

Système Programmable Nios II ALTERA

Travaux pratiques

Introduction par P.NOUEL

Ces travaux pratiques ont juste la prétention de fournir une approche au problème des SOC (System On Chip) dans ce qu'il a de particulier: mener conjointement les développement matériel et logiciel d'un projet.

Pour des raisons évidentes de souplesse d'utilisation, la plateforme matérielle est basée sur un FPGA transformant notre SOC en SOPC (System On Programmable Chip). Ceci permet d'obtenir des prototypes fonctionnels dans le minimum de temps dont on dispose.

Mais que doit-on trouver dans un enseignement des SOC ? Le système est composé d'éléments standard non originaux et bien connus. Un processeur, des mémoires, des périphériques, une interface Ethernet etc. Tous ces éléments ont déjà fait l'objet d'enseignements spécifiques préalablement : cours microprocesseur, cours réseau, cours VHDL. Mais dans tous ces enseignements, le matériel était parfaitement connu lorsqu'il s'agissait d'y associer du logiciel.

On le voit, ce qui va caractériser le SOC est en premier sa capacité d'optimiser une solution en choisissant ce qui doit revenir au matériel et ce qui restera au logiciel, l'art de passer de l'un à l'autre. Créer ses propres périphériques ou ses propres instructions en VHDL, voilà en particulier ce que permet les SOC. On peut partir d'un processeur intrinsèquement peu performant et obtenir un système aux performances remarquables.

On a choisi pour ces travaux pratiques de manière assez arbitraire un certain nombre d'applications mais toutes apportent un angle de vue différent et enrichissant.

D'autres applications ne pourront être vues pour des raisons de temps mais on pourra trouver quelques réponses sur les sites suivants:

<http://vhdl33.free.fr>

<http://www.niosforum.com/>

1) Les séances de travail

1.1) Découverte du système NIOS

Cette première séance, incontournable, offre la possibilité de découvrir les outils de développement du matériel et du logiciel. On construira un premier système simple, on en fera la synthèse complète, on y associera une application logicielle. L'ensemble sera chargé et testé sur la carte de développement. On aura de ce fait une première approche du **SOPC Builder** pour la génération du système, de **Quartus** pour la synthèse et le placement/routage, de l'**IDE** pour le développement logiciel, compilation et debug.

1.2) Création d'un composant

Afin de mettre en valeur la double compétence (matériel et logiciel) de l'ingénieur travaillant dans le monde des systèmes embarqués, il est essentiel de bien appréhender le niveau où se fait le lien entre le matériel et le logiciel. On va ainsi offrir l'exemple de périphériques décrits en langage VHDL et qui grâce à l'éditeur de composant présent dans le SOPC Builder vont pouvoir être spécifiés et rendus compatible au système de bus NIOS. Pour que le composant soit complet, il faudra ensuite constituer un fichier de description des registres qui sera la couche basse de toute application logicielle sur ce composant.

