

École Polytechnique Montréal

INF1600

Architecture des micro-ordinateurs

TP2 – Architecture à deux bus et introduction à l'assembleur IA-32

Paul Clas 1846912

Gaspar Faure 1903623

Groupe 2 – B2

Présenté à

Giovanni Beltrame

26 février 2017

Exercise 1

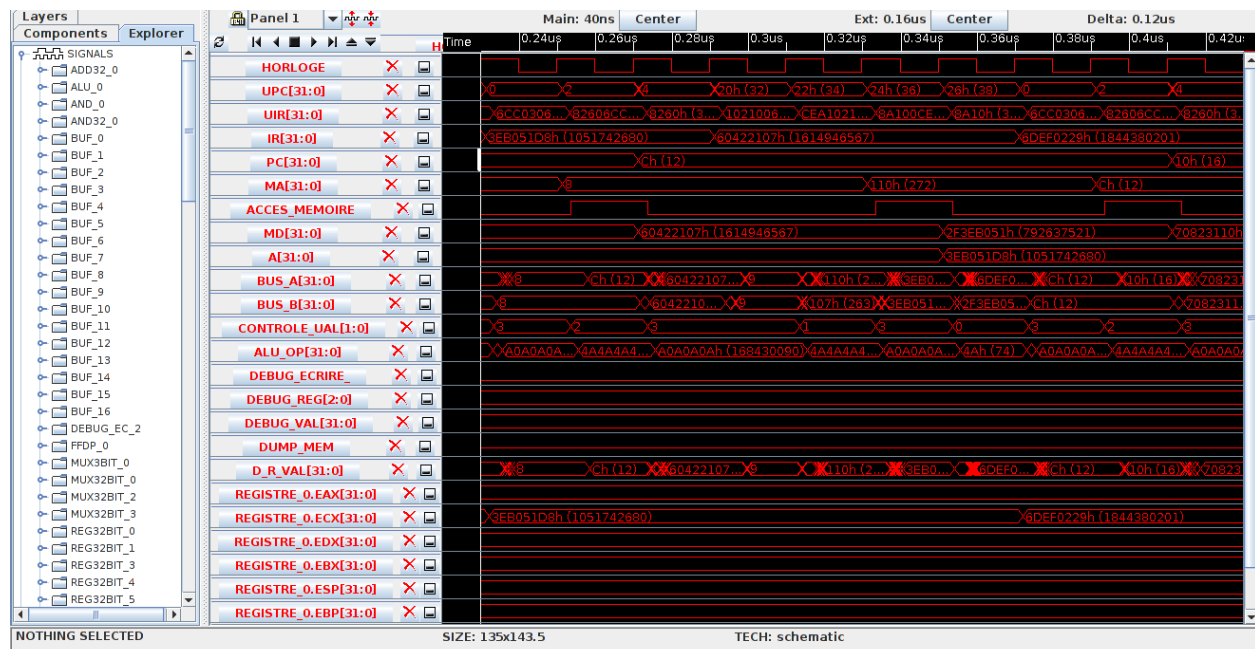
1)

RTN concret	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	hexa
MA <- PC;	0	0	1	1	0	0	0	0	0	1	1	0	0	0	0	0	0x3060
MD <- M[MA]; PC <- PC + 4;	0	1	1	0	1	1	0	0	1	1	0	0	0	0	0	0	0x6CC0
IR <- MD	1	0	0	0	0	0	1	0	0	1	1	0	0	0	0	0	0x8260

2)

RTN concret	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	hexa
A <- R[rc];	0	0	0	0	0	0	0	0	0	1	1	0	1	1	1	0	0x006E
MA <- A + IR<11..0>;	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	1	0x1021
MD <- M[MA]; A <- R[rb];	0	0	0	0	1	1	0	1	1	1	0	0	1	0	1	0	0x0CEA
R[ra] <- A oper MD	1	0	0	0	1	0	1	0	0	0	0	1	0	0	0	0	0x8A10

3) Voici la capture d'écran de notre simulation :



Cette capture montre bien le résultat placé dans R[1] soit 0x6DEFF0229.

A <- R[rc];	0x3EB051D8
MA <- A + IR<11..0>;	0x110
MD <- M[MA]; A <- R[rb];	0x2F3EB051
R[ra] <- A oper MD	0x6DEFF0229

4)

La valeur de op[6 :0] pour que l'opération finale soit un NAND est 0x01. En effet lorsque nous faisons l'analyse de l'utilisation des bits de l'UAL et observons d'avantage ce qui se passe dans l'op logique 32, nous réalisons que l'op[0] doit être égal à 1. La valeur de NAND doit donc être 0x01.

La valeur de l'EDX est alors 101C56.

5)

a) 0x55555555=0000 0101 0101 0101 0101 0101 0101 0101

op :0000= 0x0 =0 = nop

Ra :0101= 0x05=5

Les données des deux derniers octets et spécifiquement le 8^{ème} octet indique l'opcode 0 qui est l'opération nop qui ne fait rien. Le 7^{ème} octet et les suivants servent à repérer/valider plus facilement l'instruction. Ces valeurs spécifiques dans les bits sont inutilisées. Toutes les instructions avec l'opcode 0 ou plus petit que 0x0FFFFFFF feront exactement la même chose.

b) Un avantage de l'architecture à deux bus est que les deux bus nous permettent de faire plus de transfert de données (micro-instructions en binaire) durant le même cycle d'horloge. Oui nous nous sommes servis de cet avantage dans notre microprogramme développé en 2.

c) Les instructions de l'architecture du Tp2 peuvent être aussi flexibles voir même plus en termes d'opérations arithmétiques/logiques que celles étudiées à l'exercice du Tp1 car le processeur du TP2 comporte 2 bus. Nous pouvons écrire nos opérations une à la suite de l'autre comme dans le TP1 et utilisé un registre temporaire T ou alors utilisé les deux bus de l'assembleur IA-32 pour réaliser nos opérations.

Exercice 2

```
Terminal
Fichier  Édition  Affichage  Rechercher  Terminal  Aide
[paclab@l4712-24 exo2] $ gcc -m32 -gdwarf-2 -o tp2_2 tp2_2.c tp2_2.s
[paclab@l4712-24 exo2] $ ./tp2_2
Le résultat devrait être le même les 3 fois :
Version en langage C : -1.485682
Version en assembleur : -1.485682
Version en langage C : -1.485682
[paclab@l4712-24 exo2] $
```

Exercice 3

```
Terminal
Fichier  Édition  Affichage  Rechercher  Terminal  Aide
[paclab@l4712-24 exo3] $ gcc -m32 -gdwarf-2 -o tp2_3 tp2_3.c tp2_3.s
[paclab@l4712-24 exo3] $ ./tp2_3
a = 0      b = 5000    c = 3000    d = 500    e = 1000
Le résultat devrait être le même les 3 fois :

Version en langage C : 2500
Version en assembleur : 2500
Version en langage C : 2500
[paclab@l4712-24 exo3] $
```