

CEG 3556: Conception Avancée des Systèmes
Informatiques (Hiver 2018)
Prof. Rami Abielmona
Laboratoire #1: Multiplication a Point-Mobile

11 Janvier, 2018

1 Objectif

L'objectif de ce laboratoire est de concevoir et construire un multiplicateur à point mobile (floating-point multiplier) en VHDL.

A la fin de cet exercice, l'étudiant(e) doit être capable de:

- Concevoir, réaliser et tester une unité d'addition à point mobile (floating-point adder unit);
- Concevoir, réaliser et tester une unité de multiplication à point mobile (floating-point multiplier unit);
- Démontrer une compréhension complète de l'arithmétique à point mobile (floating-point arithmetic).

2 Pré-Lab

Habituellement, un pré-lab serait associée à tous les laboratoires, mais en raison de contraintes de temps, ce laboratoire ne nécessitera pas un pré-laboratoire. Les laboratoires successives, cependant, auront une section pré-laboratoire qui doit être complétée par le groupe avant les heures de laboratoire alloués.

3 Introduction aux Nombres à Point Mobile (Floating-Point Numbers)

Travailler avec arithmétique en point fixe (fixed-point arithmetic) limite la convivialité d'un processeur. Si les opérations sur les nombres avec des fractions

(e.x. 10.2445), de très petits nombres (e.x. 0.000004), ou de très grands nombres (e.x. 42.243×10^5) sont nécessaires, une représentation différente doit être utilisée: saisir l'arithmétique à point mobile (floating-point arithmetic). Ce dernier terme est utilisé quand le point binaire n'est pas fixe, comme c'est le cas avec l'arithmétique des nombres entiers (integer fixed-point arithmetic).

Afin d'obtenir une partie de la terminologie, prenons un exemple simple d'un nombre à point mobile, -2.42×10^3 . Le symbole '-' indique le champ signe du nombre, tandis que '242' indique le champ des chiffres significatifs du nombre, et enfin le '3' indique le champ de facteur d'échelle du nombre. Il est intéressant de noter que la chaîne de chiffres significatifs est techniquement appelé la mantisse du nombre, tandis que le facteur d'échelle est appelé l'exposant du nombre.

3.1 Standard IEEE pour les Nombres à Point Mobile

Au lieu d'avoir une myriade de représentations de point mobile, il a été décidé qu'une norme doit être convenue. Cette norme est le standard IEEE pour les représentations de nombres à point mobile. La forme générale de la représentation est la suivante:

$$(-1)^S * M * 2^E, \text{ ou} \quad (1)$$

S représente le bit de signe,

M représente la mantisse, et

E représente l'exposant.

Dans les ordinateurs d'aujourd'hui, 32 bits est la longueur standard des mots, et cela, l'IEEE a décidé qu'une représentation de 32 bits pour les nombres à point mobile pourrait bénéficier des caractéristiques de transfert de bus d'aujourd'hui. Ainsi, un *nombre de précision simple* (référer à 1) a été défini à 32 bits, tandis qu'un *nombre de précision double* (référer à 2) a été défini à 64 bits. Le numéro de précision simple a trois champs principaux :

Signe 1 bit utilisé pour désigner le signe du nombre (0 signifie + et 1 signifie -);

Exposant 8 bits exposant signé en représentation de excès-de-127 (à voir plus tard) ;

Mantisse champ fractionnaire de 23 bits.

La mantisse de 24 bits (le premier bit de 1 est implicite) peut représenter environ un nombre décimal de sept chiffres, tout en exposant de 8 bits à une base implicite de 2 fournit un facteur d'échelle avec une gamme raisonnable. Ainsi, un total de 32 bits est nécessaire pour la représentation de précision simple, qui fournit un facteur d'échelle de 2^{-126} à 2^{127} . Dans le cas de précision double, l'exposant augmente à 11 bits, tandis que le champ de mantisse augmente à 52 bits, ce qui permet un facteur d'échelle de 2^{-1022} à 2^{1023} , tout en offrant une précision équivalente à 16 chiffres décimaux.

