

## Corrigé TD 9 – Contrôle de l'exécution des instructions

Dans ce TD, on utilise le jeu d'instruction NIOS II (processeur « logiciel » pour FPGA Altera).

### Version 1 : structure 3 bus

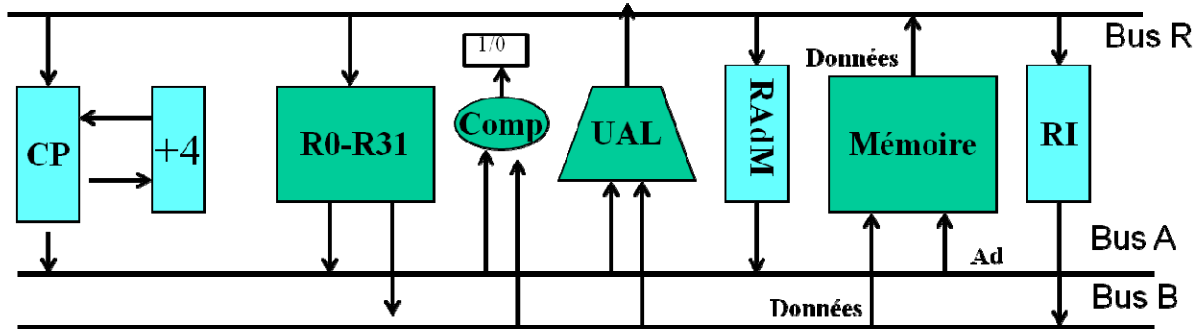


Figure 1 : Chemin de données (version 1).

Pour le chemin de données de la Figure 1, on veut réaliser le contrôle de l'exécution d'un certain nombre d'instructions du processeur NIOS.

- Add rk, ri, rj
- Or rk,ri,rj
- Ldw rk, (ri + IMM16)
- Blt ri,rj,IMM16
- Jmp ri
- Callr ri

### Q 1) Définir l'ensemble des commandes pour contrôler l'exécution des instructions

Rappel du TD8

Registres vers bus

Des barrières de bus (interrupteurs)

Bus A : TCP, TARi, TRAdM

Bus B : TBri, TRI (pour IMM16 avec extension)

Ecriture dans les registres

Des commandes d'écriture

WCP, WRi (sauf R0), Wcomp (booléen) WRAdM, WRI

Commandes mémoire

Lecture, Ecriture

Commandes UAL

Addition, Soustraction, Opérations logiques (AND, OR, etc)

Ecriture dans CP

Commande d'un multiplexeur : Bus R ou Sortie additionneur +4

Q 2) Remplir la table donnant les actions à exécuter à chaque cycle d'horloge pour exécuter les instructions.

INST	OP	OPX	Barrière BUS	MEM	UAL	Comp	BUS R	Mux CP	Ecriture registre
TC			TACP	Lec			MEM	+4	WCP, WRI
ADD	3A	31	TAR <sub>i</sub> , TBR <sub>j</sub>		+		UAL		WR <sub>k</sub>
OR	3A	16	TAR <sub>i</sub> , TBR <sub>j</sub>		OR		UAL		WR <sub>k</sub>
LDW-1	17		TAR <sub>i</sub> , TBR <sub>i</sub>		+		UAL		WRAdM
LDW-2			TRAdM	Lec			MEM		WR <sub>k</sub>
BLT-1	16		TAR <sub>i</sub> , TBR <sub>j</sub>			LT			WComp
BLT-2			TACP, TBR <sub>i</sub>		+		UAL	Bus R	Booleen et WCP
JMP	3A	0D	TAR <sub>i</sub> , TBR <sub>0</sub>		+		UAL	Bus R	WCP
CALLR -1	3A	1D	TACP, TBR <sub>0</sub>		+		UAL		WR31
CALLR -2			TAR <sub>i</sub> , TBR <sub>0</sub>		+		UAL	Bus R	WCP

### Remarques

1. Pour les commandes TAR<sub>i</sub> et TBR<sub>j</sub>, les numéros de registres source 1 et 2 (contenus dans RI) servent d'adresse pour la « mémoire » de 32 registres et les barrières de bus contrôlent l'accès du contenu du registre sélectionné vers le bus
2. Pour les commandes WR<sub>k</sub>, le numéro de registre destination sert d'adresse pour la mémoire de 32 registres et WR sert de commande d'écriture.

### Q 3) Définir la partie Etat futur = fonction (Etat présent, Entrées) de l'automate de contrôle

Les instructions utilisent deux ou trois cycles.

LDW, BLT et CALLR utilisent 3 cycles, les autres deux cycles.

Le graphe de transition est donné par la figure 3.

Trois correspond aux instructions utilisant 3 cycles donc

$$\text{Trois} = (\text{op}=16) + (\text{op}=3A \text{ et } (\text{OPX} = 0D \text{ ou } 1D))$$

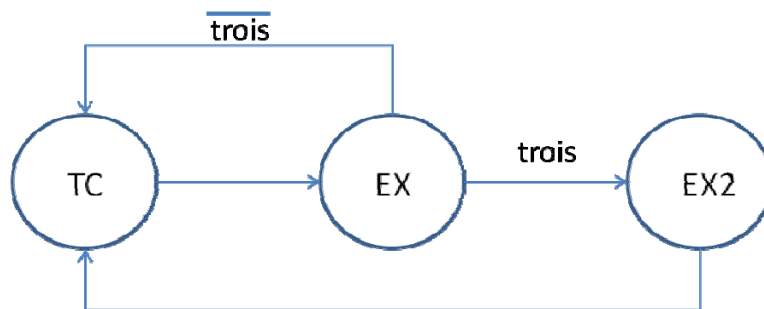


Figure 2 : Graphe de transition

On a donc la table de transition suivante

trois	Q1	Q0	D1	D0
	0	0	0	1
0	0	1	0	0
1	0	1	1	0
	1	0	0	0

$$D0 = \overline{Q1} \cdot \overline{Q0} \quad ]$$

$$D1 = \text{trois} \cdot Q0$$

Q 4) Donner les sorties de l'automate en fonction des entrées et de l'état présent.  
Dans la suite, on utilise les notations suivantes :

INST	OP = RI <sub>5</sub> -RI <sub>0</sub>	OPX= RI <sub>16</sub> -RI <sub>6</sub>
TC		
ADD	3A	31
OR	3A	16
LDW	17	
BLT	16	
JMP	3A	0D
CALLR	3A	1D

- TACP = TC + EX1.[Callr]+EX2. [Blt]
- TARi= EX.[ Add+Or+Ldw+Jmp+Blt]+ EX2 [Callr]
- TBRj = EX.[Add+Or+Blt]
- TBR0= EX.[Jmp+Callr] + EX2[Callr]
- TARADM = EX2. (Ldw)
- TBRI = EX.(Ldw) + EX2. (Blt)
- + UAL = EX [Add+Ldw+Jmp+Callr] + EX2 [Blt+Callr]
- OR UAL = EX.(Or)
- Lecture = TC + EX2. (LdW)
- TMEM = TC+ EX2.(Ldw)
- TUAL =  $\overline{TMEM}$
- WCP = TC + EX.[Jmp] + EX2. [Callr+Blt.Booleen]
- WRI=TC
- WRk= EX.[Add+Or]+EX2.[Ldw]
- WR31=EX.[Callr]
- WComp=EX.[Blt]

## Version 2

Q 5) Quels sont les différentes étapes pour l'exécution des types d'instructions suivantes

Rappel du TD8

Tronc commun

$$NCP \leftarrow CP+4 \quad // \quad RI \leftarrow MI(CP) \quad // \quad 1 \text{ cycle}$$