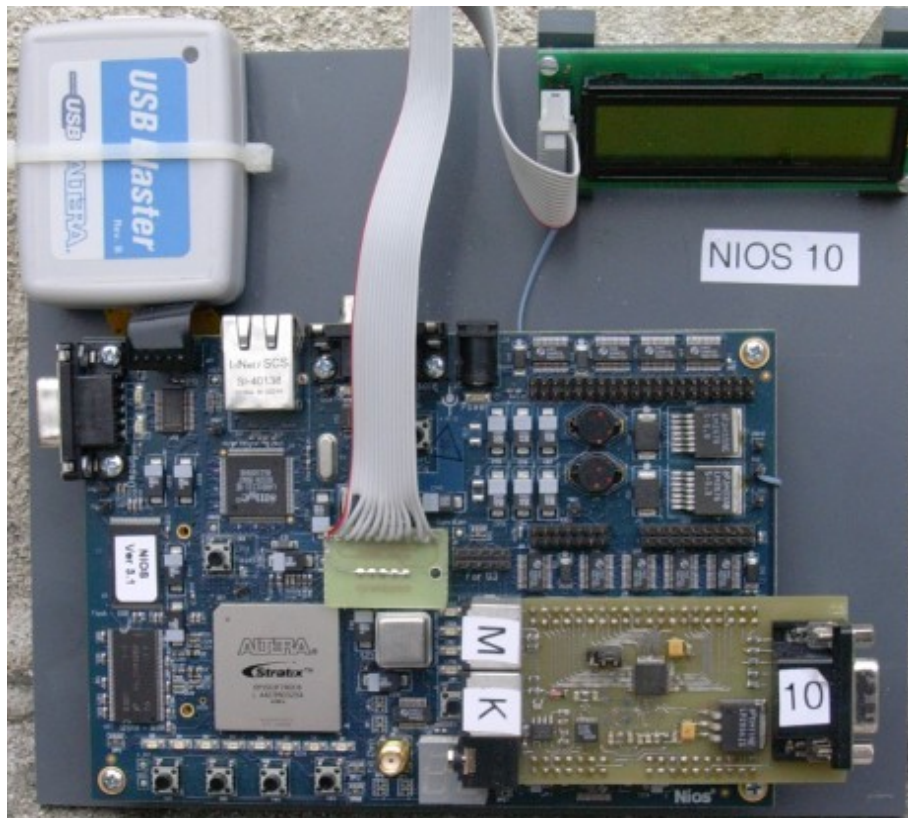
1.3) Outils d'inspection et de développement du matériel

Lorsque l'on associe un processeur et des composants périphériques, tous les échanges se concentrent au niveau du bus. Celui-ci a en général un fonctionnement et des possibilités complexes qu'il n'est pas simple d'appréhender. Comme on le fait pour un système de composants discrets, le premier outil essentiel pour inspecter le système matériel est l'analyseur logique. Chez Altera, l'analyseur logique intégré au FPGA est dénommé « **Signal TAP** » et il permet de visualiser en temps réel divers échanges sur un quelconque signal du projet. En particulier, c'est le moyen de visualiser les signaux du bus Avalon.

Le deuxième moyen de mise au point du matériel est , bien entendu la **simulation HDL**. Compte tenu de la complexité de la plupart des systèmes que l'on cherche à mettre au point, elle ne nous semble utile que dans des applications très spécifiques et à coup sûr suffisamment dépouillées pour donner des résultats exploitables. En effet, dans l'univers simulation, le programme qui s'exécute se retrouve soit en hexadécimal, soit au mieux en assembleur alors qu'il a été écrit en C. L'outil montre ici sa limite, sa seule force restant la visualisation possible de tous les signaux.

