

École Polytechnique de Montréal
Département de génie informatique et génie logiciel

INF1600

Architecture des micro-ordinateurs

TP1

Soumis par :

William Harvey (1851388) et
Mathieu Bélanger (1850591)
Section 4

16 février 2017



Exercice 1 : Révision de logique et arithmétique numérique

1. a)
$$\begin{array}{r} 11110101 \\ - \quad 1 \\ \hline 11110100 \end{array} \rightarrow 00001011 \rightarrow -11$$

b) 42

c) -1713

d) -13 570

e) -128

2. a)

ID	Numéros	Bin	Oct	Dec	Hex
a	5781			x	x
b	10000000	x	x	x	x
c	1600		x	x	x
d	B747				x
e	00000000	x	x	x	x

3. On affecte à la variable y le résultat de la comparaison logique binaire **ET** entre la variable x et le résultat du décalage du nombre 3 (0011) décalé de 4 bits vers la gauche (0011 0000).

4. a) $1111\ 1011\ 0010\ 1110$ $0x\text{FB2E}$

b) $0111\ 1111\ 1111\ 1111$ $0x\text{7FFF}$

c) $1111\ 1111\ 1110\ 0000$ $0x\text{FFE0}$

5. a) $1100\ 1011$ $0x\text{CB}$
Il n'y a aucun débordement signé.

b) $1001\ 1010$ $0x\text{9A}$
Il n'y a aucun débordement signé.

6. a)

Big-endian :

oc ₀	oc ₁	oc ₂	oc ₃	oc ₄	oc ₅	oc ₆	oc ₇
08	61	C2	BB	38	A0	9E	EC

$\rightarrow 1100\ 0010\ 1011\ 1011\ 0011\ 1000\ 1010\ 0000 = 3\ 267\ 049\ 632$

b)

Little-endian :

OC ₀	OC ₁	OC ₂	OC ₃	OC ₄	OC ₅	OC ₆	OC ₇
EC	9E	A0	38	BB	C2	61	08

→ 2 688 072 642

Exercice 2 : Disque dur

a. $512 \text{ (B/secteur)} * 792 \text{ (secteurs/piste)} * 624 \text{ (pistes/zones)} * \text{zone 1} = 253\,034\,496 \text{ octets}$

zone 2 = 568 688 640 octets

zone 3 = 653 721 600 octets

zone 4 = 669 081 600 octets

Espace total = espace zone 1 + espace zone 2 + espace zone 3 + espace zone 4
= 2 144 526 336 octets = 2045,18 Mo = 2,00 Go

b. Taux de lecture moyen = 33,25 Mo/s = 265,99 Mbits/s

c. Cela ne changerait rien au taux de lecture moyen, car ce bus permet largement à ce débit d'information de circuler sans aucune limite.

d. Oui, l'espace totale du disque sera agrandie, ainsi que le taux de lecture, car chaque zone contiendra plus de pistes et de secteurs par pistes.

Exercice 3 : Description RTN

1.

$(IR < 31...27 > = 5) : (IR < 16...0 > = 8) \rightarrow R[IR < 26...22 >] \leftarrow (R[IR < 26...22 >] - R[IR < 21...17 >]);$
 $R[IR < 26...22 >] \leftarrow (R[IR < 26...22 >] \times R[IR < 16...0 >]);$

2.

$(IR < 31...27 > = 13) : (IR < 16...0 > = 1) \rightarrow R[IR < 26...22 >] \leftarrow (R[IR < 26...22 >] - R[IR < 16...0 >]);$
 $R[IR < 21...17 >] \leftarrow (R[IR < 21...17 >] - R[IR < 16...0 >]);$

Exercice 4 : Architecture d'un microprocesseur

a) 0x4A2C0000

b) $T \leftarrow IR < 20..18 >;$
 $IR < 23..21 > \leftarrow RAM2[T] + T;$

c)

	$T \leftarrow IR\langle 20..18 \rangle;$	$IR\langle 23..21 \rangle \leftarrow RAM2[T] + T;$
A	X	X
B	01	X
C	0	0
D	0	0
E	1	0
F	0	1
G	0	0
UAL	0x0A	0x4A
EcrireEIP	X	X
EcrireT	1	0
EcrireRegistre	0	1

- d) Lorsqu'on sait que le registre r3 a une valeur de 0x1F (31), on peut soustraire notre résultat final, qui est stocké dans r1, par ce registre r3. Ainsi, on obtient la valeur de mémoire2[r3]. Cette valeur est de 0x6261605F. Cela correspond exactement aux données retrouvées dans la mémoire à partir de la 32^e adresse (0x1F). On peut aussi voir que sur l'image de notre simulation, X et Y ont les valeurs adéquates.



2. a) 0x5B298023

b) $T \leftarrow IR\langle 17..15 \rangle;$
 $T \leftarrow RAM2[T];$
 $T \leftarrow T + IR\langle 12..0 \rangle;$
 $IR\langle 23..21 \rangle \leftarrow IR\langle 20..18 \rangle \gg T$

c)

	$T \leftarrow IR\langle 17..15 \rangle;$	$T \leftarrow RAM2[T];$	$T \leftarrow T + IR\langle 12..0 \rangle;$	$IR\langle 23..21 \rangle \leftarrow IR\langle 20..18 \rangle \gg T$
A	X	X	X	X
B	10	X	X	01
C	0	0	0	0
D	0	0	1	0
E	1	0	0	1
F	0	1	0	0
G	0	0	0	0
UAL	0x0A	0x0A	0x4A	0x11
EcrireEIP	X	X	X	X
EcrireT	1	1	1	0
EcrireRegistre	0	0	0	1

d) Cette réponse est bonne, car la donnée stockée à l'adresse r3 (0x1F) est 0x6261605F. On additionne cette donnée à 0x23 ce qui donne 0x6261605F. On additionne cette donnée à 0x23 ce qui donne 0x62616082, puis on décale de r2 (0xB) vers la droite cette même donnée, ce qui nous permet d'obtenir 0xC4C2C comme réponse finale.

