

# Rapport du projet LC-3

## Avancement atteint :

Toutes les instructions demandées sont implémentées dans le fichier Logisim.  
Toutes les routines assembleur demandées sont implémentées de manière à ne pas modifier d'autres registres que celui utilisé pour la valeur de retour de la fonction (R0)

---

## Logisim :

### Flags NZP, calcul de l'état :

Si WriteReg = 1, alors on met à jour les flags NZP, sinon ceux-ci gardent leur valeur précédente. Cela est fait grâce à un multiplexeur permettant de sélectionner l'entrée désirée.

Le résultat du calcul est négatif si le bit de poids fort est à 1 (complément à deux), il est nul si tous ses bits sont à zéro et positif si aucune des deux conditions précédentes n'est remplie.

### Évaluation de la condition de saut :

La condition de saut est évaluée vraie si l'instruction courante est un branchement et si l'état des flags NZP spécifié dans l'instruction est effectivement leur état dans le circuit.

### Decodage Instruction :

Les instructions étant catégorisées par type dans leur tableau de Karnaugh, il suffit de regarder les deux premiers bits de leur code d'opération pour pouvoir les différencier.

Seuls les chargements et les opérations arithmétiques peuvent modifier les registres.

### ALU :

L'instruction SCAN est branchée sur l'emplacement laissé vide du LC-3.

### WriteVal :

La valeur de PC est modifiée si l'instruction courante est un saut et que les conditions sur NZP sont satisfaites.

PC prend la valeur de BaseR si l'instruction courante est JMP ou JSRR et est incrémentée de l'offset spécifié sinon. Cet offset est codé sur 9 bits dans le cas de BR et sur 11 bits dans le cas de JSR.

### **Write R7 :**

Ce circuit permet de décider si PC doit être sauvegardé ou non.

Le registre PC doit être sauvegardé dans R7 lors d'appels à JSR ou JSRR afin de permettre un retour à l'appelant.

Lorsque le pin WriteR7 de ce circuit est à 0, ce circuit est transparent.

Lorsque ce pin est à 1 en revanche:

- la valeur de PC incrémentée de 1 est envoyée vers les registres
- le pin d'écriture des registres est positionné à 1
- l'adresse du registre à écrire est forcée à 7

Ces actions ne sont effectuées que lors de la phase d'exécution de l'instruction (pin Exec à 1).

---

## **Partie Assembleur :**

### **Routines de test des instructions :**

Les fichiers de test des instructions du fichier logisim sont dans le dossier /LC-3/MemFiles, un fichier assembleur est associé à certains de ces fichiers pour des questions de compréhension.

### **index :**

La routine prend en argument :

- la chaîne str "hello World !" située à l'adresse 3020
- le caractère 'W'

et retourne l'adresse de la première occurrence de 'W' dans str (3026 dans ce cas)

### **rindex :**

La routine prend en argument :

- la chaîne str "Hello World !" située à l'adresse 3021
- le caractère 'W'

et retourne l'adresse de la dernière occurrence de 'W' dans str (302B dans ce cas)

### **strcmp :**

La routine prend en argument: - La chaîne str1 "Hello World !" - La chaîne str2 "Healo"

et retourne la différence entre les codes ASCII des deux premiers caractères différents entre str1 et str2

**strcpy :**

La routine prend en argument :

- La chaîne str1 “Hello World !”
- La chaîne str2 “Here starts the trash”

et copie le contenu de str1 dans str2, str2 contient donc “Hello World !he trash” après son exécution

**strncpy :**

La routine prend en argument :

- La chaîne str1 “Hello World !”
- La chaîne str2 “Here starts the trash”
- La valeur 7

et copie les 7 premiers caractères de str1 dans str2, str2 contient donc “Hello Warts the trash”