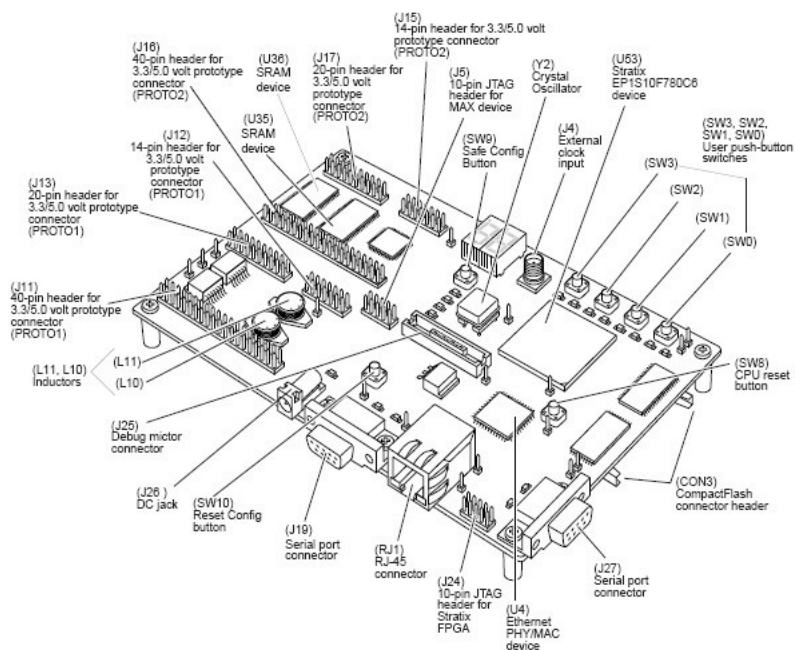
2) La plateforme matérielle

Le support pour tous les travaux pratiques NIOS est une carte appelée uPC (microPC) constituée de la carte de développement Altera **Stratix1S10** à laquelle on a rajouté une carte d'interface Clavier, souris PS2 et sortie audio, dénommée **Lancelot**.



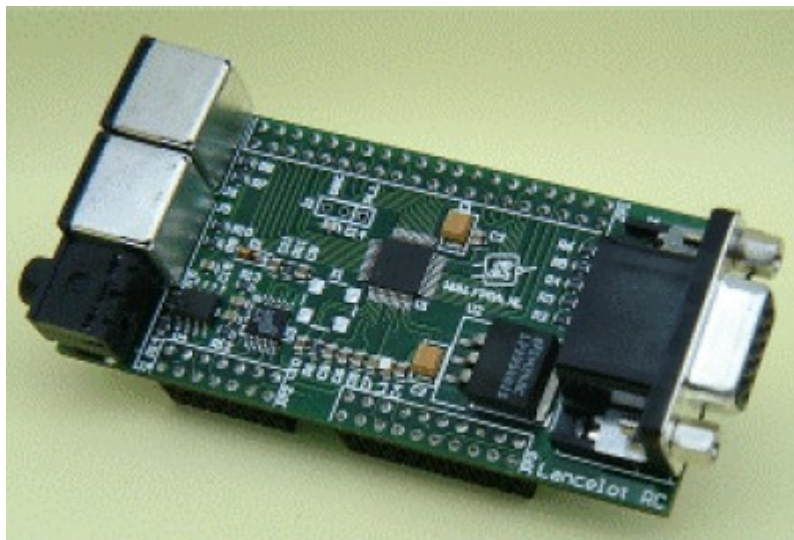
2.1) La carte de développement Stratix1S10

Elle comporte



- Un composant Stratix EP1S10F780C6. Cette famille correspond à une technologie cuivre 1,5 volt 0,13 μ m (fréquence limite 300 Mhz). On y recense 10570 éléments logiques, 920448 bits de RAM et 6 blocs DSP, 6PLLs et 780 broches. C indique une série commerciale et 6 une échelle de rapidité moyenne.
- Une mémoire Flash de 8 Moctets
- Une mémoire RAM statique de 1 Moctets
- Une mémoire SDRAM de 16 Moctets
- Un contrôleur Ethernet MAC/PHY LAN91C11 et connecteur RJ45
- 2 ports série RS-232
- 4 boutons poussoirs libres d'utilisation
- 8 leds
- 2 afficheurs 7-segments
- Un oscillateur 50 Mhz et un circuit de distribution d'horloge
- Un connecteur JTAG utilisé pour le téléchargement, la programmation et le développement
- Une extension CompactFlash
- Un circuit de type CPLD MAX7128A qui contrôle les différentes possibilités d'initialisation de la carte: par le JTAG, par la FLASH, sur un design de référence ...
- Des connecteurs d'extension utilisé en particulier pour enficher la carte « Lancelot »

2.2) La carte d'extension Lancelot



Cette carte, réalisée à l'Enseirb à partir de plans fournis par le site <http://www.fpga.nl> apporte une interface VGA constituée par un connecteur et un circuit triple convertisseur numérique analogique 8 bits (pour les « couleurs R, G, B), deux interfaces PS2 (qui seront dédiées à une souris et un clavier) et enfin deux circuits filtres analogiques associés à un connecteur audio stéréo. La carte est prévue pour pouvoir s'enficher sur les connecteurs d'extension de la carte de développement comme on peut le voir sur la photo de la carte uPC (ci-dessus).

