

TD 6 : Micro-architecture non pipelinée

1. Architecture 3 bus

Soit le processeur RISC 16 bits dont le chemin de données est représenté figure 1. Tous les registres sont sur 16 bits. Il y a 8 registres généraux, de R0 à R7. Le registre R0 est câblé à 0. La mémoire est commune pour les instructions et les données. Le format d'instruction est fixe, selon la figure 2. Les instructions sont représentées figure 3.

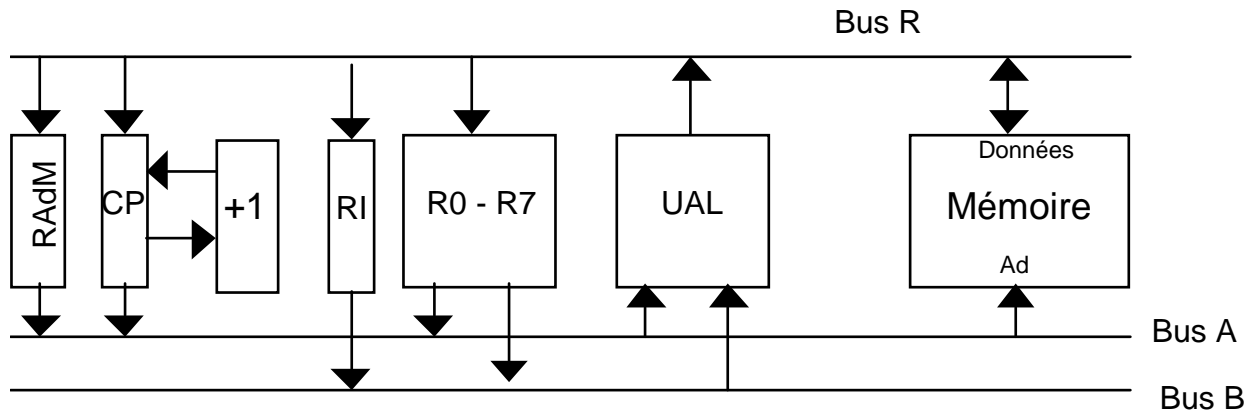


Figure 1 : Microarchitecture du processeur

F1 : reg/reg	4	3	3	3	
	00xx	a	d	b	xxx
F2 : reg/imm	4	3	3	6	
	01xx	a	d	immédiat	
F3 : branchement	4	3	9		
	1xxx	a	immédiat		

Figure 2 : format des instructions

Mnémonique	codop	action
ADD	0000	Rd <- Ra + Rb
SUB	0001	Rd <- Ra - Rb
ADI	0100	Rd <- Ra + ES(imm)
SUBI	0101	Rd <- Ra - ES(imm)
LD	0010	Rd <- Mem (Ra + Rb)
LDI	0110	Rd <- Mem (Ra + ES(imm))
ST	0011	Rd -> Mem (Ra + Rb)
STI	0111	Rd -> Mem (Ra + ES(imm))
Bxxx	1xxx	CP <- CP + ES(imm) si condx vraie

Figure 3 : Instructions

On considère que l'exécution d'une instruction se déroule en deux phases : lecture de l'instruction, avec incrémentation de CP, et exécution effective de l'instruction.

1) Préciser les transferts de données élémentaires associés à chacune de ces phases, pour chaque mode d'adressage :

Tous les registres sont opaques

Une seule donnée par cycle par bus

Commandes UAL :
 Bus R <- Bus A + B us B
 Bus R <- Bus A + ES6 (B us B)
 BUS R <- Bus A - Bus B
 Bus R <- Bus A + ES6 (B us B)
 Bus R <- BusB
 Bus R <- Bus A + ES9 (B us B)

Banc de Registres :

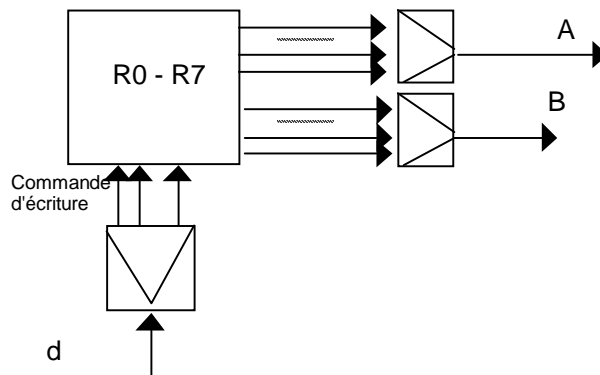


Figure 4 : Détail du matériel

2) On veut définir complètement la partie contrôle de processeur par automate selon le principe décrit en figure 5.
 Définir les contenus du séquenceur et du PLA de décodage.

