

**TP1 M41****Système NIOS II**

Nous utiliserons dans ce TP le logiciel **Intel FPGA Monitor Program** pour compiler, charger et exécuter les programmes d'application sur un système embarqué à base du microprocesseur NIOS II d'Intel. Le système est mis en œuvre en tant que circuit qui est téléchargé dans le FPGA Cyclone V. Ceci permettra que des programmes écrits en langage assembleur Nios II ou en langage C puissent être exécutés sur la carte DE1 SOC.

**Partie a**

Dans cette partie on va coder et exécuter un programme qui affiche des motifs mobiles sur les afficheurs 7 segments.

Procéder comme suit:

1. Ouvrir le logiciel *Intel Monitor Program*, qui conduit à la fenêtre de la figure 1

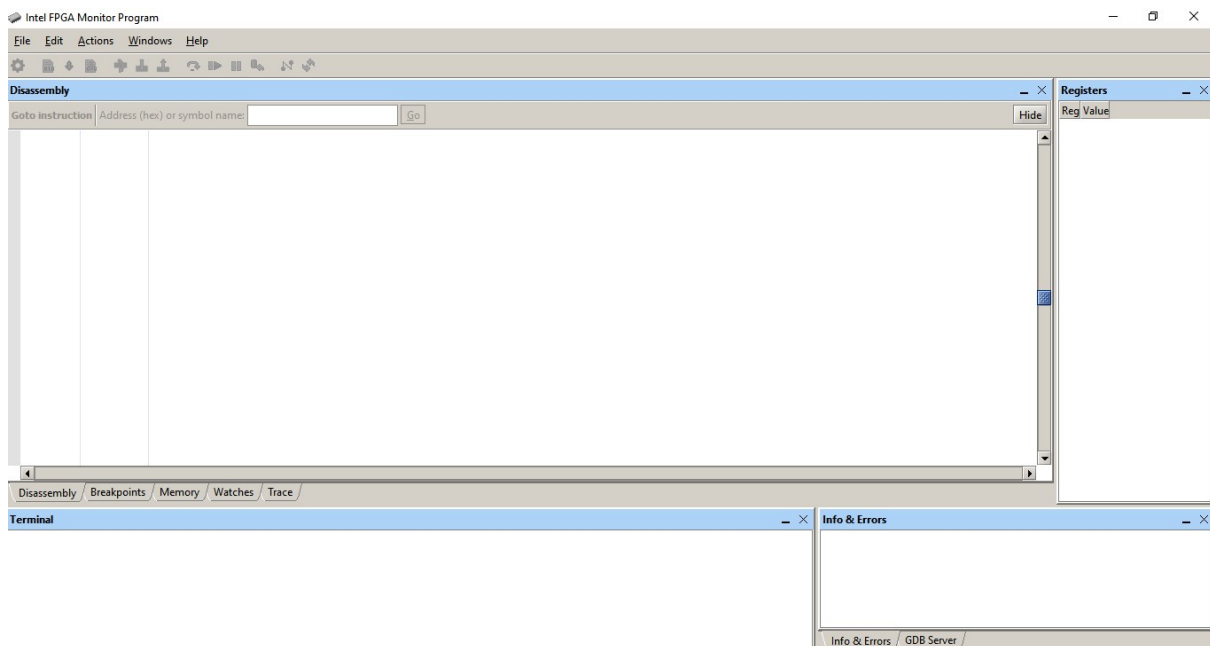


Figure 1

2. Pour exécuter un programme, il est nécessaire de créer un nouveau projet. Sélectionner *File> New Project* (figure 2a) pour atteindre la fenêtre dans la figure 2b. Choisir le dossier comme indiqué sur la figure 3 (C:\GE41\tp1a) et donner TP1a comme nom de projet. Sélectionner l'Architecture NIOS II. Cliquer sur « Next », pour atteindre la figure 3.

