

# INF1600 Architecture des micro-ordinateurs

Laboratoire 1

Soumis par: Nguyen Nicolas- 2031636 Do Minh-Tri – 2030231 Groupe de laboratoire 05

Le 11 février 2020

## Exercice 1 Révision de logique et arithmétique numérique

- 1.
- a) 10110101 (binaire) = 01001011 (binaire positif) = -75 (décimal)
- b) 00110110 (binaire) = 54 (décimal)
- c) 7027 (octal) = 111 000 010 111 (binaire) = 000 111 101 001 (binaire positif) = -234 (décimal)
- d) FACE (hexadécimal) = 1111 1010 1100 1110 (binaire) = 0000 0101 0011 0010 (binaire positif) = 1024 + 256 + 32 + 16 + 2 (décimal positif) = -1330 (décimal)
- e) 10000001 (binaire) = 01111111 (binaire positif) = -127 (décimal)

2.

ID	Numéros	BIN	OCT	DEC	HEX
a)	817			X	X
b)	A10101				X
c)	911			X	Х
d)	0	X	X	X	X
e)	123		Χ	Χ	X

- x fait tout d'abord un ou logique avec 2 qu'on bit shift de 4 vers la gauche, soit 10 qu'on décale de 4 positions vers la gauche, qui devient 100000. Ce ou logique renvoie x auquel son 6° bit le moins significatif est affecté à 1. Ensuite, on effectue un et logique avec 254, ce qui affecte tous les bits sauf les 8 derniers à 0, à l'exception du 2° qui lui aussi est affecté à 0. Puis, on affecte à y la valeur de x modifiée. De façon générale, y = 0000 0000 xx1x xx0x.
- 4.
- a) -1999 = 1111 1000 0011 0001 (binaire) = F831 (hexa)
- b) -32 = 1111 1111 1110 0000 (binaire) = FFE0 (hexa)
- c) 5432 = 0001 0101 0011 1000 (binaire) = 1538 (hexa)
- 5.
- a) DAD + ACE = 1101 1010 1101 + 1010 1100 1110 = (1) 1000 0111 1011 = 87B (débordement signé)
- b) 10A + F50 = 0001 0000 1010 + 1111 0101 0000 = (1) 0000 0101 1010 = 05A (débordement signé)
- 6.
- a) 140 615 404
- b) 3 969 802 504

### Exercice 2 Disque dur

- a) 512 \* 783 \* 541 + 512 \* 870 \* 937 + 512 \* 532 \* 1210 + 512 \* 841 \* 1853 = 1,761 733 632 GB
- b) 278,876 160 Mb/s
- c) Étant donné que la vitesse du bus PCle est plus grande que le taux de lecture moyen du disque dur, la vitesse effective moyenne est inchangée.
- d) Oui, l'espace total sur le disque serait plus grand, mais le taux de lecture resterait inchangé.

## Exercice 3 Description RTN

#### Exercice 4 Architecture d'un microprocesseur

1.

- a) 16 00 30 10, où 0x16 est LSB, donc correspond à IR<0..7>
- b) T <- R[IR<20..18>];

T <- Mémoire2[T];

R[IR<23..21>] <- T << IR<12..0>;

c)

Α	В	С	D	Е	F	G	UAL	ecrireEIP	ecrireT	ecrireRegistre
0	01	0	0	1	0	0	0x0a	0	1	0
0	00	0	0	0	1	0	0x0a	0	1	0
0	00	0	1	0	0	0	0x10	0	0	1

d)

HORLOGE	×				
A	×				
B[1:0]	×		<u>√0</u>		
B[0]	×				
B[1]	×				
С	×				
D	×				
E	×				
F	×				
G	×				
ECRIRE_R	×				
ECRIRE_T	×				
ECRIRE_IR	×				
ECRIRE_EIP	×				
ALU_OP[6:0]	×		(Ah (10)	10h (16)	
IR[31:0]	×		X10300016h (271581206)		
Y[31:0]	×		(ZZZZZZZZZZZZZZZZZZZZZ)(0 )(4	3424X0 X16h (22)	
ALU[31:0]	×		<b>XXX</b> 0	43424)0 <b>////////////////////////////////////</b>	
T[31:0]	×		<b>⊘</b> ©	X43424140h (1128415552)	
EFFACER_TOUT	×		1		
DEBUG_ECRIRE_	×				
DEBUG_REG[2:0]	×		0203		
DEBUG_VAL[31:0]	×		(X X X1Fh (31)		
REGISTRE_0.EAX[31:0]	×		X0 XCCCC0004h (3435921412)		
REGISTRE_0.ECX[31:0]	×		X0 X137DDh (79837)	∕50000000h (13421	77280)
REGISTRE_0.EDX[31:0]	×		(√0 XBh (11)		
REGISTRE_0.EBX[31:0]	×	□	X0 X1Fh (31)		

On voit que le registre ECX (r1) a été modifié après les instructions, donc on sait que l'écriture s'est bien faite. Étant donné que l'on ne connait pas la valeur de Memoire2[r4], on ne peut que supposer que le résultat du calcul est bon.

2.

- a) 08 00 31 1a , où 0x08 est LSB , donc correspond à IR<0..7> On invente aussi un opcode, 0X1a, étant donné qu'il n'en n'existe pas pour décrire cette opération.
- b) T <- R[IR<20..18>];

T <- Mémoire2[T];

T <- T + R[IR < 17..15 >];

R[IR<23..21>] <- T >> IR<12..0>;

c)

Α	В	С	D	Ε	F	G	UAL	ecrireEIP	ecrireT	ecrireRegistre
0	01	0	0	1	0	0	0x0a	0	1	0
0	00	0	0	0	1	0	0x0a	0	1	0
0	10	0	0	1	0	0	0x4a	0	1	0
0	00	0	1	0	0	0	0x11	0	0	1

d)

I → N ≜ ▽			Time	s 20ns	40ns	60ns	Sons	0.1us	0.12us	0.14us	0.160
HORLOGE	×										
A	×										
B[1:0]	×			D .	XI	X		X2	X .		
B[0]	×										
B[1]	×								<u> </u>		
С	×										
D	×										
E	×										
F	×										
G	×								<u> </u>		
ECRIRE_R	×										
ECRIRE_T	×										
ECRIRE_IR	×										
ECRIRE_EIP	×								<del> </del>		
ALU_OP[6:0]	×				¥Ah (1			X4Ah (74)	11h (17)		
IR[31:0]	×			<u>(0</u>	X1A310008h						
Y[31:0]	×	_			ZZZZZZXO		X <u>43424</u> X0		_X€		
ALU[31:0]	×			<b>*</b> (0			X43424X0	XX434 <b>X</b> 4.	143424		
T[31:0]	×						X <u>4342</u>	4140h ( <b>X</b> 434	24 <u>14Bh (112</u>	8415563)	
EFFACER_TOUT	×										
DEBUG_ECRIRE_	×										
DEBUG_REG[2:0]	×										
DEBUG_VAL[31:0]	×			.X X X1Fh (31)							
REGISTRE_0.EAX[31:0]	×				n (3435921412)						
REGISTRE_0.ECX[31:0]	×			0 X137DDh (79837	)					134241h (44078	
REGISTRE_0.EDX[31:0]	×			0 \8h (11)							
REGISTRE_0.EBX[31:0]	×	_		0 X1Fh (31)							

On voit que le registre ECX (r1) a été modifié après les instructions, donc on sait que l'écriture s'est bien faite. Étant donné que l'on ne connait pas la valeur de Memoire2[r4], on ne peut que supposer que le résultat du calcul est bon.