Sec 4.2 Computer Applications of Integers

June 12, 2016 10:22 AM

Section 4.2 – Computer Applications of Integers

Comp 232
Instructor: Robert Mearns

1. The basic memory unit of a digital computer can be interpreted many ways. We have previously looked at Boolean use of the integers: $0 \equiv$ False, $1 \equiv$ True In this section we add some of the other ways that integers 0, 1 are used. A summary of these applications include:

Boolean Variables	Integers other than 0,1	Addresses For memory	Character: Alphabetic and Special characters	Real numbers

2. For integers other than 0, 1 we first look at our Base Ten number system closely:

   a) When we state 965 we mean: 9X10^2 + 6X10^1 + 5X10^0
   this is why we say 9 hundred sixty five

   b) We have ten symbols in our number system: 0,1,2,3,……9 and we use the powers of ten to give meaning to the digits. Hence we call the system Base Ten or Decimal system

3. a) In a digital computer circuit we only have two states so: we only have 2 different symbols

   b) We must construct an arithmetic system that has only has two symbols. We will still use the symbols 0 and 1 (they come from power off or power on) but we will interpret them differently. They will not represent False, True as in Boolean Algebra.
   Since we only have two symbols 0,1 we will use powers of two to give meaning to the digits. We will call the system Base Two or Binary system

1

4. Comparing the Decimal and Binary number systems

Topic	Base Ten System	Base Two System
Number of symbols	10	2
Typical Example	6X10^3 + 0X10^2 + 8X10^1 + 5X10^0	1X2^3 + 1X2^2 + 0X2^1 + 1X2^0
Meaning in base 10	6000 + 0 + 80 + 5 = 6085	8 + 4 + 0 + 1 = 13
How do we count	0 represents zero 1 one .... 9 nine 10 ten (we need 2 digits) 11 eleven	0 represents zero 1 one 10 two (we need 2 digits) 11 three 100 four (we need 3 digits) 101 five
Basic addition tables	We learned our addition "tables" and we do not need to think very hard to get our answers: 3 + 4 = 7	We will have to learn our addition "table" for Binary arithmetic: 0+0=0, 0+1=1, 1+1=10 because 10 represents two <u>Source of confusion:</u> 1+1=1 in Boolean algebra 1+1 = 2_{10} in Base ten arithmetic 1+1 =10_2 in Base two arithmetic

3

Topic	Base Ten System	Base Two System
How do we add	1 1 cary digits 4 3 + 5 9 1 0 2	1 1 cary digits 1 1 + 1 1 1 1 0

5. Next we look at the use of bits 0, 1 to express an address for a computer memory location

a) A typical address looks like: 0111101010111111

b) Since this is difficult to read, the bits (0,1), are listed as:

0111 1010 1011 1111 (groups of 4 or groups of 3)

c) Since this was still inconvenient a search was made so we could express each group of 4 bits with a single character

Ex: 0111 = 0X2^3+1X2^2+1X2^1+1X2^0 = seven (7 is a signle character, so 7 is used)

$1010 = 1\times2^3 + 0\times2^2 + 1\times2^1 + 0\times2^0 =$ ten (10 is not a single character, so A is used)

$1011 = 1\times2^3 + 0\times2^2 + 1\times2^1 + 1\times2^0 =$ eleven (11 is not a single character, so B is used)

$1111 = 1\times2^3 + 1\times2^2 + 1\times2^1 + 1\times2^0 =$ fifteen (15 is not a single character, so F is used)

so the memory address above is written as: 7 A B F

d) Summary of the representation of the content of 4 bits together:

0,1,2,3,4,5,6,7,8,9 representation is not changed.
A represents ten, B=eleven, C=twelve, D=thirteen, E=fourteen, F=fifteen

d) Since four bits can be at most 1111 = 15(10) . Hence for 16(10) we use more that 4 bits and do not need a single character to represent 16.

10, 11,12, 13, 14,15

e) This number system has symbols 0,1,2,3………..,9, A, B, C, D, E, F.

There are 16 symbols and hence system is called Base Sixteen or the Hexadecimal system

6. a) Bits can also represent characters from language alphabets and special characters.

To do this a Binary code was developed that, at first, used seven bits and gave a code for each English alphabetic character (A..Z and a..z), the numerals (0,1…9), some special characters (example: @, +, *) and keyboard keys (example: space, enter). The code was called the

ASCII code (American Standard Code for Information Interchange)

Ex: How many possibilities are there with a 7-bit binary code 2^7 - 1 = 127 (excluding all 0 code)

b) The ASCII code has been expanded using more than 7 bits to represent many more characters. Then the complete binary code was assigned to other languages and applications. Depending on the Character Set selected, a single code value can represent many different characters. We can now quickly change a complete character set from one language or application to another. Potential problem exists if you input using one character set and output device uses a different set.