2.3) Les entrées-sorties de la carte uPC

A carte fixe, entrées-sorties fixes. Afin de gérer au mieux les différentes versions de chaque projet, on a tout intérêt une bonne fois pour toutes à figer les noms associés aux entrées-sorties de la carte puisque leur emplacement par rapport au FPGA est fixé par le câblage de celle-ci. On a choisi de la faire par l'intermédiaire du fichier de commande TCL *uPC1S10.tcl* ci-après. Ce fichier est documenté et doit être consulté et pourquoi pas enrichi.

```
# Load Quartus II Tcl Project package
package require ::quartus::project

set need_to_close_project 0
set make_assignments 1

# Check that the right project is open
if {[is_project_open]} {
    if {[string compare $quartus(project) "uPC1S10"]} {
        puts "Project is not open"
        set make_assignments 0
    }
} else {
    # Only open if not already open
    if {[project_exists uPC1S10]} {
        project_open -revision uPC1S10 uPC1S10
    } else {
        project_new -revision uPC1S10 uPC1S10
    }
    set need_to_close_project 1
}

# Make assignments
if {$make_assignments} {
    set_global_assignment -name FMAX_REQUIREMENT "60.0 MHz"
    set_global_assignment -name FAMILY Stratix
    set_global_assignment -name DEVICE EP1S10F780C6
    set_global_assignment -name RESERVE_ALL_UNUSED_PINS "AS INPUT TRI-STATED"
    set_global_assignment -name RESERVE_DATA0_AFTER_CONFIGURATION "USE AS REGULAR IO"
    set_global_assignment -name ON_CHIP_BITSTREAM_DECOMPRESSION OFF
    set_global_assignment -name RESERVE_ALL_UNUSED_PINS_NO_OUTPUT_GND "AS INPUT TRI-STATED"
    set_global_assignment -name GLITCH_INTERVAL "1 "
```

```
# Clock and PLL
```

```
set_location_assignment PIN_K17 -to PLD_CLOCKINPUT  
set_location_assignment PIN_AC17 -to PLD_CLOCKINPUT2  
set_location_assignment PIN_AC9 -to PLD_CLEAR_N  
set_location_assignment PIN_AB19 -to video_clk  
set_location_assignment PIN_E15 -to SDRAM_CLKOUT
```

```
# Ethernet controler
```

```
set_location_assignment PIN_A4 -to FSE_A[0]  
set_location_assignment PIN_A3 -to FSE_A[1]  
set_location_assignment PIN_B3 -to FSE_A[2]  
set_location_assignment PIN_B5 -to FSE_A[3]  
set_location_assignment PIN_B4 -to FSE_A[4]  
set_location_assignment PIN_C4 -to FSE_A[5]  
set_location_assignment PIN_A5 -to FSE_A[6]  
set_location_assignment PIN_C5 -to FSE_A[7]  
set_location_assignment PIN_D5 -to FSE_A[8]  
set_location_assignment PIN_E6 -to FSE_A[9]  
set_location_assignment PIN_A6 -to FSE_A[10]  
set_location_assignment PIN_B7 -to FSE_A[11]  
set_location_assignment PIN_D6 -to FSE_A[12]  
set_location_assignment PIN_A7 -to FSE_A[13]  
set_location_assignment PIN_D7 -to FSE_A[14]  
set_location_assignment PIN_C6 -to FSE_A[15]  
set_location_assignment PIN_C7 -to FSE_A[16]  
set_location_assignment PIN_B6 -to FSE_A[17]  
set_location_assignment PIN_D8 -to FSE_A[18]  
set_location_assignment PIN_C8 -to FSE_A[19]  
set_location_assignment PIN_E8 -to FSE_A[20]  
set_location_assignment PIN_D9 -to FSE_A[21]  
set_location_assignment PIN_B9 -to FSE_A[22]  
set_location_assignment PIN_H12 -to FSE_D[0]  
set_location_assignment PIN_F12 -to FSE_D[1]  
set_location_assignment PIN_J12 -to FSE_D[2]  
set_location_assignment PIN_M12 -to FSE_D[3]  
set_location_assignment PIN_H17 -to FSE_D[4]  
set_location_assignment PIN_K18 -to FSE_D[5]  
set_location_assignment PIN_H18 -to FSE_D[6]  
set_location_assignment PIN_G18 -to FSE_D[7]  
set_location_assignment PIN_B8 -to FSE_D[8]  
set_location_assignment PIN_A8 -to FSE_D[9]  
set_location_assignment PIN_A9 -to FSE_D[10]
```

```
set_location_assignment PIN_C9 -to FSE_D[11]
set_location_assignment PIN_E10 -to FSE_D[12]
set_location_assignment PIN_A10 -to FSE_D[13]
set_location_assignment PIN_C10 -to FSE_D[14]
set_location_assignment PIN_B10 -to FSE_D[15]
set_location_assignment PIN_A11 -to FSE_D[16]
set_location_assignment PIN_C11 -to FSE_D[17]
set_location_assignment PIN_D11 -to FSE_D[18]
set_location_assignment PIN_B11 -to FSE_D[19]
set_location_assignment PIN_D10 -to FSE_D[20]
set_location_assignment PIN_G10 -to FSE_D[21]
set_location_assignment PIN_F10 -to FSE_D[22]
set_location_assignment PIN_H11 -to FSE_D[23]
set_location_assignment PIN_G11 -to FSE_D[24]
set_location_assignment PIN_F8 -to FSE_D[25]
set_location_assignment PIN_J9 -to FSE_D[26]
set_location_assignment PIN_J13 -to FSE_D[27]
set_location_assignment PIN_L13 -to FSE_D[28]
set_location_assignment PIN_M11 -to FSE_D[29]
set_location_assignment PIN_L11 -to FSE_D[30]
set_location_assignment PIN_G7 -to FSE_D[31]
set_location_assignment PIN_M18 -to SRAM_BE_N[0]
set_location_assignment PIN_F17 -to SRAM_BE_N[1]
set_location_assignment PIN_J18 -to SRAM_BE_N[2]
set_location_assignment PIN_L17 -to SRAM_BE_N[3]
set_location_assignment PIN_F19 -to FLASH_OE_N
set_location_assignment PIN_G19 -to FLASH_RW_N
set_location_assignment PIN_B26 -to SRAM_OE_N
set_location_assignment PIN_K19 -to FLASH_CS_N
set_location_assignment PIN_B24 -to SRAM_CS_N
set_location_assignment PIN_C24 -to SRAM_WE_N
set_location_assignment PIN_T22 -to enet_be_n[0]
set_location_assignment PIN_U26 -to enet_be_n[1]
set_location_assignment PIN_U25 -to enet_be_n[2]
set_location_assignment PIN_T19 -to enet_be_n[3]
set_location_assignment PIN_V25 -to enet_ads_n
set_location_assignment PIN_V27 -to ENET_INTRQ[0]
set_location_assignment PIN_V28 -to enet_aen
set_location_assignment PIN_T23 -to enet_ior_n
set_location_assignment PIN_T24 -to enet_iow_n
```

