

MI01 TP7b – Traitement d'image – Troisième partie.

1 Travail à réaliser

Le but est de reprendre l'algorithme de conversion en niveaux de gris que vous avez réalisé dans le TP4 et de le réimplémenter en utilisant des instructions MMX afin d'établir le gain éventuel de vitesse de traitement obtenu.

1.1 Mise en place

Reprenez le fichier *image.asm* dans lequel vous avez implémenté la conversion en niveaux de gris pour le TP5. Ajoutez la directive **.MMX** comme suit afin d'informer l'assembleur de l'usage d'instructions MMX :

```
; IMAGE.ASM
;
; MI01 - TP Assembleur 2 à 5
;
; Réalise le traitement d'une image 32 bits.

.686
.MMX
.MODEL FLAT, C

.DATA

.CODE

PUBLIC      process_image_asm

process_image_asm  PROC NEAR          ; Point d'entrée du programme

    push    ebp
    mov     ebp, esp

    push    ebx
    push    esi
    push    edi
...
...

    ; Libérer l'unité MMX
    emms

    pop     edi
    pop     esi
    pop     ebx

    pop     ebp

    ret                                ; Retour à la fonction MainWndProc

process_image_asm  ENDP

END
```

1.2 Programme MMX

Il s'agit simplement de réaliser la conversion en niveaux de gris en utilisant des calculs en virgule fixe sur des mots de 16 bits, de la même manière que dans le TP6, en traitant un pixel complet (d'un seul bloc de 32 bits) à chaque itération.

Afin de suivre le déroulement de votre programme, vous pouvez utiliser Turbo Debugger (TD32). Pour afficher le contenu des registres MMX, allez dans le menu *View→Numeric Processor*, puis en utilisant un clic du bouton droit de la souris, dans le menu sélectionnez *Show→MM Extension*. Vous pouvez changer la base d'affichage au moyen de ce menu, en sélectionnant *Radix*.

Sachant que les trois composantes R, V et B d'un pixel sont chacune représentées sur 8 bits, comment utiliser les instructions de transformation de données (PACK/PUNPCK) pour les transformer en un vecteur de mots de 16 bits ? Quelle opération supplémentaire doit-on réaliser ?

Par quel moyen réaliser la conversion proprement dite ? Faites attention aux décalages nécessaires avant de stocker le résultat.

Comparez la vitesse d'exécution du calcul MMX et du calcul C.

Remarques

Il n'existe pas d'instruction permettant de charger une valeur immédiate dans un registre MMX. Il faut donc passer par un registre intermédiaire. Par exemple, pour charger la valeur 0FFFFFFFh dans la partie basse du registre mm0, on utilisera le code suivant :

```
mov    eax, 0fffffffh
movd   mm0, eax
```

N'oubliez pas de libérer l'unité de calculs flottants à la fin de votre programme mmx, sous peine d'avoir des surprises lors du retour de sous-programme !

L'instruction PMADDWD considère des vecteurs de 4 mots de 16 bits **signés**.

Important : le compilateur Borland C s'attend à ce que les registres EBX, ESI et EDI soient préservés par les appels de sous-programmes.