

# Examen d'architecture des ordinateurs

---

1IN, 21 janvier 2013 – 2h

## 1. Arithmétique (2 pts)

Poser et effectuer les opérations suivantes, en arithmétique signée sur 8 bits :

$$(+117) + (-78) \quad ; \quad (-45) + (-93)$$

Indiquer dans chaque cas s'il y a débordement et expliquer pourquoi.

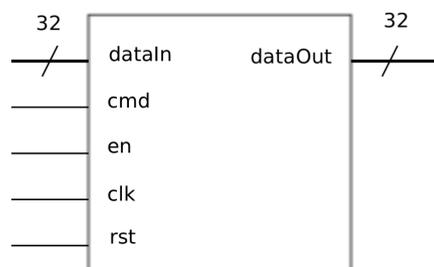
## 2. Logique combinatoire (3 pts)

Dans une banque, 4 employés ont accès au coffre-fort (CF). Ces 4 employés A, B, C et D possèdent chacun une clé différente mais aucun ne peut ouvrir le coffre-fort tout seul.

- L'employé A a besoin que l'employé B soit présent.
  - Les employés B et C ont besoin qu'au moins deux autres employés soient présents.
  - L'employé D ne peut ouvrir qu'en présence de tous les autres employés.
1. Dresser la table de vérité de la sortie CF (en fonction des entrées A,B,C,D), qui vaut 1 lorsque le coffre-fort est ouvert et 0 sinon
  2. Trouver une expression de CF grâce à une table de Karnaugh
  3. Écrire le module SHDL `coffrefort(A, B, C, D : CF)` correspondant

## 3. Circuit séquentiel – synthèse fonctionnelle (4 pts)

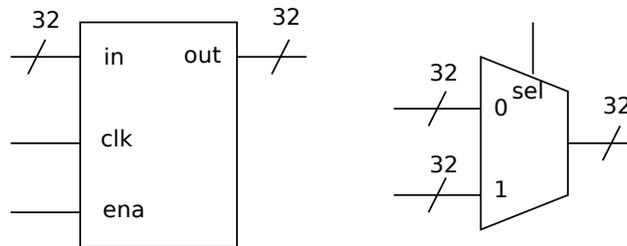
Concevoir une pile matérielle de 4 mots de 32 bits, avec l'interface suivant :



- lorsque `en=0`, quelle que soit la valeur de `cmd`, l'état de la pile ne change pas au front d'horloge
- Lorsque `cmd=1` (push) et `en=1`, la donnée `dataIn` est empilée au front d'horloge
- `dataOut` fournit toujours la dernière valeur empilée

- lorsque cmd=0 (pop) et en=1, le mot mémoire au sommet de la pile est retiré
- on ne gèrera pas les situations de pop lorsque la pile est vide ou de push lorsque la pile est pleine

Dessiner le schéma du module. On pourra utiliser des registres 32 bits avec enable et des multiplexeurs 32 bits :

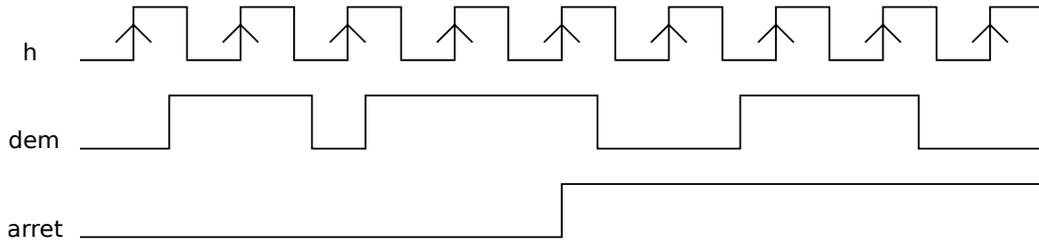


#### 4. Circuit séquentiel – graphe d'états (4 pt)

On veut concevoir un circuit séquentiel qui déclenche un signal d'arrêt lorsqu'on active une entrée de demande d'arrêt durant une seconde. L'interface du circuit sera :

```
module circuit_arret(rst, h, dem : arret)
```

On utilise une horloge à 1Hz, de sorte qu'il suffit que la commande d'arrêt `dem` soit présente durant deux fronts (montants) d'horloge consécutifs pour déclencher l'arrêt. Une fois le signal `arret` activé, il ne peut plus être désactivé (sauf lors d'un reset).



Concevoir cette réalisation sous forme d'un circuit séquentiel synchrone pur de type MOORE. Fournir le graphe d'états et les équations SHDL du circuit.

#### 5. CRAPS : programmation (3 pts)

Écrire le sous-programme qui calcule le maximum d'un tableau d'entiers signés (relatifs) dont l'adresse et la taille sont passées en paramètre.

- indiquer clairement vos choix en terme de passage des paramètres et du résultat
- quel changement effectuer si le tableau est constitué d'entiers naturels (positifs) ?

#### 6. CRAPS : ajout d'une instruction (5 pts)

On souhaite ajouter au processeur CRAPS un autre groupe d'instructions du type :

*opération-arithmétique*      *rs, [rad1+rad2]*

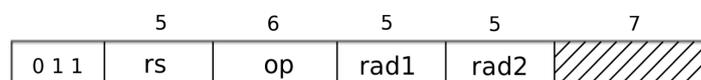
où l'opération arithmétique est réalisée entre le registre `rs` et le mot mémoire dont l'adresse est égale à `rad1+rad2`, et stocke le résultat dans le même mot mémoire.

On pourra écrire en assembleur par exemple :

```
subcc    %r1, [%r5+%r7]
```

Cette instruction lira le mot mémoire de contenu `M` situé à l'adresse `A=%r5+%r7`, effectuera l'opération `%r1 - M`, et stockera le résultat en mémoire à l'adresse `A`.

Ce groupe d'instructions a pour format :



- Dans quel(s) module(s) effectuer ces modifications ? Les détailler

- Quel gain réalise-t-on (en %) par rapport à la même opération réalisée avec des instructions CRAPS classiques ?