Push button

```

set_location_assignment PIN_W5 -to USER_PB[0]
set_location_assignment PIN_W6 -to USER_PB[1]
set_location_assignment PIN_AB2 -to USER_PB[2]
set_location_assignment PIN_AB1 -to USER_PB[3]
# LCD Display: cablage d'origine avec conflit sur compact Flash
# : compatible avec la safe configuration Altera
# set_location_assignment PIN_L4 -to LCD[0]
# set_location_assignment PIN_M6 -to LCD[1]
# set_location_assignment PIN_J1 -to LCD[2]
# set_location_assignment PIN_K2 -to LCD[3]
# set_location_assignment PIN_K4 -to LCD[4]
# set_location_assignment PIN_G1 -to LCD[5]
# set_location_assignment PIN_V18 -to LCD[6]
# set_location_assignment PIN_J4 -to LCD[7]
# set_location_assignment PIN_AB18 -to LCD_E
# set_location_assignment PIN_AB17 -to LCD_RS
# set_location_assignment PIN_M8 -to LCD_RW
# New pin assignement ( mictor connector)
set_location_assignment PIN_N19 -to LCD[0]
set_location_assignment PIN_N20 -to LCD[1]
set_location_assignment PIN_N28 -to LCD[2]
set_location_assignment PIN_M27 -to LCD[3]
set_location_assignment PIN_G27 -to LCD[4]
set_location_assignment PIN_G28 -to LCD[5]
set_location_assignment PIN_K21 -to LCD[6]
set_location_assignment PIN_K22 -to LCD[7]
set_location_assignment PIN_P26 -to LCD_E
set_location_assignment PIN_L28 -to LCD_RS
set_location_assignment PIN_L27 -to LCD_RW
# 7 segments display
set_location_assignment PIN_C21 -to Display_7_Segment[0]
set_location_assignment PIN_B21 -to Display_7_Segment[1]
set_location_assignment PIN_A21 -to Display_7_Segment[2]
set_location_assignment PIN_C20 -to Display_7_Segment[3]
set_location_assignment PIN_A20 -to Display_7_Segment[4]
set_location_assignment PIN_B20 -to Display_7_Segment[5]
set_location_assignment PIN_B18 -to Display_7_Segment[6]
set_location_assignment PIN_D21 -to Display_7_Segment[7]
set_location_assignment PIN_E19 -to Display_7_Segment[8]
set_location_assignment PIN_C19 -to Display_7_Segment[9]
set_location_assignment PIN_B19 -to Display_7_Segment[10]

