IN 302

TD02

Représentation d'entiers signés

1) 36_{10} → $36 = 18 \times 2 + 0$
$18 = 9 \times 2 + 0$
$9 = 4 \times 2 + 1$
$4 = 2 \times 2 + 0$
$2 = 1 \times 2 + 0$
$1 = 0 \times 2 + 1$
→ $(00100100)_2$ dans les 3 écritures

[écrire en 8 bits avec 1er bit pour le signe]

- -113_{10} → $113 = 64 \times 1 + 49$
$64 = 30 \times 2 + 4$
$30 = 15 \times 2 + 0$
$15 = 7 \times 2 + 1$
$7 = 3 \times 2 + 1$
$3 = 1 \times 2 + 1$
$1 = 0 \times 2 + 1$
$(1110001)_2$
→ $(11110001)_2$ en signe+abs, $(10001110)_2$ en C1

en C2 :
1 1 1 1 1 1 1
1 0 0 0 1 1 1 0
1 1 1 0 0 1 0
0 0 0 0 0 1 0 1
→ $(10000101)_2$

- 45_{10} → $45 = 22 \times 2 + 1$
$22 = 11 \times 2 + 0$
$11 = 5 \times 2 + 1$
$5 = 2 \times 2 + 1$
$2 = 1 \times 2 + 0$
$1 = 0 \times 2 + 1$
$(101101)_2$
→ $(00101101)_2$ en signe+abs, $(11010010)_2$ en C1

en C2 :
1 1 1 1 1 1 1 2
1 0 0 0 0 0 0 0
1 0 1 1 0 1
1 0 1 0 0 1 1
→ $(11010011)_2$

$x = 2^7 - C_2$

h) $(11111111)_2$ → $2^7 - (1111111)_2$
→ 1 0 0 0 0 0 0 0
1 1 1 1 1 1 1
0 0 0 0 0 0 1

signe négatif → $-(1)_2$
en C2 → -1_{10}

$(11111111)_2$ → 0_{10} et signe néga. → -128_{10}
en C1 et $(11111111)_2 = 128_{10}$ car abs

$(10100+1000)_2 \rightarrow (1000+1000)_2$ dans les 3 modes
$\rightarrow (1001000)_2 = 2^3 + 2^6 = 72_{10}$

$(10000111)_2 \rightarrow 2^{\text{ème}} - (111)_2$ signe néga. $\rightarrow -(11+1000+1)_2$
$\rightarrow$ 1 0 0 0 0 0 0 0 en $C_2 \rightarrow -113_{10}$
− 1 1 1 1
1 1 1 0 0 0 1

$(1000+111+1)_2 \rightarrow (11110000)_2 \rightarrow$ signe néga. et $(11+10000)_2$
$\rightarrow -112_{10}$ en C1
signe neg et $(1111)_2 = -15_{10}$ val. abs.

3) $x = 11111111$ / 100+10+01 (8 bits)
$y = 00000000$ / 0+100+10 (6 bits)
$z = 11111111$ / 1+0+01 (6 bits)
Pour x 8 bits, y et z 6 bits

4) $-2^{n-1} \leqslant x \leqslant 2^{n-1} - 1$

5) En signe + val abs, il suffit par 0 en complément à 2 aussi en complément à 1, le bit de signe et le dernier signe doivent être identiques

→ Additions / soustractions d'autres signes

1) $120_{10} - 45_{10} = 120_{10} + (-45)_{10} =$

$120 \rightarrow 120 = 60 \times 2 + 0 = (1111000)_2$
$60 = 30 \times 2 + 0$
$30 = 15 \times 2 + 0$
$15 = 7 \times 2 + 1$
$7 = 3 \times 2 + 1$
$3 = 1 \times 2 + 1$

$45 \rightarrow 45 = 22 \times 2 + 1$
$22 = 11 \times 2 + 0$
$11 = 5 \times 2 + 1$
$5 = 2 \times 2 + 1$
$2 = 1 \times 2 + 0$
$1 = 0 \times 2 + 1$
$= (101101)_2$

→ en C2 : $(120)_{10} = (01111000)_2$

en C2 : $(-45)_{10} = 11010011$

$-00101101 \rightarrow (11010011)_2$

$\rightarrow (01111000)_2 + (11010011)_2 = (01001011)_2$

2) en C2 : $(00100100)_2 + (01000000)_2$

$= (01100100)_2$

[2] $(00111001)_2 + (11001110)_2$

$= (11100111)_2$

[3] $(11110110)_2 + (10100110)_2$

$= (11001100)_2$

[4] $(00101101)_2 + (01101111)_2$

$= (01001100)_2$ overflow

[5] $(10000001)_2 + (11000000)_2$

$= (10100000)_2$ overflow

3) $125_{10} = (1111101)_2$ → 9 bits (sans overflow) pour le signe

$5_{10} = (101)_2$

4) $CA2(x) = 2^n - x$

$CA2(x) = \overline{x} + 1$

Soit $x = x_{n-1} 2^{n-1} + \dots + x_0 2^0$, poser

$2^n - x = 2^n - (x_{n-1} 2^{n-1} + \dots + x_0 2^0)$

$= (2^{n-1} + \dots + 2^0 + 1) - (x_{n-1} 2^{n-1} + \dots + x_0 2^0)$

$= (1 - x_{n-1}) 2^{n-1} + \dots + (1 - x_0) 2^0 + 1$

$= \overline{x_{n-1}} 2^{n-1} + \dots + \overline{x_0} 2^0 + 1$

$= \overline{x} + 1$

→ Un peu de programmation :

1)
```
char opt_zero(uint32_t M)
{
  char count = 0;
  while (M)
  { count += M and 1;
    M = M >> 1;
  }
  return count;
}
```

2) $M'' = 00000000000000001000000100010000_2$ and
$00000000000000001000000100101111_2$
$= 00000000000000001000000100000_2$
$M''' = 00000000000000001000000100000000_2$
à chaque application de la formule on met à zéro le bit le plus à droite

3)
```
char opt_zero_loop(uint32_t M)
{ char count = 0;
  while (M)
  { M = M and (M-1);
    count += 1;
  }
  return count;
}
```

4) Si on précalcule dans un tableau le nombre de bits à 1 dans des mots de 8 bits, une possibilité bien plus rapide consiste à découper l'entier en 4 mots de 8 bits et utiliser la table pour obtenir le résultat. Bien entendu, on pourrait faire la même chose directement avec une table de 32 bits, mais la table serait alors prohibitive.