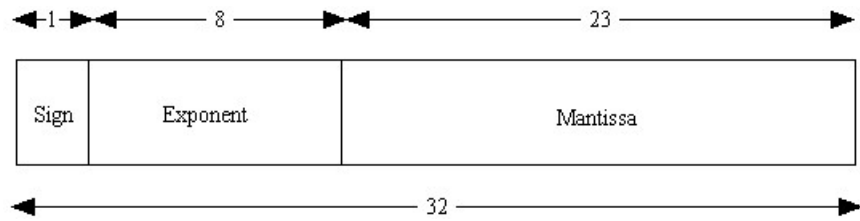


Figure 1: Représentation des Nombres de Précision Simple à Point Mobile

La représentation d'excès-de-127 mentionné ci-dessus lors de l'analyse de la partie exposant est utilisée pour comparer efficacement les tailles relatives des deux nombres à point mobile. Au lieu de stocker l'exposant (E) comme un nombre signé, nous stockons sa représentation comme nombre entier non signé ($E' = E + 127$). Cela nous donne une gamme E' of $0 \leq E' \leq 255$. Alors que les valeurs des extrémités 0 et 255 sont utilisées pour représenter des nombres spéciaux (0 exacte, infini, et nombres dénormalisés), la gamme de fonctionnement de E' devient $1 \leq E' \leq 254$, ainsi, ce qui limite la gamme de E à $-126 \leq E \leq 127$. (Noter que si l'on travaille avec des nombres de précision double, une présentation d'excès-de-1023 est utilisée).

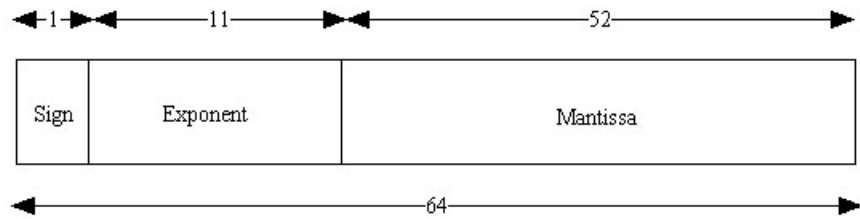


Figure 2: Représentation des Nombres de Précision Double à Point Mobile

Une note finale sur le standard IEEE est que le champ de la mantisse est toujours **normalisé**. Ce dernier implique que le point décimal est placé à droite du premier chiffre significatif (non nulle). Par conséquent, les 23 bits stockés dans le champ M représentent en fait la partie fractionnaire de la mantisse, qui est les bits à la droite du point binaire. Comme mentionné précédemment, le bit le plus significatif de la mantisse est toujours égal à 1, en raison de la normalisation binaire.

Si nous revisitons notre formule originale, nous formalisons maintenant la représentation d'un nombre a 32 bits a précision simple en point mobile comme:

$$(-1)^S * (1 + M) * 2^{E-127}, \text{ ou} \quad (2)$$

S représente le bit de signe (1 bit),
M représente la mantisse (23 bits), et
E représente l'exposant (8 bits).

3.2 Un Exemple Simple

Essayons de représenter le nombre décimal $(-0.75)_{10}$ en format point mobile IEEE. Tout d'abord, nous remarquons que $(-0.75)_{10} = (-3/4)_{10} = (-3/2^2)_{10}$. En notation binaire, nous avons $(-0.11)_2 = (-0.11)_2 \times 2^0 = (-1.1)_2 \times 2^{-1}$.

En se référant à l'équation (2), nous pouvons représenter notre nombre comme :

$$(-1)^1 * (1 + .100000000000000000000000_2) * 2^{126-127} \quad (3)$$

Ainsi, notre représentation du nombre à précision simple est $(101111110100000000000000000000)_{\text{2}}$,
où,

- Le bit de signe est $(1)_2$, pour les nombres négatifs;
- L'exposant est $(01111110)_2$, pour représenter $(126)_{10}$;
- La mantisse est $(1000000000000000000000)_2$, pour représenter la partie fractionnaire $(0.1)_2$.

