Recitation 1

CS 3843 Fall 2013

Representation: Hexadecimal

- Binary <-> hexadecimal:
  - 0001~1001 <-> 1~9
  - 1010 ~1111 <-> A~F
- Decimal-> hexadecimal:

  x = q*16 + r, e.g: 71 = 4*16 + 7, $71_{10} = 47_{16}$
- Hexadecimal -> Decimal :
  - E.g : $ABC_{16} = A*16^2 + B*16^1 + C*16^0$
    =2560+176+12

Conversion of base R to decimal

- $(a_n \ldots a_1 a_0)_R = (?)_{10}$
- $? = a_0 * R^0 + a_1 * R^1 + \ldots\ldots a_n * R^n$

- Generally
- $(a_n \ldots a_1 a_0 . a_{-1} a_{-2} \ldots a_{-m})_R$

$= a_0 * R^0 + a_1 * R^1 + \ldots\ldots a_n * R^n +$

$a_{-1} * R^{-1} + a_{-2} * R^{-2} + \ldots\ldots a_{-m} * R^{-m}$

Conversion of decimal to base R

- Division method(Integer part)
  - Divide the decimal by R and then the quotient each time generated until the quotient is zero.
  - record the remainders a_0 , a_1 ,… a_n
  - reverse the remainders
  - E.g: 53_{10} to base 4 number
    - 53/4 =13*4 + 1 -> a_0 = 1
    - 13/4 = 3*4 + 1 -> a_1 = 1
    - 3/4 = 0*4 + 3 -> a_2 = 3 so $53_4 = 311_4$

- Multiply method(fraction part)
- Multiply the number by base R and get the integer part of the result as a_0
- Repeat using the fractional part of each production to multiply R : a_1 , a_2… a_n
- Result : (. a_0 , a_1… a_n), sometimes it is infinite
- E.g: $0.25_{10} = (.01)_2$

Data sizes

C	32bit	64bit
Char	1	1
Short int	2	2
Int	4 (1 word)	4 (1 word)
Long int	4 (1 word)	8 (double word)
Long long int	8 (double word)	8 (double word)
Char *	4 (1 word)	8 (double word)
float	4 (1 word)	4 (1 word)
double	8 (double word)	8 (double word)