IUT de Montpellier, Dép. Informatique
Intro. Archi V. Poupet

Examen (durée 1h30)

- Le sujet comporte 2 pages
- Les documents ne sont pas autorisés
- Les calculatrices (et autres appareils) ne sont pas autorisés. Vous pouvez donner vos réponses en donnant une expression non simplifiée du résultat (par exemple « $14 \times (2^{12} - 128)$ »)

Rappels sur la norme IEEE 754 en simple précision

- 1 bit de signe (S)
- 8 bits d'exposant (E)
- 23 bits de mantisse (M)

Si le champ de l'exposant ne contient que des zéros (représentation dénormalisée), la valeur représentée est

$$(-1)^S \times 0{,}M \times 2^{-126}$$

Sinon (représentation normalisée), la valeur représentée est

$$(-1)^S \times 1{,}M \times 2^{E-127}$$

Exercice 1. *Changements de bases*

Effectuez la conversion des valeurs indiquées de leurs bases d'origine vers les bases demandées.

1. $1011001_{(2)} = ???_{(10)}$

Solution : $1011001_{(2)} = 89_{(10)}$

2. $134_{(6)} = ???_{(10)}$

Solution : $134_{(6)} = 58_{(10)}$

3. $1111_{(3)} = ???_{(2)}$

Solution : $1111_{(3)} = 40_{(10)} = 101000_{(2)}$

4. $4426_{(10)} = ???_{(2)}$

Solution : $4426_{(10)} = 1000101001010_{(2)}$

5. $4426_{(10)} = ???_{(8)}$

Solution : $4426_{(10)} = 10512_{(8)}$

6. $4426_{(10)} = ???_{(16)}$

Solution : $4426_{(10)} = 114A_{(16)}$

Exercice 2. *Représentations des nombres*

Donnez les représentation du nombre -204 dans les formats suivants :

a. sur 16 bits en complément à 2

b. sur 16 bits avec un bit de signe

c. en IEEE 754 simple précision (nombre à virgule flottante sur 32 bits).

Solution : En complément à 2 sur 16 bits : 1111 1111 0011 0100.
Avec un bit de signe sur 16 bits : 1000 0000 1100 1100.
En IEEE 754 simple précision (nombre à virgule flottante sur 32 bits) :
— Signe négatif : 1
— $204_{(10)} = 11001100_{(2)} = 1{,}1001100 \times 2^7$
— Exposant réel $e_r = 7$, exposant biaisé $e_b = e_r + 127 = 134 = 10000110_{(2)}$
— Mantisse : 10011000000000000000000
Donc l'écriture complète de -204 en IEEE 754 sur 32 bits est

1 10000110 10011000000000000000000

Exercice 3. *Opérations binaires*

Dans cet exercice on considère des nombres entiers relatifs représentés sur 8 bits en complément à 2.

Donnez le résultat des opérations suivantes (toujours sur 8 bits, en complément à 2) en précisant pour chacune si le résultat obtenu est correct ou s'il y a eu un dépassement de capacité :

a. 11100010 + 11001001

b. 11001110 - 01110100

c. 00010110 × 11111010

d. 01110011 / 00000101

Solution :

a. $11100010 + 11001001 = 10101011$ On additionne deux nombres négatifs (commencent par 1) et on obtient un résultat négatif donc le résultat est correct (malgré la retenue sortante qui est ignorée).

b. $11001110 - 01110100 = 01011010$ C'est un nombre négatif moins un nombre positif. Le résultat devrait être négatif mais il est positif. Il y a donc eu un dépassement de capacité.

c. $00010110 \times 11111010 = 10110_{(2)} \times -110_{(2)} = -10000100_{(2)}$ La valeur absolue du nombre est supérieure à 128 (8 chiffres) donc il y a un dépassement de capacité.

d. $01110011/00000101 = 00010111$ C'est une division donc il n'y a pas de problème de dépassement de capacité (et ici le reste est 0).

Exercice 4. *Précision flottante*

1 Donnez la représentation IEEE 754 simple précision du nombre 1.

Solution : 0 01111111 00000000000000000000000

On note η le plus petit nombre strictement supérieur à 1 représentable en IEEE 754 simple précision.

2 Donnez la représentation de η en IEEE 754 simple précision.

Solution : 0 01111111 00000000000000000000001

3 Quelle est la valeur de $(1 - \eta)$?

Solution :

$$\begin{aligned} \eta &= 1{,}00000000000000000000001_{(2)} = 1 + 2^{-23} \\ 1 - \eta &= -2^{-23} \end{aligned}$$

4 Donnez la représentation en IEEE 754 simple précision de $(1 - \eta)$

Solution :

— Signe négatif : 1
— $1{,}0 \times 2^{-23}$
— Exposant réel $e_r = -23$, exposant biaisé $e_b = e_r + 127 = 104 = 01101000_{(2)}$
— Mantisse : 00000000000000000000000
Donc la représentation IEEE 754 de $(1 - \eta)$ sur 32 bits est

$$1\ 01101000\ 00000000000000000000000$$

Exercice 5. *Multiplexeur*

Un multiplexeur (MUX) est un circuit qui permet de sélectionner un signal parmi un ensemble d'entrées qu'il reçoit. Le signal d'entrée qui doit être transmis en sortie est déterminé par un autre ensemble d'entrées appelées *sélecteurs*. Avec k sélecteurs, on peut sélectionner un signal parmi 2^k entrées possibles.

