

TP – Comparaison AES HW et AES SW



Antoine Aubert

Yann Birembaux

28/05/2023

| | |
|---|----|
| Implémentation..... | 3 |
| Programmation de la carte..... | 14 |
| Lancement et configuration | 14 |
| Test de performance et comparaison..... | 19 |

Implémentation

Installation de Quartus

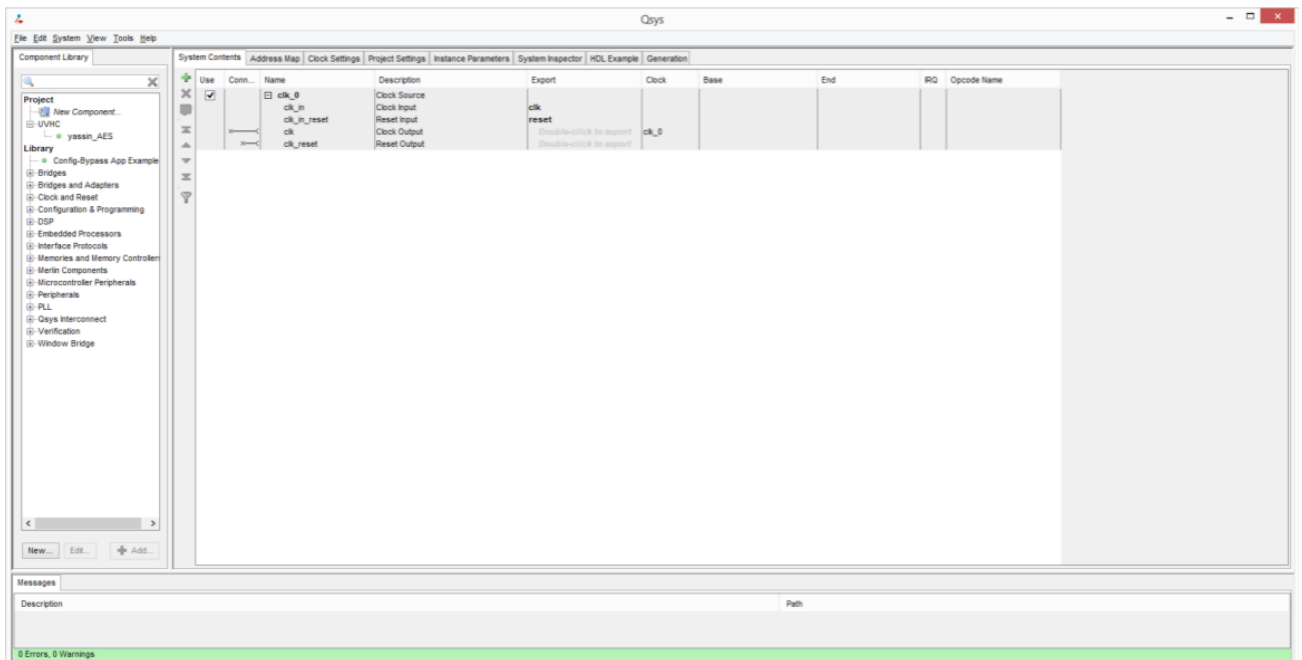
Décompression du fichier « designs_start_0.rar » dans le répertoire
C:/altera/13.0sp1/designs/

Copie du contenu du répertoire « copier_dans_HW_SW_AES » dans le répertoire

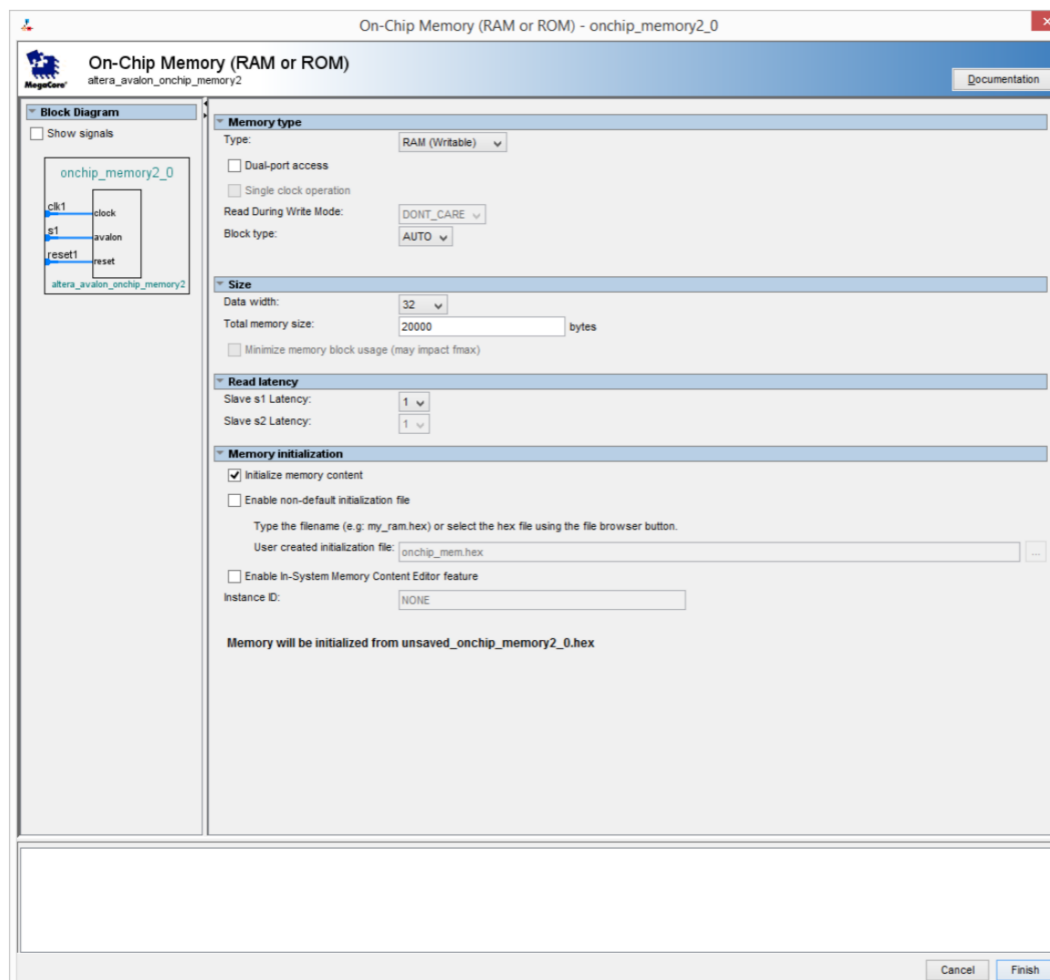
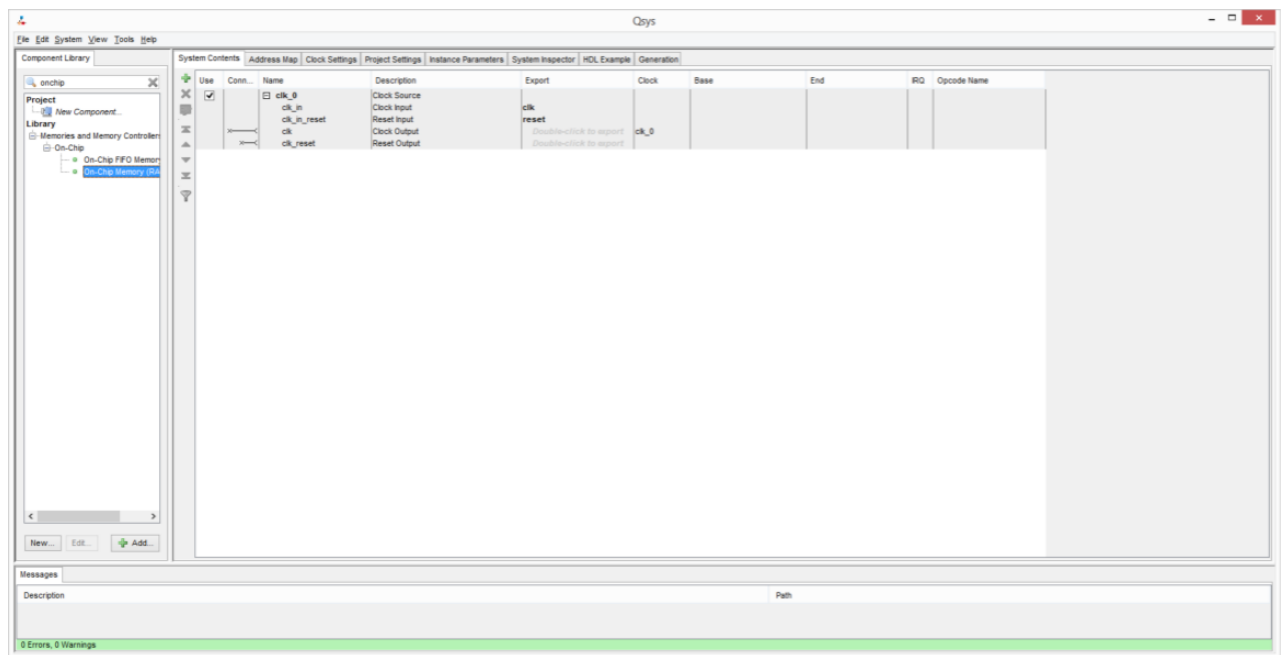
C:/altera/13.0sp1/designs/ HW_SW_AES/

