



Zhao Long LIU

Yun Ji LIAO

TP2

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

INF1600 – Architecture des micro-ordinateurs

Groupe 03

Département de génie informatique et génie logiciel

Le 23 février 2020

École Polytechnique de Montréal

Exercice 1:

a) Recherche d'instructions

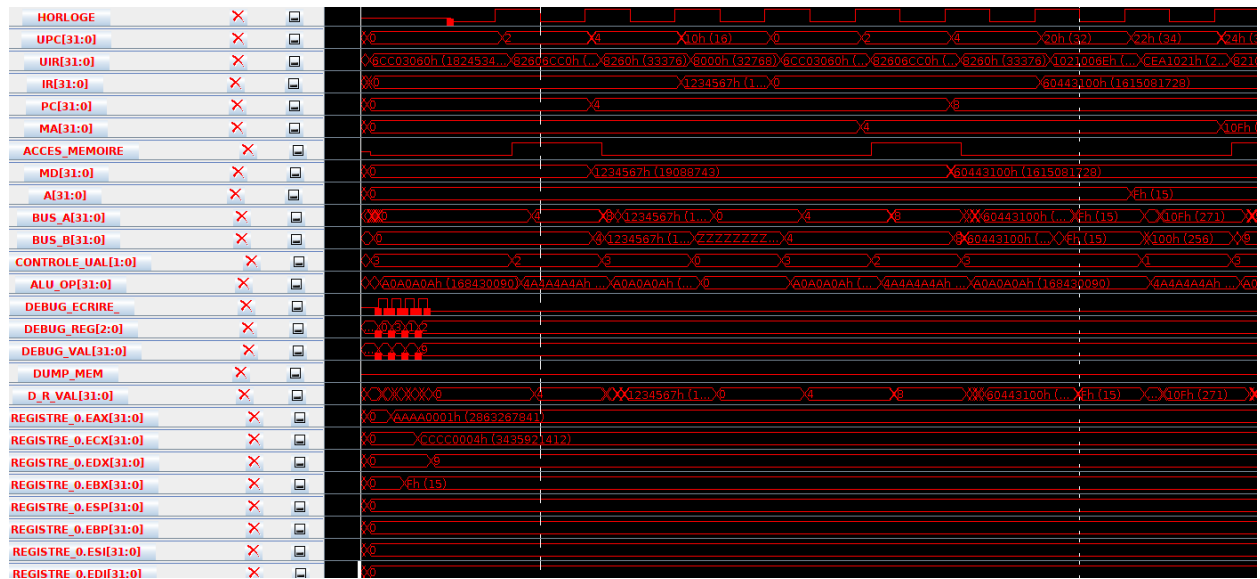
RTN Concret	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	hexa
MA <- PC;	0	0	1	1	0	0	0	0	0	1	1	0	0	0	0	0	0x3060
MD <- M[MA] : PC <- PC + 4;	0	1	1	0	1	1	0	0	1	1	0	0	0	0	0	0	0x6CC0
IR <- MD;	1	0	0	0	0	0	1	0	0	1	1	0	0	0	0	0	0x8260

b) Exécution d'une instruction générique

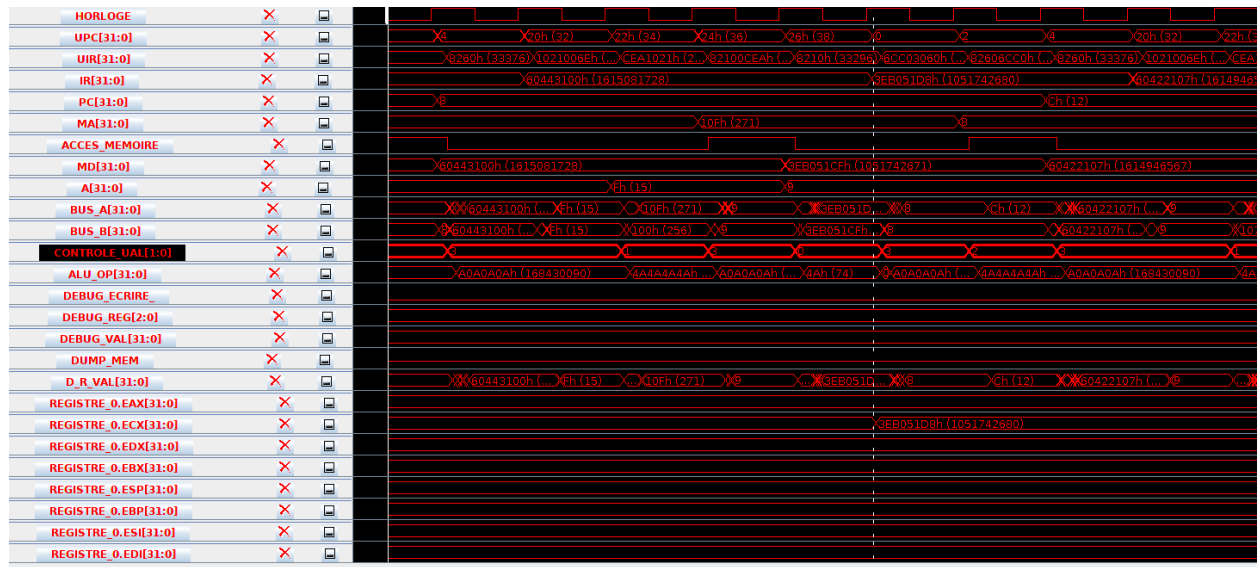
RTN Concret	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	hexa
A <- R[rc];	0	0	0	0	0	0	0	0	0	1	1	0	1	1	1	0	0x006E
MA <- A + IR<11...0>;	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	1	0x1021
MD <- M[MA] : A <- R[rb];	0	0	0	0	1	1	0	0	1	1	1	0	1	0	1	0	0x0CEA
R[ra] <- A oper MD;	1	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0x8210