```



```
set_location_assignment PIN_A19 -to Display_7_Segment[11]
set_location_assignment PIN_D18 -to Display_7_Segment[12]
set_location_assignment PIN_C18 -to Display_7_Segment[13]
set_location_assignment PIN_A18 -to Display_7_Segment[14]
set_location_assignment PIN_D19 -to Display_7_Segment[15]
# LEDs
set_location_assignment PIN_H27 -to LEDG[0]
set_location_assignment PIN_H28 -to LEDG[1]
set_location_assignment PIN_L23 -to LEDG[2]
set_location_assignment PIN_L24 -to LEDG[3]
set_location_assignment PIN_J25 -to LEDG[4]
set_location_assignment PIN_J26 -to LEDG[5]
set_location_assignment PIN_L20 -to LEDG[6]
set_location_assignment PIN_L19 -to LEDG[7]
# PLD reconfig
set_location_assignment PIN_U2 -to PLD_RECONFIGREQ_N
# RS232 connections
set_location_assignment PIN_Y28 -to RXD1
set_location_assignment PIN_U21 -to TXD1
set_location_assignment PIN_AA28 -to RXD2
set_location_assignment PIN_V24 -to TXD2
# Lancelot group
set_location_assignment PIN_W19 -to ps2_sel
#VGA
set_location_assignment PIN_AG23 -to b[0]
set_location_assignment PIN_AE23 -to b[1]
set_location_assignment PIN_AH24 -to b[2]
set_location_assignment PIN_AG24 -to b[3]
set_location_assignment PIN_AE24 -to b[4]
set_location_assignment PIN_AF25 -to b[5]
set_location_assignment PIN_AG25 -to b[6]
set_location_assignment PIN_AH25 -to b[7]
set_location_assignment PIN_AD19 -to g[0]
set_location_assignment PIN_AE19 -to g[1]
set_location_assignment PIN_AF18 -to g[2]
set_location_assignment PIN_AH20 -to g[3]
set_location_assignment PIN_AH21 -to g[4]
set_location_assignment PIN_AF20 -to g[5]
set_location_assignment PIN_AE20 -to g[6]
set_location_assignment PIN_AF21 -to g[7]
```

```

set_location_assignment PIN_AD23 -to r[0]
set_location_assignment PIN_AF23 -to r[1]
set_location_assignment PIN_AH23 -to r[2]
set_location_assignment PIN_AE22 -to r[3]
set_location_assignment PIN_AF22 -to r[4]
set_location_assignment PIN_AH22 -to r[5]
set_location_assignment PIN_AG22 -to r[6]
set_location_assignment PIN_AG20 -to r[7]
set_location_assignment PIN_AA19 -to m1
set_location_assignment PIN_Y17 -to m2
set_location_assignment PIN_AG21 -to sync_t
set_location_assignment PIN_AD21 -to blank_n
set_location_assignment PIN_AH26 -to vga_hs
set_location_assignment PIN_AE21 -to sync_n
set_location_assignment PIN_AG26 -to vga_vs

# mouse
set_location_assignment PIN_AC19 -to ps2_ms_clk
set_location_assignment PIN_AF24 -to ps2_ms_dat

# keyboard
set_location_assignment PIN_W18 -to ps2_kb_clk
set_location_assignment PIN_Y19 -to ps2_kb_dat

# audio
set_location_assignment PIN_AC21 -to audio_l
set_location_assignment PIN_AA20 -to audio_r

# SDRAM
set_location_assignment PIN_AE4 -to sdram_a[0]
set_location_assignment PIN_W12 -to sdram_a[1]
set_location_assignment PIN_AC11 -to sdram_a[2]
set_location_assignment PIN_W10 -to sdram_a[3]
set_location_assignment PIN_AA11 -to sdram_a[4]
set_location_assignment PIN_AC10 -to sdram_a[5]
set_location_assignment PIN_AB11 -to sdram_a[6]
set_location_assignment PIN_AC8 -to sdram_a[7]
set_location_assignment PIN_AB10 -to sdram_a[8]
set_location_assignment PIN_V11 -to sdram_a[9]
set_location_assignment PIN_Y11 -to sdram_a[10]
set_location_assignment PIN_AB7 -to sdram_a[11]
set_location_assignment PIN_AG19 -to sdram_ba[0]
set_location_assignment PIN_AF19 -to sdram_ba[1]
set_location_assignment PIN_AH4 -to sdram_dq[0]
set_location_assignment PIN_AE5 -to sdram_dq[1]

