Corrigé TD 8 – Exécution des instructions

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

Hypothèses

Le séquencement des instructions est réalisé par une horloge.

Figure 1 : signal d'horloge

En un cycle d'horloge, on peut effectuer un transfert d'un ou plusieurs registres vers un autre registre à travers un opérateur combinatoire (UAL, comparateur) ou la mémoire. On peut bien évidemment avoir en un cycle un transfert d'un registre vers un autre registre, sans opération intermédiaire.

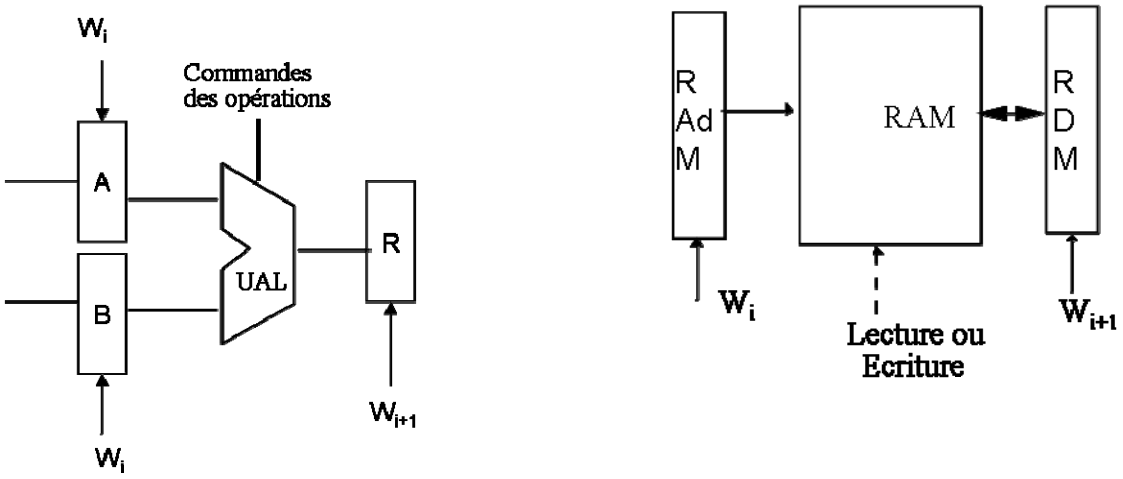

Figure 2 : Transferts en un cycle

Version 1 : structure 3 bus

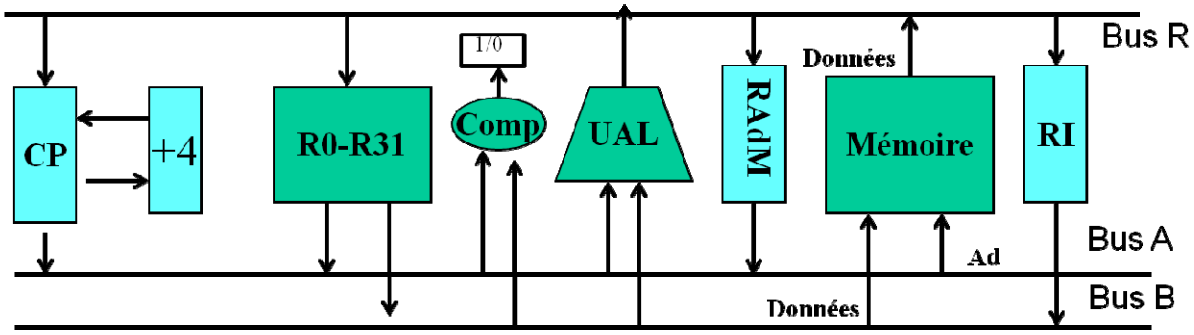

Figure 3 : Chemin de données (version 1).

Q 1) Quels sont les différentes étapes pour l'exécution des types d'instructions suivantes

Toutes les instructions commencent par un tronc commun
RI ← Mem (CP) // CP ←CP+4 // 1 cycle

Actions
Bus A ← CP
Lecture Mémoire et +4
RI ← Données Mémoire et CP ← Sortie +4

Instructions arithmétiques/logiques/comparaison de format R
Rk ← Ri Opération Rj // 1 cycle
Actions
Bus A ← Ri // Bus B ← Rj
Opération UAL
Rk ← sortie UAL

Instructions arithmétiques/logiques/comparaison de format I
Rk ← Ri Opération ExtensionIMM16 // 1 cycle
Actions
Bus A ← Ri // Bus B ← IMM16 avec extension signe
Opération UAL
Rk ← sortie UAL

Instructions mémoire
RAdM ← Ri Opération ExtensionIMM16 // 1 cycle
Actions
Bus A ← Ri // Bus B ← IMM16 avec extension signe
Opération UAL
RAdM ← sortie UAL

Rk← Mem (RAdM) // 1 cycle ou Mem(RAdM) ←Rk // 1cycle
Actions
Bus A ← RAdM
Lecture ou Ecriture Mémoire
Rk ← sortie Mémoire (Lecture) ou rien (Ecriture)

Instructions de branchement conditionnel
Booléen ← (Ri cond Rj) // 1 cycle
Actions
Bus A ← Ri // Bus B ← Rj
Comparaison selon condition
Booléen ←Sortie comparateur
Si Booléen == vrai alors CP ← CP + ExtensionIMM16 // 1 cycle
Actions
Bus A ← CP // Bus B ← IMM16 avec extension signe
Addition UAL
Si Booléen == vrai alors CP ←sortie UAL

Instructions de saut (JMP Ri)
CP ← Ri // 1 cycle
Actions
Bus A ← Ri // Bus B ← R0
Addition UAL
CP ← sortie UAL

