

TP 1: Prise en main de CCS. Programmation d'une convolution

Objectifs : Se familiariser avec le logiciel Code Composer et examiner les outils d'édition de lien et de compilation d'un programme en assembleur.

Les TP seront réalisés en créant un projet à chaque TP. Voici comment créer un projet:

1- Ouvrir un nouveau projet qui sera nommé TPn.pjt. La création du projet entraîne la création d'un répertoire dans lequel vous enregistrerez les fichiers composant le projet. Ce projet nécessite au minimum deux fichiers:

- Le fichier d'édition de lien nommé TPn.cmd et qui précise l'utilisation souhaitée de la mémoire du DSP
- Le programme écrit en assembleur (TPn.asm) ou en C ((TPn.c)

Pour créer ces fichiers, il suffit d'exécuter *FILE -> NEW*, d'écrire les fichiers puis de les enregistrer dans le répertoire TPn.

2- En faisant un clic droit sur le nom du projet dans la fenêtre projet, sélectionner *Add files to project* et ajouter les deux fichiers créés. Puis enregistrer le projet *Project-> Save*.

3- Dans le menu projet → build options, sélectionner les options du compilateur et de l'éditeur de lien.

4- Compiler le projet avec l'instruction *rebuild all*. Si un des fichiers a été modifié on peut recompiler juste les fichiers modifiés à l'aide de l'instruction *incremental build*. La compilation fournit un fichier exécutable .o (.out) qu'on retrouve dans le répertoire DEBUG du répertoire projet.

Exercice n°1:





1- Créez le projet TP1. Ouvrez le fichier C6713dsk.cmd (qui se trouve sur Moodle) puis ajoutez le:

```
MEMORY
{
    IVECS: org=0h,      len=0200h
    IRAM:  org=0x00220, len=02FDE0h
    SDRAM: org=0x08000000, len=0100000h
    FLASH: org=0x09000000, len=020000h
}
```

SECTIONS

```
{
    .EXT_RAM : > SDRAM
    .vectors : > IVECS
    .text    : > IRAM
    .bss     : > IRAM
    .cinit   : > IRAM
    .stack   : > IRAM
    .sysmem  : > IRAM
    .const   : > IRAM
    .switch  : > IRAM
    .far     : > IRAM
    .cio     : > IRAM
    .csldata : > IRAM
}
```

2- Que précise ce fichier à votre avis?

- 3- Créez un fichier nommé add.c qui réalise la somme de $a=10$ et $b=a+5$. Le résultat sera affiché à l'écran.
- 4- Ajoutez les fichiers précédent au projet à l'aide d'un clic droit sur le nom du projet dans le manager de projet.
- 5- Dans le menu project, sélectionnez *Build options*. Il y a deux onglets à compléter:
 - onglet Compiler: écrire CHIP_6713 dans l'option -d
 - onglet Linker: vérifier qu'un fichier est nommé pour les options -o et -m. Ajouter rts6700.lib dans l'option -l.
- 6- Dans le menu Option/Customize cochez Load Program After Build de l'onglet Program/Project Load.
- 7- Compilez en cliquant sur l'icône Rebuild all.  Si tout ce passe bien, il n'y a pas d'erreur.
Remarque: Si vous modifiez un des fichiers du projet, vous pouvez recompiler seulement les changements en cliquant sur incremental build: 
- 8- Editez le fichier add.map et vérifiez l'allocation mémoire des variables et des différentes sections.
- 9- Editez les contenus mémoire en cliquant sur  ainsi que la fenêtre des registres en cliquant sur 
 Essayez de retrouver a et b dans la section .stack
- 10- Arrangez les fenêtre avec la commande *Window -> Tile Vertically*. Faites un clic droit dans la fenêtre du programme et sélectionnez *Mixed mode*. Expliquez le code 0200052A de l'instruction MVK.S2 0x000a,B4 en utilisant l'aide de MVK.
- 11- Exécuter le programme en mode pas à pas en observant la fenêtre des registres. En observant le plan mémoire, retrouvez les adresses de a, b et somme.
- 12- Remplacez $a=10$ par $a=-10$. Comment est modifié le code de a en mémoire. Expliquez grâce à l'aide de MVK.

Exercice n°2:

- 1- Créer une application qui effectue la multiplication terme à terme de deux tableaux $x=\{1,2,3,4\}$ et $y=\{0,2,4,6\}$. Le programme utilisera une fonction nommée conv et affichera le résultat.
- 2- Exécuter le programme en mode pas à pas après avoir lancer la commande Go main du menu Debug. Puis observer l'évolution de variables x, y et resultat. Pour cela faites un clic droit sur le nom de la variable et sélectionnez Add to Watch Window.
 Examinez la mémoire pour retrouver les variables x et y du programme.

3 Profilage: Enregistrez le projet et ouvrez le programme Setup CCSStudio v3.1. Ajoutez la configuration C6713 Device CycleAccurate Simulator. Ouvrez ensuite CCSStudio.
 Rechargez votre projet. Pour ouvrir une fenêtre de profilage, dans le menu Profile/setup, cliquez sur la montre afin de valider le profilage.

Puis cliquez sur Profile/Viewer

Placez un point d'arrêt sur l'accolade fermante du main.

Lancez l'exécution du programme.

Observez les nombres de temps de cycle moyen, max et min.

Exercice n°3:

Réaliser l'exercice 2 en assembleur. Vous vous intéresserez aux directives assembleur .def, .ref, .text, .short dans l'aide de CCS.