

Examen d'architecture des ordinateurs

1IN, 23 janvier 2012

1. Algèbre de Boole (2 pts)

On veut réaliser un circuit qui indique qu'un nombre non signé de 2 bits a_1, a_0 est strictement supérieur à un nombre non signé de 2 bits b_1, b_0 , d'interface:

```
module superieur(a1, a0, b1, b0 : sup)
```

Par exemple $a_1, a_0=10$ est strictement supérieur à $b_1, b_0=01$

- dessiner la table de vérité du circuit
- simplifier le circuit à l'aide d'une table de Karnaugh
- écrire le circuit sous forme d'un module SHDL

2. Circuit séquentiel - compteur (4 pts)

Ecrire un module SHDL qui réalise un compteur sur 4 bits (a, b, c, d), d'interface :

```
module count4(clk, rst : a, b, c, d)
```

Modifier ce compteur pour qu'il puisse charger une valeur de 4 bits (a_i, b_i, c_i, d_i) lors de l'arrivée d'une commande load. Son interface sera :

```
module count4i(clk, rst, load, ai, bi, ci, di : a, b, c, d)
```

Lorsque $load=0$, le compteur compte normalement. Lorsque $load=1$, au front d'horloge, le compteur se charge avec la valeur (a_i, b_i, c_i, d_i) et continuera ensuite de compter à partir de cette valeur. On pourra décrire le compteur modifié en SHDL ou sous forme d'un schéma, au choix.

3. Circuit séquentiel – graphe de Mealy (3 pts)

On souhaite réaliser un distributeur de canettes sous forme d'une machine séquentielle synchrone de type MEALY. Une horloge gouverne les changements d'états ; elle est supposée suffisamment rapide pour ne manquer aucun changement des entrées. La canette coûte 2€ ; on introduit dans la machine des pièces de 0,5€, 1€ ou 2€ et la machine délivre la canette et la monnaie dès que le montant de 2€ est atteint (pas de bouton spécial à appuyer). On suppose que la machine dispose de toute la monnaie nécessaire. Après chaque délivrance de canette et de monnaie, la machine revient dans l'état initial.

Les entrées e_1 et e_2 codent l'entrée des pièces. Lorsqu'on introduit une pièce, les codes suivants sont présentés en entrée durant un cycle d'horloge :

e1,e2	
00	rien
01	0,5€
10	1€
11	2€

Les sorties s1,s2,s3 codent la sortie de la canette et le rendu de la monnaie:

s1,s2,s3	
000	rien
100	canette seule
101	canette + 0,5€
110	canette + 1€
111	canette + 1,5€

Dessinez un graphe de MEALY qui traduit le fonctionnement de cette machine. On ne dessinera que le graphe sans chercher à concevoir la machine.

4. Code machine d'une instruction (1 pt)

Donner le code machine de l'instruction suivante, en binaire et en hexadécimal :

```
subcc    %r2, 10, %r5
```

5. Microcommandes associées à une instruction (2 pts)

Compléter le tableau suivant avec la suite des microcommandes qui sont envoyées par le séquenceur lors de l'exécution de l'instruction :

```
subcc    %r2, 10, %r5
```

(On pourra répondre directement sur le sujet ou recopier le tableau sur la copie)

areg	breg	dreg	cmd_ual	oe_num	write	commentaire

6. Programmation de CRAPS (4 pts)

Écrire un programme en assembleur CRAPS qui affiche sur les afficheurs 7-segments la valeur d'un compteur, et qui fonctionne de la façon suivante :

- à chaque levée du switch le plus à droite, la valeur du compteur est incrémentée
- à chaque levée du deuxième switch en partant de la droite, le compteur est remis à zéro

7. CRAPS – ajout d'instructions (4 pts)

Dans la machine CRAPS construite en TP, les instructions `push %ri` et `pop %ri` étaient des instructions synthétiques, équivalentes à :

```
push %r1 == sub %r29, 1, %r29; st %ri, [%r29]
pop %ri  == ld [%r29], %r1; add %r29, 1, %r29
```

Pour améliorer les performances, on désire implémenter `push` et `pop` sous forme de 2 véritables instructions :

- proposer de nouveaux codes en langage machine pour ces 2 instructions
- indiquer en détail toutes les modifications à faire dans chacun des modules de CRAPS pour les implémenter
- quel est le gain de temps d'exécution obtenu par rapport aux instructions synthétiques ?