

Devoir Surveillé no 2

Durée : 1h20 — Sans document

1 Circuit combinatoire simple

On souhaite construire un circuit combinatoire à trois entrées (de 1 bit chacune) et une sortie (de 1 bit) réalisant la fonction « majorité » (notée `maj` sur le schéma). Le principe est que la sortie vaut 1 si une *majorité* d'entrées vaut 1 (c'est-à-dire deux d'entre elles ou les trois), et vaut 0 sinon.



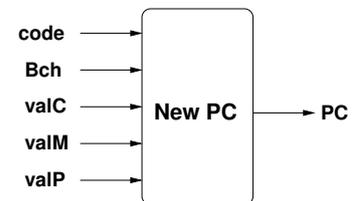
Q1 (échauffement) Donnez la table de vérité de la fonction `maj`.

Q2 Proposez une implémentation du circuit `maj` à l'aide de portes logiques simples.

Q3 En vous appuyant sur la table de vérité de `maj` (cf question 1), expliquez¹ comment vous pourriez utiliser ce circuit pour calculer la retenue au sein d'un additionneur (sur 1 bit) à 3 entrées.

2 Version séquentielle du simulateur Y86

La figure ci-contre montre les entrées du circuit combinatoire « New PC » chargé de calculer l'adresse de la prochaine instruction que le processeur devra charger (l'entrée `Bch` désigne la valeur « *Branch Flag* »). Le code HCL « partiel » décrivant ce circuit est donné sous le schéma.



Q1 Expliquez le code HCL fourni.

Q2 Complétez ce code pour traiter les quelques cas non gérés actuellement. Pour chaque ligne ajoutée, expliquez précisément le choix de la valeur résultat.

```
PC = [
  code == Jxx && Bch : valC ;
  ...
  1 : valP ;
]
```

3 Version pipelinée du simulateur Y86

On considère la version pipelinée du processeur Y86 à 5 étages (*Fetch*, *Decode*, *Execute*, *Memory*, *Write Back*) telle que vue en cours.

On souhaite examiner le fonctionnement du processeur lors de l'exécution de la séquence d'instructions ci-contre.

```
irmovl $1, %eax
mrmovl variable, %edx
addl %eax, %edx
rmmovl %edx, variable
```

Q1 En supposant une version naïve du processeur se limitant à l'insertion automatique de *bulles* pour conserver une exécution correcte, dessinez un chronogramme montrant la progression des instructions à l'intérieur du pipeline pour chaque cycle.

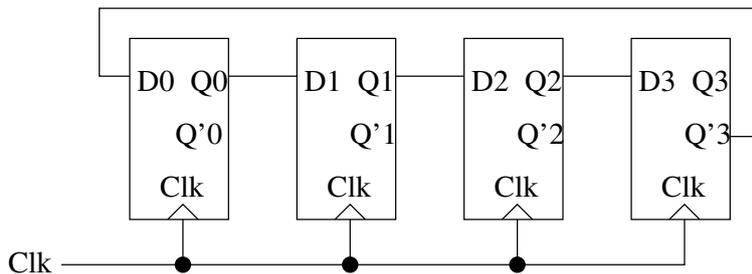
Q2 En supposant maintenant une version du processeur capable de propager des valeurs des étages *E*, *M* ou *WB* vers l'étage *E*, redessinez le chronogramme. Y a-t-il encore apparition de *bulles*? Expliquez.

Q3 Cela change-t-il quelque chose d'inverser l'ordre des deux premières instructions dans le programme d'origine? Expliquez.

¹NB : On ne demande pas de dessiner l'additionneur.

4 Compteur de Johnson

Le compteur de Johnson ci-dessous est composé de quatre bascules D (rappel : $Q'_i = \bar{Q}_i$).



On suppose qu'à l'initialisation on a l'état $Q_0 = Q_1 = Q_2 = Q_3 = 0$.

Q1 Dresser une table des états Q_i en fonction du temps (une ligne par période d'horloge). Combien de situations différentes peuvent survenir (noté N) ?

Q2 On souhaite à partir de ce circuit construire un circuit possédant N sorties étant mises à 1 tour à tour de manière cyclique. Écrivez les expressions de $s_1 \dots s_N$ en fonction de $Q_0, Q'_0, \dots, Q_3, Q'_3$.

Q3 On rappelle qu'une bascule D peut être implémenté avec 6 portes simples. Combien de portes nécessite le circuit établi à la Q2 ? De manière générale, combien de portes sont nécessaires (en fonction du N voulu) ?

Q4 Proposer une autre solution utilisant huit bascules D et supposant un état initial approprié. Combien de portes sont nécessaires ? De manière générale, combien de portes sont nécessaires ?

Q5 (difficile, complémentaire) Proposer une autre solution utilisant trois bascules D et quelques circuits supplémentaires. Estimez un ordre de grandeur du nombre de portes nécessaire de manière générale.