3.3 Addition à Point Mobile

Pour effectuer l'addition à point mobile, un algorithme simple est réalisé:

1. Choisissez le nombre avec le plus petit exposant et décaler sa mantisse vers la droite un nombre de positions égal à la différence des exposants;
2. Définir l'exposant du résultat égal à la plus grande exposant;
3. Effectuer l'addition sur les mantisses et déterminer le signe du résultat ;
4. Si nécessaire, normaliser la valeur résultante.

Reportez-vous à la figure 4 pour un diagramme pour l'opération d'addition des nombres à point mobile. La figure est un extrait de notre livre de cours, se il vous plaît se référer à la section 4.8 du livre pour une explication détaillée de l'interfonctionnement du diagramme, ainsi que l'addition des nombres a point mobile.

Egalement montré dans ce document est l'implémentation d'un schéma d'un additionneur de nombres a point mobile (figure 5). Ce diagramme indique le chemin de données de l'addition, et élimine le chemin de contrle de l'unité.

3.4 Multiplication à Point Mobile

Pour effectuer la multiplication à point mobile, un algorithme simple est réalisé:

1. Ajouter les exposants et soustraire 127;
2. Multiplier les mantisses et de déterminer le signe du résultat;

3. Si nécessaire, normaliser la valeur résultante.

Reportez-vous à la figure 6 pour un diagramme pour l'opération de multiplication des nombres à point mobile. La figure est un extrait de notre livre de cours, se il vous plaît se référer à la section 4.8 du livre pour une explication détaillée de l'interfonctionnement du diagramme, ainsi que la multiplication des nombres a point mobile.

4 Laboratoire

Dans ce laboratoire, vous allez premièrement procéder à mettre en place un additionneur à point mobile comme désigné en figure 5, ainsi que faire la conception et l'implémentation d'un multiplicateur à point mobile comme indiqué dans la figure 6. La spécification des entrées/sorties de l'additionneur et du multiplicateur sont présentés dans le tableau 1. Nous allons travailler avec un exposant de **7 bits de largeur** et une mantisse de **8 bits de largeur**, tandis que le bit de signe reste évidemment à 1 bit de largeur. Notez également que l'exposant de 7 bits sera représenté dans un **format d'excès-de-63**.

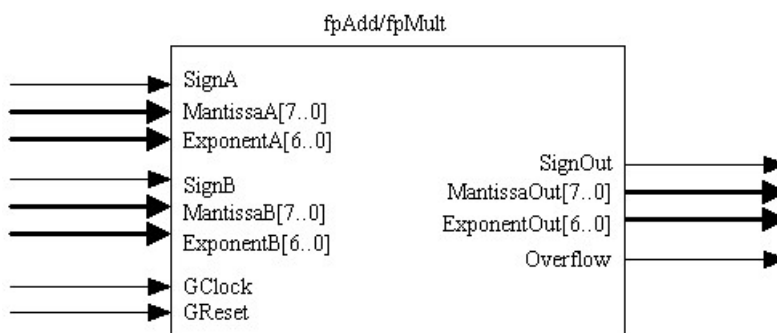


Figure 3: L'entité d'un Additionneur/Multiplicateur à Point Mobile

5 Restrictions de Conception

- Implémentations utilisant Verilog ne seront pas acceptées. Effectuer toutes les implémentations utilisant VHDL seulement.
- La modélisation au niveau comportemental ne sera pas acceptée. La conception doit être faite au niveau de la modélisation structurelle.
- La conception utilisant Register Transfer Logic (RTL) et la volonté de codage sont obligatoire.