Ecriture registres : RAdM, CP, RI
 Entrées Bus : BA, BB, BR
 Entrée CP : CCP
 Lecture Banc de registres : MUXA, MUXB
 Action UAL : voir détails architecture
 Mémoire : R ou W, Lecture ou Ecriture

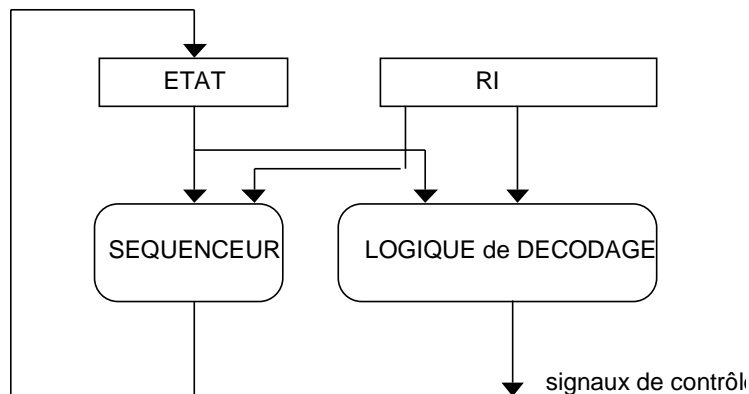


Figure 5 : Schéma de principe de la partie contrôle du processeur

3) On veut diviser l'exécution complète d'une instruction en quatre phases : lecture instruction, acquisition des opérandes, exécution, rangement. Comment faut-il modifier le chemin de données ? Quel peut être l'intérêt de cette modification ?

2. Architecture 1 bus

Le chemin de données du processeur POLD a une largeur de 16 bits, et une mémoire de 64 K mots de 16 bits. Toutes les instructions sont sur 16 bits. Les registres sont :

- CP : le Compteur de programme
- SI, un registre d'index
- SP : Pointeur de pile
- AD : Registre adresse mémoire
- RI : Registre instruction
- AX (accumulateur), BX (base), CX (compteur) et DX (supplémentaire)

Les modes d'adressage sont :

- **registre** : opérande = contenu d'un registre spécifié dans CODOP

8	8
Code Op	Inutilisé

- **immédiat** : opérande dans l'instruction

8	8
Code Op	Opérande

- **direct page 0** : adresse de l'opérande dans l'instruction

8	8
Code Op	Adresse

- **registre indirect** : adresse de l'opérande contenue dans un registre spécifié dans CODOP

8	8
Code Op	Inutilisé

- **indexé** : adresse de l'opérande = SI + déplacement

8	8
Code Op	déplacement

- **basé** : adresse de l'opérande = BX + déplacement

8	8
Code Op	déplacement

- **relatif** : CP = CP + déplacement

8	8
Code Op	déplacement

L'architecture interne est représentée figure 7. On suppose que

- Tous les registres sont opaques (registres D).
- Les échanges avec la mémoire (lecture ou écriture) se font en un cycle avec le contenu actuel du registre d'adresse. L'adresse mémoire doit être placée dans le registre d'adresse dans un cycle précédent. (cf figure 6). Ces transferts peuvent avoir lieu concurremment avec un transfert sur le bus interne.
- Les commandes de la boîte à opérations sont :

Addition :	notée +	E1 + E2
Incréméntation :	notée +1	E2 + 1
Addition retenue :	notée + R	E2 + R
Décréméntation :	notée -1	E2 - 1
Soustraction :	notée -	E2 - E1
Soustraction retenue :	notée - R	E2 - R
Ou exclusif	noté XOR	E1 xor E2

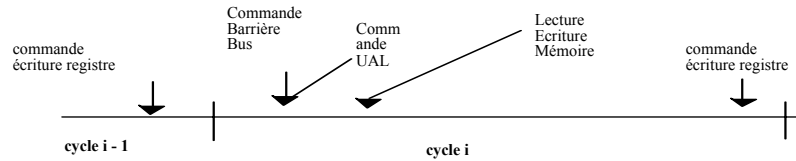


Figure 6

Décrire les transferts de données élémentaires qui interviennent pendant

- La lecture de l'instruction
- Le rangement de l'opérande (contenu de registre ou de case mémoire) dans le registre Y.
- L'exécution des instructions ADD, SUB, XOR.
- L'exécution des instructions JMP, PUSH BX, POP CX

