

---

# Travaux pratiques d'architecture des ordinateurs

---

Processeur CRAPS : la micromachine

## 1. Etude de la micromachine

---

Expliquer à partir du schéma général de la micromachine :

- pourquoi un court-circuit est impossible sur le bus D
- comment lire et écrire en RAM
- comment effectuer un calcul sur 2 registres, et stocker le résultat dans un troisième
- comment agir sur les leds, les afficheurs 7 segments, le module PWM
- comment lire l'état des switches
- comment utiliser les signaux du moniteur

## 2. Construction de la micromachine

---

Assembler tous les morceaux qui composent la micromachine. En plus de l'UAL et des registres, on utilisera les modules prédéfinis ou de librairie suivants :

- une mémoire de 512 mots de 32 bits :

```
rams_asyn_read512x32(clk, write, addr[8..0], din[31..0] :  
    dout[31..0])
```

- un registre 8 bits pour les leds et les switches, d'interface :

```
reg8(rst, clk, en, d[7..0] : q[7..0])
```

- un module d'affichage 7 segments, d'interface :

```
segs7(rst, clk, ssgClk, en, selec, data[15..0] :  
    anodes[3..0], ssegs[7..0])
```

Lors d'un front d'horloge et si  $en = 1$  :

- si  $selec = 1$  alors les afficheurs sont alimentés (c'est à dire s'allument) ssi la donnée écrite  $data$  est non nulle
- si  $selec = 0$  alors les 16 bits de poids faible de  $data$  sont projetés sur les afficheurs (et visibles si ceux ci sont alimentés)

Du point de vue du programmeur qui souhaite se servir de ces afficheurs, il faut commencer en début de programme par activer les afficheurs (qui sont initialement éteints) en écrivant une valeur

non nulle à l'adresse 0xA0000001. Ensuite, à chaque écriture à l'adresse 0xA0000000, les 16 bits de la donnée écrite seront visualisés sur les afficheurs.

- un module PWM, d'interface :

```
pwm_module(rst, clk, pwmClk, en, selec, data[15..0] : out)
```

Lors d'un front d'horloge et si en = 1 :

- si selec = 0, on écrit dans le registre P du module PWM
- si selec = 1, on écrit dans le registre N du module PWM

Le module génère sur sa sortie out un signal rectangulaire périodique paramétré par les valeurs P et N, qui seront décrites ultérieurement.

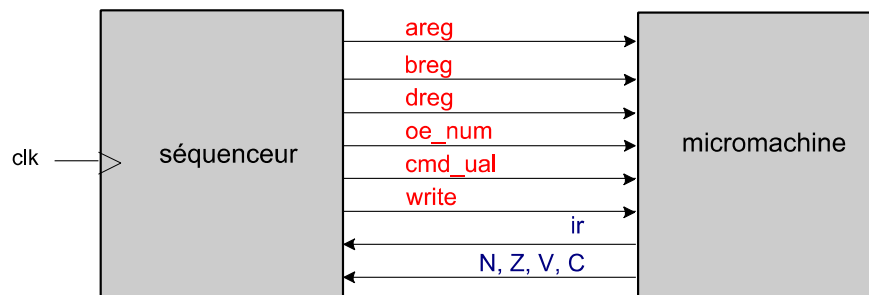
L'interface générale sera :

```
module micromachine(rst, clk, areg[4..0], breg[4..0], dreg[4..0],
  cmd_ual[5..0], oe_num[1..0], write, monitor[31..0],
  switch[7..0], swClk, ssgClk, pwmClk :
  dbus[31..0], N, Z, V, C, ir[31..0], break, leds[7..0],
  ssegs[7..0], anodes[3..0], pwm_out)
```

Synthétiser le circuit avec les entrées-sorties à distance et tester la lecture et l'écriture des registres, la lecture et l'écriture en mémoire RAM, la lecture des switches, l'affichage sur les leds.

### 3. Usage de la micromachine

Les microcommandes de la machine seront envoyées automatiquement par le séquenceur, au vu de l'instruction à exécuter contenue dans %ir et de la valeur des flags N,Z,V,C



### 3. Test de la micromachine

En utilisant les entrées monitor, initialisez le mot mémoire d'adresse 0 de la RAM avec la valeur 0x8400A005, c'est à dire le code machine de l'instruction :

```
0000 0000    add %r2, 5, %r2    8400 A005
```

Appliquer manuellement les microcommandes nécessaires à l'exécution de cette instruction ; remplir la feuille fournie.