L'exemple le plus simple est le multiplexeur à un unique sélecteur (cf. figure 1). Ce circuit comporte 3 entrées x_0, x_1 (signaux à sélectionner) et s_0 (sélecteur) et une unique sortie y. Si le sélecteur est à 0, la sortie y prend la valeur de l'entrée x_0 tandis que si le sélecteur est à 1, la sortie y prend la valeur de x_1.

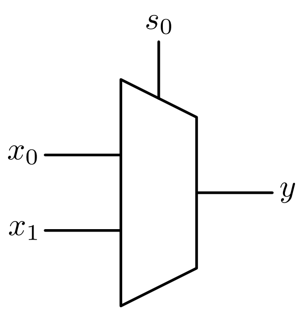


FIGURE 1 – Multiplexeur à un selecteur

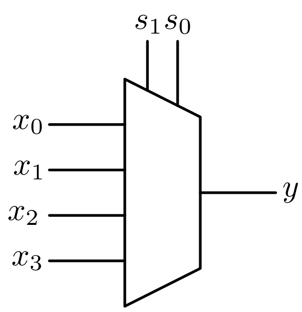


FIGURE 2 – Multiplexeur à deux selecteurs

1 Donnez la table de vérité du multiplexeur à 1 sélecteur.

Solution :

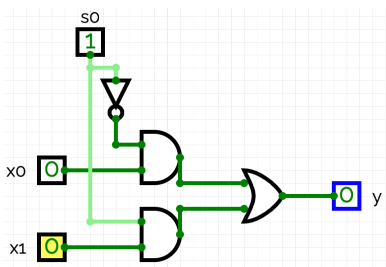
s_0	x_0	x_1	y
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

2 Dessinez un circuit correspondant au multiplexeur à 1 sélecteur en n'utilisant que les portes logiques élémentaires (NON, ET, OU).

Solution : Le multiplexeur à un sélecteur correspond à la formule booléenne

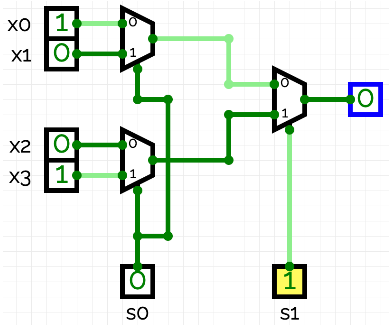
$$(\overline{s_0} \wedge x_0) \vee (s_0 \wedge x_1)$$

On peut utiliser le circuit



❸ Dessinez un circuit correspondant au multiplexeur à 2 sélecteurs (cf. figure 2). Le signal sélectionné est celui dont l'indice correspond au nombre binaire s_1s_0 (par exemple si $s_1 = 1$ et $s_0 = 0$ alors c'est l'entrée x_2 qui est sélectionnée car $10_{(2)} = 2$).
Indication : Vous pouvez utiliser 3 copies du multiplexeur à 1 sélecteur pour réaliser ce circuit.

Solution :



Exercice 6. *Questions de cours*

❶ Quel est le composant d'un microprocesseur qui est responsable d'effectuer les additions sur des entiers en binaire ?

Solution : L'unité arithmétique et logique (UAL) (ou ALU en anglais).

❷ Pourquoi a-t-on ajouté des nombres en représentation dénormalisée dans la norme IEEE 754 ? Quels sont les avantages d'avoir cette représentation ?

Solution : Il y a plusieurs intérêts aux nombres dénormalisés (bits d'exposant à 00000000, pas de 1 implicite dans la mantisse, exposant réel vaut -126). Entre autres :
— On peut représenter des nombres plus petits que si tous les nombres étaient normalisés. Avec des nombres normalisés, à cause du 1 implicite le plus petit nombre strictement positif représentable serait 2^-127, en dénormalisé on peut représenter 2^{-149}.
— On peut représenter le nombre 0.
— La différence entre deux nombres à virgule flottantes distincts peut toujours être représentée. Si on ne représentait que des nombres normalisés, la différence entre certains nombres serait trop petite pour être représentée.

❸ Combien de mémoire faut-il pour représenter (sans compression particulière) une image quelconque de 3000×2000 pixels si chaque pixel est représenté par une couleur au format RGB avec 256 valeurs possibles par composante ?

Solution : Il faut 3 octets par pixel de l'image, donc $3000 \times 2000 \times 3 = 18000000$ octets $= 18$ Mo.