

Les problèmes d'ordonnancement cyclique dans la synthèse des circuits numériques

Imed Eddine Bennour et El Mostapha Aboulhamid

- $O = \{o_i\}$ est l'ensemble de noeuds du graphe; les noeuds représentent les opérations que le circuit doit réaliser. Ces opérations sont dites génériques étant donné qu'elles peuvent être exécutées plusieurs fois sur des données différentes. Une itération correspond à une exécution de l'ensemble des opérations génériques.
- $A \subset O \times O$ est l'ensemble des arcs du graphe reliant toute paire d'opérations ayant une contrainte de précédence entre elles. On note par a_{ij} l'arc de o_i vers o_j .
- $H : A \rightarrow \mathbb{N}$ (l'ensemble des entiers naturels) appelée fonction de hauteur. La valeur $H(a_{ij})$ indique que le résultat de l'opération o_i d'une itération n quelconque est utilisé par l'opération o_j de l'itération $(n + H(a_{ij}))$. On dit que l'opération o_i est liée à l'opération o_j par une contrainte de précédence intra-itération (resp. par une contrainte de précédence inter-itérations) si $H(a_{ij}) = 0$ (resp. si $H(a_{ij}) > 0$).
- $D : A \rightarrow \mathbb{N}^*$ appelée fonction de durée. La valeur $D(a_{ij})$ est égale à la durée d'exécution de l'opération o_i . Notons que la fonction de durée aurait pu être associée aux noeuds du graphes, mais par souci de conformité et de généralité nous l'avons associée aux arcs.

De $n = 4$ à ∞ faire

Debut

$o_0: a[n] \leftarrow E[n] \times d[n-2];$

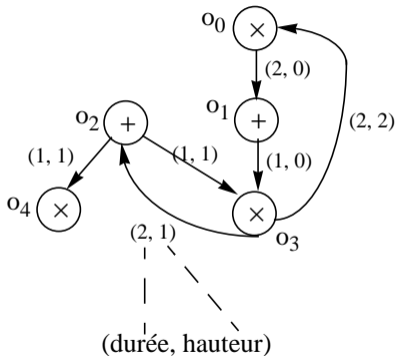
$o_1: b[n] \leftarrow c_1 + a[n];$

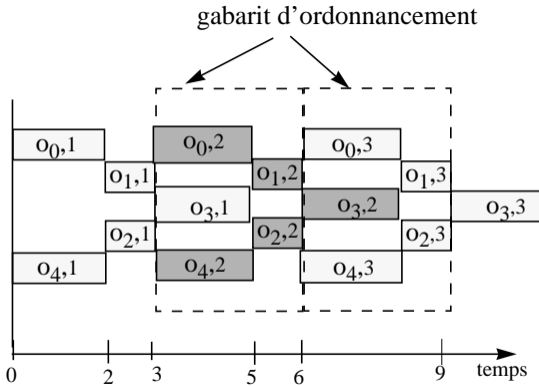
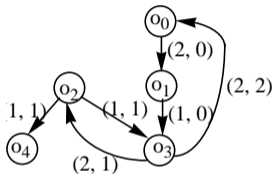
$o_2: c[n] \leftarrow c_2 + d[n-1];$

$o_3: d[n] \leftarrow b[n] \times c[n-1];$

$o_4: S[n] \leftarrow c_3 \times c[n-1];$

Fin





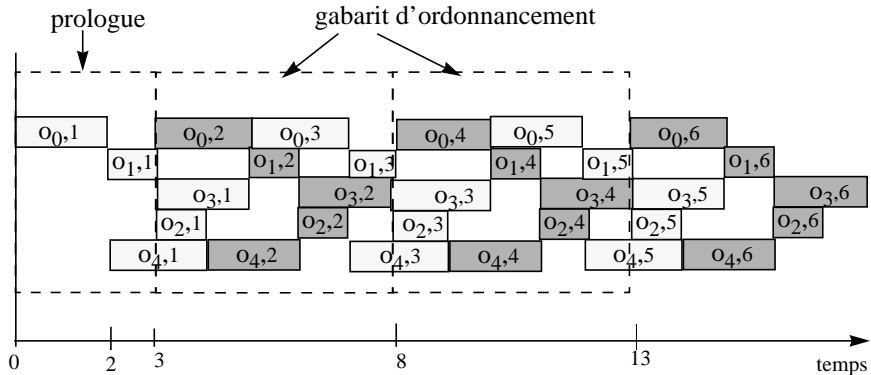


Figure 5. Un ordonnancement K -périodique de fréquence $2/5$.

