Week 2 Number Bases Continued Lecture Note

Notebook:	Computational Mathematics		
Created:	2020-04-21 2:48 PM	**Updated:**	2020-04-24 4:56 PM
Author:	SUKHJIT MANN		

Cornell Notes	**Topic:** Number Bases Continued	Course: BSc Computer Science Class: Computational Mathematics[Lecture] Date: April 23, 2020

Essential Question:

How do convert non-integer numbers and what are arithmetic operations possible with binary numbers?

Questions/Cues:

- How do we convert a decimal fractional number to binary?
- How do we perform addition in binary?
- How do we perform subtraction in binary?
- How do we perform multiplication in binary?
- How do we perform division in binary?

Notes

- Decimal (Fractional) to Binary = Separate whole integer part of number and convert to binary as normal. For fractional part, multiple the fractional decimal number by 2 writing the answer as a sum of a integer and fractional decimal remainder. eg. 0.375 x 2 = 0.75 = 0 + 0.75. Continue this step until you obtain an answer that a sum of a integer part plus 0 and the last multiplication should resulting in 0 and this is where you stop the process and bring together your answer. Your answer for the fractional decimal part is the integer parts in the conversion read top to bottom. Finally you join your whole integer part from the beginning with the fractional part you just converted to produce a final binary number value

Non-integer numbers: decimal to binary

Example: $17.375_{10} = 1\times10^{1}+7\times10^{0}+3\times10^{-1}+7\times10^{-2}+5\times10^{-3}$

$= 10 + 7 + 3/10 + 7/100 + 5/1000$

(Positional value of digits)

How to convert to binary? $17_{10}=10001_2$

Separate integer part 17 and fractional part 0.375

$0.375\times2=0.75=0+0.75$
$0.75\times2=1.5=1+0.5$
$0.5\times2=1.0=1+0$
$0 \rightarrow$ STOP

$0.375_{10}=0.011_2$

$\Rightarrow 17.375_{10}=10001.011_2$

Non-integer numbers: binary to decimal

Example: $1101.101_2 = 1\times2^{3}+1\times2^{2}+0\times2^{1}+1\times2^{0}+$
$+1\times2^{-1}+0\times2^{-2}+1\times2^{-3}$
$= 8+4+1+1/2+1/8=13.625_{10}$

(Positional value of digits)

in general $a_n\, a_{n-1}\, a_{n-2} \ldots a_0 .\, c_{-1}\, c_{-2} \ldots c_{-k}$ in base b

In decimal units corresponds to

$a_n\times b^{n}+a_{n-1}\times b^{n-1}+\ldots+a_0\times b^{0}+$
$+c_{-1}\times b^{-1}+c_{-2}\times b^{-2}+\ldots+c_{-k}\times b^{-k}$

- Binary addition = Similar to decimal addition, you add each digit in each column and carrying over "1" when adding together two ones or three ones. In the case of two ones, it is zero carry over 1 and in the case of three ones it is one carry over one.

```
  1          1         1 1
1 0 1+     1 0 1 +     1 0 1 +
1 1 1=     1 1 1 =     1 1 1 =
    0      0 0 0     1 1 0 0
```

$101_2+111_2=1100_2$ *in decimal* $5+7=12$

Operations with binary numbers

Other example: 1010011+1110110=?

1	1 1 1	1 1 1 1
1 0 1 0 0 1 1+	1 0 1 0 0 1 1+	1 0 1 0 0 1 1+
1 1 1 0 1 1 0=	1 1 1 0 1 1 0=	1 1 1 0 1 1 0=
0 1	0 0 1 0 0 1	1 1 0 0 1 0 0 1

1010011+1110110=11001001

In decimal 1010011=83 1110110=118 118+83=201

11001001=1+8+64+128=201

- Binary subtraction = Similar to decimal subtraction, subtraction in binary is quite straightforward; where 1-1 = 0, 1-0 = 1. In the case where subtraction is not immediately possible like 0 - 1, you must borrow one unit from the digit to the left of the zero and the zero becomes a 2 and if the left digit was initially a one it becomes a zero to complete the subtraction operation because it a base 2 number instead of base 10 like decimal. If you borrow a unit from the left digit neighboring a zero that is already promoted to a 2 it becomes a one.

