**Présentation du micro-ordinateur (Partie 2)**

2.Gère les communications avec ses périphériques (composantes qui l’entourent et qui l’aident pour exécuter les programmes).

Le CPU n’exécute pas lui-même toutes les instructions des programmes. Il va exécuter les opérations mathématiques (add, sous,mul, div, exp, racine carrée….), les tests et les boucles.

Le reste (entrées (cin) ,sorties (cout) ,accès aux fichiers (disque dur….) sont faits par les périphériques. Le CPU va communiquer avec les périphériques grâce aux contrôleurs qui sont des processeurs spécialisés. Les contrôleurs sont représentés par des rectangles bleus sur le schéma suivant.

Exemple : instruction cout << somme;

|  |  |  |  |
| --- | --- | --- | --- |
| CPU |  | RAM |  |
| |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | |  | |  | | --- | |  | | cout << somme; | |  | |  | |  | |  |   UAL  Unité de contrô |  | |  |  |  | |  |  | |  | |  | | --- | |  | | |  | |  | |  | |  |  | | --- | --- | |  |  | | . . . |  | | cout << somme; |  | | . . . |  | |  |  | |  |  | |  |  | |  |  |   L’unité de contrôle s’aperçoit que l’instruction doit être exécutée par l’écran. Elle envoie donc un signal au contrôleur d’écran pour qu’il affiche le contenu da la variable **somme**. | programme |
|  |  |
|  | variables |

Clavier

Disque dur

Écran

**Exercices récapitulatifs**

1. Le tableau de gauche vous donne des adresses et des contenus en RAM (les adresses et les contenus sont en hexadécimal). Le tableau de droite vous informe sur les types des variables contenues à ces adresses.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | **Adresse** | Contenu | | **1AE7** | 10 | | **1AE8** | AF | | **1AE9** | FF | | **1AEA** | FF | | **1AEB** | FF | | **1AEC** | FF | | **1AED** | 01 | | **1AEE** | 12 | | **1AEF** | 00 | | **1AF0** | FF | | **1AF1** | 4D | | **1AF2** | 45 | | **1AF3** | 28 | | **1AF4** | 63 | | **1AF5** | 68 | | **1AF6** | 61 | | **1AF7** | 74 | | **1AF8** | 2E | | **1AF9** | 12 | | **1AFA** | 15 | |  | |  | | --- | | Adresse Type | | | 1AE8 Entier signé 8 bits | | 1AEA Entier signé16 bits | | 1AEE Entier signé 32 bits | | 1AF2 Un caractère | | 1AF4 Suite de caractères se terminant par un point | |

1. SI on demande à l’ordinateur d’afficher le contenu des adresses 1AE8,1AEA,1AEE,1AF2 et 1AF4, que sera-t-il affiché ?

1AE8 : AF 🡺 -81 | 1AEA : FFFF 🡺 -1 | 1AEE : 1200FF4D 🡺 302055245 | 1AF2 : 45 🡺 E | 1AF4 : 63 68 61 74 2E 🡺 chat.

1. On vous dit que deux entiers de 16 bits sont emmagasinés un à la suite de l'autre. Si l'adresse du premier est 1AFE, quelle est l'adresse du second ?

1 : 1AFE et 1AFF | 2 : 1B00 et 1B01

1. Quelles sont les adresses des 2 dernières cases mémoire que vous voyez dans RAM ci-dessus ? (celles qui contiennent 1216 et 1516)

1AF9 et 1AFA

1. Soient les adresses et les contenus mémoire suivants : (les adresses et les contenus sont en hexadécimal)

|  |  |
| --- | --- |
| **Adresse** | Contenu |
| **………** |  |
| **C2A7** | 10 |
| **C2A8** | 00 |
| **C2A9** | F2 |
| **C2AA** | 48 |
| **C2AB** | 44 |
| **C2AC** | 96 |
| **C2AD** | 5A |
| **C2AE** | 40 |
| **C2AF** | 40 |
| **C2B0** | 12 |
| **C2B1** | CD |
| **C2B2** | C6 |
| **C2B3** | 9A |
| **C2B4** | 94 |

1. Si on vous dit qu’à l’adresse C2AB, il y a un réel point flottant 32 bits, quelle sera la valeur affichée en base 10 ?

C2AB : 44 96 5A 40 🡺 1202.82

1. Si on vous dit qu’à l’adresse C2AB, il y a un entier signé 8 bits, quelle sera la valeur affichée en base 10 ?

C2AB : 44 🡺 68

1. Si on vous dit qu’à l’adresse C2AB, il y a un entier signé 16 bits, quelle sera la valeur affichée en base 10 ?

C2AB : 44 96 🡺 17558

1. Si on vous dit qu’à l’adresse C2AB, il y a un entier non signé 8 bits, quelle sera la valeur affichée en base 10 ?

C2AB : 44 🡺 68

1. Si on vous dit qu’à l’adresse C2AB, il y a un caractère, quelle sera le caractère affiché ?

C2AB : 44 🡺 D

1. Quels sont les deux caractères codés à partir de l'adresse C2AD ?

CSAD : 5A 40 🡺 Z @

1. Soient les adresses et les contenus mémoire suivants : (les adresses et les contenus sont en hexadécimal)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Adresse** | Contenu | **Adresse** | Contenu | **Adresse** | Contenu |
| **………** |  | **2AE7** | 10 | **2AF6** | 6E |
| **2AD9** | 56 | **2AE8** | AF | **2AF7** | DB |
| **2ADA** | 8A | **2AE9** | FF | **2AF8** | D9 |
| **2ADB** | 00 | **2AEA** | C8 | **2AF9** | A0 |
| **2ADC** | 10 | **2AEB** | FF | **2AFA** | 43 |
| **2ADD** | FF | **2AEC** | FF | **2AFB** | 6F |
| **2ADE** | 41 | **2AED** | 01 | **2AFC** | 75 |
| **2ADF** | 9F | **2AEE** | E2 | **2AFD** | 72 |
| **2AE0** | C0 | **2AEF** | 00 | **2AFE** | 73 |
| **2AE1** | 00 | **2AF0** | FF | **2AFF** | 42 |
| **2AE2** | BA | **2AF1** | FF | **2B00** | 31 |
| **2AE3** | 18 | **2AF2** | 15 | **2B01** | 32 |
| **2AE4** | 79 | **2AF3** | 28 | **2B02** | 2E |
| **2AE5** | 5C | **2AF4** | 8D | **2B03** | 78 |
| **2AE6** | 5D | **2AF5** | 12 | **2B04** | 22 |

1. Si on vous dit qu’à l’adresse 2ADE, il y a le contenu d’une variable de type FLOAT, quelle sera la valeur affichée en base 10 si on demande de l’imprimer ?

FLOAT = 4 octets = 32 bits donc 419FC000 = 19,96875

1. Si on vous dit qu’à l’adresse 2ADE, il y a le contenu d’une variable de type LONG, quelle sera la valeur affichée en base 10 ?

LONG = 64 bits mais uniquement nombre entier donc 419FC000BA18795C = 4,728709243 \* 10^18

1. Si on vous dit qu’à l’adresse 2AEE, il y a le contenu d’une variable de type UNSIGNED SHORT, quelle sera la valeur affichée en base 10 ?

UNSIGNED SHORT = 16 BITS uniquement entier positif = E200 = 57856

1. Si on vous dit qu’à l’adresse 2AEE, il y a le contenu d’une variable de type SHORT, quelle sera la valeur affichée en base 10 ?

SHORT = 16 bits entier = E200 = -7680

1. Quels sont les deux caractères codés à partir de l'adresse 2AFB ?

6F = o 75 = u

1. On vous dit que deux valeurs de type « FLOAT » sont emmagasinées une à la suite de l'autre. Si l'adresse de la première est 2AE6, quelle est l'adresse de la seconde ?

