INF1600
Architecture des micro-ordinateurs

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

Adam Martin-Côté - 1798345
Abdoulaye Fall - 1825176
Groupe 1

Débuté : 3 octobre 2016
Remis : 16 octobre 2016

Polytechnique Montréal

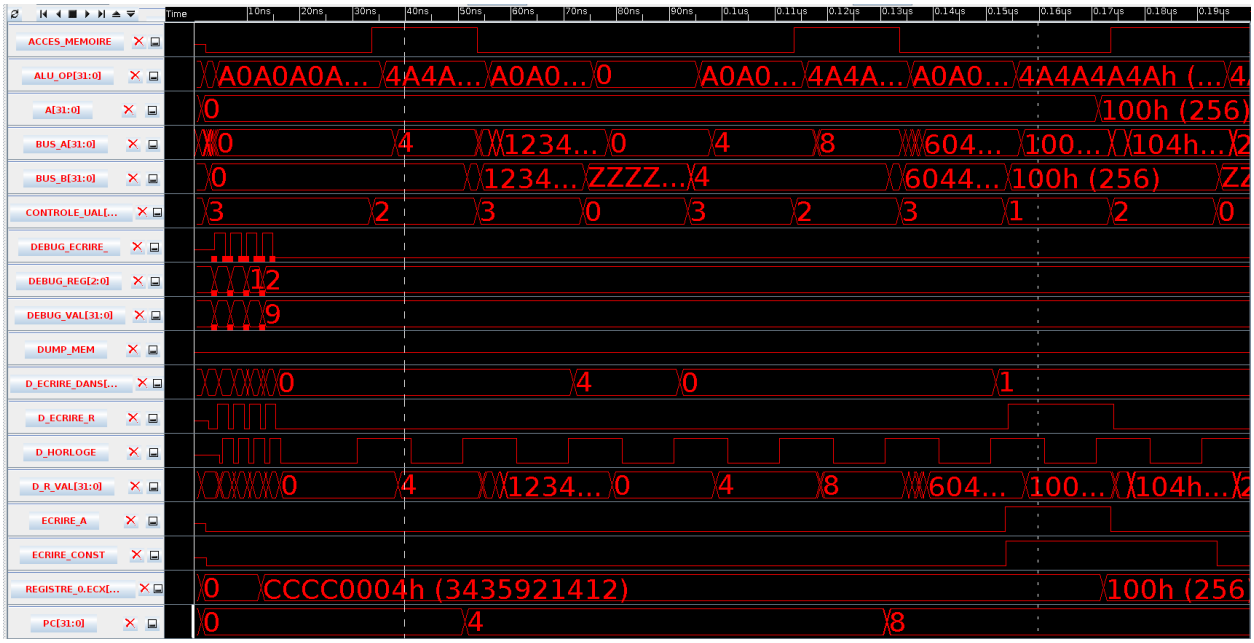
Exercice 1
#1

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

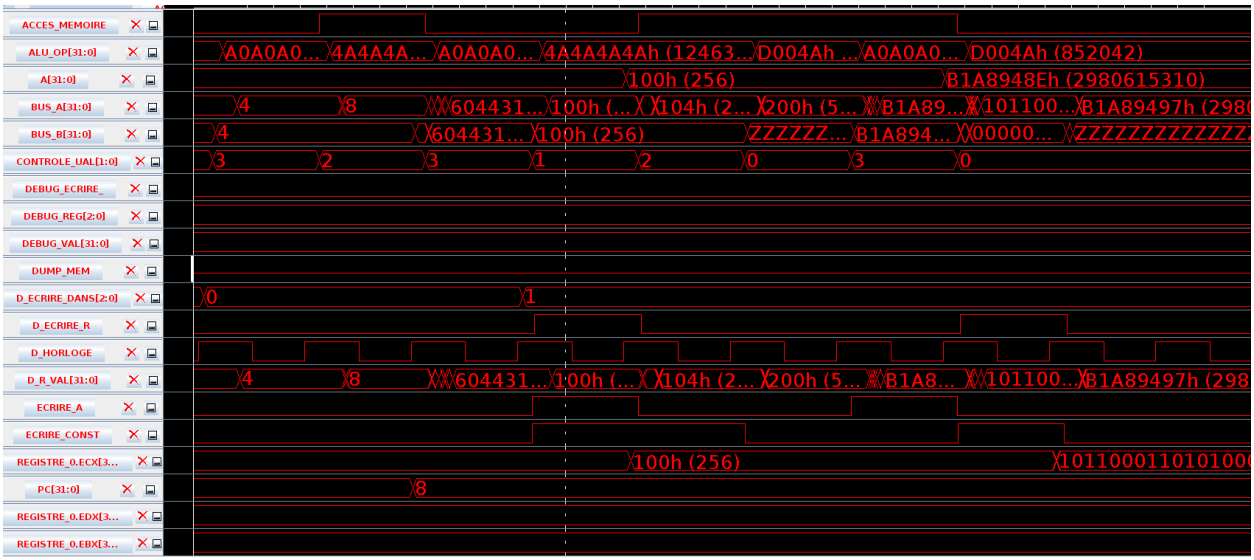
RTN concret	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	hexa
A<- R[IR<<16...12>	0	0	0	0	0	0	0	0	0	1	1	0	1	1	1	0	0X0037
MA<- A+IR<11...0>	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	1	0X10C1
MD <- M[MA]	0	0	0	0	1	1	0	0	1	0	0	0	0	0	0	0	0xC80
A<-MD	0	0	0	0	0	0	1	0	1	1	1	0	0	0	1	0	0X02E2
R[IR<26...22>]< R[IR<21...17>] oper A	1	0	0	0	0	0	0	0	0	0	0	1	1	0	0	1	0X0019

#3


Time
ACCES_MEMOIRE
ALU_OP[31:0]
A[31:0]
BUS_A[31:0]
BUS_B[31:0]
CONTROLE_UAL[...
DEBUG_ECRIRE_
DEBUG_REG[2:0]
DEBUG_VAL[31:0]
DUMP_MEM
D_ECRIRE_DANS[...
D_ECRIRE_R
D_HORLOGE
D_R_VAL[31:0]
ECRIRE_A
ECRIRE_CONST
REGISTRE_0.ECX[...
PC[31:0]
A0A0A0A...
4A4A...
A0A0...
4A4A4A4Ah (...
100h (256)
1234...
ZZZZ...
6044...
100...
104h...
CCCC0004h (3435921412)

#4

NAND = 0x0D


ACCES_MEMOIRE
ALU_OP[31:0]
A[31:0]
BUS_A[31:0]
BUS_B[31:0]
CONTROLE_UAL[1:0]
DEBUG_ECRIRE_
DEBUG_REG[2:0]
