Subtraction With Complement

The subtraction of two n-digit unsigned numbers $M - N$ in base r can be done as follows:

1- Add the minuend, M, to the r's complement of the subtrahend, N.

2- If $M \geq N$, the sum will produce an end carry, r^n, which can be discarded; what is left is the result $M-N$

3- If $M < N$, the sum does not produce an end carry and is equal to $r^n - (N-M)$, which is the r's complement of $(N-M)$. To obtain the answer in a familiar form, take the r's complement of the sum and place a negative sign in front.

Ex: Using 10's complement, subtract 72532 - 3250

Sol^n

$M = 72532$

10's Comp. of $N = +96750$

Sum = 169282

Discard end carry $10^5 = -100000$

Answer = 69282

The occurence of the end carry signifies that $M \geq N$ and that the result is positive.

Ex: Using 10's complement, subtract 3250 - 72532

$M = 03250$

10's complement of $N = +27468$

Sum = 30718

There is no end carry

Therefore, the answer is -(10's complement of 30718)

$= -69282$

Binary Addition

To compute the sum of two binary numbers, say

```
  1 1
  0 1 1 0     6
  0 1 1 1    +7
-----------  ---
  1 1 0 1    13
```

$0+0=0$
$0+1=1$
$1+0=1$
$1+1=10$ (2, or a sum of 0 and a carry of 1 to the next bit)

We add one digit at a time, producing a sum and a carry to the next bit.

Binary Subtraction

Subtraction is generally accomplished by first taking the two's complement of the second operand, and then adding. Thus $(a-b)$ is computed as $a+(-b)$.

Ex: Given the two binary numbers $X=1010100_2$ and $Y=1000011$, perform the subtraction (a) $X-Y$ and (b) $Y-X$ using 2's complements.

$X=84$
$Y=67$
17

soln a)

$X = 1010100$
2's complement of $Y = +0111101$
sum $= 10010001$
Discard end carry $2^7 = -10000000$
Answer: $X-Y = 0010001$

$Y = 1000011$
1's comp $Y = 0111100$
2's comp $Y = 0111101$

b)

$Y = 1000011$

2's complement of $X = +0101100$

sum $= 1101111$

$X = 1010100$

1's comp of $X = 0101011$

$+1$

2's comp. of $X = 0101100$

There is no end carry.

Therefore, the answer is $Y - X = -(2\text{'s complement of } 1101111)$

$= -0001000$

Signed Binary Numbers

Positive numbers can be be represented as unsigned numbers. It is customary to represent the sign with a bit placed in the leftmost position of the number. The convention is to make the sign bit 0 for positive and 1 for negative.

If the binary number is signed, then the leftmost bit represents the sign and the rest of the bits represent the number. If the binary number is assumed to be unsigned, then the leftmost bit is the most significant bit of the number.

Ex: 01001 can be considered as 9 (unsigned binary) or as +9 (signed binary) because the leftmost bit is 0.

Ex: 11001 represent the binary equivalent of 25 when considered as an unsigned number or as -9 when considered as a signed number, because the 1 in the leftmost position designates a negative and the other four bits represents 9.

The following table shows all possible 4-bit signed binary numbers in three representations, signed-magnitude, signed 1's complement, and signed 2's complement.

Decimal	signed 2's complement	signed 1's complement	signed magnitude
+7	0111	0111	0111
+6	0110	0110	0110
+5	0101	0101	0101
+4	0100	0100	0100
+3	0011	0011	0011
+2	0010	0010	0010
+1	0001	0001	0001
+0	0000	0000	0000
−0	—	1111	1000 ← ?
−1	1111	1110	1001
−2	1110	1101	1010
−3	1101	1100	1011
−4	1100	1011	1100
−5	1011	1010	1101
−6	1010	1001	1110
−7	1001	1000	1111
−8	1000	—	—

Ex: Repeat Ex* using 1's complement.

