by b to get the quotients and remainders until it cannot be divided. That means the quotient must be less than the base b.
5. (Optional) If you are not good in number conversion, you can do the division one more time until the quotient becomes 0 to get the last remainder.
6. Present all the remainders from the bottom to the top, then the number will be the representation of the base b.

Example 1 : Convert 7910 to binary.

1. 79 > 0
2. 79 > 2 also (Binary, the base is 2.)
3. Show the calculation steps in conversion.  
   79 / 2 = 39 ... 1 (1st turn, we get the first quotient 39 and the remainder 1.)
4. The quotient is still larger than the base, keep dividing.  
   39 / 2 = 19 ... 1 (2nd turn, we pick the quotient 39 from the 1st turn to calculate the 2nd quotient and remainder.)  
   19 / 2 = 9 ... 1 (Repeat the division until the quotient cannot be divided by the base 2.)  
   9 / 2 = 4 .... 1  
   4 / 2 = 2 ... 0 (The quotient from the last turn is 4 then it is divisible by 2, but we still need to get the remainder 0.)  
   2 / 2 = 1 ... 0 (We can stop here since the quotient cannot be divided anymore.)

--------------

1. 1 / 2 = 0 ... 1 (This step is optional and for the classmates who are not familiar.)

--------------

1. Result presentation

79 = 100 11112

We get (the last quotient 1 and) all the remainders to become the representation of the number 7910 in binary format 100 11112. Generally, we will add a space for every 4 bits from right to left.

**Method 2 : Convert the base b number to decimal**

We use the remainder theorem to convert decimal integers into different numbering systems. We know that in each carry in a numbering system, the value increases one exponent of the base. The rightmost must be a unit, which means base to the power of 0 (base0). Then increment the exponent by 1 to the base in each carry, we get base1, base2, etc.

1. Starting from the rightmost digit, the digit times to the base0, the 2nd rightmost digit times to the base1, etc. until the digit placed at n times to the basen-1.
2. Sum all the values.

Example 2 : Convert 1 10102 to decimal

1. (1 x 24) + (1 x 23) + (0 x 22) + (1 x 21) + (0 x 20)
2. = 16 + 0 + 8 + 0 + 2 + 0 = 2610

We can do it in a simple way,

Step 1 and 2 : (1 x 24) + (1 x 23) + (1 x 2) = 2610 OR 24 + 23 + 2 = 26­­10

(We can omit the terms which results in zero.)

We explored how to convert a number to another numbering system. On the computer, there are four main numbering systems. They are binary, decimal, octal (base 8) and hexadecimal (base 16). Octal should be easy, the range is 0~7 in a digit, each carry is increment by 1 to the exponent of the base 8. Like the decimal system, 108 = 1 x 81 + 0 x 80 = 810.

In hexadecimal, since the range is from 0~15. However, we cannot use 10~15 to represent its values.

E.g. 1016 = 1 x 161 = 1610. If we use a "10" to represent the value of 10 (ten), the "10" in representing the value of 1010 is conflicted to the "10" in representing the value of 1610. The issue is the same in 11~15. So, to clarify the representation, we use letter "A~F" to represent the values of 10~15 in hexadecimal representation.

No matter octal or hexadecimal, their representation is related to binary representation. We group the largest bits to the largest value in each digit (i.e. 7, 1112 in octal and 15, 11112 in hexadecimal) We can convert those numbers into binary form efficiently.

Reversely, we can re-order a binary number with a 3-bit / 4-bit groups starting from the rightmost that the number can be converted into octal and hexadecimal easily.

Example 3 : Convert 10010101102 into octal.

Step 1 : Re-order the binary number into 3-bit groups. If there is insufficient bit(s), fill "0"(s) to the leftmost (must be, 1~2 zero(s) should be filled only) group.

1001010110 = "00"1 001 010 110

Step 2 : Convert each binary group into decimal then combine them together to become the result in octal.

001 001 010 110 = 1 1 2 6 = 11268

Example 4 : Convert 3168 into binary.

Similarly, we can convert an octal number into its binary form by reversing the steps above.

Step 1 : Convert each digit into 3-bit binary form. (if the value does not contain 3-bit, e.g. 0~3, fill zero(s) until 3 bits)

3168 = “0”11 “00”1 1102

Step 2 : Combine them together to become the result. If zero(s) is/are at the leftmost, delete it/them.

011 001 110 = 0 1100 1110 = 1100 11102

Example 5 : Convert 1011000101012 into hexadecimal.

Step 1 : Convert the binary group in 4-bit groups.

101100010101 = 1011 0001 0101

Step 2 : Convert each group into decimal, if the value is 10~15, use A~F instead.

