Algorithmique IV (UE-41) - TD 5.

TD 5. Calcul multiprécision [1]

EXERCICE 1. Soit k un entier naturel non-nul. On cherche à calculer le produit de deux entiers naturels A et B codés comme des entiers non-signés en machine sur des registres de 2^k bits.

(1) Quelle est l'entier *naturel* le plus grand que l'on peut représenter avec un registre de 64 bits ?

(2) Quelle est la condition sur les tailles de deux entiers A et B pour lesquels on peut réaliser l'opération de multiplication $A \cdot B$ en machine ?

On se propose d'écrire un algorithme qui calcule le produit $R := A \cdot B$ de deux entiers non-signés A et B de $n := 2^k$ bits, en segmentant les $2n$ bits du résultat R dans deux registres de n bits chacun.

Si X est un registre codé sur un nombre pair de bits, alors X_H (resp. X_L) désigne le registre contenant les bits de la première moitié « haute » (resp. de la deuxième moitié « basse ») de X. Donc $X = X_H|X_L$ si $|$ est l'opérateur de concaténation binaire, ou exprimé autrement

$$X = 2^p X_H + X_L$$

si X est codé sur $n = 2p$ bits.

Ce sont ces deux registres que nous allons calculer $R_H|R_L := A \cdot B$ pour dépasser la limitation du produit.

On dispose des algorithmes suivants dont les paramètres sont toujours codés sur n bits (exemples pour $k = 2$ et $n = 4$) :

- `MUL(X,Y)` : renvoie $X_L \cdot Y_L$.
  Exemple : $\texttt{Mul}(7, 10) = 6$ $((01\underline{11}, 10\underline{10}) \mapsto 0110)$.
- `ADD(X,Y)` : renvoie $X + Y$ en opérant sur les $n - 1$ bits faibles.
  Exemple : $\texttt{Add}(7, 13) = 12$ $((0\underline{111}, 1\underline{101}) \mapsto 1100)$.
- `HIGH(X)` : renvoie $0^{\frac{n}{2}}|X_H$.
  Exemple : $\texttt{H}(13) = 3$ $(\underline{11}01 \mapsto 0011)$.
- `LOW(X)` : renvoie $0^{\frac{n}{2}}|X_L$.
  Exemple : $\texttt{L}(13) = 1$ $(11\underline{01} \mapsto 0001)$.
- `LOWLOW(X,Y)` : renvoie $X_L|Y_L$.
  Exemple : $\texttt{LOWLOW}(7, 9) = 13$ $((01\underline{11}, 10\underline{01}) \mapsto 1101)$.

Le principe est le suivant : on partitionne chaque opérande X en deux moitiés X_H et X_L sur $\frac{n}{2}$ bits :

$$\begin{aligned} A \cdot B &= (A_H 2^{\frac{n}{2}} + A_L)(B_H 2^{\frac{n}{2}} + B_L) \\ &= A_H B_H\, 2^n + A_H B_L\, 2^{\frac{n}{2}} + A_L B_H\, 2^{\frac{n}{2}} + A_L B_L. \end{aligned} \tag{1}$$

Ces quatre produits sont calculables sans débordement par l'algorithme `MUL(X,Y)`, il reste à déterminer comment les exploiter pour déterminer le contenu des deux registres R_H et R_L codés sur n bits chacun.

(3) Écrivez les algorithmes `LOW(X)`, `HIGH(X)` et `LOWLOW(X,Y)` à l'aide des opérateurs logiques bit à bit.

(4) Pour $k = 2$, c'est-à-dire quand les opérandes A et B sont codés sur $n = 4$ bits, représentez les contenus des 4 produits (cf. (1)) pour $A = B = 2^n - 1$ ainsi que ceux des registres R_H et R_L en binaire.

(5) Dressez une table des valeurs binaires des 4 termes de la somme (1) pour mettre en évidence le calcul à réaliser pour obtenir le contenu des registres R_H et R_L contenant respectivement la moitié haute et la moitié basse du produit AB. Généralisez à des opérandes A et B codés sur n bits pour écrire un algorithme qui calcule R_H et R_L.

EXERCICE 2. Soit n un entier naturel. On veut estimer le nombre de chiffres nécessaires pour représenter n en base 2 (asymptotiquement).

(1) À l'aide de la formule de Stirling ci-dessous :

$$\lim_{n \to +\infty} \frac{n!}{\sqrt{2\pi n}(n/e)^n} = 1, \tag{2}$$

donnez une estimation asymptotique du nombre de chiffres nécessaires pour représenter la factorielle $n!$ d'un entier naturel n en base 2.

(2) Même question, mais en comparant avec une somme intégrale.

EXERCICE 3. Écrivez un algorithme qui réalise la soustraction $A - B$ de deux entiers naturels en multiprécision et en supposant que $A \geqslant B$. Calculez sa complexité dans le meilleur des cas et dans le pire des cas.

1. Version du 25 février 2025, 13 : 11

EXERCICE 4. On considère deux entiers A et B de n chiffres en base b et on note $\ell := \frac{n}{2}$. On scinde A (resp. B) à l'aide de deux nombres de ℓ chiffres A_H et A_L (resp. B_H et B_L) :

$$A = A_H\, b^\ell + A_L \quad \text{et} \quad B = B_H\, b^\ell + B_L.$$

(1) Calculez les produits $A\,B$ et $(A_H + A_L)(B_H + B_L)$.

(2) À partir des résultats de la question (1), vérifiez que

$$A\,B = A_H B_H b^n + \big((A_H + A_L)(B_H + B_L) - (A_H B_H + A_L B_L)\big) b^\ell + A_L B_L \quad (3)$$

(3) En supposant que le nombre de multiplications pour calculer le produit de deux nombres à n chiffres est $P(n)$ et que le produit de deux nombres à un chiffre a un coût constant de $\Theta(1)$ (c'est une recherche en table), exprimez la fonction P avec une relation de récurrence. Pour simplifier les calculs, on suppose que n est une puissance de 2. Indication : utilisez l'égalité (3).

(4) Montrez que $P(n) = \Theta(n^{\log_2(3)})$.