$$\forall a_{ij} \in A, \forall n \in \mathbb{N}^* . \tau \langle o_j, H(a_{ij}) + n \rangle \geq \tau \langle o_i, n \rangle + D(a_{ij}) \quad [1]$$

(2) Contraintes de ressources: il existe une allocation des unités fonctionnelles aux opérations sans qu'il y ait un conflit de ressources.

La fréquence d'un ordonnancement cyclique τ est égale à

$$\lim_{n \rightarrow \infty} \frac{n}{\max_{o_i} \{ \tau \langle o_i, n \rangle + D(a_{ij}) \}}$$

Le problème d'ordonnancement cyclique (POC). Etant donné un graphe de précedence, une affectation des opérations génériques aux types d'unité fonctionnelles et un nombre d'instances de chaque type d'unité, le POC est de déterminer un ordonnancement réalisable de fréquence maximale.

Ordonnancement K-périodique. Un ordonnancement τ est dit K-périodique s'il existe deux entiers n_0 , K et un rationnel positif P tel que:

$$\forall n \geq n_0, \forall o_i \in O . \tau \langle o_i, n + K \rangle = \tau \langle o_i, n \rangle + P$$

K et P sont appelés, respectivement, la périodicité et la période de τ . La fréquence de τ est égale à $\frac{K}{P}$.

Ordonnancement périodique. Un ordonnancement τ est périodique s'il existe un rationnel positif P tel que

$$\forall n \geq 2, \forall o_i \in O. \quad \tau \langle o_i, n \rangle = \tau \langle o_i, n - 1 \rangle + P \quad [2]$$

La fréquence de τ est égale à $\frac{1}{P}$. Un ordonnancement périodique est un ordonnancement K -périodique de périodicité égale à un.

Exemple: Le graphe de précédence de la figure 3 contient deux circuits

$$C_1 = o_0 \rightarrow o_1 \rightarrow o_3 \rightarrow o_0, C_2 = o_2 \rightarrow o_3 \rightarrow o_2$$

La valeur de la fréquence critique est égale à $\min\left(\frac{2}{5}, \frac{2}{3}\right) = \frac{2}{5}$. Le circuit C_1 est critique.

La fréquence maximale due aux dépendances entre les opérations est appelée *fréquence critique* et notée F_{crit} . La valeur de F_{crit} est donnée par la formule suivante [CHR 83]:

$$F_{crit} = \min_{C_k} \left\{ \frac{\sum_{a_{ij} \in C_k} H(a_{ij})}{\sum_{a_{ij} \in C_k} D(a_{ij})} \right\} \quad [4]$$

Soient $G = (O, A, D, H)$ un graphe de précédence et F un rationnel positif. F est une fréquence réalisable de G si le graphe $\hat{G} = (O, A, D, \hat{H})$ où

$$\hat{H}(a_{ij}) = D(a_{ij}) - \frac{1}{F} \cdot H(a_{ij})$$

ne contient pas de circuit de hauteur positive.

$$\forall n \geq 2, \forall o_i \in O. \tau\langle o_i, n \rangle = \tau\langle o_i, n-1 \rangle + \frac{1}{F_{crit}} \quad [6]$$