Instructions Callr (format R)
R31 ← CP (CP est en fait NCP comme résultat du tronc commun) // 1 cycle

CP ← Ri // 1 cycle

Actions
    Bus A ← CP // Bus B ← R0
    Addition UAL
    R31 ← sortie UAL
Actions
    A ← Ri // Bus B ← R0
    Addition UAL
    CP ← sortie UAL

Q 2) Donner pour chaque type d'instructions le temps d'exécution en nombre de cycles d'horloge

Type d'instructions	Nombre de cycles
Arithmétique et logique R	2
Arithmétique et logique I	2
Mémoire	3
Branchement conditionnel	3
Saut	2
Callr	3

Q 3) Définir l'ensemble des commandes pour contrôler l'exécution des instructions

Registres vers bus
    Des barrières de bus (interrupteurs)
    Bus A : TCP, TRi, TRAdM
    Bus B : Tri, 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)
Lecture du banc de registres
    Numéro de registre
Ecriture dans CP
    Commande d'un multipleur : Bus R ou Sortie additionneur +4

Version 2

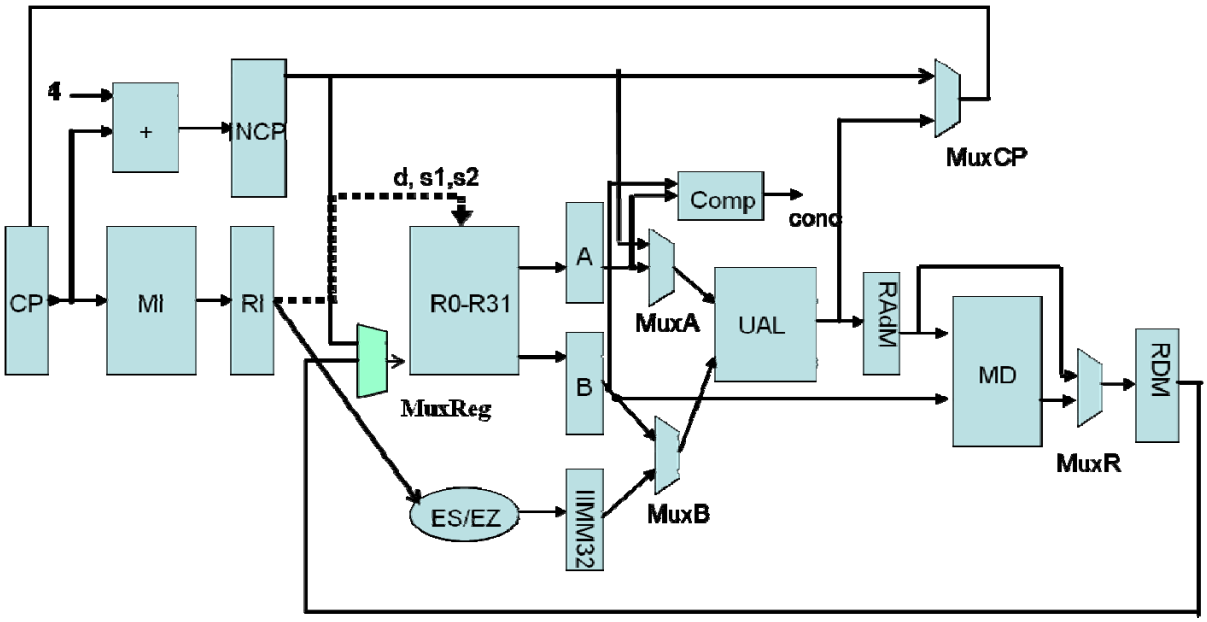

Figure 4 : chemin de données (version 2)

Q 4) Quels sont les différentes étapes pour l'exécution des types d'instructions suivantes

Tronc commun

NCP ← CP+4 // RI ← MI(CP) // 1 cycle

Instructions arithmétiques/logiques/comparaison de format R

A ← Ri // B← Rj // 1 cycle
RADM ← A+B // 1cycle
RDM ←RADM // 1 cycle
Rk ← RDM // 1 cycle

Instructions arithmétiques/logiques/comparaison de format I

A ← Ri // IMM32 ← IMM16 (dans RI) avec extension // 1 cycle
RADM ← A+IMM32 // 1cycle
RDM ←RADM // 1 cycle
Rk ← RDM // 1 cycle

Instructions mémoire

A ← Ri // IMM32 ← IMM16 (dans RI) avec extension // 1 cycle
RADM ← A+IMM32 // 1cycle
RDM ←MD(RADM) en lecture et MD (RADM) ← B en écriture // 1 cycle
Rk ← RDM // 1 cycle

Instructions de branchement conditionnel

A ← Ri // IMM32 ← IMM16 (dans RI) avec extension // 1 cycle
Si (Ri cond Rj) CP ← 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 ← Ri // B← R0// 1 cycle
CP ← Ri + R0 // 1 cycle

Instructions Callr

A ← Ri // B← R0// 1 cycle
CP ← Ri + R0 // R31 ←NCP // 1 cycle

Q 5) Donner pour chaque type d'instructions le temps d'exécution en nombre de cycles d'horloge

Type d'instructions	Nombre de cycles
Arithmétique et logique R	5
Arithmétique et logique I	5
Mémoire	5
Branchement conditionnel	3
Saut	3
Callr	3

Q 6) Définir l'ensemble des commandes pour contrôler l'exécution des instructions

Commandes d'écriture

WCP, WNCP, WRI, WRegistre (avec numéro registre d), WA,WB,WIMM32, WRADM, WRDM

Multiplexeurs

MuxA, MuxB, MuxCP, Mux Reg.

Commande UAL

Addition, soustraction, opérations logiques

Commande MD

Lecture, Ecriture.

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