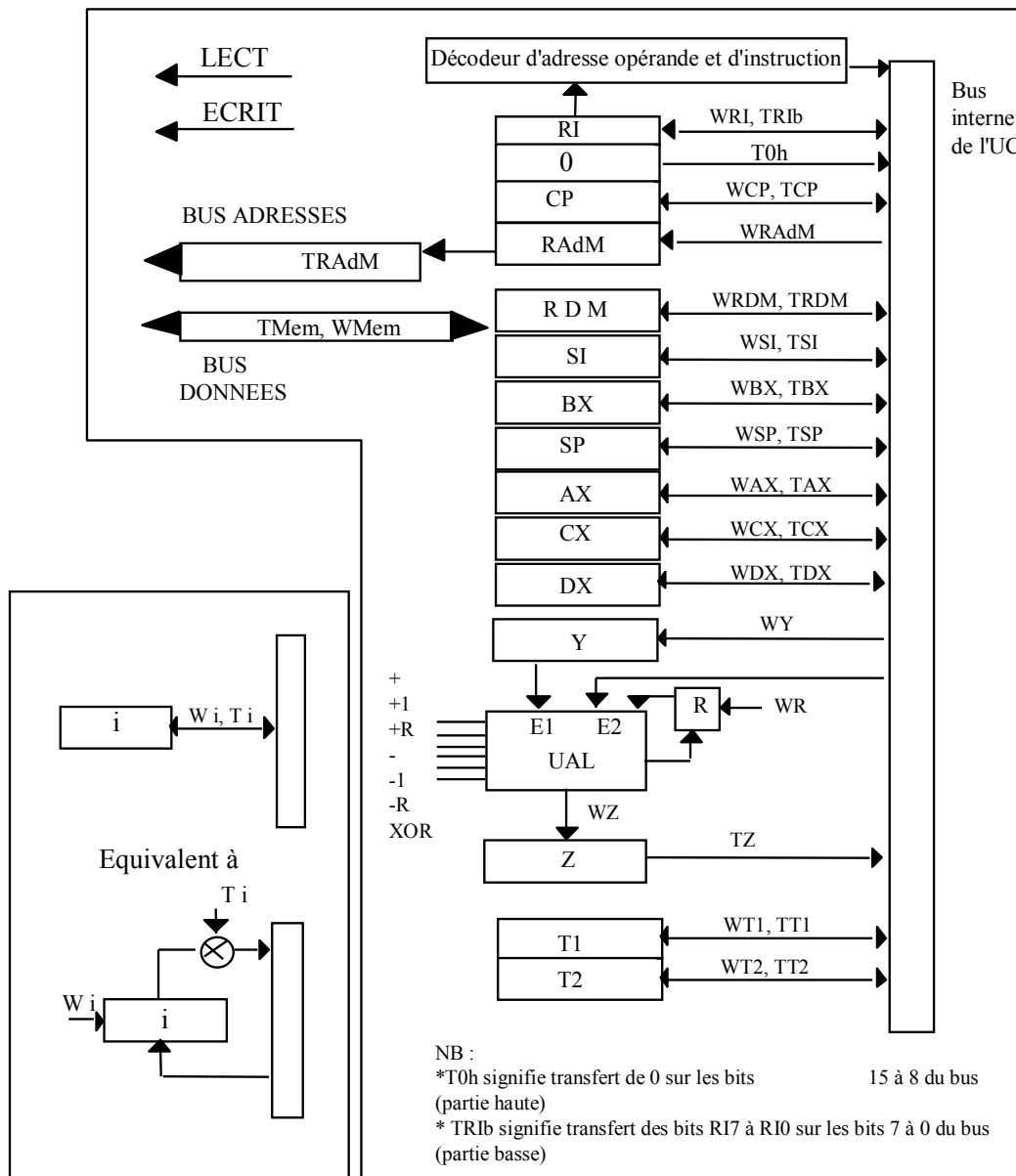


Figure 7 : Architecture interne de DOLD