

TD 9 – Électronique Numérique

Conception d'un accélérateur matériel pour le calcul d'un polynôme

Notes et Conseils

On se propose ici de concevoir un accélérateur matériel, dédié au calcul de polynômes de rang 3. L'étude commence par une réflexion algorithmique, qui suggère des optimisations. Nous poursuivons la démarche par la manipulation abstraite de cet algorithme, à l'aide de *graphes*. L'accélérateur conçu par la suite se présente comme une microarchitecture, composée classiquement d'un automate de contrôle et d'un chemin de données. On se limitera à des valeurs de x entières, codées sur 32 bits.

1 Réflexion algorithmique

La conception d'un accélérateur matériel naît souvent de la création d'un algorithme aux performances théoriques intéressantes (ex : compression par Transformée en Cosinus Discret), ou par la découverte d'une reformulation intéressante (ex : multiplication de Karatsuba¹). Nous allons ici, en miniature, suivre cette démarche.

On considère un polynôme d'ordre 3 :

$$P_3(x) = a_3x^3 + a_2x^2 + a_1x + a_0 \quad (1)$$

Questions

- **Question 1 :** Combien de multiplications et d'additions sont-elles nécessaires ?
- **Question 2 :** À l'aide du schéma de Hörner, normalement connu, montrer que l'on peut réduire le nombre de ces opérations.

2 Flot de données

L'étude du flot de données (ou DFG : *dataflow graph*) permet d'établir les dépendances du calcul. On représente généralement ces dépendances à l'aide d'un graphe orienté : chaque nœud représente une opération (ici arithmétique) ou une entrée ou une sortie. Chaque arc entre deux nœuds est étiqueté par le nom de la variable qui circule du nœud source au nœud destination. Dans notre cas, le graphe est acyclique. Il indique clairement l'enchaînement des opérations, mais il s'agit encore d'une vision abstraite du calcul, indépendante de choix d'architecture matérielle. On peut toutefois noter que ce DFG peut être vu comme une première tentative de réalisation purement combinatoire du circuit (mais son chemin critique est très certainement rédhibitoire).

Questions

Question 3 : Établir le graphe de flot de données issu du schéma de Hörner.

1. Nous vous invitons à vous renseigner sur ces deux curiosités mathématiques

3 Ordonnancement. Assignment et partage des ressources

La notion d'**ordonnancement** permet de découper le DFG en différentes étapes de calculs, appelées **control-step** (CSTEP). Prosaiquement, cela revient à dessiner différentes lignes qui représentent effectivement cette découpe du DFG.

Chacune des étapes s'exécute en un seul cycle d'horloge. Selon la durée physique du cycle d'horloge, il est toutefois possible de réaliser des découpes du DFG plus ou moins "longues" : plus le cycle est long, plus il est possible d'envisager de **chaîner** un ensemble d'opérations. À la fin du cycle, on devra veiller à échantillonner les données produites : formulé autrement, on peut dire que la découpe en CSTEPs détermine où devront se trouver les *registres intermédiaires* de calcul.

En général, les opérateurs de calculs et les registres de calcul représentent la majeure partie de la surface occupée sur silicium. Il est souvent souhaitable de les utiliser avec parcimonie et notamment de les partager dès que cela est possible. On réalise l'**assignment** des opérations aux opérateurs afin d'établir où se réaliseront effectivement les calculs du DFG abstrait. Bien évidemment, à la fin d'un CSTEP tous les opérateurs utilisés sont à nouveau disponibles pour le CSTEP suivant, ce qui donne des opportunités de partage de ressources entre opérations ordonnancées sur deux CSTEPs différents. Ces informations d'assignment peuvent être également annotées sur le DFG : à côté de l'opération abstraite du DFG, on peut indiquer sur quel opérateur matériel elle s'exécutera.

Avant même la phase d'assignment, le concepteur se donne par ailleurs un objectif en terme de nombre de ressources utilisées au total : c'est la phase d'**allocation de ressources**. Il peut se donner le droit d'utiliser tel nombre de multiplieurs, tel nombre d'additionneurs, tel nombre de registres, etc. Parmi les contraintes, on retrouve d'autres paramètres variés et en particulier la fréquence attendue du circuit. On se fixe ici une fréquence d'horloge de 100 MHz.

Questions

- **Question 4** : On s'alloue 1 multiplieur et un additionneur. Le multiplieur a un temps de calcul de 7 ns et l'additionneur de 2 ns. Concernant les registres, on considère que leur temps de setup et hold sont très inférieurs, et non significatifs par rapport aux opérateurs. Proposer un ordonnancement qui vous semble judicieux. À combien de CSTEPs aboutissons-nous ?
- **Question 5** : Réaliser l'assignment des ressources et notamment des registres.

4 Microarchitecture : automate de contrôle et datapath

Dès lors qu'un ordonnancement est choisi, il est alors aisé de décrire un automate et un chemin de donnée. L'automate permet de séquencer les différentes étapes. Il réalise en outre, pour chacune d'entre elles, les actions appropriées. À l'issue de l'assignment, ces actions reviennent à contrôler les multiplexeurs d'un chemin de données (datapath) : ces multiplexeurs servent soit à échantillonner une donnée à l'entrée d'un registre, soit à router une donnée vers un opérateur matériel.

Questions

- **Question 6** : Dessiner l'automate de contrôle.
- **Question 7** : Dessiner un chemin de données.
- **Question 8** : Compléter l'automate pour indiquer les actions réalisées dans chaque état.

5 Dimensionnement des registres

Une difficulté récurrente lorsqu'on traduit un algorithme en microarchitecture réside dans le dimensionnement des signaux, c'est-à-dire la détermination du nombre de bits nécessaires à la représentation des nombres, sans perte d'information préjudiciable. On parle aussi de leur *dynamique*. Il y a perte d'information lors d'une troncature (élimination brutale de bits à un certain rang).

Questions

- **Question 9 :** Soient deux signaux s_n et $s_{n'}$ entiers codés respectivement sur n et n' bits. Déterminer le nombre de bits nécessaires à l'addition $s_n + s_{n'}$, sans perte d'information.
- **Question 10 :** Même question pour $s_n \times s_{n'}$.
- **Question 11 :** En déduire la dynamique des signaux apparaissant dans le datapath.

6 Codage VHDL

Le code VHDL de cet accélérateur est fourni sur Moodle, ainsi qu'un banc de test. La syntaxe de VHDL est complexe et nouvelle pour vous, mais vous êtes à même d'en comprendre les grandes lignes.

Questions

- **Question 12 :** Observez attentivement le code de l'accélérateur. Dessiner l'architecture générale du système. Discutez avec votre encadrant afin de bien vérifier que vous avez compris la correspondance avec vos schémas précédents.
- **Question 13 :** Observez attentivement le code du banc de test. Discutez avec votre encadrant afin de bien vérifier que vous avez compris son fonctionnement.
- **Question 14 :** La simulation ne sera pas réalisée en séance, mais une vidéo sera mise à disposition.