INF1600

Architecture des micro-ordinateurs

Travail pratique 4 :

Architecture interne du processeur

Soumis par :

Chowdhury, Rasel - 2143023

Fellah, Rayan - 2147523

Groupe 4

4 avril 2022

TP 4		**/4,00**
		/4,00
Q1	/0,25	
Q2	/0,75	
Q3	/0,5	
Q4	/0,25	
Q5	/0,5	
Q6	/0,25	
Q7	/0,50	
Q8	/0,50	
Q9	/0,25	
Q10	/0,25	

Q1: Le processeur PolyRisc qui vous est fourni suit-il une architecture de von Neuman ou de Hardvard ? Justifiez votre réponse.

Pour répondre à la question du haut, le site ci-dessous a été utilisé. Ainsi, il peut être dit que le processeur PolyRisc utilise l'architecture de Harvard considérant que d'**un**, << Des bus séparés sont utilisés pour le transfert de données et d'instructions >>. De **deux**, << Une instruction est exécutée en un seul cycle >> et **enfin**, << Une adresse mémoire physique distincte est utilisée pour les instructions et les données >>. Ainsi, il peut être affirmé que le processeur PolyRisc utilise bel et bien l'architecture de Harvard considérant qu'il y a une séparation physique de la mémoire des données de même que celle du programme.

➤ ***Source*** *: https://fr.acervolima.com/difference-entre-von-neumann-et-harvard-architecture/*

Q2: Complétez le programme. Décrivez brièvement le principe de fonctionnement du programme.

Le code se trouve dans le fichier Q2.s !

Ce programme permet de déterminer le nombre de fois que les nombres contenus dans l'intervalle entre x et y sont divisibles par 4, 8 et 16. Les réponses sont stockées dans les variables a.b et c.

Q3: Donnez le RTN abstrait de toutes les instructions utilisées dans le programme. Notez Mi la mémoire d'instructions et Md la mémoire des données.

ldi : R[dst] ⇐ Imm

ld : R[dst] ⇐ Md[R[scr1]] * Md => Mémoire d'instructions

st : Md[R[dst]] ⇐ R[scr2] * Md => Mémoire d'instructions

add : R[dst] ⇐ R[src1] + R[src2]

sub : R[dst] ⇐ R[scr1] - R[scr2]

and : R[dst] ⇐ R[scr1] & R[scr2]

brnz : if (Z == 0) PC ⇐ disp

brgez : if (N == 0) PC ⇐ disp

stop : Arrêter le programme ⇔ IR ⇐ 0x0f000000; PC ⇐ PC

Q4: Inspectez les signaux de la simulation. Selon vos observations, quel est le CPI des instructions suivantes : ldi , ld , and ?

Selon nos observations, on remarque que le CPI de ldi = 3, que celle du ld = 3 et que celle du and = 3. Du coup, le CPI des trois instructions énumérées dans la question nous donne la même valeur, soit de 3 (<< Fetch, Decode, Execute >>).

Q5: Donnez le RTN concret des instructions suivantes : sub , brz , st ?

RTN concret de sub : sub : R[dst] <= R[scr1] - R[scr2]

IR ⇐ Mi[PC] : PC ⇐ PC + 1;

;

R[dst] ⇐ R[scr1] - R[scr2];

RTN concret de brz : if (Z == 1) PC <= disp

IR ⇐ Mi[PC] : PC <= PC + 1;

;

if (Z = 1) PC ⇐ disp

RTN de st : Md[R[dst]] <= R[scr2]

IR ⇐ Mi[PC] : PC ⇐ PC + 1;

;

Md[R[dst]] ⇐ R[scr2] ;

Q6: Donnez l'encodage hexadécimal des instructions suivantes :

mv r5, r0

- op (mv) = 7
- En se basant sur le diagramme des << Format des types d'instructions du processeur PolyRisc >>, plus précisément la ligne << op_alu >>, on arrive à la réponse suivante:

➢ Réponse: 0x07050000

st (r7), r2

- op (mv) = 1001
- En se basant sur le diagramme des << Format des types d'instructions du processeur PolyRisc >>, plus précisément la ligne << write_mem >>, on arrive à la réponse suivante :

➢ Réponse: 0x09000702

ld r3, (r20)

- op (mv) = 1000
- En se basant sur le diagramme des << Format des types d'instructions du processeur PolyRisc >>, plus précisément la ligne << read_mem >>, on arrive à la réponse suivante:

➢ Réponse: 0x08031400

Q7: Modifiez votre programme pour qu'il vérifie les multiples de 5,10 et 15. Gardez les mêmes valeurs d'initialisations que vous avez utilisés auparavant. Les résultats doivent être écrit dans la mémoire d'adresse a,b et c respectivement.

Le code se trouve dans le fichier Q7.s !

Q8: Considérez la liste des instructions hexadécimales suivantes : 0x0a010018, 0x0a02000c, 0x0a040008, 0x01030102, 0x0c01000a, 0x0c030008, 0x01010102, 0x0c000003, 0x01020201 0x0c000003, 0x09000401, 0x0f000000.Donnez l'équivalent assembleur de l'encodage ci-dessus et valider votre réponse en l'exécutant sur le processeur PolyRisc sous Code machine.

Le code se trouve dans le fichier Q8.s !

Q9: Que calcule le programme? Justifiez votre réponse Remplacez les champs des valeurs immédiates (Imm) des 2 premières instructions du programme par les valeurs ci-dessous :

1ere instruction : champ Imm = (MATR ETU 1 + MATR ETU 2) % 150

2eme instruction : Champ Imm = (MATR ETU 1 + MATR ETU 2) % 12

- Réponse : Ce programme calcule le PGCD entre les deux valeurs contenues dans les registres r1 et r2.

Q10: En considérant une fréquence hypothétique du processeur de 100 MHz, donnez le nombre d'instructions de votre programme et déduisez son temps d'exécution.

- **Fréquence** = 100 MHz = 10 ns
- ***1ere instruction :*** *champ Imm = (MATR ETU 1 + MATR ETU 2) % 150 :*

⇒ *(2143023 + 2147523) % 150 = 96*

- ***2eme instruction :*** *Champ Imm = (MATR ETU 1 + MATR ETU 2) % 12 :*

⇒ *(2143023 + 2147523) % 12 = 6*

- **Nombre d'instructions** = 98
- **CPI = 3**

- Ainsi, **le temps d'exécution** = (98*3) / 100 MHz = **2.94 MHz = 340.14 ns**