CSC 320
Computer Organization

Number Systems and Codes

Dr. Sanaa Sharafeddine

Positional Number Systems

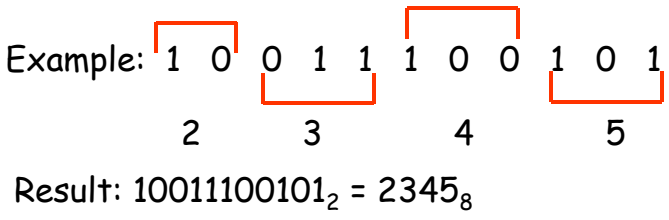
- Number represented by a string of digits
  - Each digit has an associated weight
  - Number = weighted sum of digits
  - Each weight is a power (-ve or +ve) of the radix
  - Radix r (also called base) maybe any integer ≥ 2
- General form of a number is:
  - $d_{p-1}d_{p-2} \ldots d_1 d_0 . d_{-1}d_{-2} \ldots d_{-n}$
  - d_{p-1} = Most significant digit
  - d_{-n} = Least significant digit
- The value of the number is: $D = \sum_{i=-n}^{p-1} d_i r^i$

Positional Number Systems

- Radix 10: Decimal numbers
  - Used in everyday calculations
- Radix 2: Binary numbers
  - Used to represent numbers in a digital system
- Octal (Radix 8) and Hexadecimal (Radix 16) numbers
  - Used to represent multibit numbers
  - Octal: Digits from {0,1,2,3,4,5,6,7}
  - Hexadecimal: Digits from {0,…,9,A,B,C,D,E,F}

Positional Number System Conversions

- Binary to Octal: Bits are grouped in "Threes" from right to left, and the binary value of every group represents the corresponding Octal Digit

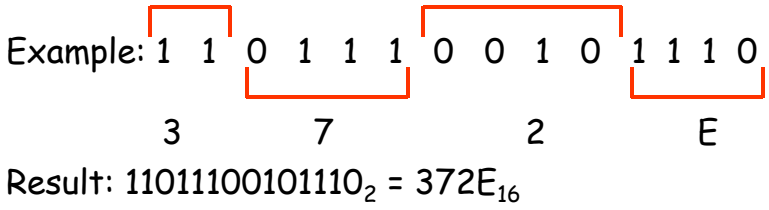


Result: $10011100101_2 = 2345_8$

- Example: Convert 1001111 from Binary to Octal?

Positional Number System Conversions

- Binary to Hexadecimal: Bits are grouped in "Fours" from right to left, and the binary value of every group represents the corresponding Hexadecimal Digit



Result: $11011100101110_2 = 372E_{16}$

- Example: Convert 1001111 from Binary to Hexadecimal?
- Example: Convert 10.1011 to Octal and Hexadecimal?
- Example: Convert 9F.4 from Hexadecimal to Binary?

Positional Number System Conversions

- Conversion to Decimal (radix-10)
  - To convert a number D to decimal, we convert each digit to radix-10 equivalent and expand the weighted-sum formula using radix-10 arithmetic
  - EX1: $1CE8.1_{16} = 1x16^3 + 12x16^2 + 14x16 + 8x1 + 1x16^{-1}$

    $= 7400.0625_{10}$
  - EX2: $F1AC_{16} = ?$

Positional Number System Conversions

- Decimal to Binary: It is done by successive divisions by r until a zero quotient is obtained. The remainders taken in reverse order constitute the binary representation

  Example: Convert 46 from Decimal to Binary

  46 / 2 = 23 and Remainder = 0
  23 / 2 = 11 and Remainder = 1
  11 / 2 = 5 and Remainder = 1
  5 / 2 = 2 and Remainder = 1
  2 / 2 = 1 and Remainder = 0
  1 / 2 = 0 and Remainder = 1

  Result: $46_{10} = 101110_2$

Positional Number System Conversions

- Decimal fraction to Binary: It is done by successive multiplications taking out each time the integer part of the result

  Example: Convert 0.3125 to binary

  0.3125 x 2 = 0.625 ➔ 0
  0.625 x 2 = 1.25 ➔ 1
  0.25 x 2 = 0.5 ➔ 0
  0.5 x 2 = 1 ➔ 1

  Result: $0.3125_{10} = 0.0101_2$

Addition and Subtraction: Binary Numbers

190	10111110
+141	10001101
331	101001011

229	11100101
- 46	00101110
183	10110111

- Subtraction can be used to compare numbers

Quiz Question

- Determine radix x:

  $(12)_x = 18_{10}$

Codes

- Binary numbers are used in internal computations of digital systems, but
  - Some digital devices process/display decimal numbers
  - Some digital devices process events, actions, conditions, etc.
- Solution: Use codes (mappings)
  - Map decimal numbers to bits (binary sequences)
  - Map events, actions, and conditions to bits
  - Codes are composed of codewords
- E.g., how many bits are required to represent the ten decimal digits?

Binary Codes for Decimal Numbers

- There are many possibilities to build binary codes for decimal digits. How many?

Table 2-9 Decimal codes.

Decimal digit	*BCD (8421)*	*2421*	*Excess-3*	*Biquinary*	*1-out-of-10*
0	0000	0000	0011	0100001	1000000000
1	0001	0001	0100	0100010	0100000000
2	0010	0010	0101	0100100	0010000000
3	0011	0011	0110	0101000	0001000000
4	0100	0100	0111	0110000	0000100000
5	0101	1011	1000	1000001	0000010000
6	0110	1100	1001	1000010	0000001000
7	0111	1101	1010	1000100	0000000100
8	1000	1110	1011	1001000	0000000010
9	1001	1111	1100	1010000	0000000001
			Unused code words		
	1010	0101	0000	0000000	0000000000
	1011	0110	0001	0000001	0000000011
	1100	0111	0010	0000010	0000000101
	1101	1000	1101	0000011	0000000110
	1110	1001	1110	0000101	0000000111
	1111	1010	1111	. . .	. . .