Instructions arithmétiques/logiques/comparaison de format R

$$A \leftarrow Ri \quad // \quad B \leftarrow Rj \quad // \quad 1 \text{ cycle}$$

$$RADM \leftarrow A+B \quad // \quad 1 \text{ cycle}$$

$$RDM \leftarrow RADM \quad // \quad 1 \text{ cycle}$$

$$Rk \leftarrow RDM \quad // \quad 1 \text{ cycle}$$

Instructions arithmétiques/logiques/comparaison de format I

$$A \leftarrow Ri \quad // \quad IMM32 \leftarrow IMM16 \text{ (dans RI) avec extension} \quad // \quad 1 \text{ cycle}$$

$$RADM \leftarrow A+IMM32 \quad // \quad 1 \text{ cycle}$$

$$RDM \leftarrow RADM \quad // \quad 1 \text{ cycle}$$

$$Rk \leftarrow RDM \quad // \quad 1 \text{ cycle}$$

Instructions mémoire

$A \leftarrow Ri // IMM32 \leftarrow IMM16$  (dans RI) avec extension // 1 cycle  
 $RADM \leftarrow A + IMM32 // 1 \text{ cycle}$   
 $RDM \leftarrow MD(RADM)$  en lecture et  $MD(RADM) \leftarrow B$  en écriture // 1 cycle  
 $Rk \leftarrow RDM // 1 \text{ cycle}$

Instructions de branchement conditionnel

$A \leftarrow Ri // IMM32 \leftarrow IMM16$  (dans RI) avec extension // 1 cycle  
 Si  $(Ri \text{ cond } Rj)$   $CP \leftarrow A + IMM32$ . L'addition est toujours effectuée. L'écriture dans CP n'intervient que si la condition est vraie // 1 cycle

Instructions de saut (JMP Ri)

$A \leftarrow Ri // B \leftarrow R0 // 1 \text{ cycle}$   
 $CP \leftarrow Ri + R0 // 1 \text{ cycle}$

Instructions Callr

$A \leftarrow Ri // B \leftarrow R0 // 1 \text{ cycle}$   
 $CP \leftarrow Ri + R0 // R31 \leftarrow NCP // 1 \text{ cycle}$

Q 6) Définir la partie Etat futur = fonction (Etat présent, Entrées) de l'automate de contrôle

