TD 1 : warmup

Les exercices marqués (*) ne seront pas traités dans les séances TD. Vous êtes invités d'essayer de resoudre ces exercices chez vous.

Représentation des nombres non signés

Exercice 1
- Ecrire les représentations du nombre $(1110\ 1101\ 0101\ 0111)_2$ en octal et en hexadécimal.
- Ecrire en binaire et en décimal la valeur de l'hexadécimal 0xFEA ?
- (*) Ecrire les nombres suivants en binaire et en hexadécimal pour $(125)_{10}$ et $(2001)_{10}$.

Représentation des entiers binaires signés

Exercice 2 Donnez les représentations binaires sur 8 bits de -115 en utilisant les trois représentations (valeur absolue et signe, complément à 1, complément à 2).

Exercice 3 Donnez la valeur en base 10 des nombres binaires 01010101, 10010001, selon que l'on les lit en considérant un codage d'entiers sur 8 bits non signé, signé par valeur absolue et signe, signé en complément à 1, et signé en complément à 2.

Exercice 4 Additionnez en binaire -115 et 92, puis -115 et -2 dans les deux représentations complément à 1 et complément à 2. Que se passe-t-il ? Qu'en déduisez-vous ?

Exercice 5 ($*$) Effectuez les additions suivantes en considérant une représentation des nombres sur 8 bits (si un 9ème bit est produit, on le considera comme retenue et il ne fera pas parti du résultat). Interprétez les résultats pour une représentation de nombres entiers en complément 2. Sous quelle condition (en prennant en compte la retenue) peut-on dire que le résultat de l'addition est correcte ?

00101101	11111111	10000001	01111111
+ 01101111	+ 11111111	+ 10000010	+ 11111111

Circuits combinatoires

Exercice 6 Etablir pour l'additionneur complet (3 entrées, 2 sorties) les tables de vérité. Déduisez les formes disjonctives des deux sorties et implémentez l'additionneur en circuit combinatoire.

Donnez une description Verilog (netlist) de l'additionneur.

Représentation des flottants

Nous parlons de nombres flottants plutot que de réels puisqu'on ne peut représenter qu'un nombre fini de réels, qui ne sont de plus que des rationnels. La méthode employée est similaire à la notation scientifique utilisée dans les calculatrices : le réel $r = -123,5$ est alors noté $r = -1,235.10^2$. En binaire, ce nombre vaut $-1111011,1$ et donc en notation scientifique $-1,1110111.2^6$. Dans ce cas, le *signe* vaut $'-'$, la *mantisse* vaut $M = 1,1110111$ et l'*exposant* vaut $E = 6$. Dans la notation scientifique binaire, le chiffre avant la virgule est toujours 1 : on peut donc l'omettre dans le codage des flottants concernant la mantisse. On parle de 1 *caché*.

Codage La représentation binaire sur 32 bits IEEE 754 d'un nombre flottant en simple précision est donnée ci-dessous

31	30 … 23	22 … 0
s	e = *exposant*	f = *partie fractionnaire*

Dans cette représentation, 1 bit est dédié au signe, 8 bits à l'exposant et 23 bits à la partie fractionnaire de la mantisse. Afin de pouvoir représenter des exposants négatifs, l'exposant est codé en excès à 127 : les huits bits codent en binaire l'entier $e = E + 127$.

Dans le cas de $r = -123,5$, le signe est négatif : $s = 1$; la partie fractionnaire de la mantisse est 1110111 et donc m vaut 1 10000101 111 0111 0000 0000 0000 0000 ; enfin, $e = E + 127 = 133 = (1000\ 0101)_2$. Le codage de r est donc

1	10000101	111 0111 0000 0000 0000 0000

c'est-à-dire en regroupant par paquets de 4

1100	0010	1111	0111	0000	0000	0000	0000

et donc, en hexadécimal :

C	2	F	7	0	0	0	0

Décodage En notant s le bit le plus à gauche, $e_7 \dots e_0$ les huits bits d'exposant et $f_1, \dots, f_{23}$ les 23 bits de la partie fractionnaire de la mantisse, alors l'exposant positif est codé en binaire par : $e = (e_7 \dots e_0)_2$ et le nombre réel représenté par les 32 bits $se_7 \dots e_0 f_1, \dots, f_{23}$ vaut

$$r = (-1)^s.(1 + \sum_{i=1}^{23} f_i.2^{-i}).2^{e-127}$$

Afin de traité certains cas spéciaux, la norme prévoit un schéma de codage comme suit :

exposant e	partie fractionnaire f	valeur de r	
$0 < e < 255$	f	$(-1)^s \times (1, f)_2 \times 2^{e-127}$	normalisé
$e = 0$	$f \neq 0$	$(-1)^s \times (0, f)_2 \times 2^{-126}$	dénormalisé
$e = 0$	$f = 0$	0	dénormalisé
$e = 255$	$f = 0$	$+/-\infty$	réservé
$e = 255$	$f \neq 0$	NaN	réservé

Lorsqu'on travaille sur de très petits nombres (exposant minimum), on n'effectue pas la normalisation (le chiffre avant la virgule est alors 0). Le nombre est alors qualifié de dénormalisé.

NaN (Not a Number) est une valeur spéciale destinée à coder le résultat d'opérations incorrectes (comme la division par zéro).

Exercice 7 Quel réel est représenté par les 8 chiffres hexadécimaux $(4148\ 0000)_{16}$ selon la norme IEEE 754 ? Détaillez vos calculs.

Exercice 8 Quelle suite de 32 bits représente le réel -4,25 selon cette norme ? Détaillez vos calculs. Indiquez votre résultat sous la forme de 8 chiffres hexadécimaux.

Exercice 9 Codez sur 32 bits : 2,5 et -4,75.

Exercice 10 $(*)$

1. Quels sont le plus grand positif et son prédécesseur ; indiquer leur écart ?
2. Quels sont les plus petits entiers normalisé et dénormalisé et leur successeur ?
3. Quels sont le plus grand et le plus petit négatifs ?

Exercice 11 $(*)$ Quels nombres simple précision correspondent aux mots de 32 bits suivants (donnés en hexadécimal) : 0x41300000, 0x41E00000, 0xBF800001, 0x00A00000 et 0x00100000 ?