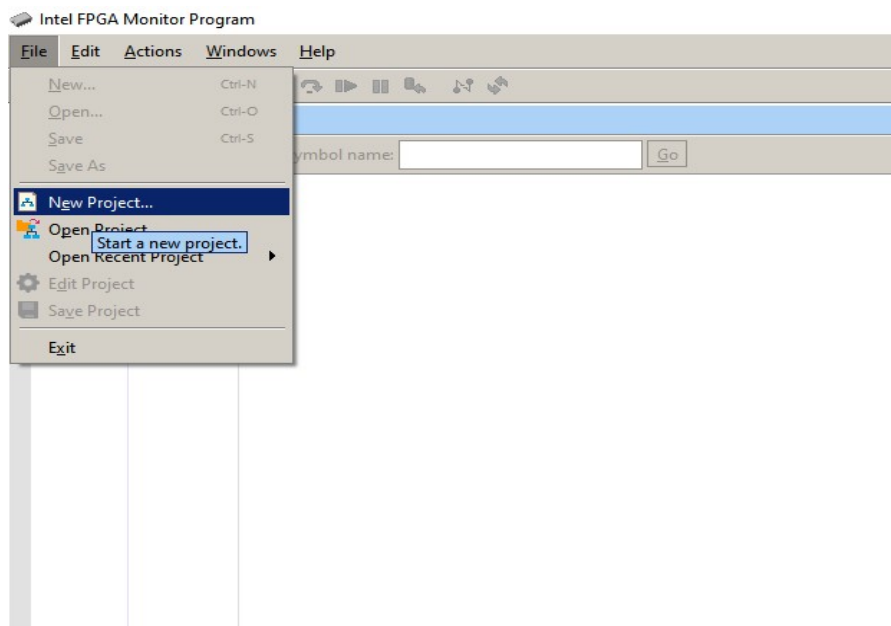


Figure 2a

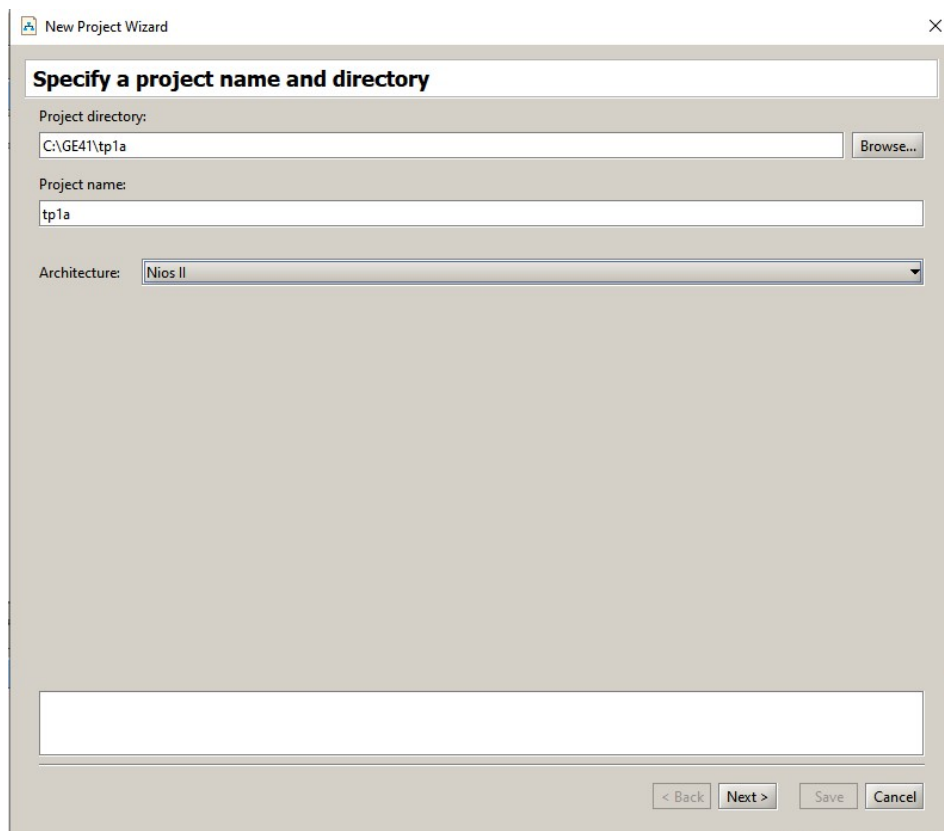


Figure 2b

3. Choisir le système « DE1 SOC Computer » dans la figure 3 et cliquer « Next »

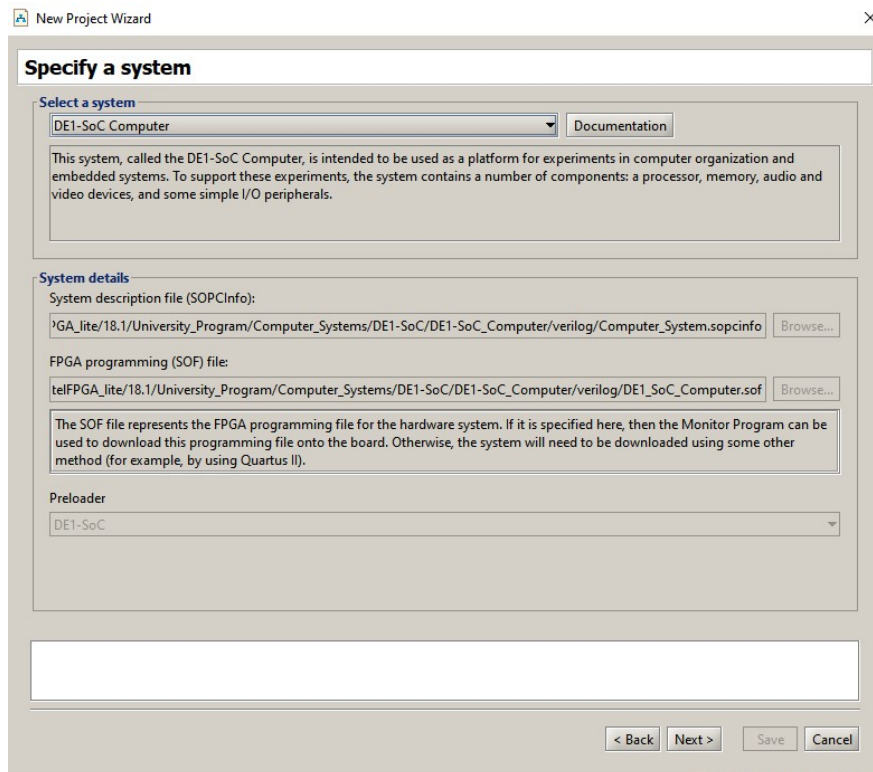


Figure 3

4. Sélectionner « Assembly program » dans la fenêtre « Specify a program type » de la figure 4