Lancement de Quartus et ouverture du projet HW_SW_AES

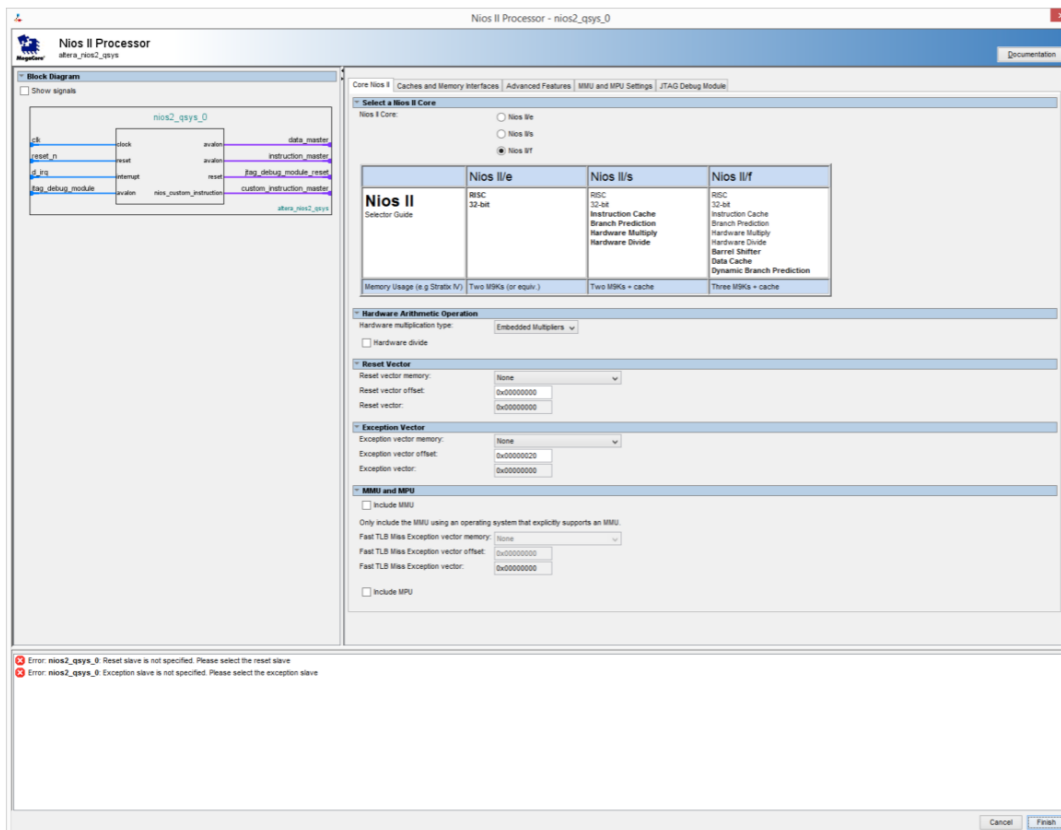
Lancement de Qsys



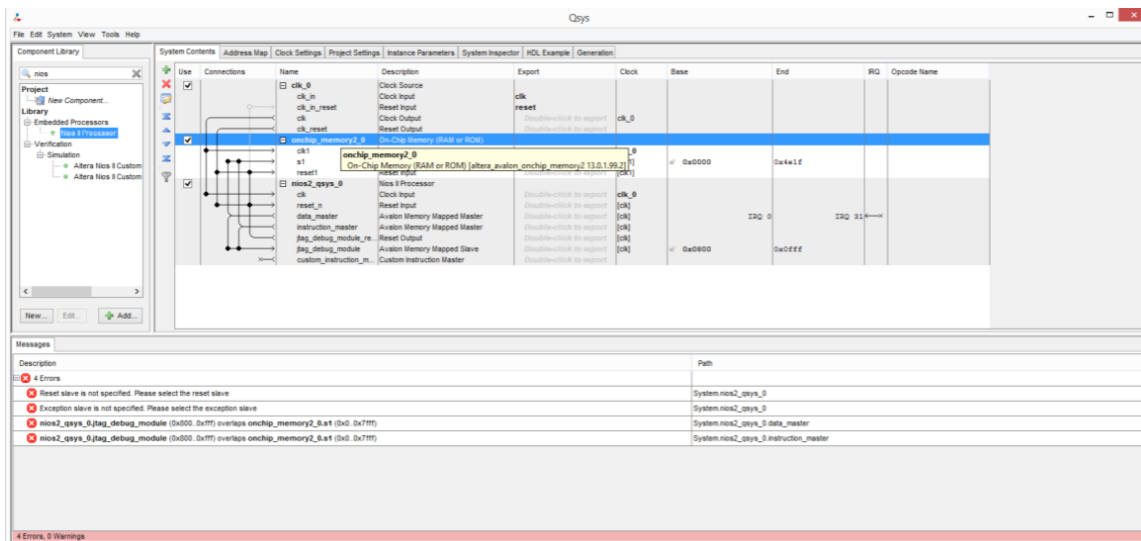
Ajout de onchip_memory2_0



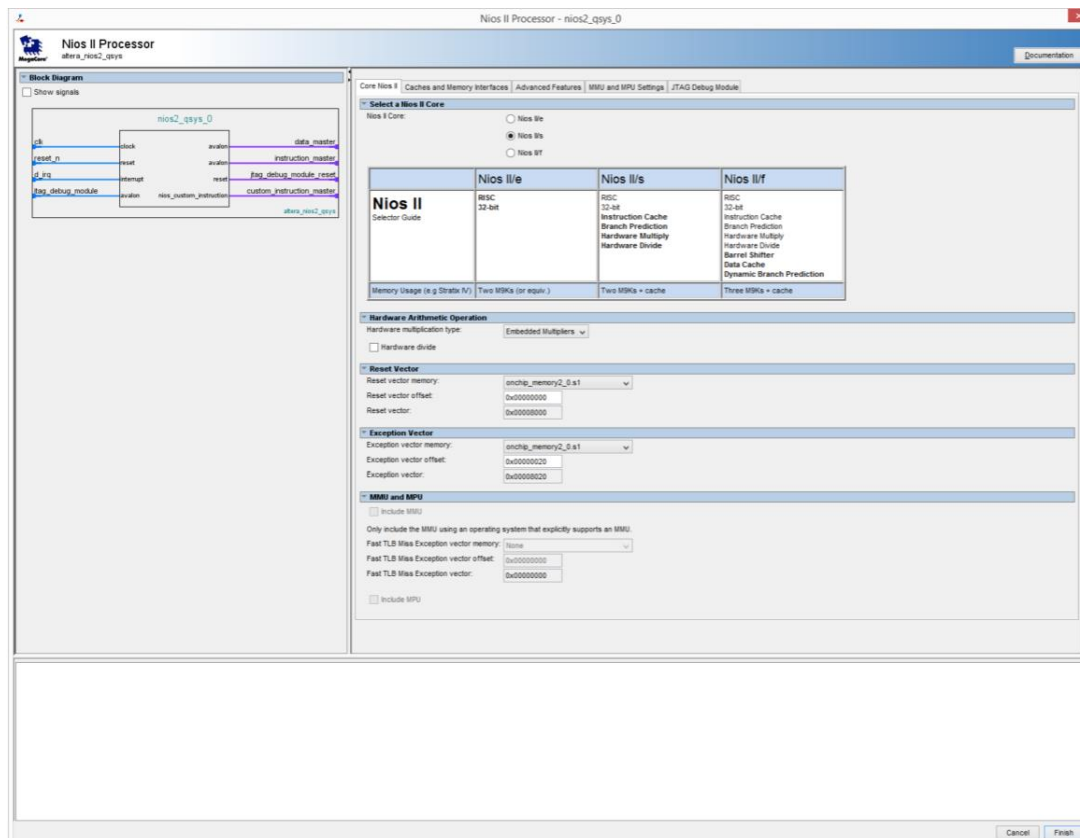
Ajout du CPU



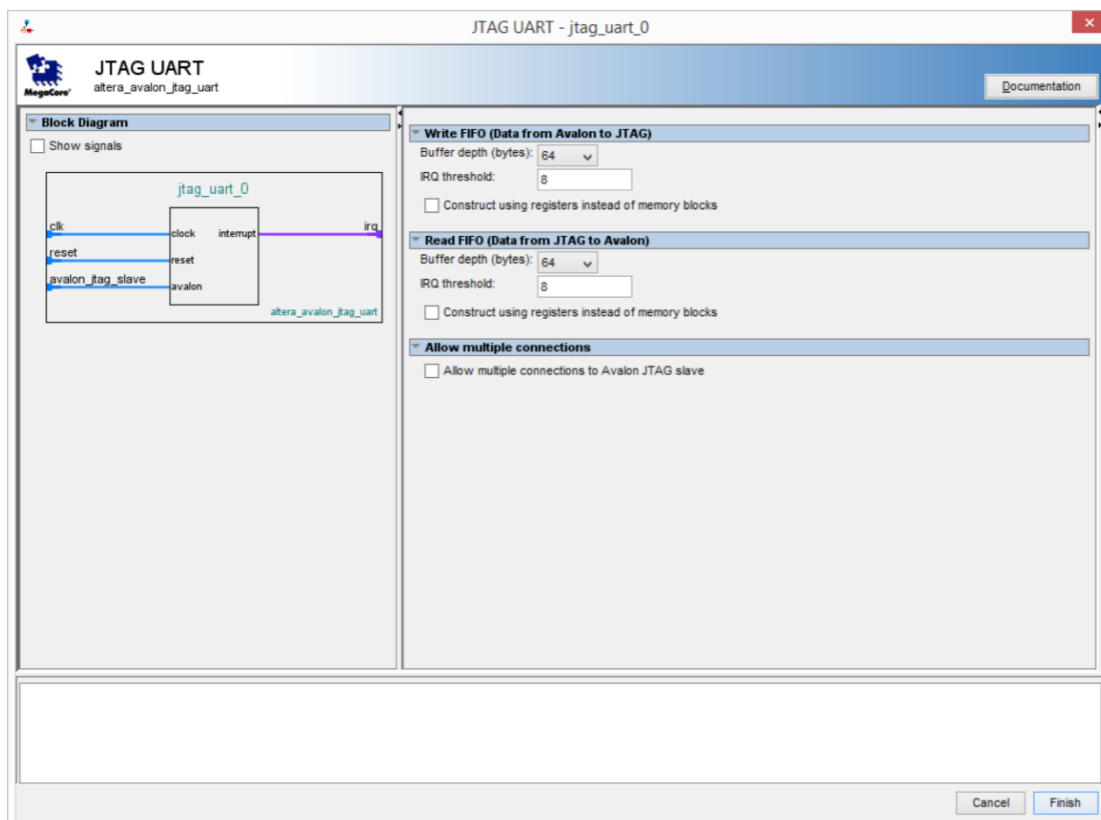
Connection du CPU



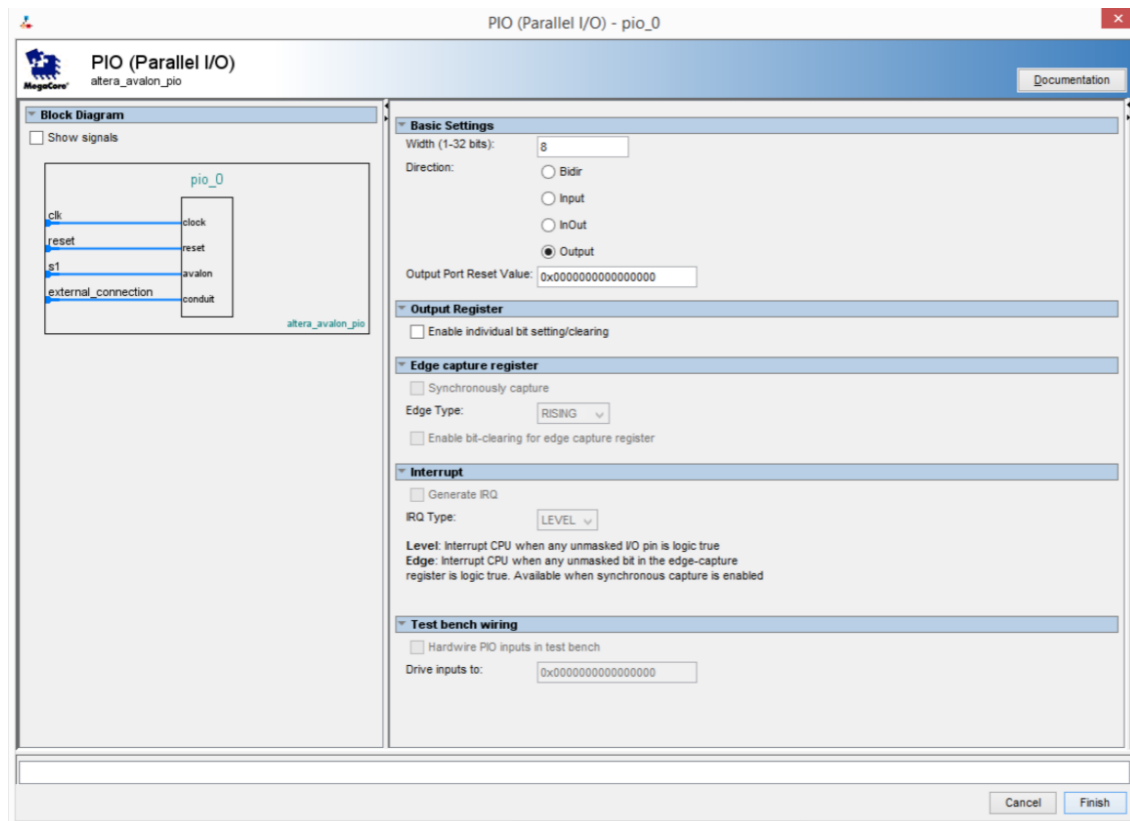
Configuration du CPU



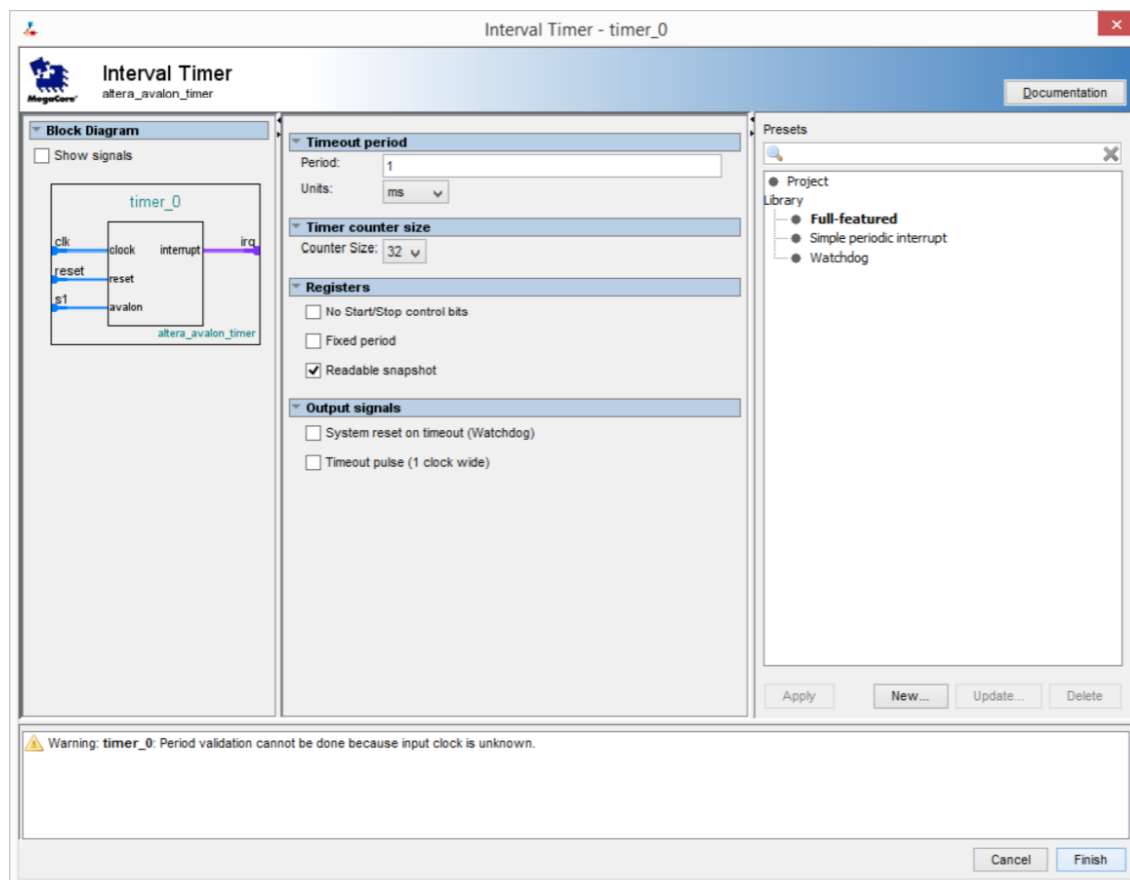
Ajout du module JTAG UART



Ajout du module PIO (Parallèle I/O)



Ajout du module interval Timer



Ajout du module system ID Peripheral

The screenshot shows the 'System ID Peripheral - sysid_qsys_0' configuration window. The window has a title bar with the text 'System ID Peripheral - sysid_qsys_0' and a close button. The main area is divided into three sections: 'Block Diagram', 'Parameters', and 'Description'. The 'Block Diagram' section shows a block diagram of the 'sysid_qsys_0' component, which is an instance of the 'altera_avalon_sysid_qsys' component. The diagram shows three input signals: 'clk', 'reset', and 'control_slave', which are connected to the 'clock', 'reset', and 'avalon' ports of the component, respectively. The 'Parameters' section shows a single parameter, '32 bit System ID', with a value of '0x00000000'. The 'Description' section contains the text 'Please use hexadecimal numbers only in System ID.' At the bottom of the window, there is a status bar with two informational messages: 'Info: sysid_qsys_0: System ID is not assigned automatically. Edit the System ID parameter to provide a unique ID' and 'Info: sysid_qsys_0: Time stamp will be automatically updated when this component is generated.'

System ID Peripheral - sysid_qsys_0

System ID Peripheral
altera_avalon_sysid_qsys

Block Diagram

☐ Show signals

sysid_qsys_0

clk
reset
control_slave

clock
reset
avalon

altera_avalon_sysid_qsys

Parameters

32 bit System ID: 0x00000000

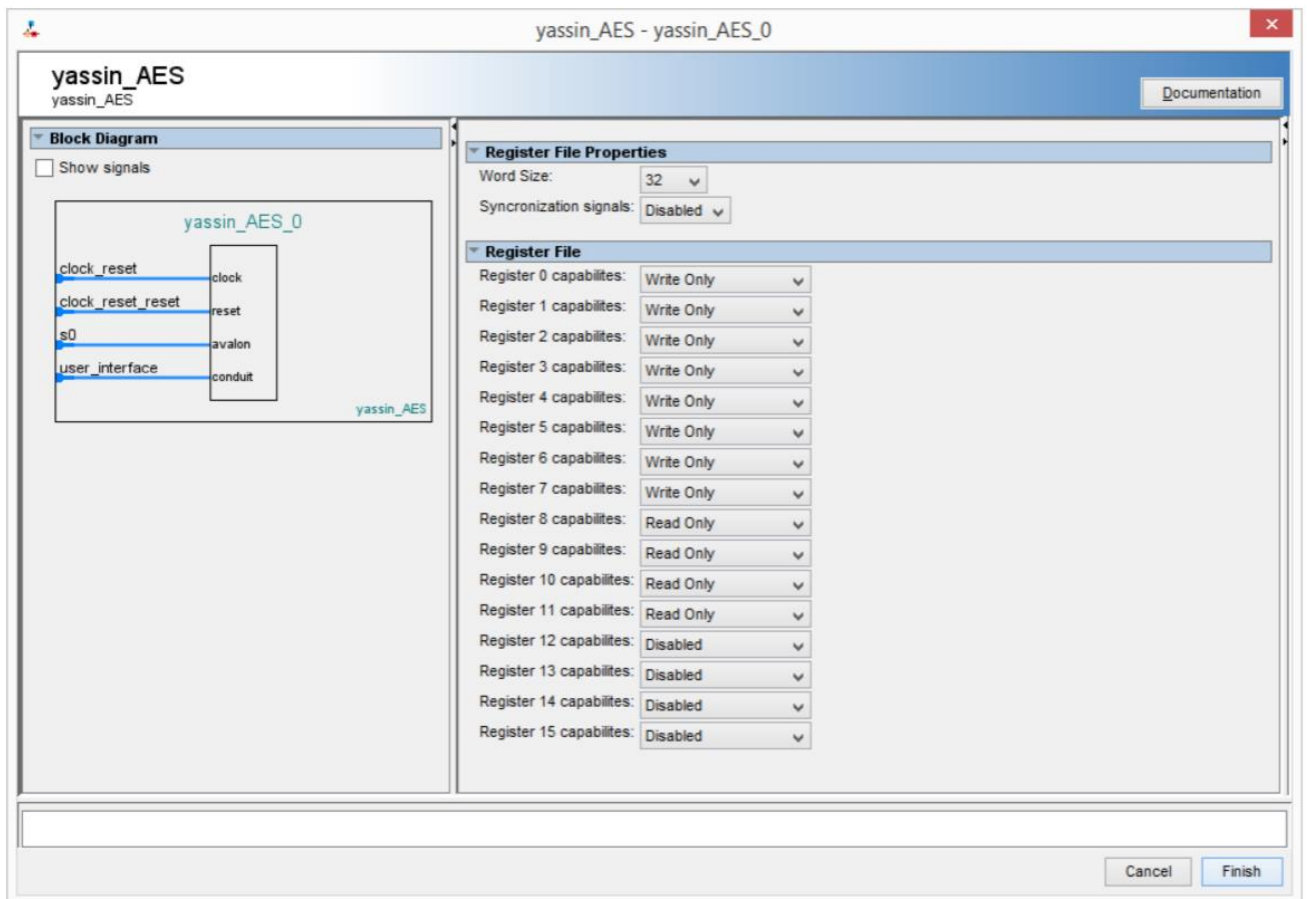
Description

Please use hexadecimal numbers only in System ID.

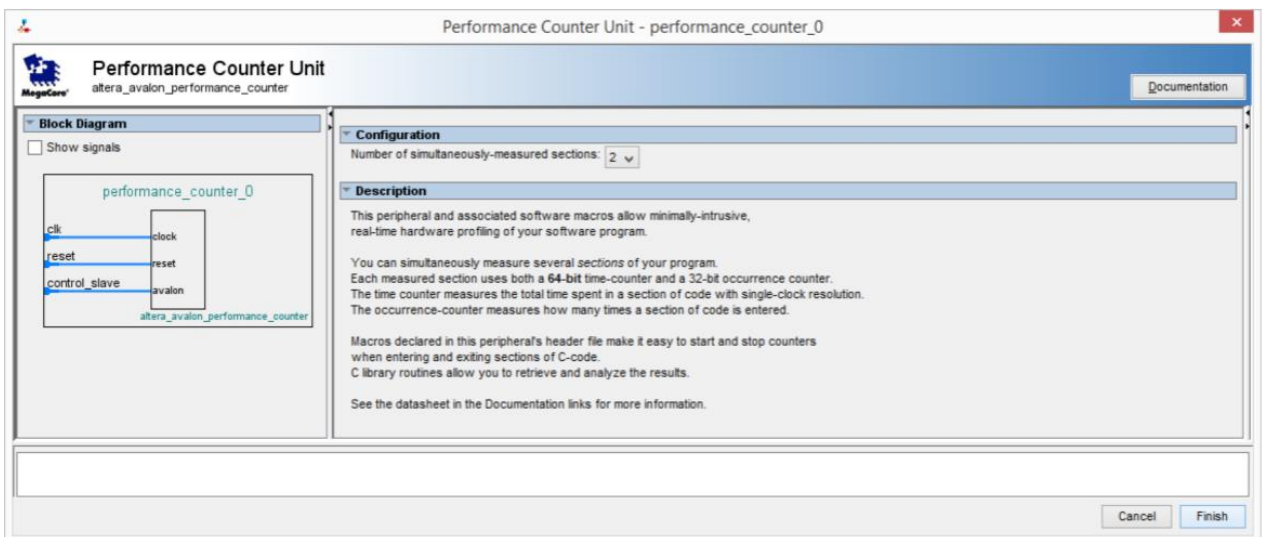
Info: sysid_qsys_0: System ID is not assigned automatically. Edit the System ID parameter to provide a unique ID

Info: sysid_qsys_0: Time stamp will be automatically updated when this component is generated.

Ajout du module Yassin_AES



Ajout du module Performance Counter UNIT



Connexion des différents modules.

Qsys

Component Library

Project

Library

Peripherals

Debug and Performance

Performance Counter

System Contents

Address Map

Clock Settings

Project Settings

Instance Parameters

System Inspector

HDL Example

Generation

| Name | Description | Export | Clock | Base | End | IRQ | Opcode Name |
|----------------------|-----------------------------|------------------------|-------|-------------|-------------|-----|-------------|
| clk_0 | Clock Source | clk_reset | clk_0 | | | | |
| clk_n | Clock Input | Double-click to export | clk_0 | | | | |
| clk_n_reset | Reset Input | Double-click to export | clk_0 | | | | |
| clk_reset | Clock Output | Double-click to export | clk_0 | | | | |
| onchip_memory2_0 | On-Chip Memory (RAM or ROM) | Double-click to export | clk_0 | 0x0000_8000 | 0x0000_0a1f | | |
| clk1 | Clock Input | Double-click to export | clk_0 | | | | |
| reset1 | Reset Input | Double-click to export | clk_0 | | | | |
| nios2_0sys_0 | Nios 2 Processor | Double-click to export | clk_0 | | | | |
| clk | Clock Input | Double-click to export | clk_0 | | | | |
| reset_n | Reset Input | Double-click to export | clk_0 | | | | |
| data_master | Avalon Memory Mapped Master | Double-click to export | clk_0 | | | | |
| instruction_master | Avalon Memory Mapped Master | Double-click to export | clk_0 | | | | |
| jtag_debug_module_n | Reset Output | Double-click to export | clk_0 | | | | |
| jtag_debug_module | Avalon Memory Mapped Slave | Double-click to export | clk_0 | 0x0001_0800 | 0x0001_0fff | | |
| custom_instruction_m | Custom Instruction Master | Double-click to export | clk_0 | | | | |
| jtag_uart_0 | JTAG UART | Double-click to export | clk_0 | | | | |
| clk | Clock Input | Double-click to export | clk_0 | | | | |
| reset | Reset Input | Double-click to export | clk_0 | | | | |
| avln_jtag_slave | Avalon Memory Mapped Slave | Double-click to export | clk_0 | 0x0001_10b8 | 0x0001_10bf | | |
| p10_0 | P10 (Parallel IO) | Double-click to export | clk_0 | | | | |
| clk | Clock Input | Double-click to export | clk_0 | | | | |
| reset | Reset Input | Double-click to export | clk_0 | | | | |
| avln_jtag_slave | Avalon Memory Mapped Slave | Double-click to export | clk_0 | 0x0001_10a0 | 0x0001_10af | | |
| external_connection | Conduit | Double-click to export | clk_0 | | | | |
| timer_0 | Interval Timer | Double-click to export | clk_0 | | | | |
| clk | Clock Input | Double-click to export | clk_0 | | | | |
| reset | Reset Input | Double-click to export | clk_0 | | | | |
| avln_jtag_slave | Avalon Memory Mapped Slave | Double-click to export | clk_0 | 0x0001_1080 | 0x0001_107f | | |
| sysid_0sys_0 | System ID Peripheral | Double-click to export | clk_0 | | | | |
| clk | Clock Input | Double-click to export | clk_0 | | | | |
| reset | Reset Input | Double-click to export | clk_0 | | | | |
| avln_jtag_slave | Avalon Memory Mapped Slave | Double-click to export | clk_0 | 0x0001_10a0 | 0x0001_10af | | |
| clock_reset | Clock Input | Double-click to export | clk_0 | | | | |
| clock_reset_reset | Reset Input | Double-click to export | clk_0 | 0x0001_1040 | 0x0001_107f | | |
| avln_jtag_slave | Avalon Memory Mapped Slave | Double-click to export | clk_0 | | | | |
| user_interface | Conduit | Double-click to export | clk_0 | | | | |
| performance_counter | Performance Counter Unit | Double-click to export | clk_0 | | | | |
| clk | Clock Input | Double-click to export | clk_0 | | | | |
| reset | Reset Input | Double-click to export | clk_0 | | | | |
| control_slave | Avalon Memory Mapped Slave | Double-click to export | clk_0 | 0x0001_1000 | 0x0001_103f | | |

Messages

2 Warnings

pio_0external_connection must be exported, or connected to a matching conduit.

System.pio_0

yassin_AES_0user_interface must be exported, or connected to a matching conduit.

System.yassin_AES_0

2 Info Messages

System ID is not assigned automatically. Edit the System ID parameter to provide a unique ID.

System.sysid_0sys_0

Time stamp will be automatically updated when this component is generated.

System.sysid_0sys_0

0 Errors, 2 Warnings

Assignation des adresses de base et exportation des sorties LED et AES

Qsys

Component Library

Project

Library

Peripherals

Debug and Performance

Performance Counter

System Contents

Address Map

Clock Settings

Project Settings

Instance Parameters

System Inspector

HDL Example

Generation

| Name | Description | Export | Clock | Base | End | IRQ | Opcode Name |
|----------------------|-----------------------------|------------------------|-------|-------------|-------------|-----|-------------|
| clk_0 | Clock Source | clk_reset | clk_0 | | | | |
| clk_n | Clock Input | Double-click to export | clk_0 | | | | |
| clk_n_reset | Reset Input | Double-click to export | clk_0 | | | | |
| clk_reset | Clock Output | Double-click to export | clk_0 | | | | |
| onchip_memory2_0 | On-Chip Memory (RAM or ROM) | Double-click to export | clk_0 | 0x0000_8000 | 0x0000_0a1f | | |
| clk1 | Clock Input | Double-click to export | clk_0 | | | | |
| reset1 | Reset Input | Double-click to export | clk_0 | | | | |
| nios2_0sys_0 | Nios 2 Processor | Double-click to export | clk_0 | | | | |
| clk | Clock Input | Double-click to export | clk_0 | | | | |
| reset_n | Reset Input | Double-click to export | clk_0 | | | | |
| data_master | Avalon Memory Mapped Master | Double-click to export | clk_0 | | | | |
| instruction_master | Avalon Memory Mapped Master | Double-click to export | clk_0 | | | | |
| jtag_debug_module_n | Reset Output | Double-click to export | clk_0 | | | | |
| jtag_debug_module | Avalon Memory Mapped Slave | Double-click to export | clk_0 | 0x0001_0800 | 0x0001_0fff | | |
| custom_instruction_m | Custom Instruction Master | Double-click to export | clk_0 | | | | |
| jtag_uart_0 | JTAG UART | Double-click to export | clk_0 | | | | |
| clk | Clock Input | Double-click to export | clk_0 | | | | |
| reset | Reset Input | Double-click to export | clk_0 | | | | |
| avln_jtag_slave | Avalon Memory Mapped Slave | Double-click to export | clk_0 | 0x0001_10b8 | 0x0001_10bf | | |
| p10_0 | P10 (Parallel IO) | Double-click to export | clk_0 | | | | |
| clk | Clock Input | Double-click to export | clk_0 | | | | |
| reset | Reset Input | Double-click to export | clk_0 | | | | |
| avln_jtag_slave | Avalon Memory Mapped Slave | Double-click to export | clk_0 | 0x0001_10a0 | 0x0001_10af | | |
| external_connection | Conduit | Double-click to export | clk_0 | | | | |
| timer_0 | Interval Timer | Double-click to export | clk_0 | | | | |
| clk | Clock Input | Double-click to export | clk_0 | | | | |
| reset | Reset Input | Double-click to export | clk_0 | | | | |
| avln_jtag_slave | Avalon Memory Mapped Slave | Double-click to export | clk_0 | 0x0001_1080 | 0x0001_107f | | |
| sysid_0sys_0 | System ID Peripheral | Double-click to export | clk_0 | | | | |
| clk | Clock Input | Double-click to export | clk_0 | | | | |
| reset | Reset Input | Double-click to export | clk_0 | | | | |
| avln_jtag_slave | Avalon Memory Mapped Slave | Double-click to export | clk_0 | 0x0001_10a0 | 0x0001_10af | | |
| clock_reset | Clock Input | Double-click to export | clk_0 | | | | |
| clock_reset_reset | Reset Input | Double-click to export | clk_0 | 0x0001_1040 | 0x0001_107f | | |
| avln_jtag_slave | Avalon Memory Mapped Slave | Double-click to export | clk_0 | | | | |
| user_interface | Conduit | Double-click to export | clk_0 | | | | |
| performance_counter | Performance Counter Unit | Double-click to export | clk_0 | | | | |
| clk | Clock Input | Double-click to export | clk_0 | | | | |
| reset | Reset Input | Double-click to export | clk_0 | | | | |
| control_slave | Avalon Memory Mapped Slave | Double-click to export | clk_0 | 0x0001_1000 | 0x0001_103f | | |

Messages

2 Info Messages

System ID is not assigned automatically. Edit the System ID parameter to provide a unique ID.

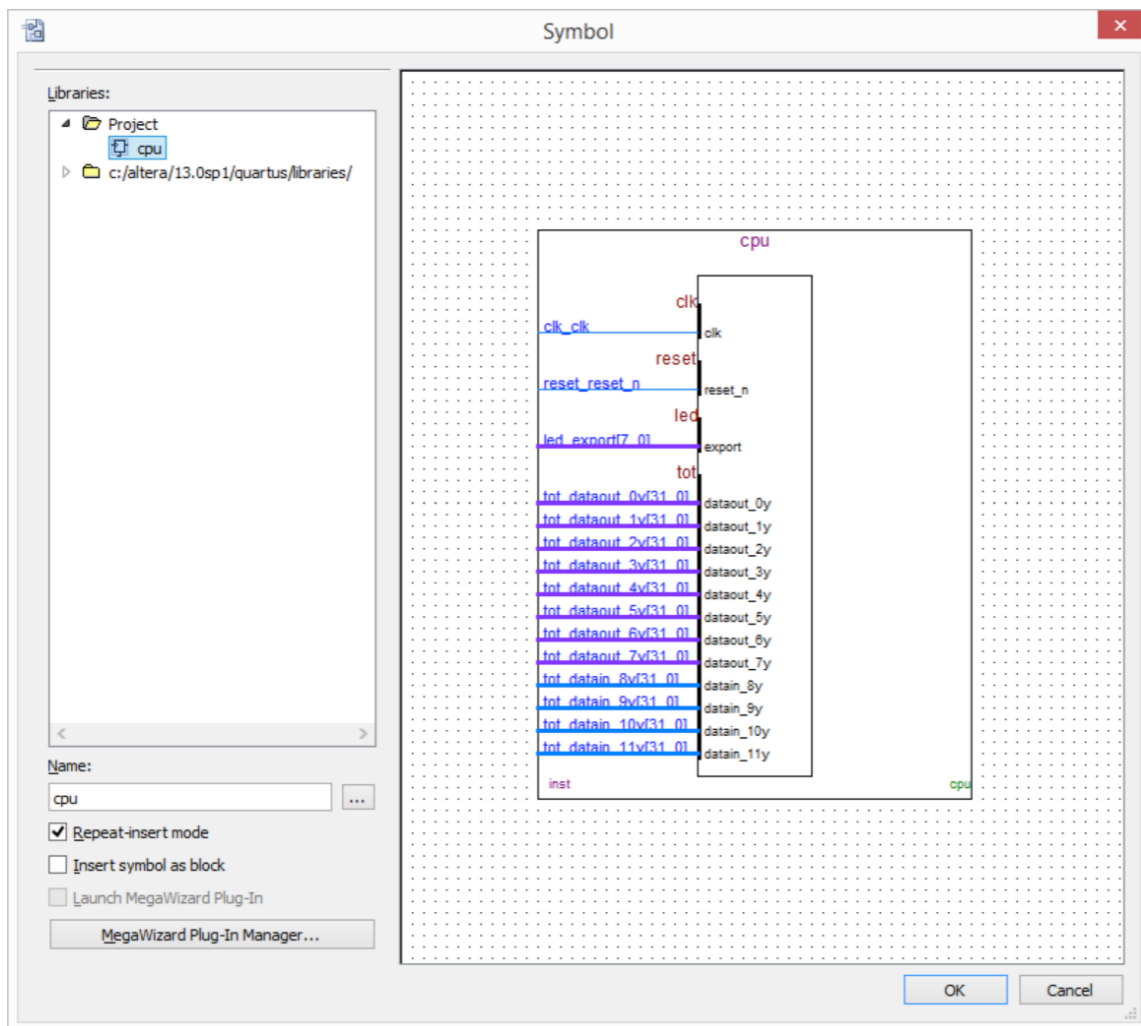
System.sysid_0sys_0

Time stamp will be automatically updated when this component is generated.

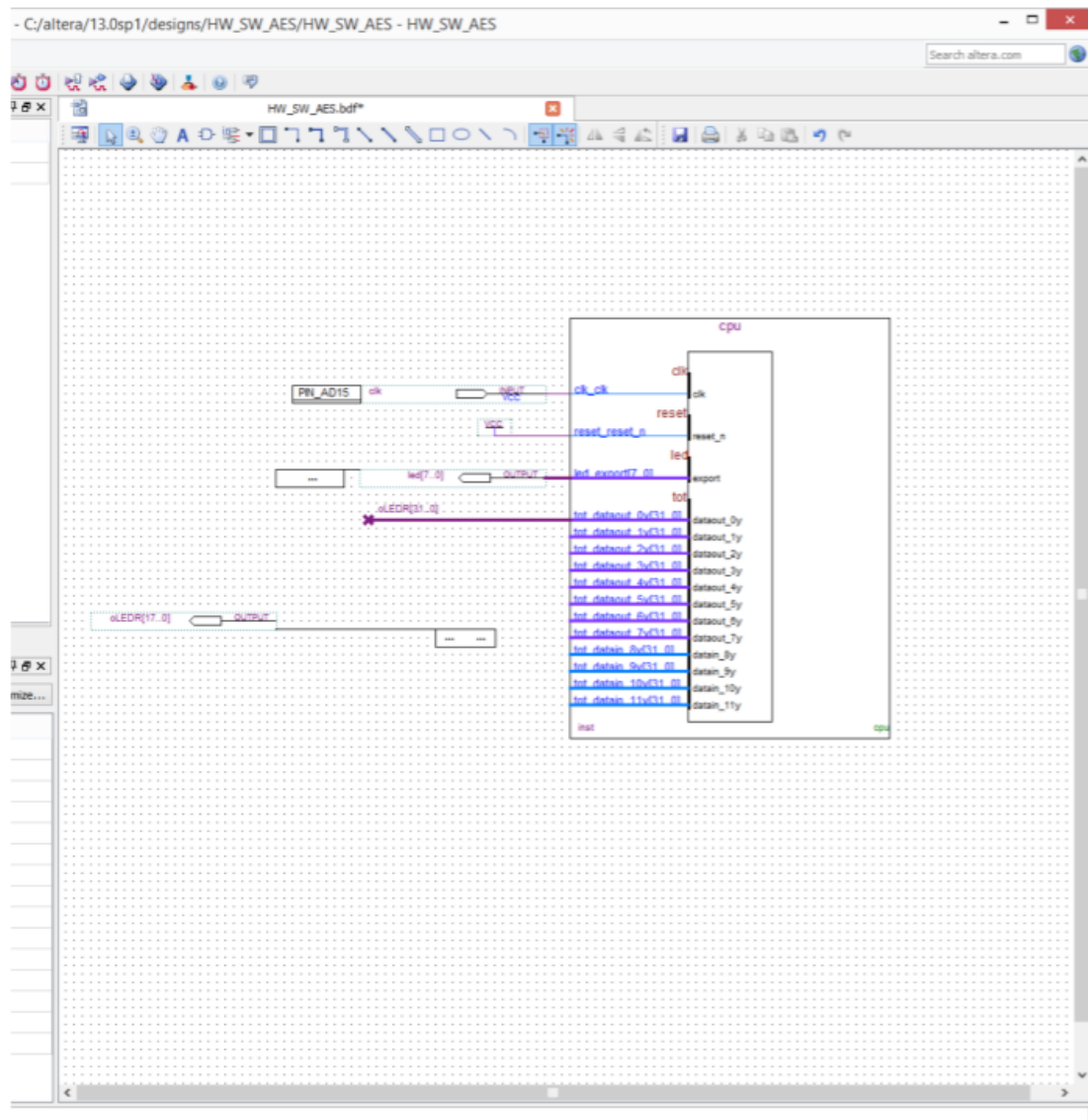
System.sysid_0sys_0

0 Errors, 0 Warnings

Génération du système



Vérification et modification des connexions CPU



Compilation du fichier et résultat :