1011 = 11 = B, 0001 = 1, 0101 = 5

1011000101012 = B1516

Example 6 : Convert F4A16 into binary.

F = 15 = 1111, 4 = 0100 (fill zero until 4 bits), A= 10 = 1010

F4A = 1111 0100 10102

Converting an octal / a hexadecimal to/from the decimal, **methods 1 and 2** stated in the previous section are always applied.

|  |  |
| --- | --- |
| Question 8 : Convert 26610 into binary.  Answer : 1 0000 10102 | Question 9 : Convert 31210 into octal.  Answer : 4708 |
| 266 / 2 = 133 … 0  133 / 2 = 66 … 1  66 / 2 = 33 … 0  33 / 2 = 16 … 1  16 / 2 = 8 … 0  8 / 2 = 4 … 0  4 / 2 = 2 … 0  2 / 2 = 1 … 0  1 | 312 = 256 + 32 + 16 + 8  = 28 + 25 + 24 + 23  = 100 111 000  = 4 7 08 |

|  |  |
| --- | --- |
| Question 10 : Convert 45110 into hexadecimal. Answer : 1C316 | Question 11 : Convert 001101012 into decimal. Answer : 5310 |
| 451 = 256 + 128 + 64 + 2 + 1  = 1 1100 0011  = 1 C 316 | 1 + 4 + 16 + 32 = 5310 |

|  |  |
| --- | --- |
| Question 12 : Convert 25178 into decimal.  Answer : 135910 | Question 13 : Convert B93A16 into decimal.  Answer : 4741810 |
| 7 + 8 + 64\*5 + 512\*2 = 1359 |  |

|  |
| --- |
| Question 14 : Convert 3167 into decimal. Answer : 16010 |
| 3 \* 72 + 1 \* 7 + 6 \* 1 = 160 |

|  |
| --- |
| Question 15 : How many bits does it require to represent 64310 in a positive binary form? |
| 512 < 643 < 1024 -> so 10 bits is required |

|  |
| --- |
| Question 16 : How many bits does it require to represent 6438 in a positive binary form? |
| 9 |

|  |
| --- |
| Question 17 : How many bits does it require to represent 64316 in a positive binary form? |
| 11 |

|  |
| --- |
| Question 18 : Evaluate 238 + 2310 and show the answer in binary form. |
| 238 = 2\*8 + 3 = 1910  19 + 23 = 42  42 = 32 + 8 + 2 = 101010 |

|  |
| --- |
| Question 19 : Evaluate 1011016 + 110118 and show the answer in hexadecimal form. |
| 110118 = 001 001 000 001 001 = 0001 0010 0000 1001 = 120916  10110 + 1209 = 1131916 |

|  |
| --- |
| Question 20 : Evaluate ACE16 + 10210 and show the answer in octal form. |
| 1010 1100 1110 = 101 011 001 110 = 53168  102 = 1100110 = 1468  54648 |

|  |
| --- |
| Question 21 : Evaluate 1011012 x 1012 and show the answer in the binary form. |
| Addition:  101101 = 32 + 8 + 4 + 1 = 45  45 x 5 = 225  225 / 2 = 162 … 1  162 / 2 = 81 … 0  81 / 2 = 40 … 1  40 / 2 = 20 … 0  20 / 2 = 10 … 0  10 / 2 = 5 … 0  5 / 2 = 2 ... 1  2 / 2 = 1 … 0  1  1010001012 |

*Section 3 : Discussion*

Consider how to be a wise PC hardware buyer.

|  |  |  |
| --- | --- | --- |
| *Hardware* | *Price* | *Time to process 1 Kbyte data* |
| **Core 2 Duo E8200 (2.66GHz)** | **$1260** | **0.019 microseconds (assuming 10 operations per Kbyte)** |
| Core 2 Duo E8400 (3 GHz) | $1310 | 0.017 microseconds (assuming 10 operations per Kbyte) |
| **DDR3 1375 (PC3-11000)** | **$1450** | **18 microseconds (~0.000018 seconds)** |
| DDR3 1800 (PC3-14400) | $1650 | 14 microseconds (~0.000014 seconds) |
| **500GB 7200 RPM (Brand 1)** | **$570** | **30 milliseconds (~0.030 seconds)** |
| 500GB 7200 RPM (Brand 2) | $615 | 27 milliseconds (~0.027 seconds) |

|  |
| --- |
| Question 22 : Currently the configuration you are considering includes components in bold as below. Assuming that when you use the computer, it spends about 1% of the time doing disk I/O transfer, think whether upgrading the CPU, the memory, or the hard disk is a smarter decision. |
|  |