Ex 1: The character "}" was given a 7 bit code that can be written as 7 D. In this code we are grouping 3 bits together then 4 bits as we did with an address with Base Sixteen.

Hence: 7X16^1 + DX16^0 = 7X16^1 + 13X16^0 = 112 + 13 = 125 in Base Ten

Ex 2: An 8 bit ASCII code of F B has what Base Ten value ?

FX16^1 + BX16^0 = 15X16^1 + 11X16^0 = 251 in Base Ten

7. There is one more number system sometimes seen in computers:
We start with groups of 3 bits instead of 4 bits

Ex: A typical address looks like: 1 111 101 010 111 111

(i) Write in groups of three Start at right, pad at left if necessary	001 111 101 010 111 111
(ii) What is the maximum Base 10 value when we Interpret three bits as base 2 Digits ?	1X2^2 + 1X2^1 + 1X2^0 = 7 (Base Ten)
(iii) How many symbols do we Need if one symbol is used For each group of 3 bits ?	Eight
(iv) What are the symbols and what is the base value ?	0, 1, 2, 3, 4, 5, 6, 7 (we do not need symbols 8, 9 why?) It is a Base Eight system. Also called Octal.
(v) What is the representation of above (i) represented in Base Eight ?	1 7 5 2 7 7 8 (written in Base Eight)
(vi) If a Base Eight value of 273_8 is given for a character code what is the value in Base Ten ?	2X8^2 + 7X8^1 + 3X8^0 = 187(written in Base Ten)

9

8. Changing from one base to another:

Non-Base Ten to Base Ten
(use powers of given base)

Base Ten to non-Base Ten
(use the Division algorithm with the desired base number as the divisor)

Step 1 Divide the number by the desired base and record the remainder
Step 2 Divide the previous step quotient by the desired base and record the remainder
Continue until the quotient equals 0

The desired number is the remainder digits in order: last to first.

Base Two to Base 16 or 8
or reverse:
(consider groups of 4 or 8 bits repectively)

Ex 1: 1110_2 = ? Base Ten

1X2^3 + 1X2^2 + 1X2^1 + 0X2^0 = 14 (written in Base Ten)

Ex 2: 11101011_2 = ? Base Sixteen

Change form 1110 1011
Group 1: 1X2^3 + 1X2^2 + 1X2^1 + 0X2^0 = 14 = E
Group 2: 1X2^3 + 0X2^2 + 1X2^1 + 1X2^0 = 11 = B
Final answer: E B

Ex 3: 745_8 = ? Base Two

Group 1: 7(Base Eight) = 1X2^2 + 1X2^1 + 1X2^0 = 111
Group 2: 4(Base Eight) = 1X2^2 + 0X2^1 + 0X2^0 = 100
Group 3: 5(Base Eight) = 1X2^2 + 0X2^1 + 1X2^0 = 101
Final answer: 111 100 101

Ex 4: 87_{10} = ? Base Two

87/2=43, remainder 1; 43/2=21, remainder 1;
21/2=10, remainder 1; 10/2=5, remainder 0;
5/2=2, remainder 1; 2/2=1, remainder 0;
1/2=0, remainder 1

Answer writing remainders starting with the last one:

1010111

9. Real Numbers

a) A Fixed Point form example: 3517.89

b) A Floating Point form example: .351789 E+04

c) In a Floating Point form the decimal part is called the: Mantissa (.351789)

d) The Binary codes for the Mantissa and Exponent are converted separately.

10. Maximum size of numbers is related to the storage configuration:

A system has a Maximum Integer:

Ex 1: If we have a system that stores Integers with a maximum of 2 bits
(i) what is the total number of different Permutations of the bit values

Each bit has two possible values (0 or 1)

Two bits implies the total number of Permutations is

Permutations are: 0 0, 0 1, 1 0, 1 1

(ii) The maximum Integer is: 1 1(Base Two) = 1X2^1 + 1X2^0 = 3(Base Ten)
= 2^2 - 1
= 2^(Number of bits) - 1

Ex 2: If we have a system that stores Integers with a maximum of 16 bits
what is the maximum base 10 integer that can be stored ?

2^16 - 1 = 65 535