

# Travaux pratiques d'architecture des ordinateurs

Processeur CRAPS : le séquenceur

## 1. Format des instructions de CRAPS

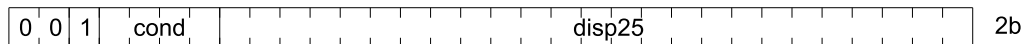
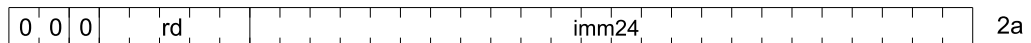
Les instructions de CRAPS ont le format binaire suivant :

Format 1: instructions arithmétiques, accès mémoire



t = 0 : instruction arithmétique, t = 1 : accès mémoire

Format 2 : sethi et branchements



Les champs *op* et *cond* ont les significations suivantes :

op (t=0)	Instr.	op (t=1)	Instr.	cond	branchement
000000	add	000000	ld	1000	ba
010000	addcc	000100	st	0001	be
000100	sub			1001	bne
010100	subcc			0101	bcs
011010	umulcc			1101	bcc
000001	and			1110	bpos
010001	andcc			0110	bneg
000010	or			0111	bvs
010010	orcc			1111	bvc
000011	xor			1010	bg
010011	xorcc			0010	ble
001101	slr			1011	bge
001110	sll			0011	bl
				1100	bgu
				0100	bleu

## 2. Langage de description d'automate d'états

On utilisera un langage spécifique pour décrire le séquenceur. Les fichiers auront le suffixe .net, et on utilisera le traducteur pour les transformer en shdl, puis en vhdl. L'exemple suivant illustre la syntaxe pour la description de la commande d'une chaudière sous forme d'une machine de MEALY :

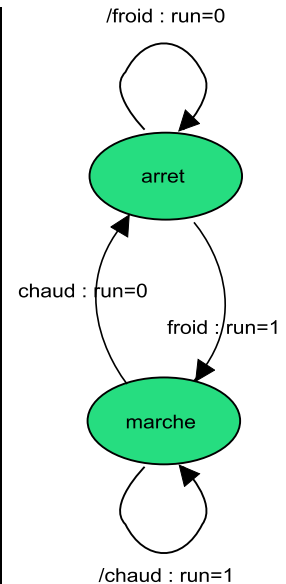
```
module: chaudiere
reset: rst
clock: clk
inputs: chaud, froid
outputs: run
added_outputs:

arret -> arret
// la chaudière reste arrêtée tant qu'il ne fait pas froid
when /froid
outputs run = 0;

arret -> marche
// la chaudière se met en marche quand il fait froid
when froid
outputs run = 1;

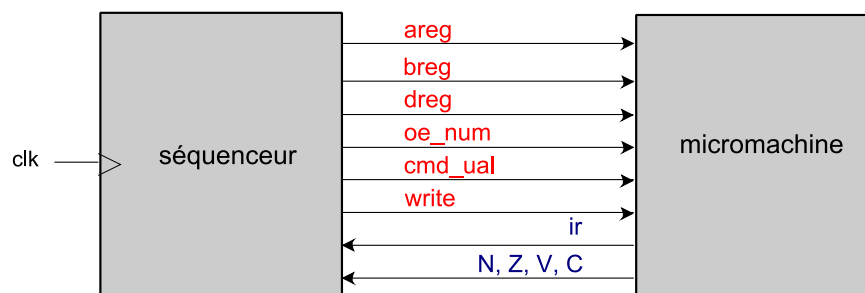
marche -> marche
// la chaudière reste en marche tant qu'il ne fait pas
chaud
when /chaud
outputs run = 1;

marche -> arret
// la chaudière s'arrête quand il fait chaud
when chaud
outputs run = 0;
```



## 3. Le module séquenceur, rôle et interface

Le séquenceur envoie automatiquement les microcommandes à la micromachine, au vu de l'instruction à exécuter contenue dans %ir et de la valeur des flags N,Z,V,C



Le séquenceur a l'interface suivant :

```
sequencer(rst, clk, mon_req, mon_cmd[3..0], monitor[37..0],
break, run, ir[31..0], N, Z, V, C :
write, oe_num[1..0], areg[4..0], breg[4..0], dreg[4..0],
cmd_ual[5..0], state[4..0], mon_ack)
```

Les signaux `mon_req`, `mon_cmd`, `monitor`, `break` et `run` servent au dialogue avec le moniteur. Eux mis à part, les entrées du module sont :

- `ir[31..0]` valeur du registre `ir` (%r15), nécessaire pour le décodage de l'instruction à séquencer.

- `N, Z, V, C`, valeurs des flags, nécessaires pour le séquencement des branchements conditionnels.

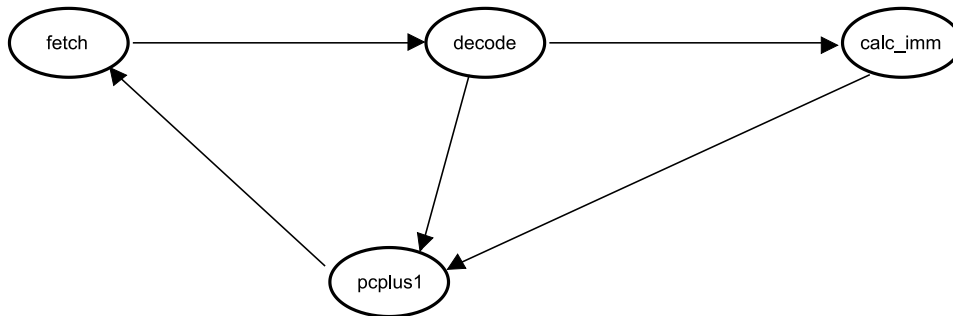
Les sorties sont les commandes de la micromachine, telles qu'elles ont été définies auparavant, ainsi que `state[4..0]` qui montre l'état et un signal de contrôle `mon_ack` pour le moniteur.