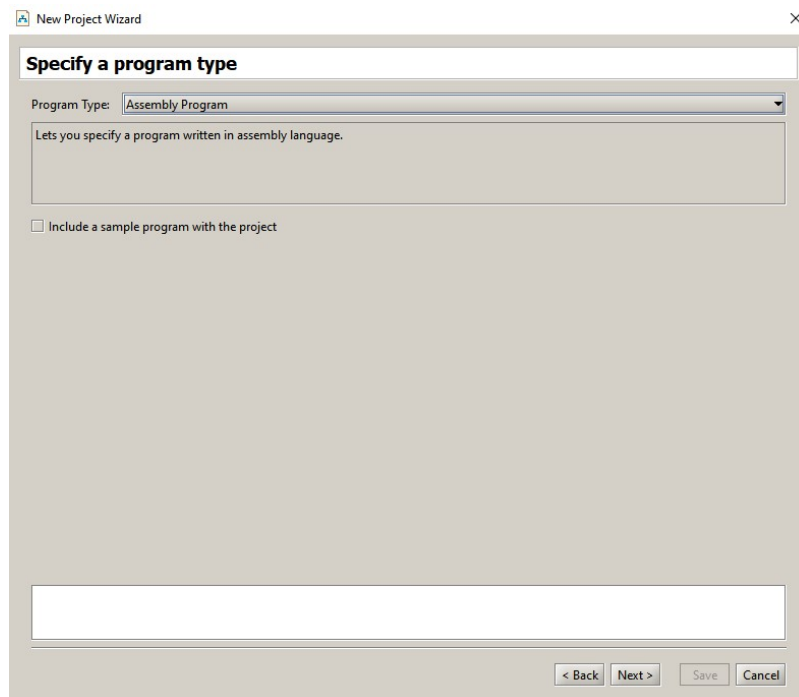


Figure 4

5. Cliquer « Add » dans la fenêtre « Specify program details » de la figure 5 et ajouter les fichiers address\_map\_nios2.s et getting\_started.s qui sont dans le dossier C:\GE41\tp1a. Cliquer « Next »