```

```
set_location_assignment PIN_AG3 -to sdram_dq[2]
set_location_assignment PIN_AG5 -to sdram_dq[3]
set_location_assignment PIN_AG4 -to sdram_dq[4]
set_location_assignment PIN_AF4 -to sdram_dq[5]
set_location_assignment PIN_AH5 -to sdram_dq[6]
set_location_assignment PIN_AF5 -to sdram_dq[7]
set_location_assignment PIN_AE6 -to sdram_dq[8]
set_location_assignment PIN_AG6 -to sdram_dq[9]
set_location_assignment PIN_AH6 -to sdram_dq[10]
set_location_assignment PIN_AD6 -to sdram_dq[11]
set_location_assignment PIN_AF7 -to sdram_dq[12]
set_location_assignment PIN_AH7 -to sdram_dq[13]
set_location_assignment PIN_AG7 -to sdram_dq[14]
set_location_assignment PIN_AF6 -to sdram_dq[15]
set_location_assignment PIN_AG8 -to sdram_dq[16]
set_location_assignment PIN_AF8 -to sdram_dq[17]
set_location_assignment PIN_AD8 -to sdram_dq[18]
set_location_assignment PIN_AH9 -to sdram_dq[19]
set_location_assignment PIN_AH8 -to sdram_dq[20]
set_location_assignment PIN_AE9 -to sdram_dq[21]
set_location_assignment PIN_AF9 -to sdram_dq[22]
set_location_assignment PIN_AG9 -to sdram_dq[23]
set_location_assignment PIN_AD10 -to sdram_dq[24]
set_location_assignment PIN_AF10 -to sdram_dq[25]
set_location_assignment PIN_AH10 -to sdram_dq[26]
set_location_assignment PIN_AE10 -to sdram_dq[27]
set_location_assignment PIN_AF11 -to sdram_dq[28]
set_location_assignment PIN_AE11 -to sdram_dq[29]
set_location_assignment PIN_AH11 -to sdram_dq[30]
set_location_assignment PIN_AG11 -to sdram_dq[31]
set_location_assignment PIN_AE14 -to sdram_dqm[0]
set_location_assignment PIN_Y13 -to sdram_dqm[1]
set_location_assignment PIN_AE7 -to sdram_dqm[2]
set_location_assignment PIN_AG10 -to sdram_dqm[3]
set_location_assignment PIN_AD18 -to sdram_cas_n
set_location_assignment PIN_AE18 -to sdram_cke
set_location_assignment PIN_AH3 -to sdram_ras_n
set_location_assignment PIN_AH19 -to sdram_we_n
set_location_assignment PIN_AG18 -to sdram_cs_n
```