<i>Type de Port</i>	<i>Nom</i>	<i>Description</i>
Entrée	GClock	Horloge globale nécessaire pour synchroniser les circuits
Entrée	GReset	Réinitialisation globale nécessaire pour mettre les internes aux états connus
Entrée	SignA	Bit de signe pour l'entrée de A: 0 = positif, 1 = négatif
Entrée	MantissaA[7..0]	Mantisse pour l'entrée de A
Entrée	ExponentA[6..0]	Exposant pour l'entrée de A
Entrée	SignB	Bit de signe pour l'entrée de B: 0 = positif, 1 = négatif
Entrée	MantissaB[7..0]	Mantisse pour l'entrée de B
Entrée	ExponentB[6..0]	Exposant pour l'entrée de B
Sortie	SignOut	Bit de signe pour la sortie: 0 = positif, 1 = négatif
Sortie	MantissaOut[7..0]	Mantisse pour la sortie
Sortie	ExponentOut[6..0]	Exposant pour la sortie
Sortie	Overflow	Débordement utilisé pour indiquer si le débordement a été détecté

Table 1: Spécification d'Entrées/Sorties

- Utiliser la conception graphique pour l'entité de haut niveau, et utiliser votre jugement pour tous les autres sous-blocs. Cependant, tous les modules atomiques doivent être implémentés en VHDL (bascule de type D, additionneur de 1 bit, comparateur de 1 bit et ainsi de suite).
- Ce laboratoire implique l'arithmétique à point mobile de 16 bits, par conséquent, les étudiants doivent concevoir et tester leurs modules en utilisant ce format, et pas les formats de précision simple ou double normalisés par l'IEEE.
- Aucune instantiation de base ne seront autorisées (LPMs de Altera ou noyaux d'IP gratuits par exemple). Tous les blocs de construction doivent être conçus et réalisés par le groupe.
- L'entité de haut niveau est donnée dans la spécification de format des entrées/sorties, mais les internes sont laissées au groupe à faire. Un échantillon de la solution schématique pour l'additionneur à point mobile a été donné, mais le groupe est libre de concevoir l'additionneur et le multiplicateur en utilisant d'autres méthodes.
- La conception doit être synchrone et globalement réinitialisable. Cela signifie que les signaux de l'horloge globale et de la réinitialisation sont nécessaires dans les deux blocs fonctionnels (le `fp_adder` et le `fp_multiplieur`).
- Simulez les deux conceptions et vérifiez vos résultats de simulation avec vos résultats théoriques (par exemple ajoutez et multipliez les deux nombres suivants: 10.1 et 6.5).
- Télécharger la conception sur la puce de Cyclone sur vos planches DE-2. Utilisez les boutons pour entrer les deux nombres à point mobile, et les seize LEDs pour démontrer la fonctionnalité de sortie correcte.

- Indiquer la détection de débordement en utilisant le point sur les décodeurs de 7 segments du FPGA Cyclone.
- Chaque groupe doit démontrer une version fonctionnelle du laboratoire au TA avant la date d'échéance du rapport.

6 Rappels pour le Rapport

- Inclure simulations de synchronisation temporelle avec des explications pour tous fichiers de source VHDL.
- Décrire et commenter tous vos fichiers de source VHDL.
- Inclure une représentation organigramme de votre solution au problème.
- Inclure un schéma de blocs de votre solution au problème.
- Si vous utilisez la conception d'ASM, inclure toutes les diagrammes et chemins appropriées (contrôle et données).
- Si vous utilisez la conception de FSM, inclure le diagramme d'états avec toutes les entrées et sorties appropriées.
- Décrivez, dans vos propres mots, votre solution au problème.
- Décrivez vos obstacles de conception et comment ils ont été résolus.
- Annexer tout code VHDL et la conception graphique dans votre rapport.
- Soumettre une copie électronique de tous les fichiers VHDL et de conception graphique dans votre rapport.

