

TD 7

Architecture PC/PO

Ex. 1 : Bresenham

Le tracé de lignes, de cercles, d'ellipse, etc, est une opération réalisée très communément sur les cartes graphiques. Des algorithmes spécifiques ont été développés pour faire ces tracés sur des grilles de pixels (par ex. un écran) en utilisant uniquement des nombres entiers et sans recourir à la division (pour la ligne) ou la racine carrée ou les fonctions trigonométriques (pour les cercles et ellipses). On se propose d'utiliser un algorithme dû à Jack Bresenham¹ qui permet de tracer un huitième de cercle pour en faire la conception PC/PO.

L'algorithme est le suivant :

```
// r est le rayon, disponible au début de l'algorithme
1 x := 0;
2 y := r;
3 m := 5 - 4 × r;
4 while x ≤ y do
    // out indique que les valeurs (x,y) sont disponibles, il n'y a pas
    // d'opérateur associé
5    out(x,y);
6    if m > 0 then
7        y := y - 1;
8        m := m - 8 × y;
9    end if
10   m := m + 8 × x + 12;
11   x := x + 1;
12 end while
```

On va utiliser la méthode présentée en cours et mise en œuvre en TD pour proposer une implantation de cet algorithme sous la forme PC/PO.

Question 1 Définir l'ensemble des opérations à réaliser ; en-déduire le type des unités fonctionnelles (registres ou opérateurs). On cherchera à utiliser les opérateurs *les plus simples* (c'est-à-dire qui implantent une opération arithmétique ou logique et une seule) pour chaque opération. On rappelle que les soustracteurs fournissent (possiblement entre autres) les informations suivantes : z qui vaut 1 si le résultat est nul, 0 sinon et s qui vaut 1 si le résultat est strictement négatif, 0 sinon.

Question 2 Quelles opérations peut-on réaliser en parallèle ?

Réécrire le programme en utilisant 1) la notion d'affectation concurrente présentée en cours et 2) en remplaçant les opérations (y compris les comparaisons) par les opérations des opérateurs « simples » déterminés précédemment. Rappel : $(\alpha, \beta) \leftarrow (3, 1)$ affecte simultanément 3 dans α et 1 dans β .

Question 3 Donner le nombre d'opérateurs de chaque type nécessaire pour exécuter une ligne de l'algorithme en 1 cycle et proposer une partie opérative interconnectant les unités

1. "A linear algorithm for incremental digital display of circular arcs", Jack Bresenham, *Communication of the ACM*, Feb 1977, vol. 20, n. 2.

fonctionnelles. Indiquer clairement les signaux de commandes (sélecteurs de multiplexeurs, signaux de chargement de registres, etc) et les comptes-rendus.

Question 4 Proposer une partie contrôle qui pilote cette partie opérative. On fait l'hypothèse que l'entrée r est magiquement disponible lorsqu'on en a besoin, et l'opération *out* se fait en un cycle (on rend les valeurs de x et y disponibles vers l'extérieur). Spécifier les opérations de transfert registre à registre (RTL) dans chaque état. Donner la fonction de sortie sous la forme d'un tableau.

Question 5 On souhaite supprimer le **if**. Pour se faire, on aimerait rendre indépendantes les affectations des lignes 7 et 8 et se ramener à une unique affectation parallèle dans le corps du **if**. Comment y arriver, i.e. comment supprimer la dépendance de données entre les lignes 7 et 8 ? De même, comment intégrer les affectations des lignes 10 et 11 au **if** ?

Question 6 Modifier l'algorithme en utilisant la notion d'affectation conditionnelle pour faire disparaître le **if**. Pour mémoire : $\nu \leftarrow \gamma ? \alpha : \beta$ affecte α dans ν si $\gamma = 1$, β sinon.

Pour aller plus loin...

Question 7 Proposer une PO, puis une PC correspondant à la question précédente.

Pour aller plus loin...

Question 8 En repartant de la forme initiale de l'algorithme, proposer une PO n'utilisant qu'un additionneur/soustracteur. Donner ensuite la PC correspondante.