
Correction du Contrôle Continu du 20 Octobre 2006
Architecture des Ordinateurs

Exercice 1 -(5 pts) -

On choisit la convention d'appel (*calling convention*) du langage C.

1. l'état de la pile lors de l'appel est donc le suivant :

ESP			
ESP-4	d	EBP+16	
ESP-8	v[]	EBP+12	adresse du tableau v
ESP-12	w[]	EBP+8	adresse du tableau w
ESP-16	adr ret	EBP+4	adresse de retour du sous-programme
ESP-20	anc EBP	EBP	sauvegarde de EBP

2. la traduction en assembleur de la fonction est :

```
; On utilise
; ESI <-> v
; EDI <-> w
; ECX <-> i
; EBX <-> sum
; on n'oubliera pas que la division en 32 bits agit sur EDX:EAX

int_vector_div:
    PUSH EBP          ; sauvegarde de EBP
    MOV  EBP,ESP      ; mise a jour de EBP pour recuperer les parametres
    XOR  ECX,ECX      ; i <- 0
    XOR  EBX,EBX      ; sum <- 0
    MOV  ESI,[EBP+12]
    MOV  EDI,[EBP+8]
while_begin:
    MOV  EAX,[ESI+ECX*4] ; EAX <- v[i]
    CMP  EAX,0
    JE   while_end      ; si v[i] == 0 alors sortir de la boucle
    XOR  EDX,EDX
    DIV  DWORD [EBP+16] ; EAX <- v[i] / d
    MOV  [EDI+ECX*4],EAX ; w[i] <- v[i] / d
    ADD  EBX,EAX        ; sum <- sum + w[i]
    INC  ECX            ; ++i
    JMP  while_begin
while_end:
    MOV  EAX,EBX        ; mettre le resultat dans EAX
    MOV  ESP,EBP
    POP  EBP
    RET
```

Exercice 2 -(5 pts) -

On rappelle que dans la notation IEEE 754, les nombres flottants (réels) sont codés sur 32 bits et sont représentés sous la forme SEM , où :

- S est le signe sur 1 bit (0 = positif, 1 = négatif)
- E est l'exposant décalé = exposant + 127, codé sur 8 bits
- M est la mantisse normalisée amputée du premier 1 représentant la mantisse exacte, elle est codée sur 23 bits

1. coder en notation IEEE 754 le nombre 5,5 et donner sa représentation hexadécimale :

$5,5_{10} = 101,1_2$, la normalisation conduit à $1,011E_2^2$

- $S_1 = 0$: il s'agit d'un nombre positif
- $E_1 = 127 + 2 = 129 = 1000.0001_2$
- $M_1 = 011$

On obtient donc la représentation :

$$\begin{array}{ccc} S_1 & E_1 & M_1 \\ \hline 0 & 1000.0001 & 11000000000000000000000 \end{array}$$

soit $40B00000_{16}$

2. coder en notation IEEE 754 le nombre 20,25 et donner sa représentation hexadécimale :

$20,25_{10} = 10100,01_2$, la normalisation conduit à $1,010001E_2^4$

- $S_2 = 0$: il s'agit d'un nombre positif
- $E_2 = 127 + 4 = 131 = 1000.0011_2$
- $M_2 = 010001$

On obtient donc la représentation :

$$\begin{array}{ccc} S_2 & E_2 & M_2 \\ \hline 0 & 1000.0011 & 01000100000000000000000 \end{array}$$

soit $41A20000_{16}$

3. montrer comment faire pour multiplier ces 2 nombres et obtenir le résultat à partir de leur représentations IEEE 754 :

On considère deux nombres en notation IEEE 754 : $S_1E_1M_1$ et $S_2E_2M_2$ et on tente de déterminer comment évaluer $S_3E_3M_3$ qui est le résultat de leur produit. Si on représente $5,5 \times 20,25 = 111,375$, on obtient :

$111,375_{10} = 1101111,011_2$, la normalisation conduit à $1,101111011E_2^6$

- $S_3 = 0$: il s'agit d'un nombre positif
- $E_3 = 127 + 6 = 133 = 1000.0101_2$
- $M_3 = 101111011$

On obtient donc la représentation :

$$\begin{array}{ccc} S_3 & E_3 & M_3 \\ \hline 0 & 1000.0101 & 10111101100000000000000 \end{array}$$

soit $42DEC000_{16}$

- visiblement $S_3 = S_1 \oplus S_2$
- l'exposant résultat est $E_3 = (E_1 - 127) + (E_2 - 127) + 127 = E_1 + E_2 - 127$
- la mantisse est le résultat de 1, $M_3 = 1, M_1 \times 1, M_2$

Exercice 3 -(5 pts) - On se place dans le cadre de la notation binaire en complément à 2 sur 8 bits (NBC2).

1. coder la valeur -40 : 1111.1011_2
2. coder la valeur -5 : 1100.1000_2
3. effectuer le produit de ces 2 valeurs et montrer comment on obtient le résultat final :

```

      1111 1011
    x 1100 1000
    -----
                                0
+                                0.
+                                0..
+      111 1101 1...
+    1111 1011 ....
+              0. ....
+  111110 11.. ....
+ 1111101 1... ....
    -----
      1100 1000

```

$-40 \times -5 = +200$, or on obtient 1100.1000_2 qui est un nombre négatif et qui correspond à -56 en NBC2 sur 8 bits. Effectivement, en NBC2 sur 8 bits, on ne peut représenter que des nombres entre -128 et $+127$. Si on essaye de coder 128 , on obtient -128 car on boucle. Or $(200 - 128) = 72$ et $(-128 + 72) = -56$.

1)	Le terme RISC signifie ☐ Reduced Innovation Set Computer ☐ Reduced Internal Set Computer ☐ Reduced Improvement Set Computer ☒ Reduced Instruction Set Computer
2)	la valeur hexadécimale $1FF_{16}$ est aussi égale à ☒ $200_{16} - 1_{16}$ ☐ $FF_{16} + FF_{16}$ ☒ $142_{16} + BD_{16}$ ☐ $17A_{16} + 75_{16}$
3)	Le 8086 possède ☐ 4 registres généraux ☒ 8 registres généraux ☒ 4 registres de segment ☐ 8 registres de segment
4)	L'hyperthreading ☐ divise de temps d'exécution par 2 d'un programme ☒ permet de simuler 2 processeurs avec un seul coeur ☐ supprime les états d'attente ☐ permet une meilleure prédiction de branchement
5)	Quelles instructions permettent une multiplication par 4 de EAX ☐ shr EAX,3 ☒ mul dword 4 ☒ shl EAX,2 ☐ shl EAX,1
6)	Un cache de niveau 2 (L2) ☐ est 2 fois plus important (en taille) qu'un cache L1 ☒ a une taille plus importante qu'un cache L1 ☐ a une taille plus importante qu'un cache L3 ☐ permet une meilleure prédiction de branchement
7)	Quel(s) processeur(s) appartiennent à la gamme Intel ☒ Celeron ☐ Duron ☒ Pentium 4 ☐ Athlon 64
8)	Pour l'Intel 8086 ☒ le registre AL a une taille de 8 bits ☐ le registre ESI a une taille de 16 bits ☐ la multiplication modifie les registres AX et BX ☐ BP est le pointeur de pile
9)	La mémoire dynamique (DRAM) doit être rafraichie ☐ car elle chauffe de manière trop importante ☐ car le dégagement de chaleur risque d'endommager les circuits ☒ car les données ont tendance à s'effacer ☐ afin d'éviter des perturbations electromagnétiques avec la montée en fréquence
10)	Les registres de l'unité SSE possèdent une taille de ☐ 32 bits ☐ 64 bits ☐ 80 bits ☒ 128 bits

Note : la question 5 était ambiguë, je l'ai comptée bonne pour ceux qui n'auraient pas coché SHL EAX,2. En effet en assembleur 8086, il n'est pas possible d'écrire SHL AX,2, il faut utiliser

MOV CL,2, SHL AX,CL. Mais avec l'assembleur Pentium, on peut écrire SHL EAX,2.