Algorithmique III. L2 Informatique I41.

TD 5. Calcul multiprécision [1]

EXERCICE 1. Soit k un entier naturel non-nul. On s'intéresse au produit de deux entiers naturels 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 $A \times B$ en machine ?

On se propose dans la suite d'écrire un algorithme qui calcule le produit de deux entiers non-signés A et B codés sur $n := 2^k$ bits. On dispose des algorithmes suivants dont les paramètres sont toujours codés sur n bits (les exemples sont donnés pour $k = 2$, i.e. pour $n = 4$:

- `Mul(X,Y)` : renvoie le produit $X \times Y$ mais en opérant uniquement sur les $\frac{n}{2}$ bits de poids faible de X et Y. Ex : $\texttt{Mul}(7, 10) = 6$.
- `Add(X,Y)` : renvoie la somme $X + Y$ mais en opérant uniquement sur les $n - 1$ bits de poids faible de X et Y. Ex : $\texttt{Add}(7, 13) = 12$.
- `H(X)` : renvoie l'entier dont les $\frac{n}{2}$ bits de poids faible sont ceux de poids fort de X et dont les $\frac{n}{2}$ bits de poids fort sont nuls. Ex : $\texttt{H}(12) = 3$.
- `L(X)` : renvoie l'entier dont les $\frac{n}{2}$ bits de poids faible sont ceux de X et dont les $\frac{n}{2}$ bits de poids fort sont nuls. Ex : $\texttt{L}(13) = 1$.
- `REG(X,Y)` : renvoie l'entier dont les $\frac{n}{2}$ bits de poids fort sont ceux de poids faible de X et dont les $\frac{n}{2}$ bits de poids faible sont ceux de poids faible de Y. Ex : $\texttt{REG}(7, 9) = 13$.

Le produit $R := A \times B$ est rangé dans deux registres R_H et R_L de n bits contenant respectivement les n bits de poids forts et de poids faibles du résultat R. Le principe est le suivant : on coupe chaque opérande X en deux moitiés X_H et X_L sur $\frac{n}{2}$ bits :

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

[1] Version du 5 juillet 2022, 12 : 17

Ces 4 produits sont réalisables sans débordement par l'algorithme `Mul(X,Y)`, il reste à déterminer comment obtenir le contenu des deux registres R_H et R_L sur n bits chacun.

3. Écrivez les algorithmes `L(X)`, `H(X)` et `REG(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.

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 AB 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 $P(n) = \Theta(n^{\log_2(3)})$.