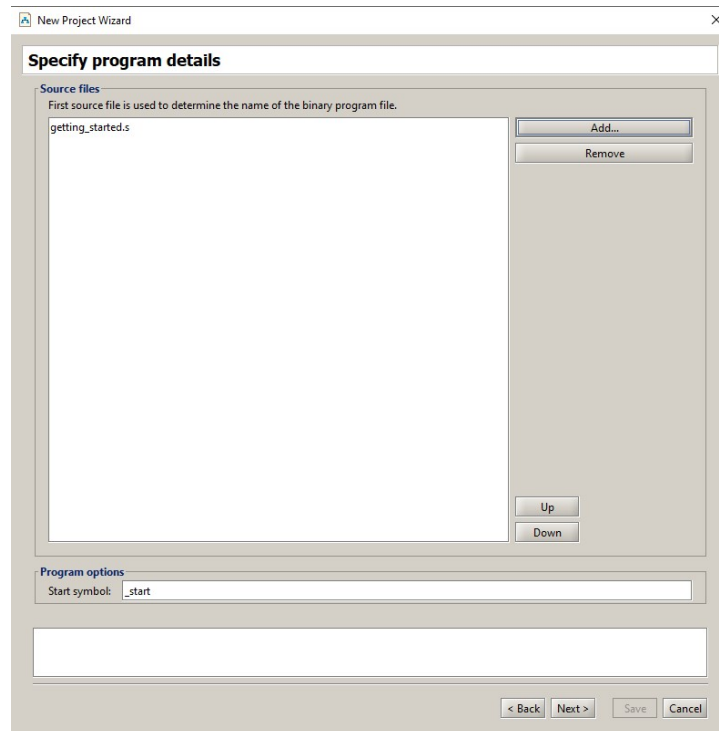


Figure 5

6. La fenêtre «Specify system parameters» doit être comme indiqué dans la figure 6

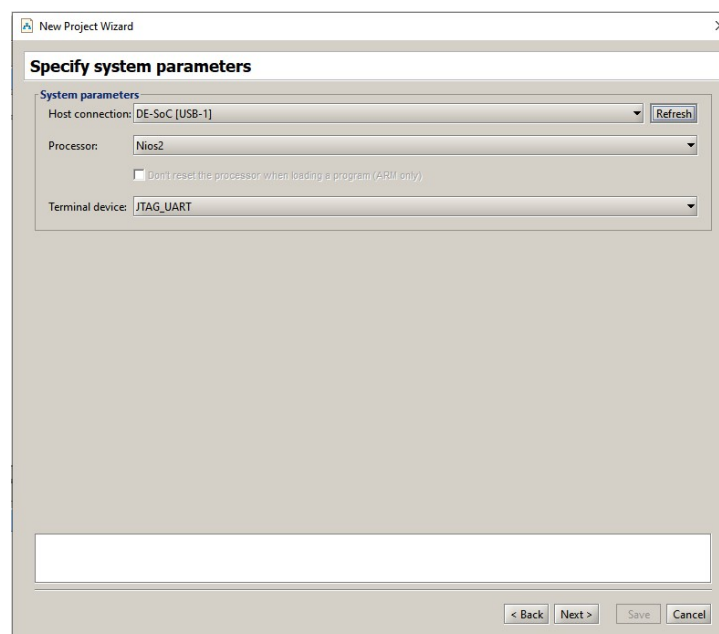


Figure 6

7. Cliquer sur « Save » dans la fenêtre « Specify program memory settings »

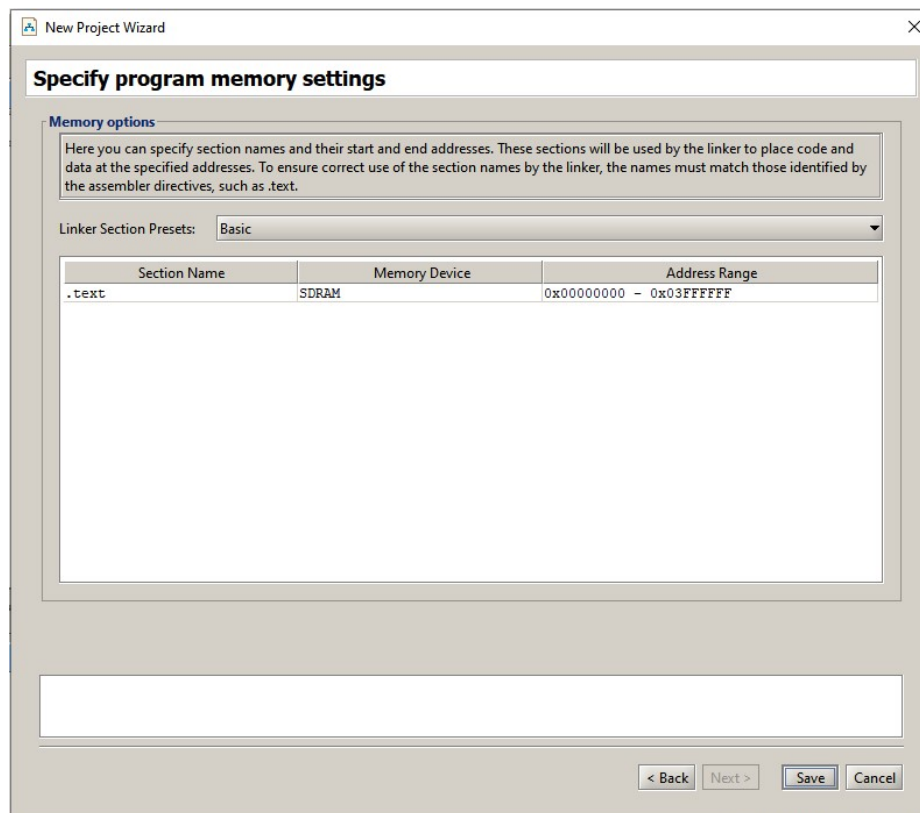


Figure 7

8. Cliquer « Oui » sur la fenêtre « Download System – Prompt » (figure 8)

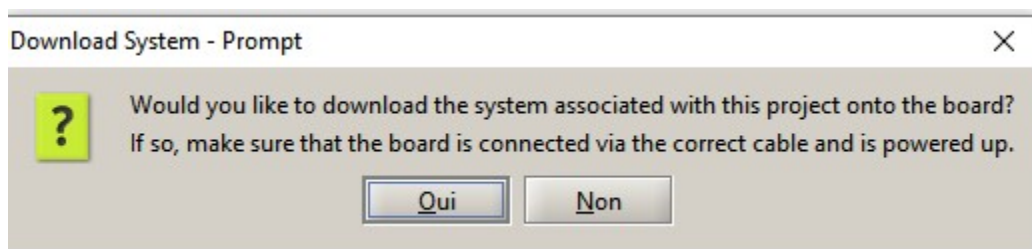


Figure 8

9. Cliquer « OK » sur la fenêtre de la figure 9

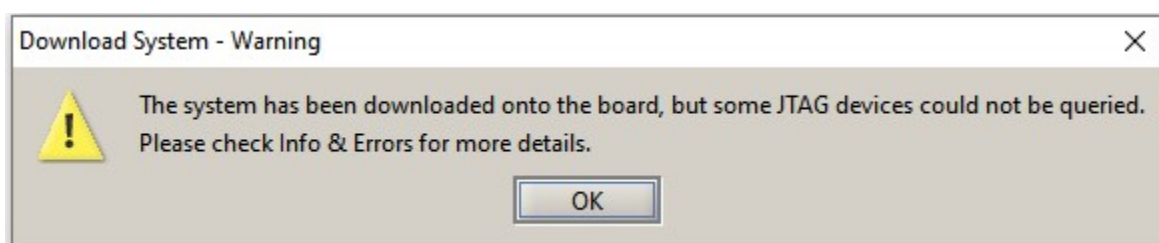


Figure 9

10. Dans le menu choisir « Actions → Compile» (figure 10)

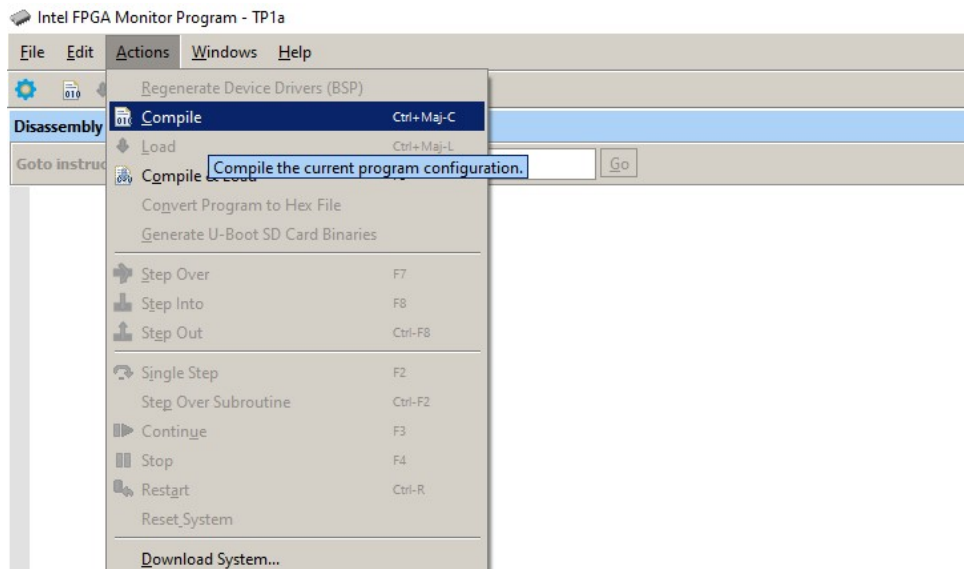


Figure 10

11. Choisir « Actions → Load» (figure 11)

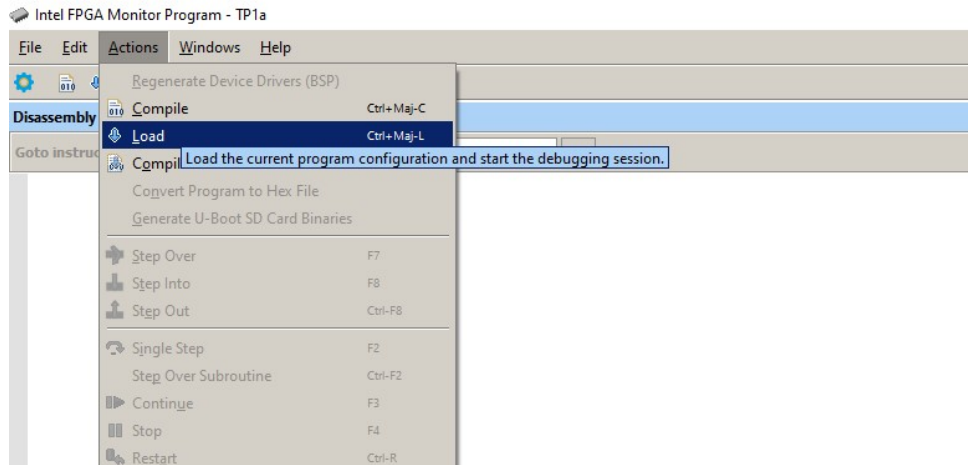


Figure 11a

Après téléchargement du programme, on obtient l'illustré dans la figure 11b

```

Intel FPGA Monitor Program - TP1a: getting_started.srec [Paused]
File Edit Actions Windows Help
Goto instruction Address (hex) or symbol name Go

Disassembly
/* executable code follows */
