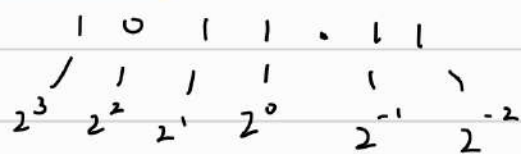


$$(1 \times 10^2) + (2 \times 10^1) + (3 \times 10^0) + (5 \times 10^{-1}) + (2 \times 10^{-2})$$

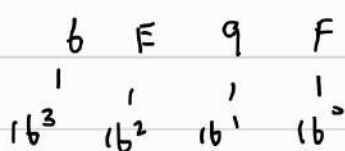
digits

Base 2 - 0 & 1



$$2^3 + 0 + 2^1 + 2^0 + 2^{-1} + 2^{-2}$$

Base 16 - 0 to 9, A to F



Conversion

Step 1: Transform number to base 10 (if not in base 10).

Step 2: list out required positional weights

eg: if base 2 \rightarrow 2^8 2^7 2^6 2^5 2^4 ... 2^0

for 356.

\downarrow \downarrow \downarrow \downarrow

256_{10} 128_{10} 64_{10} 32_{10}

Step 3: Fill in digits & subtract from number.

Final digits will be the converted number.

digits must be valid in given number system.
i.e cannot have ≥ 2 for base 2 weights.

Binary - Hex rls

if in binary: Group 4 bits & convert to hex equivalent.

if in hex: convert hex digit in 4 digit binary equivalent.

Binary - octal rls

if in binary: Group 3 bits & convert to octal equivalent.

if in octal: convert octal digit in 3 digit binary equivalent

Hex \leftrightarrow octal

if hex: convert to binary, group in 3s, convert to octal digits.

if octal: convert to binary, group in 4s, convert to hex digits

* Add '0' as required.

Diminished radix complement ($r-1$ complement)

$$(r-1) \text{ complement} = r^n - r^{-m} - N$$

eg: $N = 134.456_{10}$

$n = 3$ $m = 3$

n digits m fractional digits

given base.

$$r = 10 \text{ (base 10)}, (r-1) = 9$$

$$10^3 - 10^{-3} - 134.456 = 865.543$$

Find the 1's complement, $r = 2$ - if given N is binary, must convert to decimal for $r^n - r^{-m} - N$. then convert back

No. of integer digits.

r's complement

$$r's \text{ complement} = r^n - N$$

\uparrow given base

eg: $N = 0.3244_{10}$, $r \text{ complement} = 10^0 - 0.3244 = 0.6756$

$N = 23.12_{10}$, $\text{complement} = 10^2 - 23.12 = 76.88$

} 10's complement

$N = 1011_2$, $\text{complement} = 2^4 - 1011_2$

$= 16 - 11_{10}$

$= 5_{10} \equiv 0101_2$

} 2's complement

Purpose of complements

- in digital circuits, its faster to subtract by adding complements than by performing the subtraction.
- 2 type of complements.
 - 1) R's complement (10's & 2's complement)
 - 2) (R-1)'s complement (9's & 1's complement)

R's complement

- 10's & 2's
- addition per normal
- $a - b = a + (-b)$

$$N_1 - N_2 = N_1 + \bar{N}_2$$

$$= N_1 + (r^n - N_2)$$

$$= (N_1 - N_2) + r^n$$

if $N_1 - N_2 > 0$: ignore carry.

if $N_1 - N_2 < 0$: ans should be in complement form

(R-1)'s complement

- 9's & 1's
- same
- same.

$$N_1 - N_2 = N_1 + (r^n - r^{-m} - N_2)$$

$$= (N_1 - N_2) + (r^n - r^{-m})$$

if $N_1 - N_2 > 0$: +1 to LSB

if $N_1 - N_2 < 0$: ans in complement form

$$N_1 - N_2 = N_1 + \bar{N}_2$$

Octal subtraction

$N_1 = 7526$, $N_2 = 3142$, (8-1)'s complement

step 1: Find $\bar{N}_2 = 7777 - 3142$

$$= 4635$$

$6_8 + 5_8 = (1 \times 8)_8 + 3_8$

2: $N_1 + \bar{N}_2 =$

$$\begin{array}{r} 7526 \\ 4635 \\ \hline 4363 \end{array}$$

3: +1 to LSB = 4364

Hexa decimal subtraction, (r-1)'s complement

$N_1 = ABED$, $N_2 = 1FAD$, $\bar{N}_2 =$

$$\begin{array}{r} FFFF \\ - 1FAD \\ \hline E052 \end{array}$$

1) $N_1 + \bar{N}_2 =$

$$\begin{array}{r} ABED \\ + E052 \\ \hline 18C3F \end{array}$$

2) Add 1 to LSB $\rightarrow 8C40$

Data Representation

- unsigned is for the numbers only. (i.e the no. no need SMF)
- signed magnitude form (SMF) : MSB : 0 \rightarrow +ve 1 \rightarrow -ve

- signed 1's complement: MSB: $1 \rightarrow -ve$, Magnitude bits '0's \rightarrow '1's
- signed 2's complement = 1's c + 1 to LSB. '1's \rightarrow '0's

eg: $+10 = 01010$ (unsigned).

$-10 = 1(1010)$ (SMF)

$-10 = 1(0101)$ (1'sc): flip magnitude bits.

$-10 = 1(0110)$ (2'sc): 1'sc + 1 to LSB.

2's complement subtraction (2's complement)

$$N_1 = 75_{10} = 1001011_2 = 0 \mid \overset{\text{SMF}}{1001011}$$

$$N_2 = 25_{10} = 0011001_2 \quad 1 \mid 0011001$$

$$1'sc \text{ for } 25_{10} = 1 \mid 1100110$$

$$2'sc \text{ for } 25_{10} = 1'sc + 1 \text{ to LSB} = 1 \mid 1100111$$

$$N_1 + \bar{N}_2 = \begin{array}{r|ccccccc} & 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 1 & 1 & 1 \\ \hline 1 & 0 & & 0 & 1 & 1 & 0 & 0 \end{array} \quad \begin{array}{l} 32 \\ 16 \\ 2 \end{array} \quad 2 = 50_{10}$$

↑
ignore carry since $N_1 - N_2 \geq 0$.

Weighted code

- each binary digit assigned a weight

$$\text{eg: } \begin{array}{cccc} 1 & 0 & 1 & 0 \\ 2^3 & 2^2 & 2^1 & 2^0 \end{array}$$

↑
8 4 2 1 code. / positional weights.

Also have 2421, 5211, 5421, 4221, 3321 etc.

$$\text{eg: } 8_{10} \text{ for } 5211$$

$$1110$$

$$5 + 2 + 1 = 8_{10}$$

Non-weighted code

- value not dependant on positional weights

eg: excess-3, gray codes.

BCD (0-9)₁₀

- representing each decimal digit 0 to 9 using binary equivalent

Excess-3 (3-12)₁₀

- Add 3₁₀ to BCD, represent in binary (for each BCD)

$$\text{eg: } \begin{array}{ccc} \text{BCD} & \text{excess-3} & \\ 0000 & 0011 & \leftarrow 3_{10} \end{array}$$

$$\text{eg: } 9_{10} = 1001 \quad 0_{10} = 0000$$