

IFT 1227 – Architecture des ordinateurs

Devoir 4

- À faire en groupe de **deux** étudiants.
- Remise : Le **14 décembre** 2020 avant minuit.

I. Extension d'un processeur MIPS version un cycle

Il s'agit d'ajouter une implémentation de quelques instructions au processeur MIPS version un cycle vu en cours. Le schéma donné représente le processeur MIPS version un cycle avec les instructions supportées **lw**, **sw**, **j**, **addi**, **or**, **and**, **add**, **sub**, **beq**. Les instructions **jr**, **andi** seront ajoutées lors de la séance de démonstration.

1. **Solution schématique.** (30 pts) Étendre l'implémentation du chemin de données et du contrôleur pour traiter les instructions **jr** (2.5), **andi** (2.5), **jal** (10), **ori** (5) et **bnal** (10) (« branch if negative and link »).
 - Les modifications nécessaires du chemin de données doivent être faites d'abord sur le schéma (inclure dans le rapport à remettre). Utilisez le schéma donné ou redessinez votre schéma.
 - L'implémentation des instructions **jr**, **andi** sera montrée lors de séance de démonstration. L'intégration de ces instructions dans votre solution (schéma + VHDL) sera notée.
 - La solution schéma des modifications du chemin de données pour les instructions **jal**, **ori**, et **bnal**.
2. **Pour la partie contrôleur (tableau à compléter, rapport):**
 - (30 pts) Compléter la table de vérité de l'unité de contrôle pour incorporer les nouvelles instructions.

Notations:

- $R[reg]$ – Contenu du registre
- **imm** – Le champ Immediate de 16 bits d'instructions de type I
- **addr** – Le champ représentant une partie de l'adresse sur 26 bits de l'instruction de type J

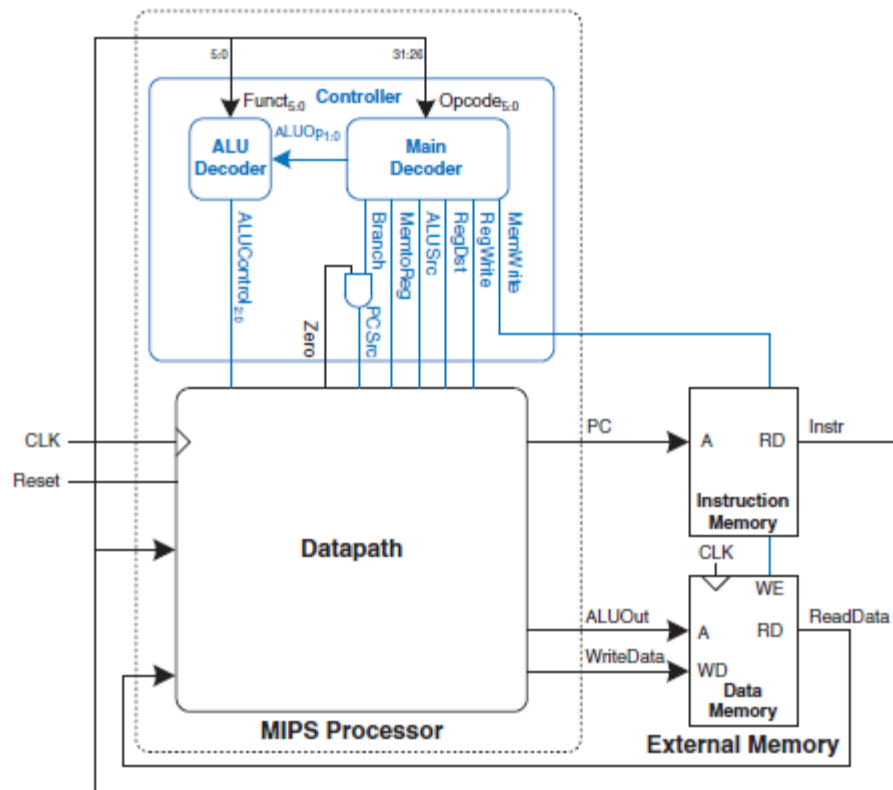
- **SignImm** – la valeur **imm** étendue par le bit le plus significatif de **imm** représentant une signe sur 32-bits (concaténer 16 fois le bit à la position 15 de **imm** et la valeur immédiate sur 16 bits **imm**) = $\{16\{\text{imm}[15]\}\}, \text{imm}\}$;
- **ZeroImm** – la valeur **imm** étendue par zéros sur 32-bits = $\{16'\text{bits } 0, \text{imm}\}$ (concaténation 16 zéros avec la valeur immédiate **imm**)
- $\text{Address} = \text{R}[\text{rs}] + \text{SignImm}$
- **Mem[Address]** – contenu de la mémoire à l'adresse **Address**
- **BTA** - branch target address = $\text{PC} + 4 + (\text{SignImm} \ll 2)$
- **JTA** - jump target address = $\{(\text{PC} + 4)[31:28], \text{addr}, 2'\text{bits } 0\}$ - concaténation de 4 bits les plus significatifs de 31 à 28 de la valeur $(\text{PC} + 4)$ + le champs **addr** sur 26 bits + 2 bits zéros (positions 1 et 0)
- **Label** – le texte qui indique une localisation de l'instruction

Les détails des instructions à implémenter :

Code d'opération	Nom	Description	Opération	funct
000000 (0)	j_r rs	Jump register	$\text{PC} = \text{R}[\text{rs}]$	001000 (08)
001100 (C)	andi rt,rs,imm	And Immediate	$\text{R}[\text{rt}] = \text{R}[\text{rs}] \& \text{ZeroImm}$	-
000011 (3)	jal label	jump and link	$\text{R}[\$ra] = \text{PC} + 4, \text{PC} = \text{JTA}$	-
001101(D)	ori rt,rs,imm	Or Immediate	$\text{R}[\text{rt}] = \text{R}[\text{rs}] \mid \text{ZeroImm}$	-
100100 (36)	b_{Nal} rs,label	Branch if negative and link	$\text{R}[\text{rs}] < 0?$ Si oui: $\text{PC} = \text{BTA};$ $\text{R}[\$ra] = \text{PC}+4$ Sinon : $\text{PC}=\text{PC}+4$	-

b_{Nal} rs,label Si le contenu du registre adressé par le champ **rs** représente une valeur négative, on doit préserver l'adresse **PC+4** dans le registre **\$ra** et sauter à l'adresse **BTA**. Sinon, l'adresse du **PC** devient **PC+4**.

3. **BTA** - branch target address = $\text{PC} + 4 + (\text{SignImm} \ll 2)$
4. **(30 pts)** Modifier le code VHDL du processeur MIPS initial (disponible sur le site du cours) pour incorporer les 3 instructions spécifiées plus haut et 2 instructions présentées en classe (**andi** et **j_r**). Le schéma structurel du code VHDL est présenté plus bas.



- a) **(5 pts)** Modifier le code du testbench et le programme dans le module `imem` pour tester toutes les instructions ajoutées et enlever les tests d'instructions non implémentées. Simuler votre processeur en observant le résultat d'exécution des instructions spécifiées plus haut.

Présentation : **5 pts**

À remettre.

- Le rapport : le schéma, la table de vérité de l'unité de contrôle, le programme test contenant le code assembleur et le code machine (voir page 437 (428) de votre livre pour un exemple).

Pour la remise du rapport: générer le document pdf et faire la remise électronique par StudiUM. Le projet compressé dans un seul fichier .zip contenant tous les fichiers sources VHDL dans le répertoire **files** pour qu'on puisse simuler et vérifier votre processeur.

[illegible]