Compact Flash IDE

```
set_location_assignment PIN_G2 -to cf_rfu
```

```
set_location_assignment PIN_L9 -to cf_addr[0]
set_location_assignment PIN_J3 -to cf_addr[1]
set_location_assignment PIN_L10 -to cf_addr[2]
set_location_assignment PIN_L6 -to cf_addr[3]
set_location_assignment PIN_H1 -to cf_addr[4]
set_location_assignment PIN_H2 -to cf_addr[5]
set_location_assignment PIN_L8 -to cf_addr[6]
set_location_assignment PIN_L7 -to cf_addr[7]
set_location_assignment PIN_H3 -to cf_addr[8]
set_location_assignment PIN_K3 -to cf_addr[9]
set_location_assignment PIN_M7 -to cf_addr[10]
set_location_assignment PIN_J2 -to cf_cs_n[0]
set_location_assignment PIN_K8 -to cf_cs_n[1]
set_location_assignment PIN_L5 -to cf_we_n
set_location_assignment PIN_M9 -to cf_iord_n
set_location_assignment PIN_M10 -to cf_iowr_n
set_location_assignment PIN_R3 -to cf_detect_n
set_location_assignment PIN_M5 -to cf_intrq
set_location_assignment PIN_K1 -to cf_iordy
set_location_assignment PIN_K7 -to cf_atasel_n
set_location_assignment PIN_H4 -to cf_power
set_location_assignment PIN_L2 -to cf_data[1]
set_location_assignment PIN_N3 -to cf_data[0]
set_location_assignment PIN_N8 -to cf_data[2]
set_location_assignment PIN_M4 -to cf_data[3]
set_location_assignment PIN_N6 -to cf_data[4]
set_location_assignment PIN_N1 -to cf_data[5]
set_location_assignment PIN_N9 -to cf_data[6]
set_location_assignment PIN_P3 -to cf_data[7]
set_location_assignment PIN_N10 -to cf_data[8]
set_location_assignment PIN_M2 -to cf_data[9]
set_location_assignment PIN_N5 -to cf_data[10]
set_location_assignment PIN_M3 -to cf_data[11]
set_location_assignment PIN_N7 -to cf_data[12]
set_location_assignment PIN_L1 -to cf_data[13]
set_location_assignment PIN_N4 -to cf_data[14]
set_location_assignment PIN_L3 -to cf_data[15]
```

#

```
# Commit assignments
export_assignments
```

```
# Close project
if {$need_to_close_project} {
    project_close
}
}
```

3) Le logiciel de développement

Les explications, images, illustrations se trouvant dans les textes e travaux pratiques proposés ont été validées avec les versions **8.1** du logiciel **Quartus**, du modèle **NIOS** et du logiciel **ide**. La raison pour laquelle il est important de spécifier la version, c'est la mauvaise compatibilité ascendante des outils Altera. En particulier, les composants n'ont plus du tout la même structure en version 8 qu'en version 6, les outils nouveaux sont souvent entachés d'erreur ou imparfaits sur la synthèse VHDL. Ces problèmes seront commentés en temps utile sur le site <http://vhdl33.free.fr>