­

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)

1. Le Tableau 1.1 donne une représentation de la conversion des valeurs Fournis à la fois en binaire, complément à deux et décimale.

Table 1.1.1 Conversion

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | Numéros | Binaire | Complément à deux | Valeur décimale |
| a | 11001101 | 1100 1101 | 0011 0011 | 205 |
| b | 01101011 | 0110 1011 | 1001 0101 | 107 |
| c | 5726 | 0101 0111 0010 0110 | 1010 1000 1101 1010 | 22310 |
| d | FADE | 1111 1010 1011 1110 | 0000 0101 0100 0010 | 64222 |
| e | 10000000 | 10000000 | 10000000 | 128 |

1. Le tableau 1.2 montre les différentes bases possibles qui pourrais représenter le numéro proposé

Table 1.2 Bases possibles pour les numéros proposés

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID | Numéro | 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 |

1. Explication de y = x & (5 << 4)

En c “<<” est utilisé pour décaler les bits vers la gauche, ce qui correspond à une multiplication par 2 autant de fois que la valeur à droite de l’expression.

Exemple : (5<<4) 0101<<0000 =0101 0000 (sur 8bits)

La fonction retourne les bits à la position (5 et 7) d'un vecteur x, c'est un masque.

1. Le tableau 1.3 affiche la conversion des nombres entiers en base décimale vers la base Hexadécimale est en complément à 2

Table 1.3 Conversion de base décimale vers hexadécimal et binaire à 16bits

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Numéros | Complément à deux | Hexadécimal |
| a | 9876 | 0010 0110 1001 0100 | 2694 |
| b | 64 | 0000 0000 0100 0000 | 40 |
| c | 12345 | 0011 0000 0011 1001 | 3039 |

1. Opérations arithmétiques

8B+6A =1000 1101 + 0110 1010 = 1111 0110 = F 6

Il n’y a pas de débordement.

52 + 49 = 0101 0010 + 0100 1001 = 1001 1011 = 9B

Il y a un débordement car la somme de 2 nombres positifs donne un nombre négatif

1. Little Endian, big Endian

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

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

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

b) Calculez le taux de lecture moyenne.

Taux de lecture moyen = 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 ,

1. SUBMUL Ra, Rb, k

op := 5:

R[k]<-8;

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

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

2) DECREM Ra, Rb

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

1. Écrivez un encodage possible (en hexadécimal) de l’instruction

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

IR<31..24>=4A : le circuit de control effectue une addition

IR<23..21>=1 : ce bus envoie le signal 1 pour écrire 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

Nous obtenons alors : 4A 19 00 00 (encodage en little-endian)

1. RTN concret des macro-instructions

-Recherche d'instructions (pas nécessaire de faire comme demandé)

-Exécution de l'instruction

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

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

R[1] <- Y + T; // addition faite par UAL et écriture dans le registre à l'adresse r1 du registre

1. Liste de valeurs des signaux de contrôle

|  |  |
| --- | --- |
| 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 | 1 |
| 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