Christophe Mouilleron
christophe.mouilleron@ensiie.fr

TD 3 : Unité arithmétique et logique

Exercice 1 - Une histoire de drapeaux

Dans cet exercice, on considère que les entiers sont codés sur 8 bits avec complément à deux.

Pour chacune des opérations suivantes, donner :
- le résultat de l'opération en binaire et en décimal,
- la valeur des drapeaux S(ign), Z(ero), C(arry) et O(verflow).

1.1 `100 + 52`

1.2 `52 + (-56)`

1.3 `(-56) + (-100)`

1.4 `(-128) + (-128)`

Exercice 2 - Additionneur *carry-select* (exam. 2021)

Le but de cet exercice est de concevoir un additionneur binaire sur n bits efficace.

Soit k un diviseur de n. On va réaliser l'addition de la manière suivante :

1. découper les entrées en $\frac{n}{k}$ paquets de k bits ;
2. pour chaque poids (sauf le plus faible), faire en parallèle deux additions (séquentielles) sur k bits, la première avec une retenue de 0 et la deuxième avec une retenue de 1 ;
3. pour chaque poids, choisir le résultat final en fonction de la retenue issue de poids précédent.

Dans la suite, on appellera *additionneur carry-select* le circuit ainsi obtenu.

2.1 Dessiner l'*additionneur carry-select* quand $n = 6$ et $k = 2$.

2.2 Rappeler la profondeur et le nombre de portes logiques pour faire un additionneur séquentiel sur k bits.

2.3 Rappeler la profondeur et le nombre de portes logiques pour faire un multiplexeur $2N \to N$.

2.4 Calculer le nombre de portes de l'*additionneur carry-select* en fonction de n et de k.
Vérifier que le résultat est bien linéaire par rapport à n.

2.5 Calculer la profondeur de l'*additionneur carry-select* en fonction de n et de k.

2.6 Trouver la valeur de k qui minimise cette profondeur (à n fixé).
Quelle profondeur obtient-on alors ?

Exercice 3 - Diviser pour regner (exam. 2019)

L'additionneur *carry-select* vu en cours utilise en interne trois additionneurs séquentiels.

Afin de diminuer la profondeur, on se propose de remplacer ces additionneurs séquentiels par des additionneurs *carry-select*, et ce récursivement jusqu'à arriver une taille 1. L'addition de taille 1 est alors effectuée à l'aide d'un *full-adder*.

Dans la suite, on notera $t(n)$ la profondeur du circuit ainsi obtenu pour faire une addition sur n bits, et $a(n)$ le nombre de portes logiques nécessaires à la réalisation de ce circuit.

3.1 Dessiner le circuit correspondant lorsque $n = 2$, puis lorsque $n = 4$.

3.2 Donner les valeurs de $t(1)$ et $a(1)$.

3.3 Justifier les équations suivantes : $\begin{array}{rcl} t(n) & = & t(n/2) + 3, \\ a(n) & = & 3\,a(n/2) + 2\,n + 4. \end{array}$

3.4 Exprimer $t(n)$ et $a(n)$ en fonction de n uniquement. Commenter.

Exercice 4 - Additionneur de Brent et Kung **(exam. 2023)**

L'additionneur de Brent et Kung est un circuit permettant de faire une addition d'entiers non signés sur n bits. Il repose sur le principe de retenues anticipées vu en cours. On rappelle que le but est de calculer efficacement, à chaque position i, le bit C_i correspondant à la retenue entrante à cette position.

Pour ce faire, on utilise les valeurs suivantes :
- $P_i = A_i \oplus B_i$, indiquant qu'une retenue sera propagée à la position i ;
- $G_i = A_i \cdot B_i$, indiquant qu'une retenue sera générée à la position i

où A_i et B_i désignent les bits des deux opérandes de l'addition à la position i.

L'idée de Brent et Kung est de généraliser ces valeurs en introduisant pour $i \leq j$:
- $P_{i,j} = P_i P_{i+1} \dots P_j$,
- $G_{i,j} = G_j + G_{j-1} P_{j,j} + \cdots + G_i\, P_{i+1,j}$.

On remarque en particulier que, lorsque $i = j$, on a bien $P_{i,i} = P_i$ et $G_{i,i} = G_i$.

4.1 Rappeler la formule liant C_i, C_{i+1}, G_i et P_i vue en cours.

4.2 En vous inspirant de la formule précédente, donnez une formule pour calculer C_j à partir de C_0, $G_{0,j-1}$ et $P_{0,j-1}$.
Expliquez pourquoi votre formule est correcte.

4.3 Proposez un circuit pour calculer efficacement $P_{0,2^\ell-1}$ à partir des valeurs P_0, P_1, …, $P_{2^\ell-1}$.
Précisez, en fonction de ℓ, la profondeur et le nombre de portes logiques de votre circuit.

4.4 Afin de calculer $G_{i,j}$ efficacement, on introduit l'opérateur $\bullet$ défini par

$$(x_1,\ y_1)\ \bullet\ (x_2,\ y_2) = (x_1 x_2,\ y_1 x_2 + y_2).$$

Un calcul rapide montre que $\bullet$ est associatif, donc que $(a \bullet b) \bullet c = a \bullet (b \bullet c)$ pour tout a, b et c.
Montrer que, si $i \leq j < k$, on a $(P_{i,k},\ G_{i,k}) = (P_{i,j},\ G_{i,j}) \bullet (P_{j+1,k},\ G_{j+1,k})$.

4.5 Expliquez comment on peut obtenir $P_{0,2^\ell-1}$ et $G_{0,2^\ell-1}$ efficacement en utilisant la formule de la question précédente.
Illustrez votre méthode en dessinant le circuit correspondant pour $\ell = 3$.
note : On considère ici que les entrées sont P_0, G_0, P_1, G_1, …, $P_{2^\ell-1}$, $G_{2^\ell-1}$. Et on utilisera une boîte $\boxed{\bullet}$ en guise de circuit pour l'opérateur $\bullet$.

4.6 On note $T(\ell)$ la profondeur du circuit permettant de calculer $P_{0,2^\ell-1}$ et $G_{0,2^\ell-1}$.
Justifiez que $\begin{cases} T(0) = 0 \\ T(\ell) = T(\ell-1) + 2 \quad \text{si } \ell > 0 \end{cases}$, puis exprimez $T(\ell)$ en fonction de ℓ uniquement.

4.7 On note maintenant $A(\ell)$ le nombre de portes logiques de ce circuit.
Justifiez que $\begin{cases} A(0) = 0 \\ A(\ell) = 2\,A(\ell-1) + 3 \quad \text{si } \ell > 0 \end{cases}$, puis exprimez $A(\ell)$ en fonction de ℓ uniquement.

4.8 [$\star$] Expliquez comment étendre le circuit précédent pour calculer toutes les valeurs de la forme $P_{0,j}$ et $G_{0,j}$, où $0 \leq j < n$.
Illustrez votre méthode dans le cas où $n = 2^\ell = 8$.
indication : On pourra s'appuyer sur l'écriture binaire de $j + 1$.

4.9 Utilisez ce qui précède pour évaluer (en ordre de grandeur par rapport à n) la profondeur et le nombre de portes logiques pour l'additionneur de Brent et Kung.
Comparez à ce qui a été vu en cours/TD et commentez.