DEBUG_VAL[31:0]
DUMP_MEM
D_ECRIRE_DANS[2:0]
D_ECRIRE_R
D_HORLOGE
D_R_VAL[31:0]
ECRIRE_A
ECRIRE_CONST
REGISTRE_0.ECX[3...
PC[31:0]
REGISTRE_0.EDX[3...
REGISTRE_0.EBX[3...
A0A0A0...
4A4A4A...
A0A0A0...
4A4A4A4Ah (12463...
D004Ah ...
A0A0A0...
D004Ah (852042)
100h (256)
B1A8948Eh (2980615310)
604431...
100h (...
104h (2...
200h (5...
B1A89...
101100...
B1A89497h (298
B1A894...
00000...
ZZZZZZ...
B1A8...
B1A89497h (298

#5
a) Les 2 derniers octets représentent une valeur immédiate (IR<11..0>), ainsi qu'une partie de valeur de d'un registre (r3 = IR<16..12> i.e. les 2 derniers octets de l'instruction englobent les 4 bits moins significatifs de ce registre)

Cette instruction porte l'opcode "0", ce qui suit est donc sans reel importance
Instruction equivalente : 0x 00 00 DE AD

b) exécution plus rapide, on gaspille moins de coup d'horloge pour incrémenter le PC.
Nous avons utilisé cette propriété dans le premier microprogramme qui effectue le chargement des instruction

c) Cette architecture est plus flexible que la précédente: les microprogrammes peuvent être utilisés pour différentes instructions. (exemple: ADD et NAND utiliser le même microprogramme).
La mémoire unique de l'architecture Von Neumann donne plus de flexibilite