.global _start
_start:
/* initialize base addresses of parallel ports */
movia r15, SW_BASE /* SW slider switch base address */
_start:
0x00000000 03FFC834 orhi r15, zero, 0xFF20
0x00000004 78C01004 addi r15, r15, 0x40
0x00000008 043FC834 movia r16, zero, 0xFF20 /* red LED base address */
0x0000000C 84000004 orhi r16, zero, 0xFF20
0x00000010 047FC834 movia r17, KEY_BASE /* pushbutton KEY base address */
0x00000014 8C401404 orhi r17, zero, 0xFF20
0x00000018 047FC834 movia r19, HEX3_HEX0_BASE /* HEX3_HEX0 base address */
0x0000001C 9C000804 orhi r19, zero, 0xFF20
0x00000020 053FC834 movia r20, HEX5_HEX4_BASE /* HEX5_HEX4 base address */
0x00000024 A5000C04 orhi r20, zero, 0xFF20
0x00000028 04800034 addi r18, r18, 0x0
0x0000002C 94801C04 orhi r18, zero, 0x0
0x00000030 91800017 ldw r6, 0(r18) /* load pattern for HEX displays */
0x00000030 91800017 ldw r6, 0(r18)

```

Figure 11b

12. Exécuter le programme en sélectionnant « Actions --> Continue » (figure 12). Observer les motifs mobiles sur les afficheurs 7 segments.