Un fichier `sequencer.net` est fourni, qui contient un squelette de séquenceur à compléter. Il s'agit d'une machine d'états de type MEALY ; la partie de la machine d'états qui gère la communication avec le moniteur est déjà écrite.

#### 4. Séquencement des instructions de calcul

---

Compléter et tester le séquenceur, pour qu'il gère le sous-graphe suivant :



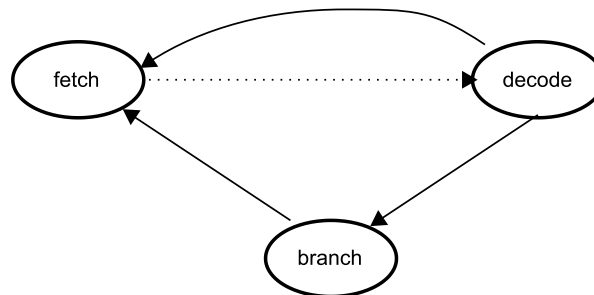
Tester le module `sequencer` avec les entrées-sorties à distance, en vérifiant que les instructions des formats 1a et 1b engendrent les bonnes séquences de microcommandes

Assembler ensuite la machine craps complète (voir module `craps.shd`) et vérifier que les instructions de calcul fonctionnent correctement.

#### 5. Séquencement des instructions de branchement

---

Compléter et tester le séquenceur, pour qu'il gère le sous-graphe suivant :



Tester le module `sequencer` avec les entrées-sorties à distance, en vérifiant qu'il engendre les bonnes séquences de microcommandes, dans le cas où le branchement est pris et dans le cas où il n'est pas pris.

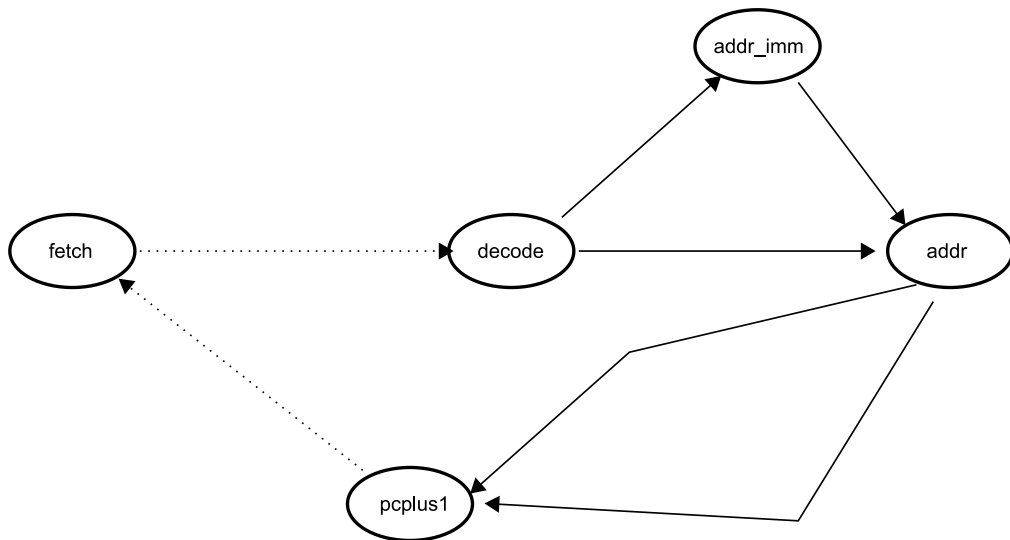
Assembler la machine craps complète et vérifier que les instructions de calculs et de branchement fonctionnent correctement.

Programmer la somme des N premiers entier.

#### 6. Séquencement des instruction de lecture et d'écriture en mémoire

---

Compléter et tester le séquenceur, pour qu'il gère le sous-graphe suivant :

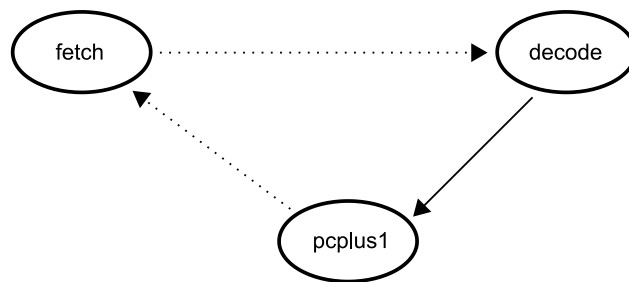


Assembler la machine craps complète et vérifier que les instructions d'accès mémoire `ld` et `st` fonctionnent correctement.

## 7. Séquencement de l'instruction `sethi`

---

Compléter et tester le séquenceur, pour qu'il gère le sous-graphe suivant :



Attention au format de `sethi` : le champ `rd` n'est pas exactement à la même place que dans les instructions de calcul et d'accès mémoire.

Assembler la machine craps complète et vérifier que l'instruction synthétique `set` fonctionne, ainsi que toutes les autres instructions.