5D10AFFF donc l’adresse de la seconde est 2AEA qui est égal à C8FFFF01

1. Quelles sont les adresses des 8 dernières cases mémoire ?

Voir tableau

**Solutions**

1.

|  |  |
| --- | --- |
| **Adresse** | Contenu |
| **1AE7** | 10 |
| **1AE8** | AF |
| **1AE9** | FF |
| **1AEA** | FF |
| **1AEB** | FF |
| **1AEC** | FF |
| **1AED** | 01 |
| **1AEE** | 12 |
| **1AEF** | 00 |
| **1AF0** | FF |
| **1AF1** | 4D |
| **1AF2** | 45 |
| **1AF3** | 28 |
| **1AF4** | 63 |
| **1AF5** | 68 |
| **1AF6** | 61 |
| **1AF7** | 74 |
| **1AF8** | 2E |
| **1AF9** | 12 |
| **1AFA** | 15 |

1. Contenu de 1AE8 : type entier signé 8 bits, donc AF

AF = 10101111 🡪 bit de signe = 1, donc négatif

son complément est 01010001 et 010100012 = 8110

Ce qui sera affiché : -81

Contenu de 1AEA : type entier signé 16 bits, donc FFFF

FFFF = 1111 1111 1111 1111 🡪 bit de signe = 1, donc négatif

🡪 son complément est 0000 0000 0000 0001

et 0000 0000 0000 00012 = 110

Ce qui sera affiché : -1

Contenu de 1AEE : type entier signé 32 bits, donc 1200FF4D

1200FF4D = 0001 0010 0000 0000 1111 1111 0100 1101 🡪 bit de signe = 0, donc positif

et 0001 0010 0000 0000 1111 1111 0100 11012 = 302 055 24510

Ce qui sera affiché : 302 055 245

Contenu de 1AF2 : type caractère, donc 45

Dans la table Ascii, 4516 = E

Ce qui sera affiché : E

Contenu de 1AF4 : suite de caractères se terminant par un point

Dans la table Ascii, le point = 2E

Donc, la suite de caractères est en RAM: 63 68 61 74 2E

Ce qui sera affiché : chat.

1. Entier de 16 bits = 2 octets (donc 2 cases en RAM)

Premier entier de 16 bits : cases 1AFE ET 1AFF

Second entier de 16 bits : cases 1B00 et 1B01, donc à l’adresse 1B00

1. 1AF9 et 1AFA

2.

|  |  |
| --- | --- |
| **Adresse** | Contenu |
| **………** |  |
| **C2A7** | 10 |
| **C2A8** | 00 |
| **C2A9** | F2 |
| **C2AA** | 48 |
| **C2AB** | 44 |
| **C2AC** | 96 |
| **C2AD** | 5A |
| **C2AE** | 40 |
| **C2AF** | 40 |
| **C2B0** | 12 |
| **C2B1** | CD |
| **C2B2** | C6 |
| **C2B3** | 9A |
| **C2B4** | 94 |

1. Contenu de C2AB : type réel point flottant 32 bits, donc 44965A40

44965A40 = 0100 0100 1 001 0110 0101 1010 0100 0000

Exposant : 100010012 = 13710

137-127 = 10

Mantisse : 1.001 0110 0101 1010 01

Forme normalisée : 1.001011001011010 01 \* 210

Ce qui donne : 10010110010. 1101001

Partie entière : 100101100102 = 120210

Partie fractionnaire : .11010012 = .5 + .25 + .0625 + .00781 = .8203110

Ce qui sera affiché : 1202.82

1. Contenu de C2AB  : type entier signé 8 bits, donc 44

4416 = 0100 01002  🡪 bit de signe = 0, donc positif

et 0100 01002= 6810

Ce qui sera affiché : 68

1. Contenu de C2AB  : type entier signé 16 bits, donc 4496

449616 = 0100 0100 1001 01102  🡪 bit de signe = 0, donc positif

et 0100 0100 1001 01102  = 17 55810

Ce qui sera affiché : 17 558

1. Contenu de C2AB  : type entier non signé 8 bits, donc 44

🡪 tous les entiers non signés sont positifs, on peut convertir directement en base 10 (il n’est pas nécessaire de vérifier le bit de signe en binaire…)