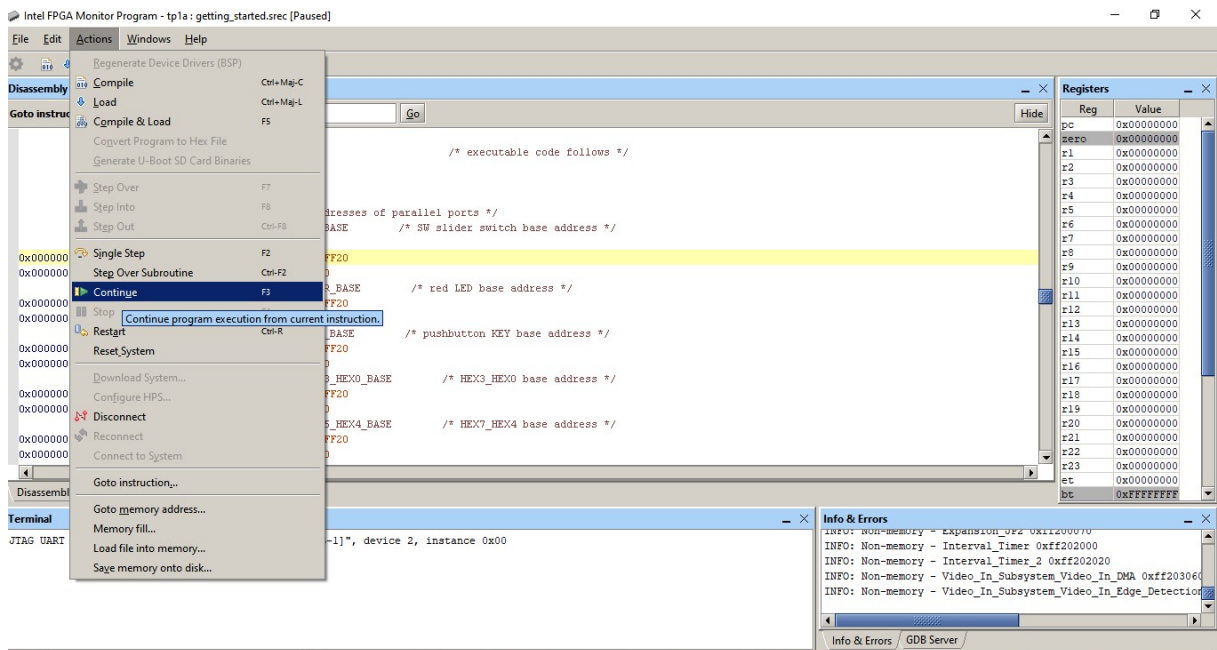


Figure 12

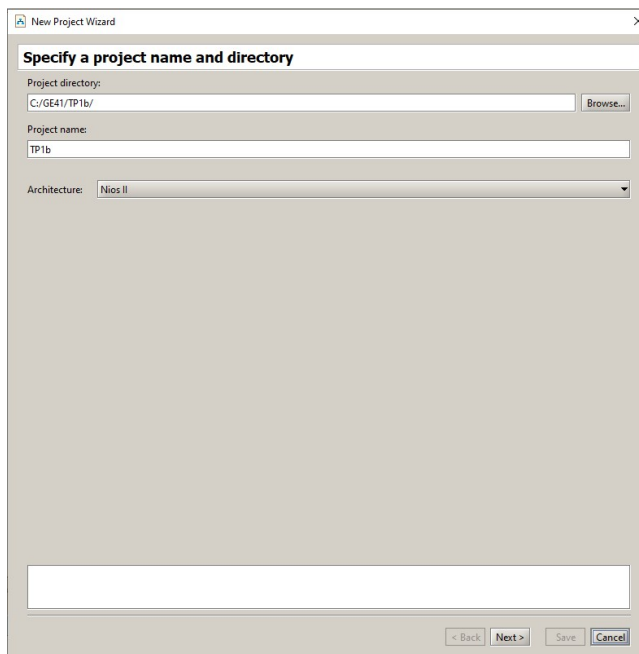
## Partie b

Nous allons maintenant explorer certaines fonctionnalités du programme Intel FPGA Monitor Program en utilisant un programme d'application simple écrit en langage c. Considérons le programme suivant, qui recherche le plus grand nombre dans une liste d'entiers de 32 bits stockée dans la mémoire

```
#include <stdio.h>
int LIST[7] = {4, 5, 3, 6, 1, 8, 2};
int biggest, i;
main()
{
    biggest = LIST[0];
    for ( i = 1; i <= 6; i++)
    {
        if ( LIST[i] > biggest )
            biggest = LIST[i];
    }
    printf( "Le nombre le plus grand est: %d \n", biggest );
}
```

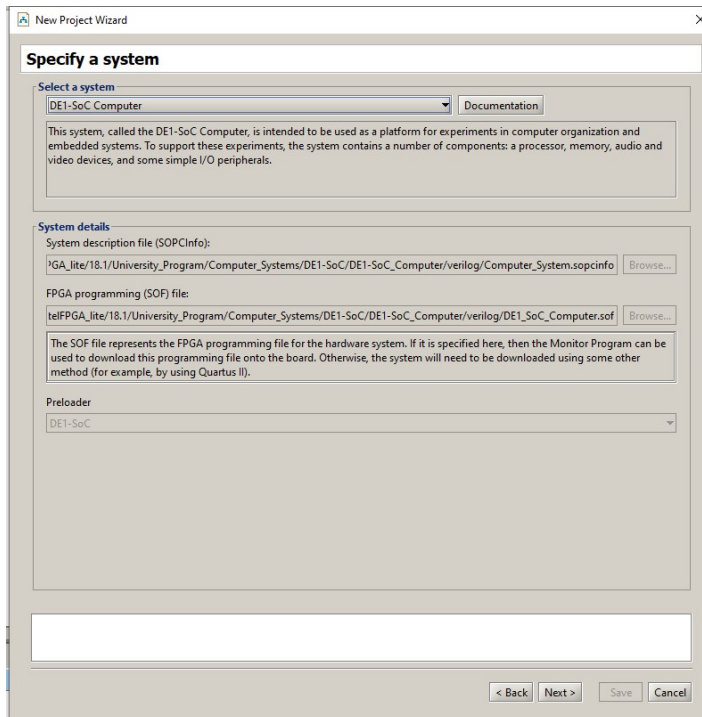
printf est utilisé pour afficher le résultat dans la fenêtre du terminal du programme Monitor.