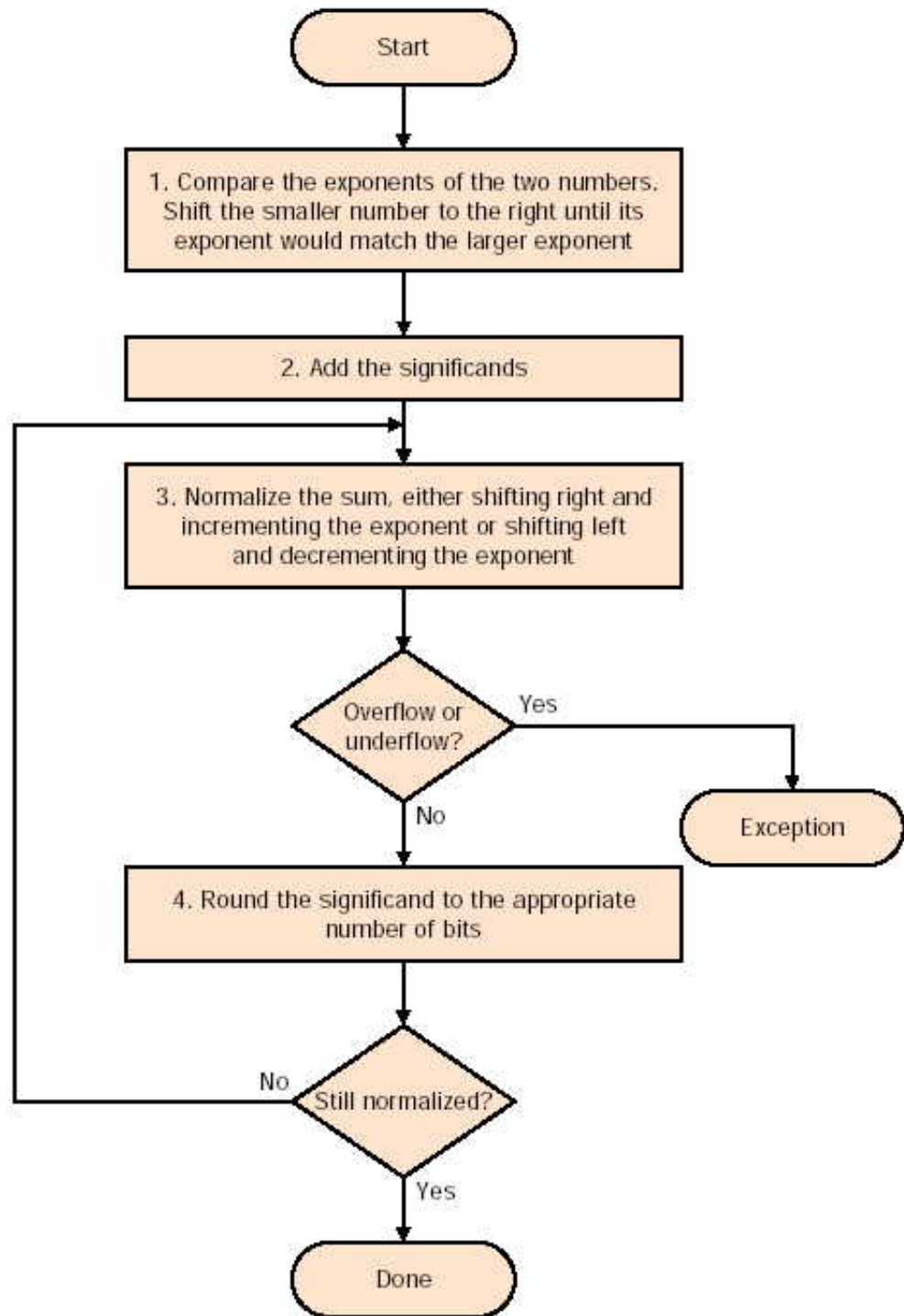


Figure 4: Diagramme d'un Additionneur de Nombres à Point Mobile