Les instructions utilisent 5 cycles ou 3 cycles

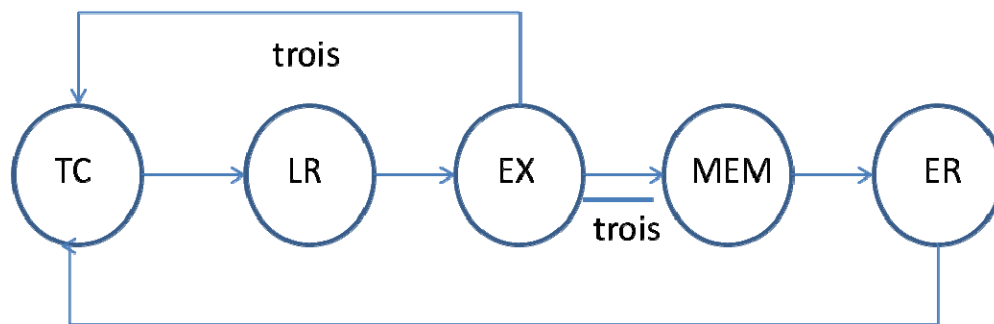


Figure 3 : Graphe de transition

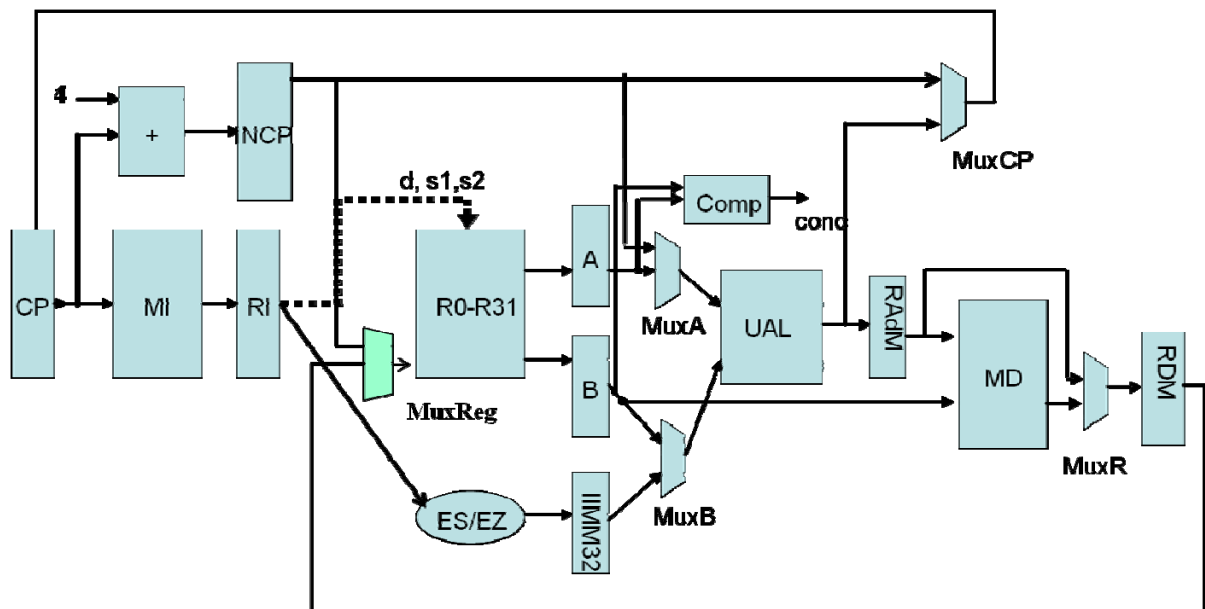


Figure 4 : chemin de données (version 2)

	Trois	Q2	Q1	Q0	D2	D1	D0
TC		0	0	0	0	0	1
LR		0	0	1	0	1	0
EX	0	0	1	0	0	1	1
EX	1	0	1	0	0	0	0
MEM		0	1	1	1	0	0
ER		1	0	0	0	0	0

$$\text{Trois} = \overline{\text{Bgt} + \text{Jmp} + \text{Callr}}$$

$$D0 = \overline{Q2} \cdot \overline{Q0} + \overline{\text{Trois}} \cdot \overline{Q2} \cdot Q1$$

$$D1 = \overline{Q1} \cdot Q0 + \overline{\text{Trois}} \cdot \overline{Q2} \cdot Q1$$

$$D2 = Q1 \cdot Q0$$

### Q 7) Définir l'ensemble des commandes pour contrôler l'exécution des instructions

Les commandes seront sous la forme Etat.W ou Etat.contrôleMux ou Etat.CommandeUAL ou Etat.commande mémoire.

Etat TC

WRI, WNCP [En fait TC.WRI et TC.WNCP]

Etat LR

WA = Add+Or+LdW+Bgt+Jmp+Callr

WB= Add+Or+Jmp+Callr

WIMM32 = LdW+Bgt

Etat EX

+UAL = Add+Ldw+Jmp+Callr+Blt

Or UAL = Or

WRAdM

MUxCP = UAL pour ( Jmp+Callr + Bgt.cond) et sinon NCP

WCP

MuxReg = NCP pour Callr

WR31 = Callr

Etat MEM

WRAdM

Lecture mémoire = Ldw

MuxR = MD si LdW et RAdM sinon

Etat ER

WRk

MuxReg = RDM sauf Bgt+Jmp+call

NB : la version 2 est une étape vers l'exécution pipeline des instructions.