(a) $X - Y = 1010100 - 1000011$

X = 1010100

1's comp of Y = +0111100

sum = 10010000

End-around carry = + 1

Answer: X − Y = 0010001

(b) $Y - X = 1000011 - 1010100$

$Y = 1000011$

1's complement of $X = +0101011$

sum $= 1101110$

There is no end carry.

Therefore, the answer is $Y - X = -$ (1's comp. of 1101110) $=$

-0001001

Binary Codes

An n-bit binary code is a group of n bits that assume up to 2^n distinct combinations of 1's and 0's, with each combination representing one element ~~assigned one~~ of the set that is being coded.

1) BCD Code (Binary Coded Decimal)

The code most commonly used for the decimal digits. It uses the first 10 binary numbers to represent the 10 decimal digits, then the number 12, for example, would be stored as 0001 0010.

Each decimal digit is represented by 4 bits, and thus a 2-digit decimal number requires 8 bits.

Ex: write the equivalent BCD and Binary Code for $(185)_{10}$

$(185)_{10} = (0001\ 1000\ 0101)_{BCD} = (10111001)_2$

Decimal symbol	BCD digit
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001

We have already discussed the simplest code, using the first 10 binary numbers to represent the 10 digits. The remaining 4-bit binary numbers (1010, 1011, 1100, 1101, 1110, 1111) are unused. This code is sometimes referred to as the 8421 code, since those are the weights of the bits. Each decimal digit is represented by $8 \times a_3 + 4 \times a_2 + 2 \times a_1 + 1 \times a_0$

Other Decimal Codes

Binary codes for decimal digits require a minimum of four Bits per digit.

The BCD and the 2421 codes are examples of weighted codes.

The 2421 and the excess-3 codes are examples of self-complementing codes. such codes have the property that the 9's complement of a decimal number is obtained directly by changing 1's to 0's and 0's to 1's in the code.

Ex: Decimal 395 is represented in the excess-3 code as 0110 1100 1000. The 9's complement 604 is represented as 1001 0011 0111

Q: Is BCD code self-complementing or not ?

Decimal digit	2421	Excess-3
0	0000	0011
1	0001	0100
2	0010	0101
3	0011	0110
4	0100	0111
5	1011	1000
6	1100	1001
7	1101	1010
8	1110	1011
9	1111	1100

Gray Code

The advantage of the Gray code over the straight binary number sequence is that only one bit in the code group changes when going from one number to the next.

Ex: In going from 7 to 8, the Gray code changes from 0100 to 1100. Only the first bit changes from 0 to 1; the other three bits remain the same. When comparing this with binary numbers, the change from 7 to 8 will be from 0111 to 1000.

Applications of Gray code are in

1- Represent the digital data when it is converted from analog data.

2- Eliminates the error or ambiguity during the transition from one number to the next.

Gray Code	Decimal equi.
0000	0
0001	1
0011	2
0010	3
0110	4
0111	5
0101	6
0100	7

Gray Code	Decimal equi.
1100	8
1101	9
1111	10
1110	11
1010	12
1011	13
1001	14
1000	15

ASCII Character Code (American Standard Code for Information Inter change)

It is the standard binary code for the alphanumeric characters & It uses seven bits to code 128 characters. The ASCII code contains 94 graphic characters that can be printed and 34 non printing characters used for various control functions. ASCII is a 7-bit, but most computers manipulate an 8-bit quantity as a single unit called a byte. Therefore, ASCII characters most often are stored one per Byte. The extra bit is sometimes used for other purposes.

Error-Detecting Code

To detect errors in data communication and processing, an eight bit is sometimes added to the ASCII character to indicate its parity. A parity bit is an extra bit ~~including~~ with a message to make the total number of 1's either even or odd.

	With even parity	With Odd parity
ASCII A = 1000001	01000001	11000001
ASCII T = 1010100	11010100	01010100

In each case, we insert an extra bit in the left most position of the code to produce an even number of 1's in the character for even parity or an odd number of 1's in the character for Odd parity.