The screenshot displays the Quartus II 64-bit software interface for a project named 'HW_SW_AES'. The main window is divided into several panes:

- Project Navigator:** Shows the project hierarchy with files like 'HW_SW_AES.qsf', 'HW_SW_AES.bdf', and 'HW_SW_AES.sdc'.
- Table of Contents:** Lists the contents of the project, including 'Flow Summary', 'Flow Settings', 'Flow Non-Default Global Settings', 'Flow Elapsed Time', 'Flow OS Summary', 'Flow Log', 'Analysis & Synthesis', 'Filter', 'Flow Messages', 'Flow Suppressed Messages', 'Assembler', and 'TimeQuest Timing Analyzer'.
- Completion Report - HW_SW_AES:** Displays the results of the compilation process, including the 'Flow Summary' and 'Flow Settings'.
- Messages:** Shows the output of the compilation process, including warnings and errors.

The Messages window contains the following text:

```

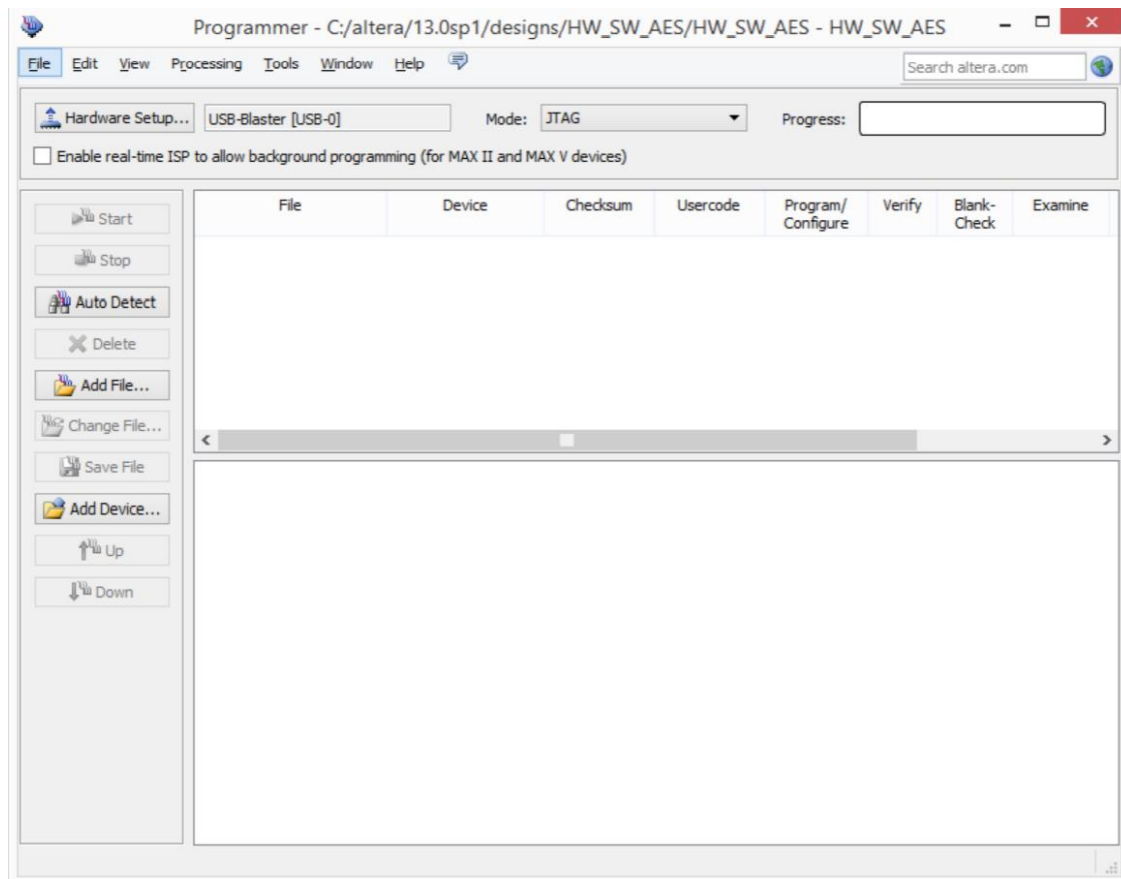
332001 The selected device family is not supported by the report_metastability command.
Analyzing Fast Model
332040 Model: clk was determined to be a clock but was found without an associated clock assignment.
332140 No Setup paths to report
332140 No Hold paths to report
332140 No Recovery paths to report
332140 No Removal paths to report
332140 Worst-case minimum pulse width slack is 97.778
332001 The selected device family is not supported by the report_metastability command.
332102 Design is not fully constrained for setup requirements
332102 Design is not fully constrained for hold requirements
Quartus II 64-Bit TimeQuest Timing Analyzer was successful. 0 errors, 3 warnings
293000 Quartus II Full Compilation was successful. 0 errors, 131 warnings