c) Simulation

Capture d'écran des premiers cycles de la simulation



Deuxième capture d'écran comportant la réponse enregistrée dans le registre ECX



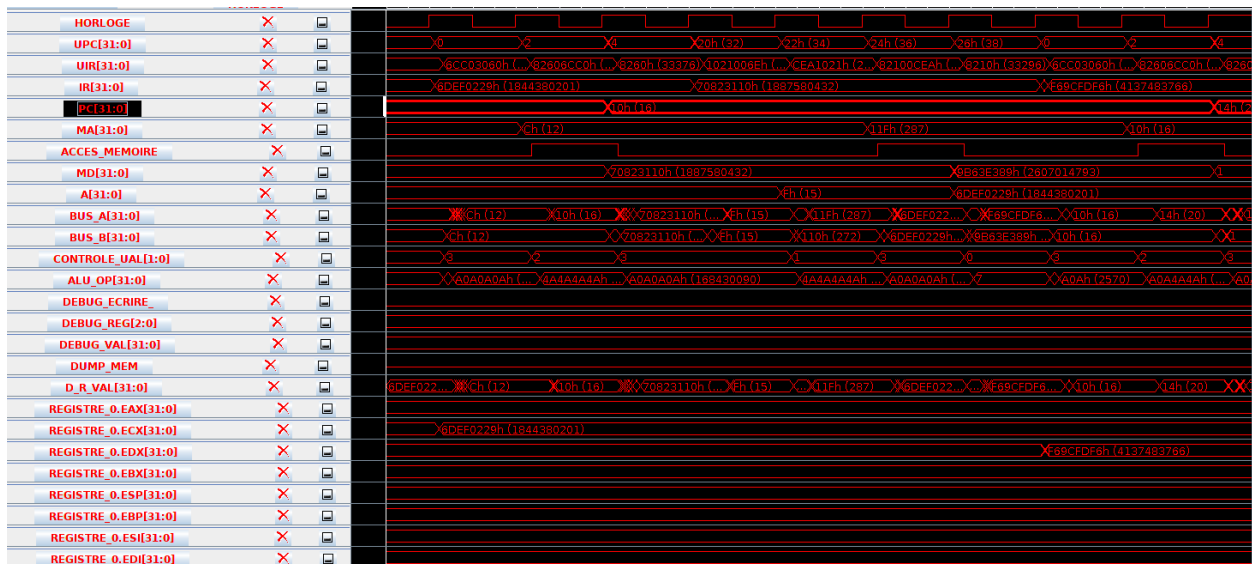
d) L'opération NAND

op[6:0] = 0000111

op[0] = 1
op[1] = 1
op[2] = 1
op[3] = 0
op[4] = 0
op[5] = 1
op[6] = 0

La réponse pour exécuter l'opération NAND est 0x07 pour l'opcode. Puisque dans l'ALU, les bits aux mêmes positions des deux chiffres « a » et « b » en entrée servent de SEL pour le multiplexeur et que les bits op[3:0] sont les valeurs en entrée du multiplexeur, il faut seulement modifier les trois bits les moins significatif.

La logique étant que le multiplexeur sert ici de table de vérité pour forcer un NAND. Si a[0] et b[0] valent 0, alors ce sera op[0] qui sera choisi. Puisque $0 \text{ NAND } 0 \rightarrow 1$, alors la valeur de op[0] est de 1. Le reste suit la logique illustrée plus haut.



La valeur de l'opération de NAND de tp2mem.txt est affiché dans le cycle 0x10 ou 0b16 de PC.

e) Compréhension

a) $0x5555 = 0b0101\ 0101\ 0101\ 0101$

Nous savons que IR<11...0> contient la valeur de la constante. Dans ce cas, cette constante s'agit de 1365 (décimal) et IR<16...12> représente R[rc]. Ainsi, pour avoir

- la même instruction, on ne devrait pas toucher les bits sauf pour $IR\langle 11\dots 0 \rangle$ ou la constante que ne change rien à la logique des opérations et des commandes exécutées.
- b) Avoir deux bus pour faire des opérations par cycle est très avantageux, car, notamment, cela permet de faire des commandes en parallèle. D'ailleurs, cette avantages nous a grandement aidé pour la réalisation du programme, puisque nous avons pu faire $MD \leftarrow M[MA] : A \leftarrow R[rb]$; en même temps ce qui sauve un cycle.
 - c) Non, car dans l'architecture du processeur à l'exercice 4 du TP1, celui pouvait utiliser deux constantes pour les calculs arithmétiques. Ainsi, il avait une meilleure flexibilité en termes d'opération arithmétiques logiques.