

Architectures & Systèmes L2-TD4

On rappelle le jeu d'instruction du microprocesseur à étudier :

<i>clr</i>	Mise à zéro du registre ACC
<i>Ld #vi</i>	Chargement de la valeur immédiate <i>vi</i> dans ACC
<i>St [ad]</i>	Chargement du mot mémoire d'adresse <i>ad</i> avec le contenu de ACC
<i>Jmp ad</i>	Saut à l'adresse <i>ad</i>
<i>Add [ad]</i>	Chargement de ACC la somme du contenu de ACC et du mot mémoire d'adresse <i>ad</i>

Les instructions peuvent être interprétées de la manière suivante.

```
PC := 0
tant que vrai
  selon Mem[PC]
    clr: Acc :=0; PC := PC + 1
    ld: Acc := Mem[PC + 1]; PC := PC + 2
    st: Mem[Mem[PC + 1]] := Acc; PC := PC + 2
    jmp: PC := Mem[PC + 1]
    add: Acc := Acc + Mem[Mem[PC + 1]]; PC := PC + 2
```

Mem[PC] correspond au code de l'instruction qui sera notée *R1* tandis que Mem[PC+1] correspond à l'opérande (donnée ou adresse) notée *R2*.

L'unité de traitement dans la figure (de la page) suivante permet d'exécuter les instructions du microprocesseur précédent.

1. Donner une unité de contrôle sous forme d'automate qui définit l'exécution des instructions. Faire apparaître dans chaque état, les valeurs des signaux de contrôle.
2. Combien de cycles sont nécessaires pour l'exécution de l'instruction *add*?
3. Certaines opérations n'utilisent pas les mêmes éléments de l'unité de traitement et peuvent donc se faire en parallèle. Proposer alors un nouvel automate ayant moins d'états pour l'unité de contrôle. Combien a-t-on besoin à présent de cycles pour exécuter l'instruction *add*.

vers la partie controle

