

## Opérations arithmétiques binaires

### 1. Opérations sur les entiers naturels

1.1 Coder en binaire sur 8 bits les entiers naturels des opérations d'addition et faire leurs additions sur 8 bits :  $28 + 30$  ;  $255 + 1$

1.2 Coder en hexadécimal sur 8 bits les opérations d'addition et faire leurs additions sur 8 bits :  $254 + 1$  ;  $128 + 128$ ,  $133 + 163$

1.3 Pour chacune de ces opérations, on donnera :

- Le résultat sur 8 bits
- L'état de l'indicateur C positionné à 1, sa signification et l'interprétation du résultat de l'opération.

### 2. Opérations sur les entiers

2.1 Donner en CC2 8 bits les nombres 61 et -61 et faire leur somme sur 8 bits.

2.2 Effectuer en CC2 sur 8 bits les opérations suivantes :  $1 + (-2)$  ;  $51 + 127$  ;  $(-3) + (-127)$  ;  $(-127) + 127$  ;  $(-63) + (-63)$ .

Pour chaque opération on donnera le résultat sur 8 bits et l'état des indicateurs C et V, la signification de chacun d'eux et on expliquera comment on les utilise pour interpréter le résultat de l'opération.

2.3. Coder sur 16 bits en CC2 l'opération  $(-127) + 127$ . Donner son résultat sur 16 bits et les états des différents indicateurs C et V. Que peut-on conclure ?

### 3. Opérations sur les entiers codés sur 32 bits

Effectuer en hexadécimal en CC2 sur 32 bits les opérations suivantes :  $(-1) + (-2)$  ;

$(-3) + (-127)$  ;  $(-127) + (127)$  ;  $(-63) + (-63)$ , ainsi que l'addition du plus grand nombre entier non signé plus 1.

Pour chaque opération on donnera le résultat sur 32 bits et l'état des indicateurs C et V, la signification de chacun d'eux et on expliquera comment on les utilise pour interpréter le résultat de l'opération

### 4. Opérations de multiplication/décalage

3.1 Effectuer l'opération :  $1001111 \times 110100$

3.2 Effectuer pour les nombres 0x01F1 et 0xFF85 les opérations suivantes :

1. Un décalage à gauche de 2 positions
2. Un décalage arithmétique à droite de 2 positions
3. Un décalage logique à droite de 2 positions
4. Donner pour chacune de ces opérations la valeur du résultat