4416  = 6810

Ce qui sera affiché : 68

1. Contenu de C2AB  : type caractère, donc 44

4416 dans la table Ascii = le caractère D

Ce qui sera affiché : D

1. 5A16 et 4016

5A16 dans la table Ascii = le caractère Z

4016 dans la table Ascii = le caractère @

Ce qui sera affiché : Z et @

3.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Adresse** | Contenu | **Adresse** | Contenu | **Adresse** | Contenu |
| **………** |  | **2AE7** | 10 | **2AF6** | 6E |
| **2AD9** | 56 | **2AE8** | AF | **2AF7** | DB |
| **2ADA** | 8A | **2AE9** | FF | **2AF8** | D9 |
| **2ADB** | 00 | **2AEA** | C8 | **2AF9** | A0 |
| **2ADC** | 10 | **2AEB** | FF | **2AFA** | 43 |
| **2ADD** | FF | **2AEC** | FF | **2AFB** | 6F |
| **2ADE** | 41 | **2AED** | 01 | **2AFC** | 75 |
| **2ADF** | 9F | **2AEE** | E2 | **2AFD** | 72 |
| **2AEO** | C0 | **2AEF** | 00 | **2AFE** | 73 |
| **2AE1** | 00 | **2AF0** | FF | **2AFF** | 42 |
| **2AE2** | BA | **2AF1** | FF | **2B00** | 31 |
| **2AE3** | 18 | **2AF2** | 15 | **2B01** | 32 |
| **2AE4** | 79 | **2AF3** | 28 | **2B02** | 2E |
| **2AE5** | 5C | **2AF4** | 8D | **2B03** | 78 |
| **2AE6** | 5D | **2AF5** | 12 | **2B04** | 22 |

1. Contenu de 2ADE : type FLOAT, donc 419FC000

Ce qui sera affiché : 19.96875

1. Contenu de 2ADE  : type LONG, donc 419FC000BA18795C

419FC000BA18795C 16 =

0100 0001 1001 1111 1100 0000 0000 0000 1011 1010 0001 1000 0111 1001 0101 11002  🡪 bit de signe = 0, donc positif

0100 0001 1001 1111 1100 0000 0000 0000 1011 1010 0001 1000 0111 1001 0101 11002  =

4 728 709 243 117 009 24410

Ce qui sera affiché : 4 728 709 243 117 009 244

1. Contenu de 2AEE : type UNSIGNED SHORT, donc E200

🡪 tous les entiers non signés sont positifs, on peut convertir directement en base 10 (il n’est pas nécessaire de vérifier le bit de signe en binaire…)

E20016  = 57 85610

Ce qui sera affiché : 57 856

1. Contenu de 2AEE : type SHORT, donc E200

E200 16 = 1110 0010 0000 00002

1110 0010 0000 00002  🡪 bit de signe = 1, donc négatif

🡪 son complément est 0001 1110 0000 00002

et 0001 1110 0000 00002  = 7 68010

Ce qui sera affiché : 7 680

1. 6F16 et 7516

6F16 dans la table Ascii = le caractère o

7516 dans la table Ascii = le caractère u

Ce qui sera affiché : o et u

1. Type FLOAT = 4 octets (donc 4 cases en RAM)

Première valeur de 32 bits : cases 2AE6,2AE7,2AE8 et 2AE9

Seconde valeur de 32 bits : cases 2AEA,2AEB,2AEC et 2AED, donc à l’adresse 2AEA

1. Adresses des 8 dernières cases mémoire :

|  |
| --- |
| **2AFD** |
| **2AFE** |
| **2AFF** |
| **2B00** |
| **2B01** |
| **2B02** |
| **2B03** |
| **2B04** |