**Université A. Mira-Bejaia Octobre 2016**

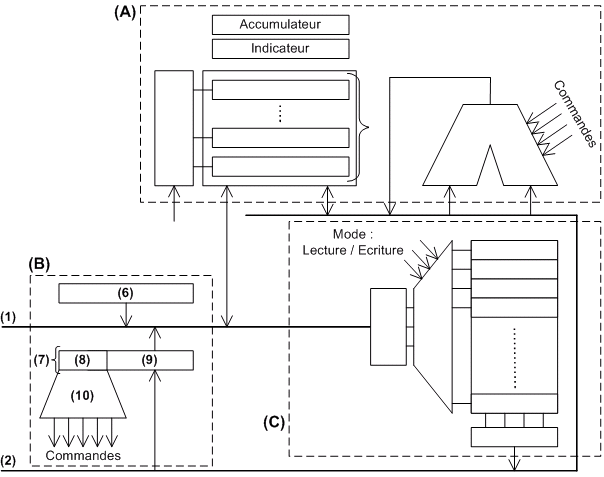
**Faculté des Sciences Exactes**

**Département Informatique**

# **Examen de rattrapage du module « SE2 »**

**Exercice1** **(8.5 pts)** :

**Q1** : Cette figure illustre l’architecture de Von-Neumann. Identifier les composants de cette architecture et compléter le tableau ci-après (ne pas utiliser d’abréviations).



**(9)**

**(5)**

**(3)**

**(4)**

**R1 :**

|  |  |  |  |
| --- | --- | --- | --- |
| (A) | Unité de traitement | (4) | Registre d’adresse mémoire |
| (B) | Unité de commandes | (5) | Registres généraux |
| (C) | Mémoire centrale | (6) | Compteur ordinal |
| (1) | Bus d’adresse | (7) | Registre d’instruction |
| (2) | Bus de données | (8) | Code opération |
| (3) | Registre d’information mémoire | (9) | séquenseur |

**Q2 :** En vous servant des notations vues en cours, dérouler l’exécution d’un rangement en mode indirect du contenu de l’accumulateur dans un mot mémoire.

**R2 :**

2.1 adresse de l’opérande(RI) 🡪 RAM

2.2 Commande de lecture

2.3 (RIM) 🡪 RAM

2.4 (acc) 🡪 RIM

2.5 Commande d’écriture

**Exercice 2** **(2.5 pts)** : On définit une nouvelle macro comme suit : **DIVI Rs , I**

**Description** : Le contenu du registre **Rs** est divisé par l’immédiat **I** défini sur 16 bits.

Réaliser cette macro en utilisant que des micro-instructions MIPS.

**R :**

**Addi $1, $0, I**

**DIV Rs, $1**

**Exercice3 (9 pts)** : Soit le programme en assembleur MIPS R3000 suivant:

|  |  |
| --- | --- |
| .**data**  x: .word 0  T: .word 23,-1,12,50  .**text**  \_start:    addi $8, $0, 3  la $9, T  lw $4, 0($9)  **test**:  beq $8, $0, fin  addi $9, $9, 4  lw $10, 0($9)  sub $11, $10, $4  blez $11, suite  add $4, $0, $10  **suite**:  addi $8, $8, -1  j test  **fin**:  la $12,x  sw $4,0($12)  addi $2, $0, 10  syscall | 1. Quelle est la valeur de l’étiquette **test** ?   **R1** : **0x00400010** (0.5)   1. Quelle est la valeur de l’étiquette **suite** ?   **R2** : **0x00400028** (0.5)   1. Donner le code machine (forme hexadécimale) de l’instruction : **add $4,$0,$10**   **R3** : **0x000A2020** (1)   1. Donner le code machine (forme hexadécimale) de l’instruction : **addi $8, $8, -1**   **R4 :** **0x2108FFFF** (1)   1. Donner le code machine (forme hexadécimale) de l’instruction: **beq $8, $0, fin**   **Indication :** Label = PC + 4 + (I\*4)  **R5** : **0x11000007** (1)   1. Donner le code machine (forme hexadécimale) de l’instruction: **blez $11, suite**   **Indication :** Label = PC + 4 + (I\*4)  **R6** : **0x19600001** (1)   1. Donner le code machine (forme hexadécimale) de l’instruction:  **j test**   **Indication :** Label = PC 31:28 || I\*4  **R7** : **0x08100004** (1)   1. Donner les instructions MIPS correspondant à la macro : **la $9, T**.   **R8** : **LUI $9,0x1000** (0.5)  **ORI $9,$9,0x0004** (0.5)   1. Quelle est la valeur de chacun des registres suivants à la fin de l’exécution de ce programme ?   **R9** : **$9**=**0x10000010** (0.5) **$4** =**0x00000032**=(**50**)10 (0.5) **PC** = 0x **00400044** (0.5)   1. Que fait ce programme ?   **R10** : Calcule le maximum des éléments du tableau T et le range en mémoire à l’adresse X  (0.5) |

**Bon courage**