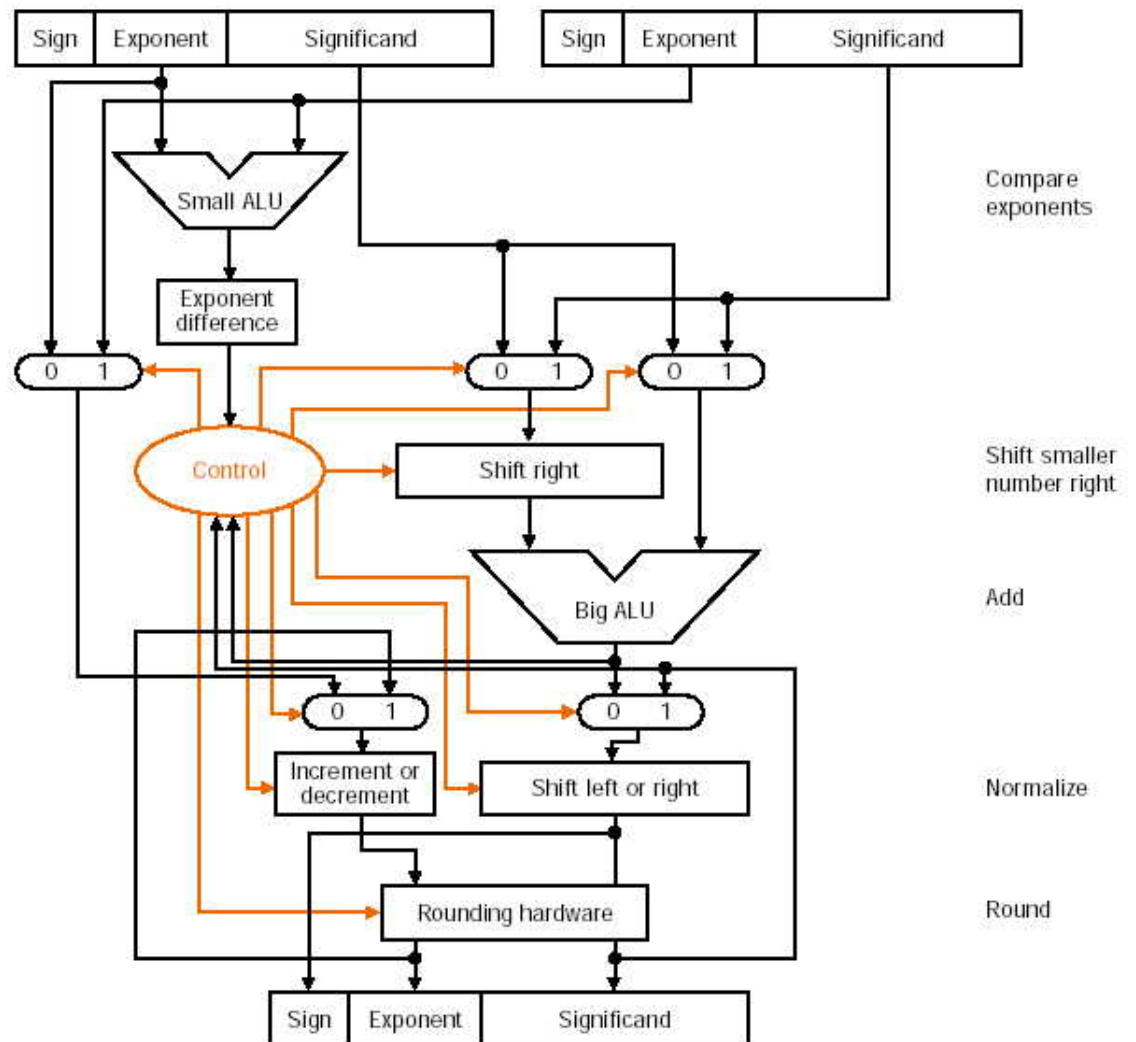


Figure 5: Diagramme de Blocs d'un Additionneur de Nombres à Point Mobile