    2
1 ~~1~~ 0-
1 0 1=
0 0 1

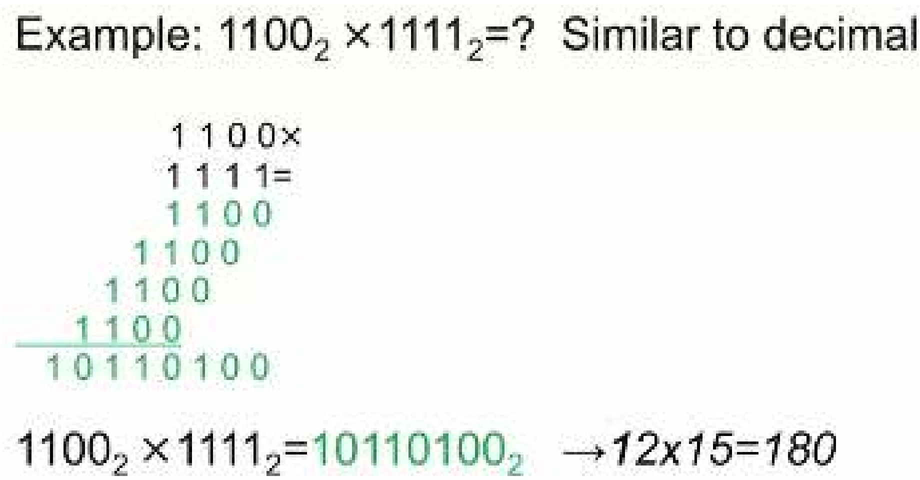
110_2-101_2=001_2 *in decimal 6-5=1*

Other example subtraction: 1110011-1010011

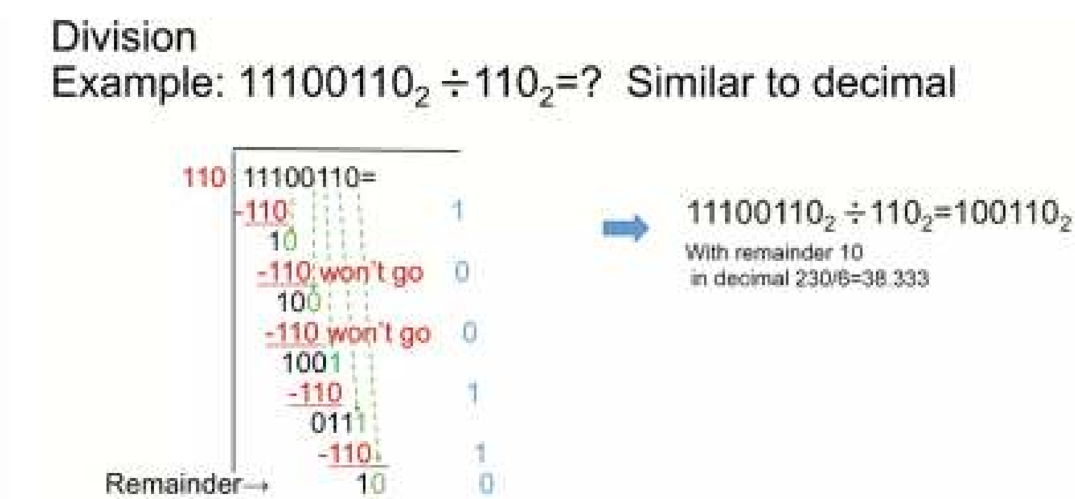
   2 ~~2~~1 2
1 ~~1~~ ~~1~~ 0 0 1 1-
1 0 1 0 1 1 0=
0 0 1 1 1 0 1

1110011=1+2+16+32+64=115
1010110=2+4+16+64=86
115-86=29
0011101=1+4+8+16=29

- Binary multiplication = Much like decimal multiplication, you line up numbers to multiplied into columns, then you multiplying by selecting the rightmost digit of the multiplier (bottom number) and multiplying that digit throughout the whole of the top number, noting the answer below the number, then you choose the second rightmost digit of the multiplier and multiply it throughout the top number but when noting the answer of the multiplication this second time you write it below the first answer and moved one digit to the left. Continue to multiply all the digits in the multiplier with the top number, noting the answer of the multiplication each time below previous answers, also moving each lower tier answer one digit to the left. In order to produce the final answer to the given problem, you must sum all the multiplication answers using binary addition and lining each of the answers from the previous multiplication step into columns; adding zeros in empty spaces and adding the resulting sum and writing the final answer in the end at the very bottom once the addition is complete.



- Binary division = Similar to decimal, you arrange the divisor on the outside and dividend on the inside of a long division brace. Then from the dividend you take a number that is larger than the divisor and from that number you subtract the divisor, noting in the quotient at the top how many times the divisor "go into" the dividend. After subtraction of the divisor from the dividend you get what is called the remainder and you bring the next digit from the dividend, here you ask the same question of how many times the divisor "goes into" this number and if the divisor is larger than the number then you add a 0 to the quotient at the top and bring the next digit from the dividend. If after this, the number is larger than the divisor you add a 1 to quotient and the divisor from number and this continue until you reach a remainder of 0 or until you have no more digits to bring from the dividend. Then your final answer is the quotient produced at the top and any remainder leftover from the subtraction.



Summary

In this week, we learned about how to convert fractional decimal numbers to binary and the different arithmetic operation that we can perform on binary numbers.