### 1. Créer un nouveau projet

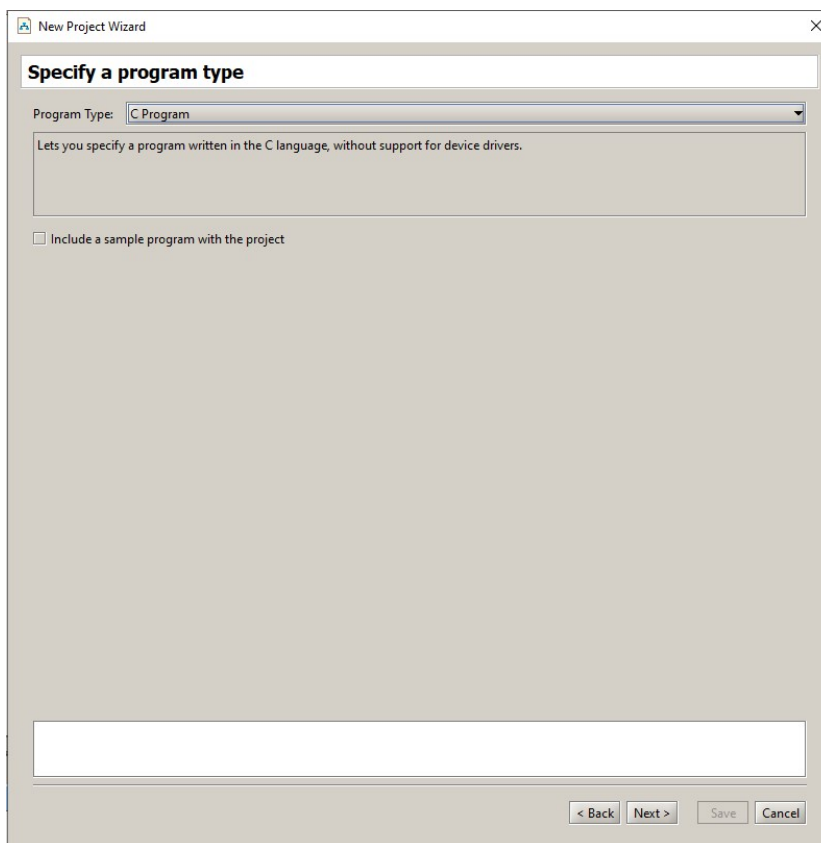




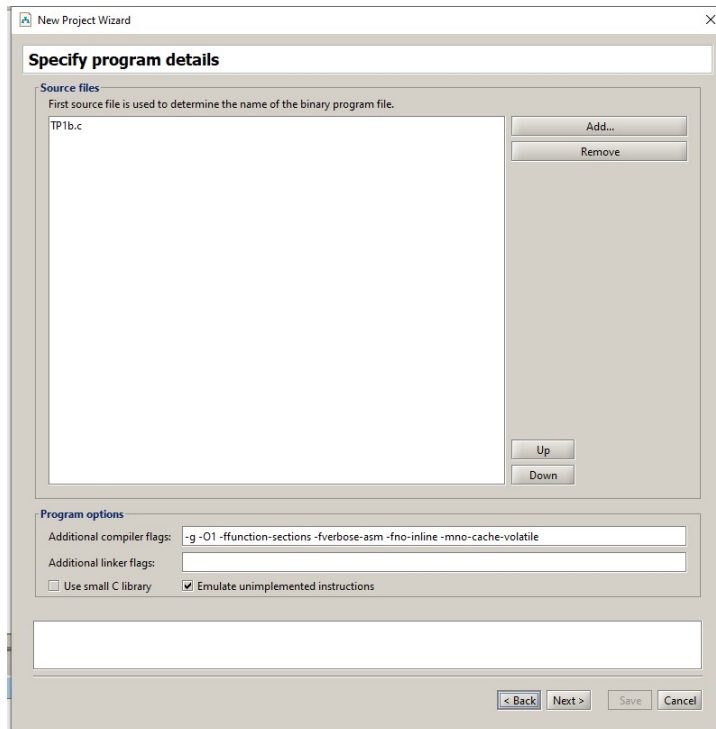
## 2. Sélectionner comme système « DE1 SOC Computer »