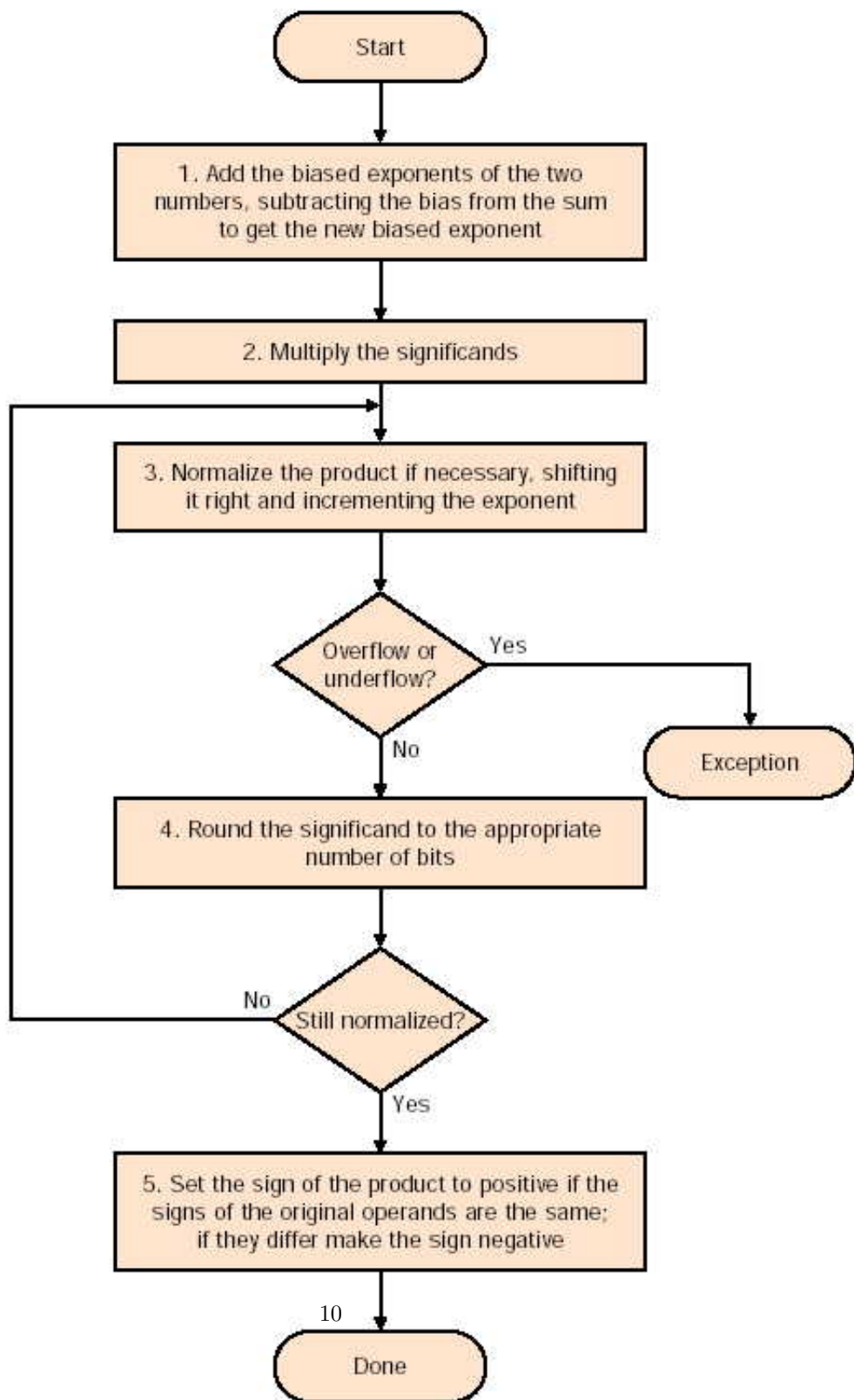


Figure 6: Diagramme d'un Multiplicateur de Nombres à Point Mobile