```

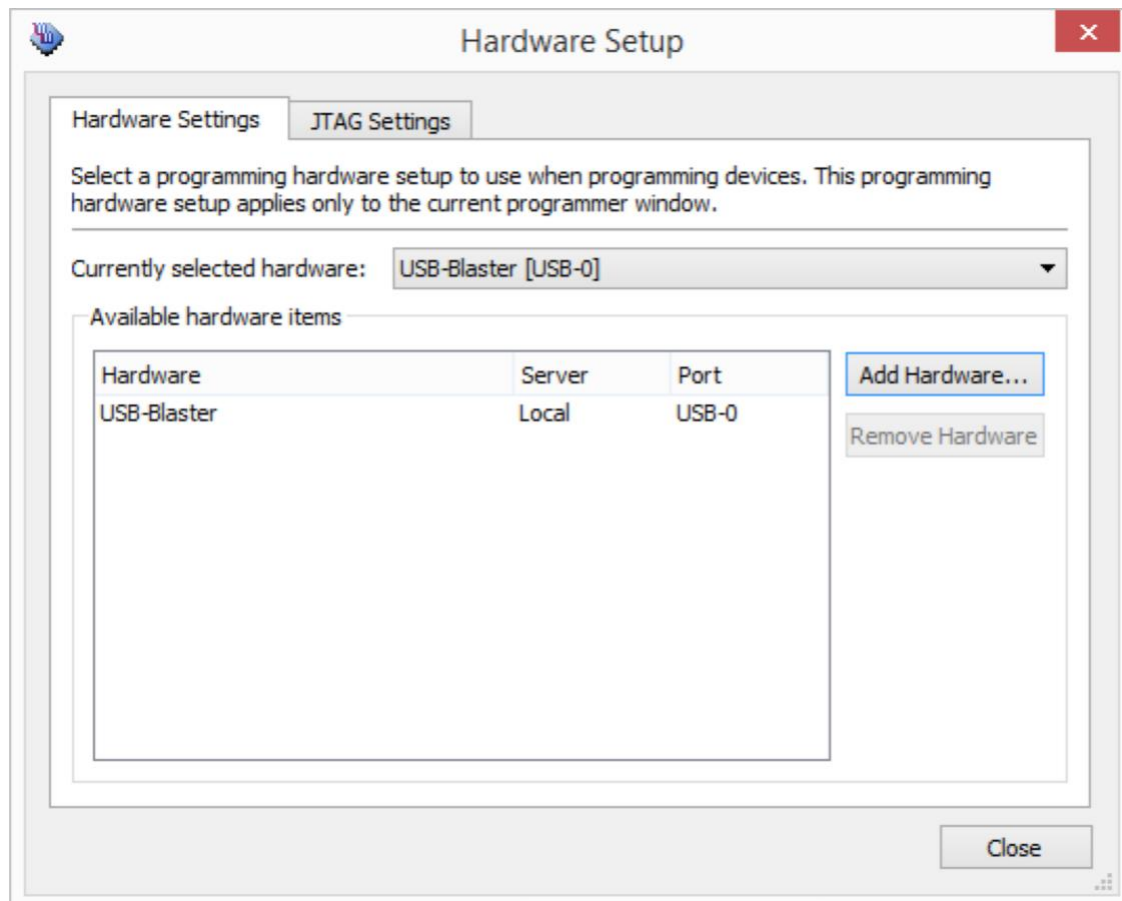
Programmation de la carte

Lancement et configuration

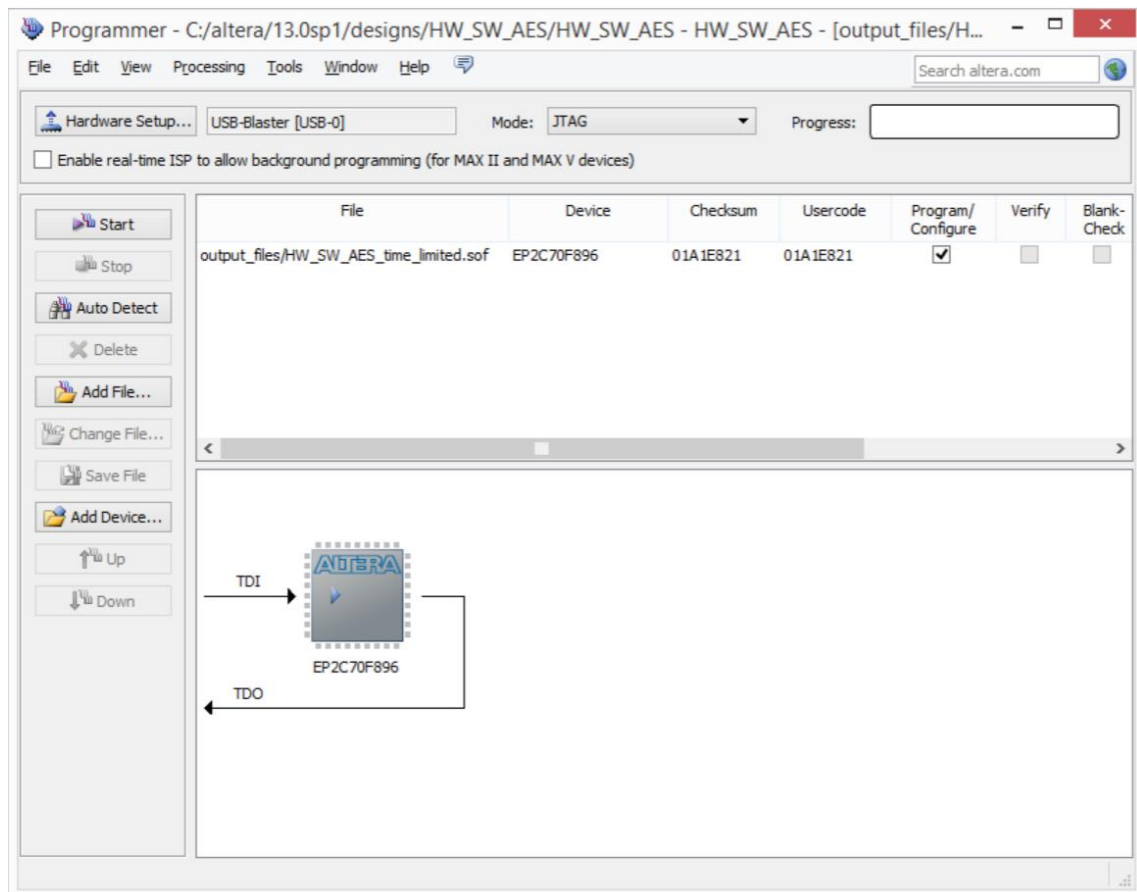
Utilisation de la fonction hardware setup pour connecter la carte



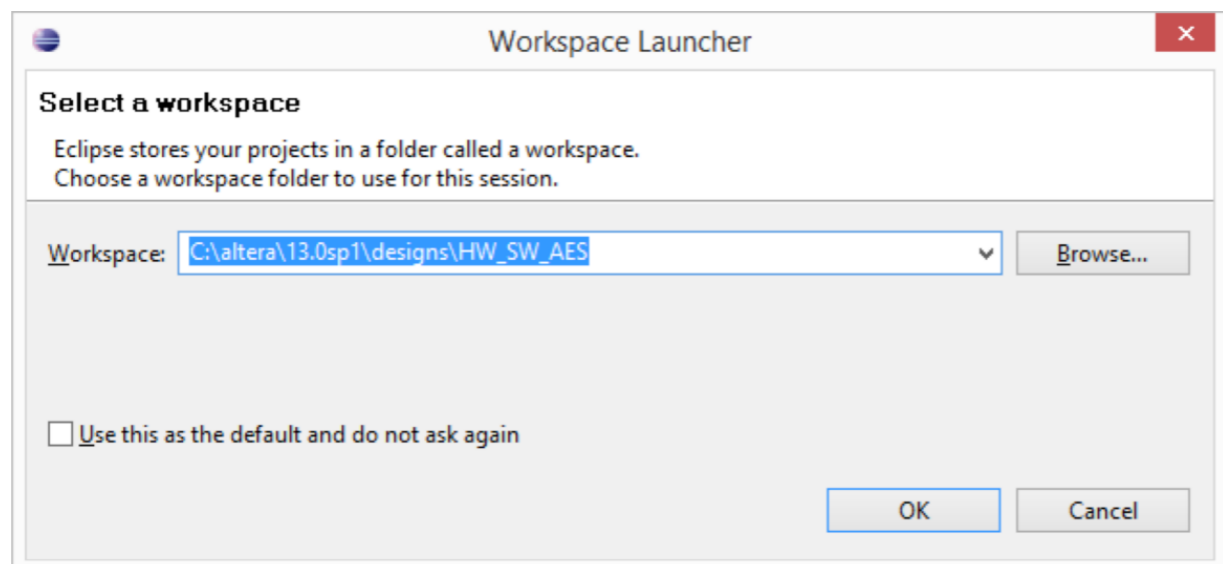
Connection via le port USB-Blaster



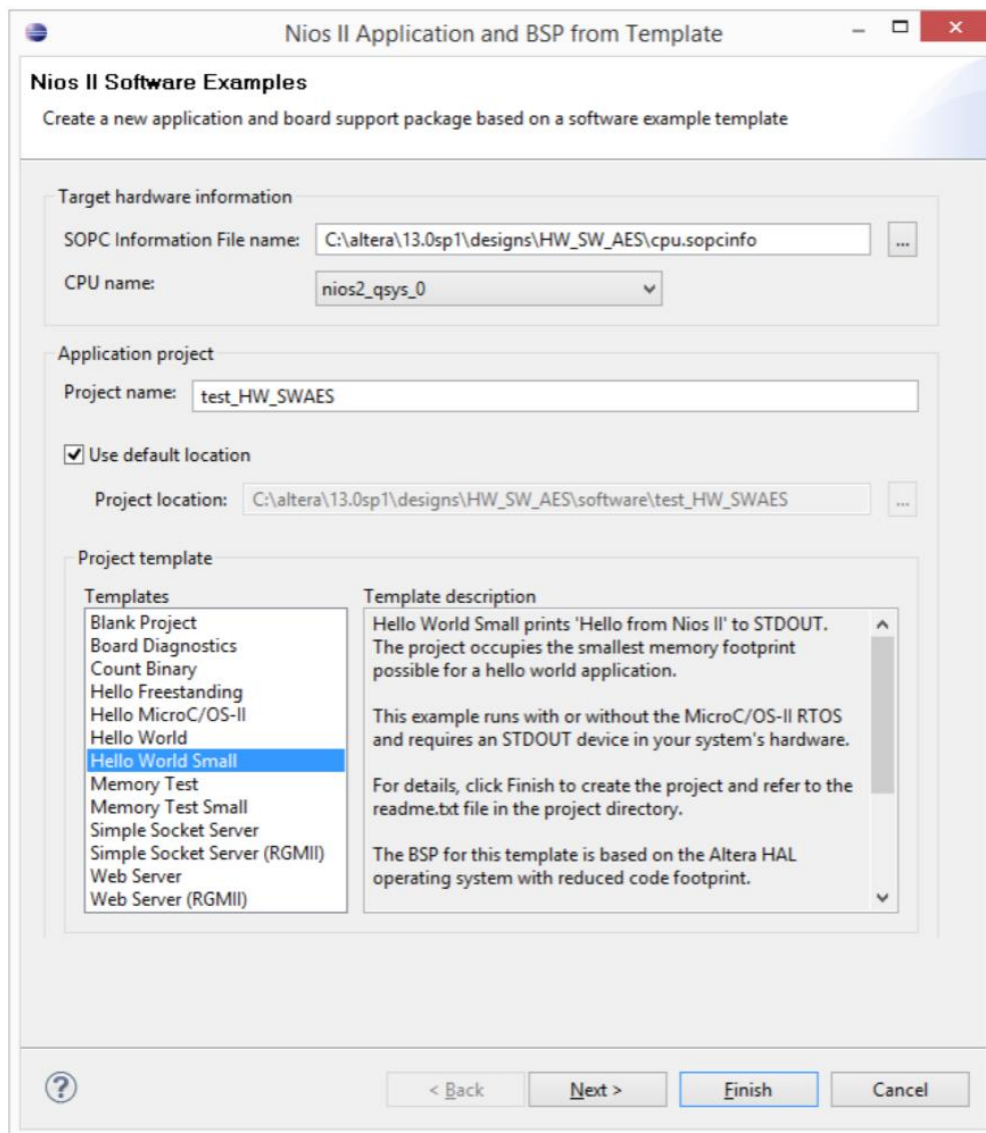
Utilisation de la fonction auto détecte pour trouver le module