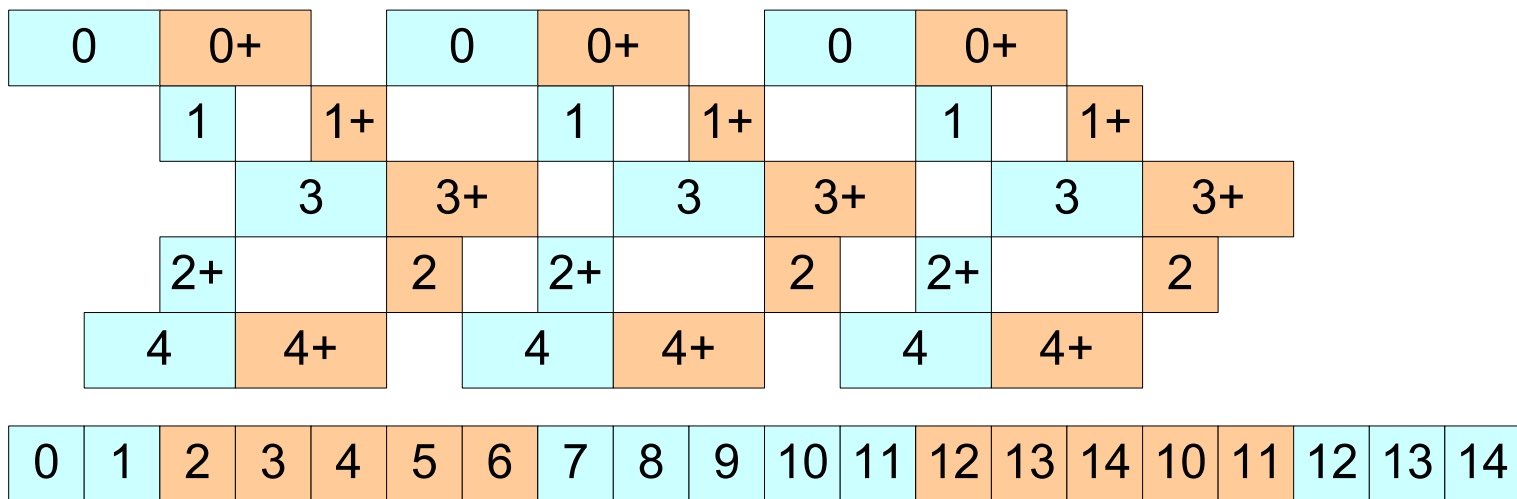
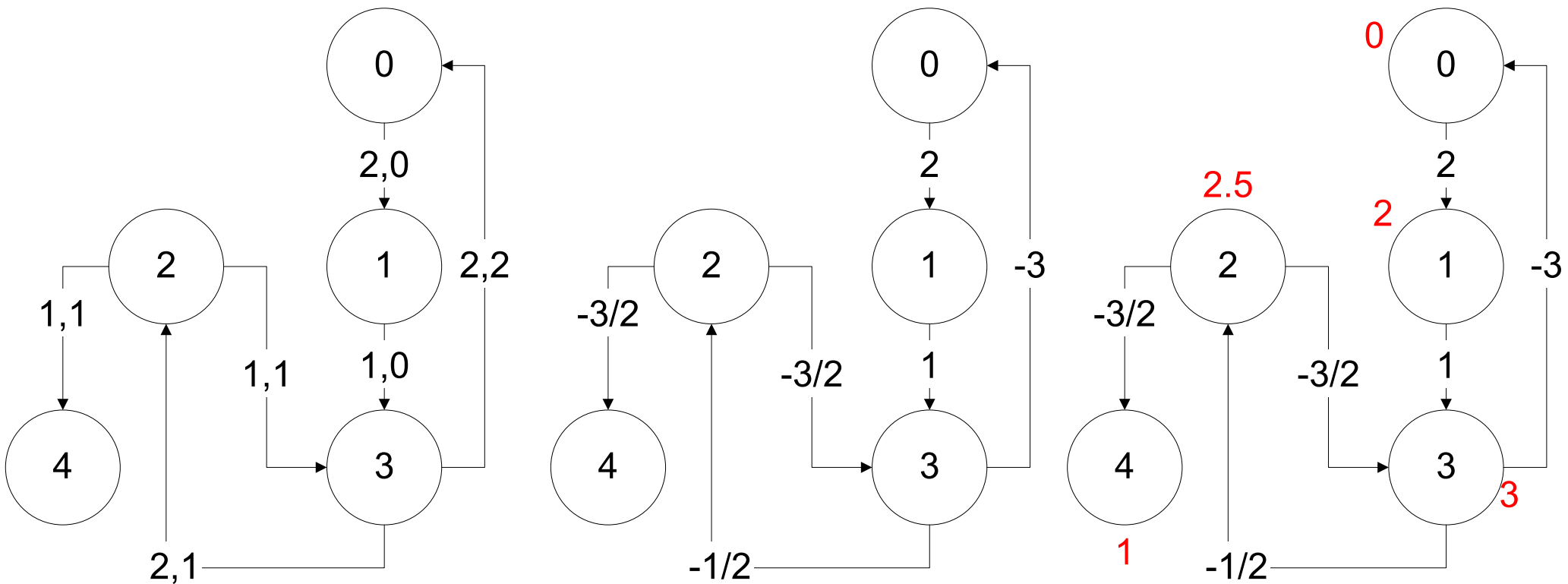
Une fois la valeur de F_{crit} est calculée, les valeurs $\tau\langle o_i, 1 \rangle$ peuvent être déduites facilement. Ces valeurs doivent satisfaire la contrainte de l'équation [1]. En combinant les équations [1] et [6], on obtient:

$$\forall a_{ij} \in A. \tau\langle o_i, 1 \rangle - \tau\langle o_j, 1 \rangle \leq \frac{H(a_{ij})}{F_{crit}} - D(a_{ij})$$

Si τ est un ordonnancement périodique, alors τ^ défini comme suit:*

$$\forall o_i \in O, \forall n \geq 1. \quad \tau^* \langle o_i, n \rangle = \lfloor \tau \langle o_i, n \rangle \rfloor$$

est un ordonnancement K -périodique et de même fréquence que τ



5.4. La resynchronisation et son utilisation dans l'ordonnancement

5.4.1. Définition de la resynchronisation

La technique de resynchronisation a été développée initialement pour réduire la période d'horloge dans les circuits synchrones [LEI 91]. Elle consiste à réorganiser le graphe de précédence en modifiant les hauteurs des arcs.

Soit $G = (O, A, D, H)$ un graphe de précédence et soit s une fonction de O dans Z (l'ensemble des entiers) dite fonction de resynchronisation. On appelle graphe resynchronisé de G par rapport à s le graphe $G_s = (O, A, D, H_s)$ où

$$\forall a_{ij} \in A \cdot H_s(a_{ij}) = H(a_{ij}) + s(o_j) - s(o_i)$$

Exemple: La figure 7 montre le graphe obtenu après la resynchronisation:
 $s(o_0) = s(o_1) = s(o_4) = -2$; $s(o_2) = s(o_3) = -1$.

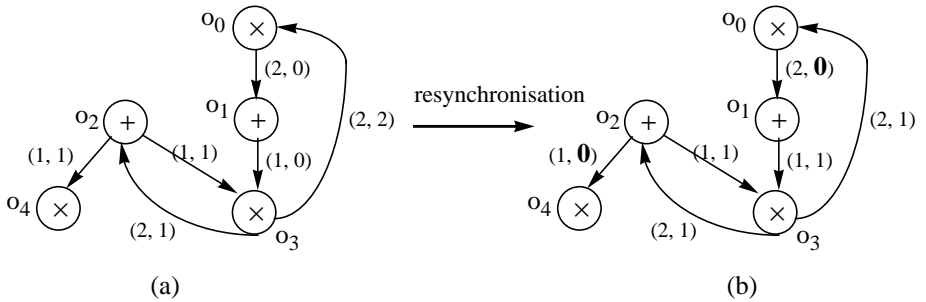


Figure 7. Graphe de précédence après une resynchronisation.

