

C9 Représentation des entiers négatifs

Complément à 2

Pour représenter un entier négatif en machine, on utilise la méthode du **complément à 2** :

Avantages de cette représentation

C9 Représentation des entiers négatifs

Complément à 2

Pour représenter un entier négatif en machine, on utilise la méthode du **complément à 2** :

- 1 on commence par écrire la représentation binaire de la valeur absolue de ce nombre

Avantages de cette représentation

C9 Représentation des entiers négatifs

Complément à 2

Pour représenter un entier négatif en machine, on utilise la méthode du **complément à 2** :

- 1 on commence par écrire la représentation binaire de la valeur absolue de ce nombre
- 2 on inverse tous les bits de cette représentation

Avantages de cette représentation

C9 Représentation des entiers négatifs

Complément à 2

Pour représenter un entier négatif en machine, on utilise la méthode du **complément à 2** :

- 1 on commence par écrire la représentation binaire de la valeur absolue de ce nombre
- 2 on inverse tous les bits de cette représentation
- 3 on ajoute 1, sans tenir compte de la dernière retenue éventuelle

Avantages de cette représentation

C9 Représentation des entiers négatifs

Complément à 2

Pour représenter un entier négatif en machine, on utilise la méthode du **complément à 2** :

- 1 on commence par écrire la représentation binaire de la valeur absolue de ce nombre
- 2 on inverse tous les bits de cette représentation
- 3 on ajoute 1, sans tenir compte de la dernière retenue éventuelle

Avantages de cette représentation

- l'algorithme d'addition classique des nombres en base 2 fonctionne

C9 Représentation des entiers négatifs

Complément à 2

Pour représenter un entier négatif en machine, on utilise la méthode du **complément à 2** :

- 1 on commence par écrire la représentation binaire de la valeur absolue de ce nombre
- 2 on inverse tous les bits de cette représentation
- 3 on ajoute 1, sans tenir compte de la dernière retenue éventuelle

Avantages de cette représentation

- l'algorithme d'addition classique des nombres en base 2 fonctionne
- tous les nombres sont représentés de façon unique (pas de double représentation pour zéro).

C9 Représentation des entiers négatifs

Complément à 2

Pour représenter un entier négatif en machine, on utilise la méthode du **complément à 2** :

- 1 on commence par écrire la représentation binaire de la valeur absolue de ce nombre
- 2 on inverse tous les bits de cette représentation
- 3 on ajoute 1, sans tenir compte de la dernière retenue éventuelle

Avantages de cette représentation

- l'algorithme d'addition classique des nombres en base 2 fonctionne
- tous les nombres sont représentés de façon unique (pas de double représentation pour zéro).
- Le bit le plus à gauche est le bit de signe, il vaut 1 lorsque le nombre est négatif, 0 sinon.

C9 Représentation des entiers négatifs

Exemples

- Sur 8 bits, donner l'écriture en complément à 2 de -12
- Même question pour 75

Exemples

- Sur 8 bits, donner l'écriture en complément à 2 de -12
 1. On écrit $12 = (8 + 4)$ en binaire sur 8 bits :
- Même question pour 75

C9 Représentation des entiers négatifs

Exemples

- Sur 8 bits, donner l'écriture en complément à 2 de -12
 1. On écrit $12 = (8 + 4)$ en binaire sur 8 bits : 00001100
- Même question pour 75

C9 Représentation des entiers négatifs

Exemples

- Sur 8 bits, donner l'écriture en complément à 2 de -12
 1. On écrit $12 = (8 + 4)$ en binaire sur 8 bits : 00001100
 2. On inverse tous les bits :
- Même question pour 75

C9 Représentation des entiers négatifs

Exemples

- Sur 8 bits, donner l'écriture en complément à 2 de -12
 1. On écrit $12 = (8 + 4)$ en binaire sur 8 bits : 00001100
 2. On inverse tous les bits : 11110011
- Même question pour 75

C9 Représentation des entiers négatifs

Exemples

- Sur 8 bits, donner l'écriture en complément à 2 de -12
 1. On écrit $12 = (8 + 4)$ en binaire sur 8 bits : 00001100
 2. On inverse tous les bits : 11110011
 3. On ajoute 1 :
- Même question pour 75

C9 Représentation des entiers négatifs

Exemples

- Sur 8 bits, donner l'écriture en complément à 2 de -12
 1. On écrit $12 = (8 + 4)$ en binaire sur 8 bits : 00001100
 2. On inverse tous les bits : 11110011
 3. On ajoute 1 : 11110100
- Même question pour 75
 1. On écrit $75 = 64 + 8 + 2 + 1$ en binaire sur 8 bits :

C9 Représentation des entiers négatifs

Exemples

- Sur 8 bits, donner l'écriture en complément à 2 de -12
 1. On écrit $12 = (8 + 4)$ en binaire sur 8 bits : 00001100
 2. On inverse tous les bits : 11110011
 3. On ajoute 1 : 11110100
- Même question pour 75
 1. On écrit $75 = 64 + 8 + 2 + 1$ en binaire sur 8 bits : 01001011

C9 Représentation des entiers négatifs

Exemples

- Sur 8 bits, donner l'écriture en complément à 2 de -12
 1. On écrit $12 = (8 + 4)$ en binaire sur 8 bits : 00001100
 2. On inverse tous les bits : 11110011
 3. On ajoute 1 : 11110100
- Même question pour 75
 1. On écrit $75 = 64 + 8 + 2 + 1$ en binaire sur 8 bits : 01001011
 2. On inverse tous les bits :

C9 Représentation des entiers négatifs

Exemples

- Sur 8 bits, donner l'écriture en complément à 2 de -12
 1. On écrit $12 = (8 + 4)$ en binaire sur 8 bits : 00001100
 2. On inverse tous les bits : 11110011
 3. On ajoute 1 : 11110100
- Même question pour 75
 1. On écrit $75 = 64 + 8 + 2 + 1$ en binaire sur 8 bits : 01001011
 2. On inverse tous les bits : 10110100

C9 Représentation des entiers négatifs

Exemples

- Sur 8 bits, donner l'écriture en complément à 2 de -12
 1. On écrit $12 = (8 + 4)$ en binaire sur 8 bits : 00001100
 2. On inverse tous les bits : 11110011
 3. On ajoute 1 : 11110100
- Même question pour 75
 1. On écrit $75 = 64 + 8 + 2 + 1$ en binaire sur 8 bits : 01001011
 2. On inverse tous les bits : 10110100
 3. On ajoute 1 :

C9 Représentation des entiers négatifs

Exemples

- Sur 8 bits, donner l'écriture en complément à 2 de -12
 1. On écrit $12 = (8 + 4)$ en binaire sur 8 bits : 00001100
 2. On inverse tous les bits : 11110011
 3. On ajoute 1 : 11110100
- Même question pour 75
 1. On écrit $75 = 64 + 8 + 2 + 1$ en binaire sur 8 bits : 01001011
 2. On inverse tous les bits : 10110100
 3. On ajoute 1 : 10110101