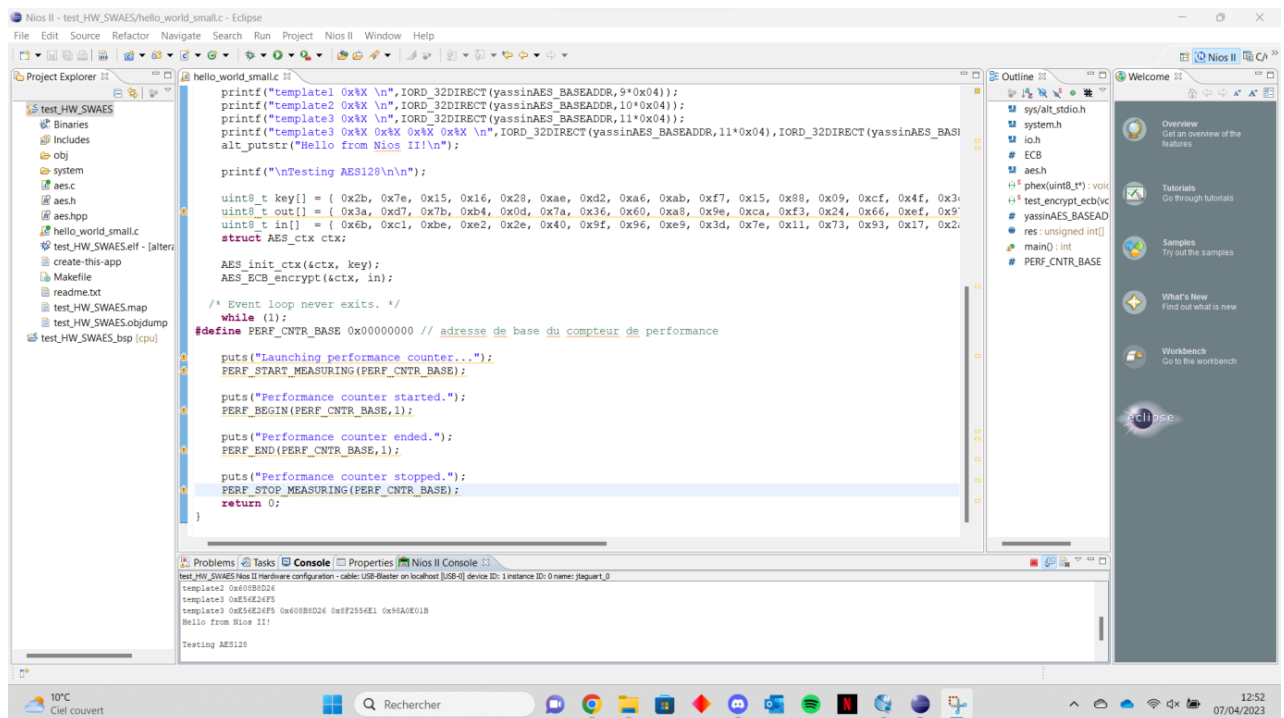
Ouverture de Eclipse et création du Workspace



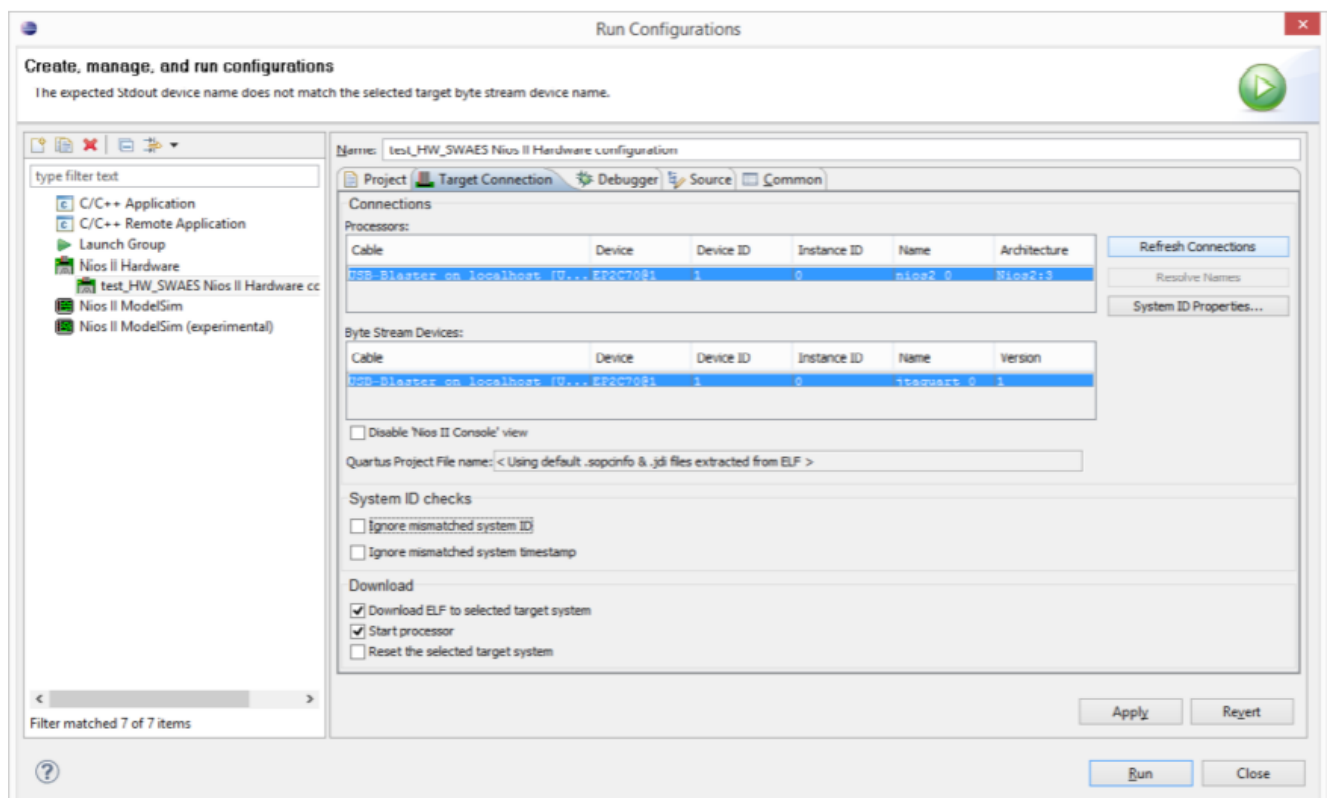
Création du nouveau projet BSP, utilisation de Template par défaut



Injection du code dans l'IDE

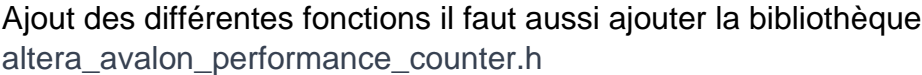


Configuration de Target connections puis Refrech connections



Test de performance et comparaison

Ajout des différentes fonctions puis vérification du Run



Ajout des différentes fonctions il faut aussi ajouter la bibliothèque `altera_avalon_performance_counter.h`

```

#include "sys/alt_stdio.h"
#include <system.h>
#include <io.h>

// Definition des adresses de base des composants
#define yassinAES_BASEADDR YASSIN_AES_0_BASE
#define PERF_CNTR_BASE 0x12345678 // Remplacez par l'adresse de base du compteur de performances

// Declaration des variables globales
unsigned int res[4];

// Declaration des fonctions
void PERF_BEGIN(unsigned int base_addr, unsigned int counter);
void PERF_END(unsigned int base_addr, unsigned int counter);
unsigned int perf_get_section_time(unsigned int base_addr, unsigned int counter);

// Fonction principale
int main()
{
    alt_putstr("Hello from Nios II!\n");
    alt_putstr("Load KEY!\n");
    IOWR_32DIRECT(yassinAES_BASEADDR, 0x00, 0x11111111);
    IOWR_32DIRECT(yassinAES_BASEADDR, 1 * 0x04, 0x11111111);
    IOWR_32DIRECT(yassinAES_BASEADDR, 2 * 0x04, 0x11111111);
    IOWR_32DIRECT(yassinAES_BASEADDR, 3 * 0x04, 0x11111111);
    alt_putstr("Load data!\n");
    IOWR_32DIRECT(yassinAES_BASEADDR, 4 * 0x04, 0x11111111);
    IOWR_32DIRECT(yassinAES_BASEADDR, 5 * 0x04, 0x11111111);
    IOWR_32DIRECT(yassinAES_BASEADDR, 6 * 0x04, 0x11111111);
    IOWR_32DIRECT(yassinAES_BASEADDR, 7 * 0x04, 0x11111111);
    alt_putstr("retrive output!\n");

    PERF_BEGIN(PERF_CNTR_BASE, 1); // D but de la mesure de performance

    printf("template0 0x%X \n", IORD_32DIRECT(yassinAES_BASEADDR, 8 * 0x04));
    printf("template1 0x%X \n", IORD_32DIRECT(yassinAES_BASEADDR, 9 * 0x04));
    printf("template2 0x%X \n", IORD_32DIRECT(yassinAES_BASEADDR, 10 * 0x04));
    printf("template3 0x%X \n", IORD_32DIRECT(yassinAES_BASEADDR, 11 * 0x04));
    printf("template3 0x%X 0x%X 0x%X 0x%X\n",
        IORD_32DIRECT(yassinAES_BASEADDR, 11 * 0x04),
        IORD_32DIRECT(yassinAES_BASEADDR, 10 * 0x04),
        IORD_32DIRECT(yassinAES_BASEADDR, 9 * 0x04),
        IORD_32DIRECT(yassinAES_BASEADDR, 8 * 0x04));

    PERF_END(PERF_CNTR_BASE, 1); // Fin de la mesure de performance

    unsigned int section_time = perf_get_section_time(PERF_CNTR_BASE, 1);
    printf("Temps de la section mesuree : %u cycles\n", section_time);

    alt_putstr("Hello from Nios II!\n");

    while (1)
    {
    }

    return 0;
}

// Fonctions de mesure de performance
void PERF_BEGIN(unsigned int base_addr, unsigned int counter)
{
    // Assurez-vous que le compteur est reinitialise en premier
    IOWR_32DIRECT(base_addr, counter * 0x04, 0);

    // Commencez la mesure de performance en activant le compteur
    IOWR_32DIRECT(base_addr, counter * 0x04 + 1, 1);
}

// Fonctions de mesure de performance
void PERF_END(unsigned int base_addr, unsigned int counter)
{
    // Arrêtez la mesure de performance en desactivant le compteur
    IOWR_32DIRECT(base_addr, counter * 0x04 + 1, 0);
}

// Fonctions de mesure de performance
unsigned int perf_get_section_time(unsigned int base_addr, unsigned int counter)
{
    // Lisez la valeur actuelle du compteur
    unsigned int counter_value = IORD_32DIRECT(base_addr, counter * 0x04);

    // Retournez la valeur du compteur, qui represente le temps ecoule pendant la section mesuree
    return counter_value;
}

```