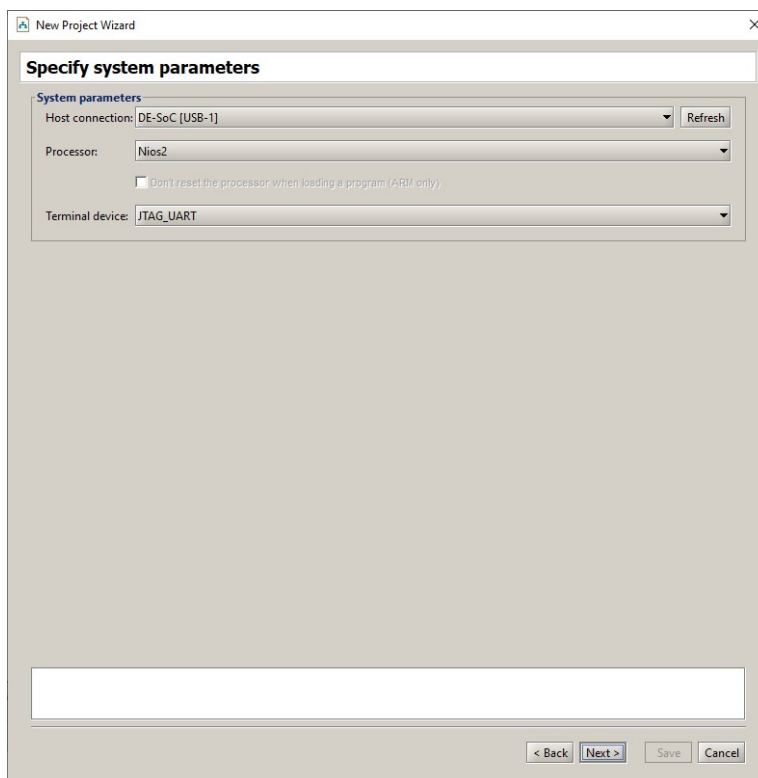
## 3. Choisir « C Program » dans la fenêtre « Specify program type »



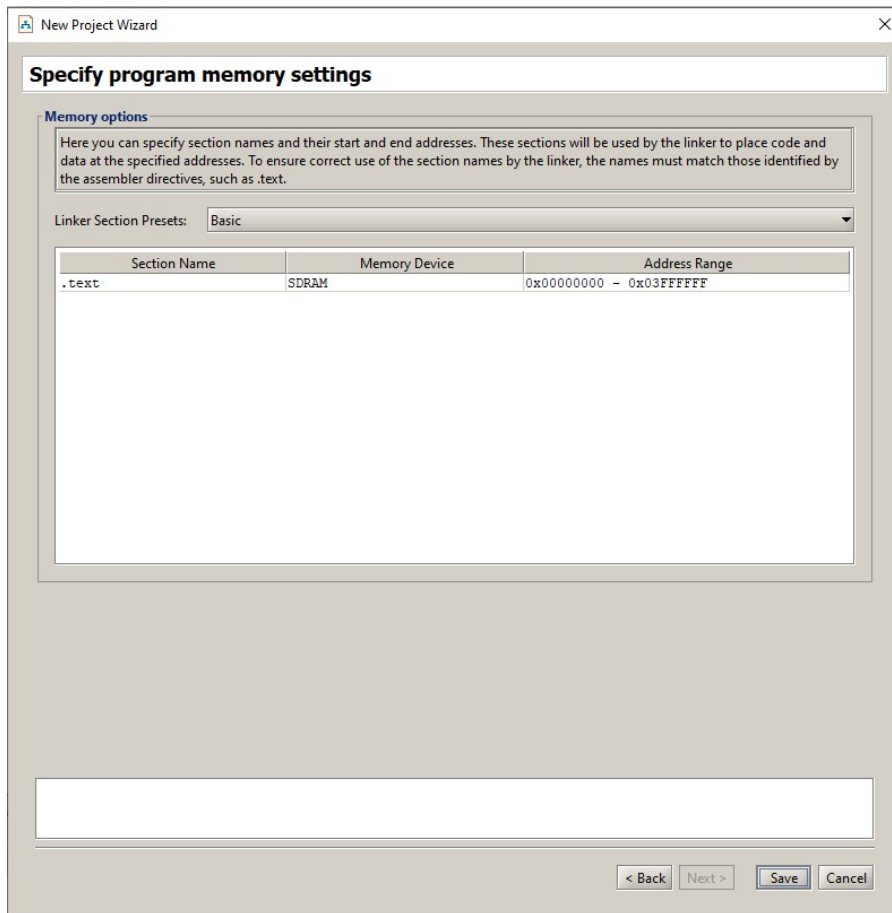
4. Cliquer sur « Add » pour ajouter TP1b.c



5. Ne rien changer sur la fenêtre suivante



6. Cliquer sur « Save » dans la prochaine fenêtre



7. Continuer les étapes suivantes comme dans la partie a
8. Exécuter le programme. Il faut avoir le résultat suivante sur la fenêtre du terminal:

```
Terminal
JTAG UART link established using cable "DE-SoC [USB-1]", device 2, instance 0x00
Le nombre le plus grand est: 8
```

9. L'utilisation de « printf » génère un nombre assez important d'instructions en langage assembleur. On va choisir plutôt d'afficher le résultat dans un emplacement mémoire spécifique, par exemple 0x7000, au lieu d'utiliser l'instruction « printf ». Faire les modifications du programme afin d'écrire le résultat à l'adresse 0x7000. Compiler et exécuter le programme. Vérifier que l'adresse 0x7000 contient le nombre le plus grand.

## Partie c

1. Les instructions logiques sont nécessaires dans de nombreuses applications embarquées. Elles sont utiles pour la manipulation de chaînes de bits et pour le traitement des données au niveau des bits, lorsque seuls quelques bits peuvent avoir un intérêt particulier. Elles sont essentielles pour gérer les tâches d'entrée/sortie. Dans cet exercice, il faut écrire un programme en langage C pour déterminer le nombre de bits ayant la valeur 1 dans un mot de 32 bits. La donnée à tester doit être à l'adresse 0x7000. Le résultat doit être stocké à l'adresse 0x7004.