

*Institut Supérieur des Etudes Technologiques en Communications de Tunis*

|  |  |
| --- | --- |
| **Enseignant :** | Nadia KHOUJA |
| **Niveau :** | ***Master*** |
| **Date :** | ***Mars 2016*** |
| **Nb pages :** | ***2*** |
| **Documents :** | ***autorisés*** |
|  |  |

**Groupe : Smart’Com1**

# Devoir surveillé

Digital Signal processor

**Durée 1h**

**Problème**

On se propose dans cet exercice d’écrire une fonction assembleur qui permet de convertir une image décrite en (Y, Cr, Cb) en image décrite en (R, G, B)

La conversion d’un pixel décrit en (Y, Cr, Cb) en pixel décrit en (R, G, B) se fait selon les équations suivantes :

R = Y + 1.402 (Cr - 128) = Y +(Cr-128) + 0.402(Cr-128)

G = Y - 0.34414 (Cb - 128) - 0.71414 (Cr - 128)

B = Y + 1.772 (Cb - 128) = Y + (Cb-128) + 0.772(Cb-128)

1. Proposer un schéma de conception de votre programme sur un processeur DSP BF-533
2. Donner le code assembleur permettant de réaliser cette fonction en vous aidant des commentaires donnés ci-dessous :

Les constantes 0.402 and -0.34414 sont 0x3374, 0xA498, respectivement.

Les constantes -0.71414 and 0.772 sont 0xD3F4, 0x62D0, respectivement.

:

* Rajouter la déclaration de buffer YCrCb , buffer de taille 8, chaque élément étant codé sur un octet
* Même question pour le buffer RGB.
* Interpréter les commentaires et compléter le code assembleur permettant de décrire la fonctionnalité voulue.

.section L1\_code;

.global \_\_YCbCrtoRGB;

\_\_YCbCrtoRGB:

// Initialize base register B1 and I1 for circular buffer (ce buffer servira à stoker les valeurs des constantes)

// Initialize length for circular buffer to 8

//mettre les valeurs 0x3374 (0.402) et -0.34414 (0xA498) dans R6

// Coefficients 0xD3F4 (-0.71414) and 0x62D0 (0.772) are stored in R7

// Address of input YCbCr array dans P0

// Address of output array to store RGB values dans P2

// Number of inputs N dans P1

// Initialize R4.H to 255 **and** fetch Y value **and** store coefficients 0.402 and -0,34414 to temp. location (buffer circulaire pointé par I1) (3 opérations en parallèle)

// Clear R5 **and** fetch Cb value **and** store coefficients -0.71414 **and** 0,772 in temp. location (buffer pointée par I1)

// Initialize R7 to 128

// R1 = Cb-128 **and** fetch Cr value

// Initialize R6 to maximum positive value (7FFF)

LSETUP(YCB\_STRT, YCB\_END) LC0 = P1;

YCB\_STRT:

// R2 = Cr-128

// Get Y value in A1 **and** A0 **and** fetch coefficients dans R3 (3 instructions)

// Multiply (Cr -128) value with coefficients 0.402 and -0.71414

// Add (Cr-128) value to A0 to get R value

// multiply (Cb-128) with -0.34414 and add to A1,

// A0= 0.772(Cb-128)

// Add Y value to A0 **and** store R value

// Leftshift to get B value in lower half **and** fetch next Y data

// Add (Cb-128) value to A0 **and** store B value **and** fetch next Cb data

// fetch next Cr value

YCB\_END:

// R1 = Cb-128 **and** store B data

RTS;

NOP; //to avoid one stall if LINK or UNLINK happens to be

//the next instruction after RTS in the memory.

\_\_YCbCrtoRGB.end: