­

INF1600

Travail pratique 1

Architecture du processeur

**Trimestre :** automne 2018

Équipier1 : Serge *GNAVO (1869985)*

Équipier2 : Fabrice *NDUI (1914377)*

Polytechnique Montréal

Date de remise (07-10-2018)

exercice 1

question 1

a)

11001101

Complement à deux :00110011

00110010+1=(Inversion des bits plus un)

00110011

Valeur décimale= 2^0+2^2+2^3+2^6+2^7=205

(b) 01101011 (binaire)

Complement a deux : 10010101

valeur decimale:107

(c) 5726 (octal)

binaire:0101 0111 0010 0110

Complement a deux:1010 1000 1101 1001 +1=1010100011011010

Valeur decimale:22310

(d) FADE (hexadécimal, 16 bits)

binaire: 1111 1010 1011 1110

Complement a deux: 0000 0101 0100 0001 +1=0000 0101 0100 0010

Valeur decimal:64222

(e) 10000000 (binaire)

Complement a deux:01111111+1=10000000

Valeur décimale:128

question 2

ID Numéros BIN OCT DEC HEX

(a) 2586 x x

(b) 00000000 x x x x

(c) 11111 x x x x

(d) 514 x x x

(e) A626 x

question 3

y = x & (5 << 4);

//(5<<4) 0101<<0000 =0101 0000 (sur 8bits)

la fonction retourne les bits a la positions (5 et 7) d'un vecteur x //c'est un masque

question 4:Reprendre le a

(a) -9876 (le mettre en decimal)

9876/2=

Binaire: 1001 1000 0111 0110

Complement a deux 16bits: 0110 0111 1000 1001+1=0110 0111 1000 1010

Complement en hexadeximal: 6 7 8 A //

(b) -64

Binaire: 01000000

Complement a deux :10111111+1=1100 0000

Complement en hexadécimal:C 0 //

(c) 12345 (est decimal)

Binaire:0001 0010 0011 0100 0101

Question 5://dans ce cas on nous demande de rester sur 8bits

8B+

6A

=1000 1101

+

0110 1010

=1111 0110 =F 6

il y a pas de bordement

52+49=

(hexadecimal)

0101 0010

+

0100 1001=

1001 1011

9 B

(IL y a un debordement car la somme de 2 nombres positifs donnent un nombre negatifs)

Question 6:

a-big endian

Le MSB est a la plus petite adresse soit oc2

0\*16^15+8\*16^14+6\*16^13+1\*16^12+12\*16^11+2\*16^10+11\*16^9+11\*16^8+3\*16^7+8\*16^6+10\*16^5+0\*16^4+9\*16^3+14\*16^2+14\*16^1+12\*16^0

b-little endian

0\*16^0+8\*16^1+6\*16^2+1\*16^3+12\*16^4+2\*16^5+11\*16^6+11\*16^7+3\*16^8+8\*16^9+10\*16^10+0\*16^11+9\*16^12+14\*16^13+14\*16^14+12\*16^15

Exercice 2:

a) Calculez l’espace total sur le disque dur (512 B par secteur)

Espace total =(792\*624+780\*1424\*2+760\*1680\*3+720\*1815\*4)\*512

b) Calculez le taux de lecture moyenne.

Taux de lecture moyenne ==5400 \*(792\*624+780\*1424\*2+760\*1680\*3+720\*1815\*4)\*512/60

c) Calculez le taux de lecture moyenne effective si le disque dur est connecté

avec un bus PCIe de vitesse 4000 Mb/s.

Taux de lecture effective =(Taux de lecture moyenne/5400) \*4000=

d) Changeriez-vous les résultats précédents si l’information sur le nombre de surfaces était disponible ?

Non ,

Exercice 3:

entete des instructions

1-

k:=8;

a<-(b-a)\*k;

op:=5:

//vrai

R[k]<-8;

SUBMUL(:=op = 5) -> R[a] <- R[b] - R[a];

R[a]<-R[a]\*R[k];

2-

DECREM(:=op = 7) -> R[a] <- R[b] - 1: R[b] <- R[b] - 1;

Exercice 4:

1)a)

Ajoutregistre3(:=op=4a)->R[1]<-Mem2[R[3]]+R[3];

//passons a l'encodage

IR<31..24>=4A//le circuit de controle effectue une addition

IR<23..21>=1//ce bus envoie le signal 1 pour ecrire dans le registre 1

IR<20..18>=3//ce bus envoie le signal 3 pour lire dans le registre 3

IR<17..15>=0//inutilise

IR<14..13>=0//inutilise

IR<12..0>=0//inutilise

4A 19 00 00 // encodage en little-endian

b)

RTN concret;

-recherche d'instructions//le prof a dit qu'il n'etait pas necessaire de faire la lecture d'instruction

-Execution de l'instruction

//envoi de l'adresse a la bascule//envoi de ladresse a la memoire 2

T<-R[IR<20..18>]; // sauvegarde de la valeur du registre dans T

Y<-Mem2[T]; // lecture de la valeur de la memoire2 a l'adresse T, assignation a Y

R[1]<-Y+T; // addition fait par ual et ecriture dans le registre a l'adresse r1

c)Liste de valeurs des signaux de controle

T<-R[IR<20..18>] ;

nom des signaux valeurs

A 0

B 1

C 0

D 0

E 1

F 0

G 0

UAL 0a

ecrireEIP 0

ecrireT 1

ecrireRegistre 0

Y<-Mem2[T];

nom des signaux valeurs

A 0

B 1

C 0

D 0

E 1

F 1

G 0

UAL 0a

ecrireEIP 0

ecrireT 0

ecrireRegistre 0

R[1]<-Y+T;

nom des signaux valeurs

A 0

B 0

C 0

D 0

E 1

F 0

G 0

UAL 4a

ecrireEIP 0

ecrireT 0

ecrireRegistre 1

2)

a-

IR<31..24>=4A//le circuit de controle effectue une addition

IR<23..21>=1//ce bus envoie le signal 1 pour ecrire dans le registre 1

IR<20..18>=3//ce bus envoie le signal 3 pour lire dans le registre 3

IR<17..15>=0//inutilise

IR<14..13>=0//inutilise

IR<12..0>=0//inutilise

r1 <- (Memoire2 [r3] + 0x23) >> r2

IR<24…31> = 0x5A = 0101 1010

IR<23…21> = r1 = 001

IR<20…18>= r2 = 010

IR<17…15> = r3 = 011

IR<14…13> = 00

IR<0…12> = 0x23 = 0 0000 0010 0011

IR<0…31> = 0101 1010 0010 1001 1000 0000 0010 0011 = 0x5A 29 80 23

Notre instruction sera donc : 0x23 80 29 5A en little endian

===============================

T <- R[IR<17…15>]

Y <- M[T] :T <- Y;

T <- IR<12…0> + T;

T <- T >> R[IR<20…18>]

R[IR<23…21>] <- T

A B C D E F G ecriT ecriRegistre ecrireEIP UAL

T <- R[IR<17…15>];

0 10 0 0 1 0 0 1 0 0 0A

T <- M[T];

0 10 0 0 0 1 0 1 0 0 0A

T <- IR<12…0> + T;

0 10 0 1 0 0 0 1 0 0 4A

T <- T >> R[IR<20…18>]

0 01 0 0 1 0 0 1 0 0 11

R[IR<23…21>] <- T

0 00 0 0 0 0 0 0 1 0 11