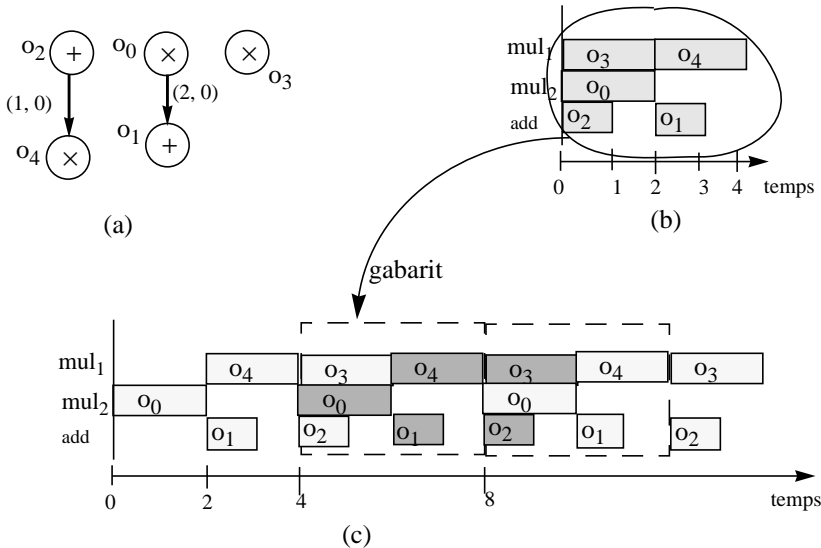


Figure 8. (a) Le graphe resynchronisé de la figure 7.b avec seulement les dépendances intra-itération (b) Un ordonnancement non cyclique de ce graphe (c) L'ordonnancement cyclique dérivé à partir de l'ordonnancement non cyclique

5.4.3. Une heuristique d'ordonnancement utilisant la resynchronisation

Dans ce qui suit on présente une heuristique illustrant l'utilisation de la resynchronisation dans un algorithme d'ordonnancement. L'heuristique consiste à effectuer d'une façon répétitive la resynchronisation suivante

$$\forall o_i \in O. s(o_i) = \begin{cases} -1 & \text{si } \forall a_{ji}, H(a_{ji}) > 0 \\ 0, & \text{autrement} \end{cases}$$

suivie d'un ordonnancement, jusqu'à l'obtention d'une fréquence satisfaisante. L'heuristique est composée de trois étapes:

- (a) Appliquer sur le graphe de précédence courant G la resynchronisation s . Le graphe résultant est G_s .
- (b) Trouver un ordonnancement des opérations de G_s qui satisfait uniquement les contraintes de précédence intra-itération et les contraintes de ressources. Le critère d'optimisation est la latence de l'ordonnancement. À ce niveau, on peut utiliser les algorithmes d'ordonnancement existants pour résoudre les problèmes d'ordonnancement non cycliques.
Rappelons que l'ordonnancement obtenu est le gabarit d'un ordonnancement cyclique de G .
- (c) Si la latence obtenue à l'étape (b) n'est pas satisfaisante, reprendre l'étape (a) avec $G \leftarrow G_s$.

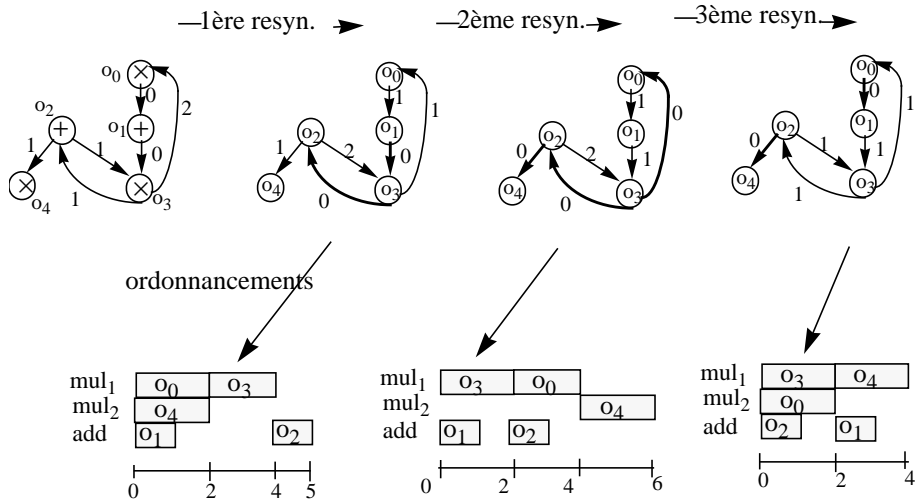


Figure 9. Illustration de l'heuristique d'ordonnancement (nous avons omis les durées des arcs).

1ère resyn. $s(o_0) = s(o_2) = s(o_4) = -1$; $s(o_1) = s(o_3) = 0$;

2ème resyn. $s(o_0) = s(o_1) = s(o_4) = -1$; $s(o_2) = s(o_3) = 0$;

3ème resyn. $s(o_1) = s(o_3) = -1$; $s(o_0) = s(o_2) = s(o_4) = 0$;