

BB-proc

Nous allons étudier un processeur théorique : « BB-proc ».

À chaque clic d'horloge, en combinant des opérations logiques élémentaires, le processeur effectue les opérations suivantes :

1. lecture du registre PC (pour Program Counter), qui vaut 00 à l'initialisation ;
2. chargement de l'instruction située à l'adresse PC depuis la mémoire vive (RAM) dans le registre CI (pour Current Instruction) ;
3. « décodage » de l'instruction, ici BB-proc décode les trois quartets de l'instruction, notés I, J et K : I indique le type d'opération à effectuer et le couple JK représente souvent un emplacement dans la RAM (voir le tableau ci-dessous) ;
4. exécution de cette instruction (ne pas passer à l'étape 5 s'il faut s'arrêter !) ;
5. si l'instruction ne résulte pas par un saut, ajouter 1 à PC.

Plus simplement :

1. où en est-on ?
2. que doit-on faire (récupération) ?
3. que doit-on faire (décodage) ?
4. faisons-le !
5. passons à la suite !

Voici une représentation schématique de la RAM avec un premier programme, sans doute le plus simple que l'on puisse écrire. Chaque « case mémoire » est repérée par l'abscisse K et l'ordonnée J (dirigée vers le bas) et peut stocker 12 chiffres binaires (donc 3 quartets).

J \ K	0	1	2	...
0	A00	???	???	...
1	???	???	???	...
2	???	???	???	...
.
.

Questions

- Combien y a-t-il d'emplacements dans la RAM ?
- Quelle est la taille de la mémoire en octets ?

Voici une liste des registres. Chacun peut stocker 12 bits, sauf PC (8 bits) et O (1 bit).

num	nom	description	valeur
0	R	sorte de carrefour	???
1	A	sert aux ops. arithm.	???
2	B	idem	???
3	O	dépassement (overflow)	?
4	PC	Program Counter	00
5	CI	Current Instruction	???

Exécutons le premier programme !

- Combien vaut PC ? 00.
- À l'adresse 00 (J=0 et K=0) on trouve l'instruction A00 que l'on charge dans CI.
- On a pour cette instruction I=A, J=0 et K=0, et d'après le tableau ci-dessous (dans la dernière ligne) il faut arrêter le programme.

I	ASM	Action (-> signifie «est copié dans»)
0	LD	@JK -> R (la valeur à l'adr. JK dans la RAM est copiée dans R)
1	STO	R -> @JK (la valeur dans R est copiée à l'adr. JK dans la RAM)
2	MOV	Reg J -> Reg K (le registre num J est copiée dans le reg num K)
3	ADD	A+B -> R (somme des registres A et B copiée dans R, voir (1))
4	SUB	A-B -> R (idem avec différence, voir (2))
5	JMP	JK -> PC (la valeur JK est copiée dans PC, c'est un saut)
6	JPZ	Si R=0 alors JK -> PC (si la condition est vraie, on saute)
7	JPO	Si O=1 alors JK -> PC (idem)
8	IN	User -> R (entrée d'une valeur par l'utilisateur)
9	OUT	R -> affichage
A	END	arrêt du déroulement du programme

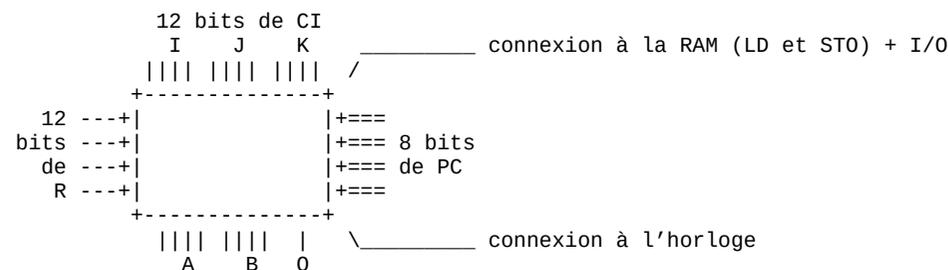
(1) si la somme A+B dépasse la capacité de R, le registre O vaut 1, 0 sinon.

(2) si la différence A-B est négative, le registre O vaut 1, 0 sinon.

Remarques

- Un registre est un type de mémoire rapide et « proche » du processeur.
- PC est aussi parfois appelé IC pour Instruction Counter.
- Le « décodage » se fait « mécaniquement » ou plutôt électroniquement avec les fonctions logiques qui ont été combinées.
- Techniquement l'étape 4 fait partie de l'étape 3 car elle est réalisée directement.
- Pour I=7 (IN) on peut imaginer que le processeur fait une pause pour qu'on ait le temps de lui communiquer la valeur.

Pour donner une image simplifiée d'un processeur qui « décode », on peut imaginer un circuit intégré branché de cette façon :



À vous d'écrire vos premiers programmes : afficher zéro et s'arrêter, afficher «HELLO», demander une valeur à l'utilisateur et afficher cette valeur plus un.