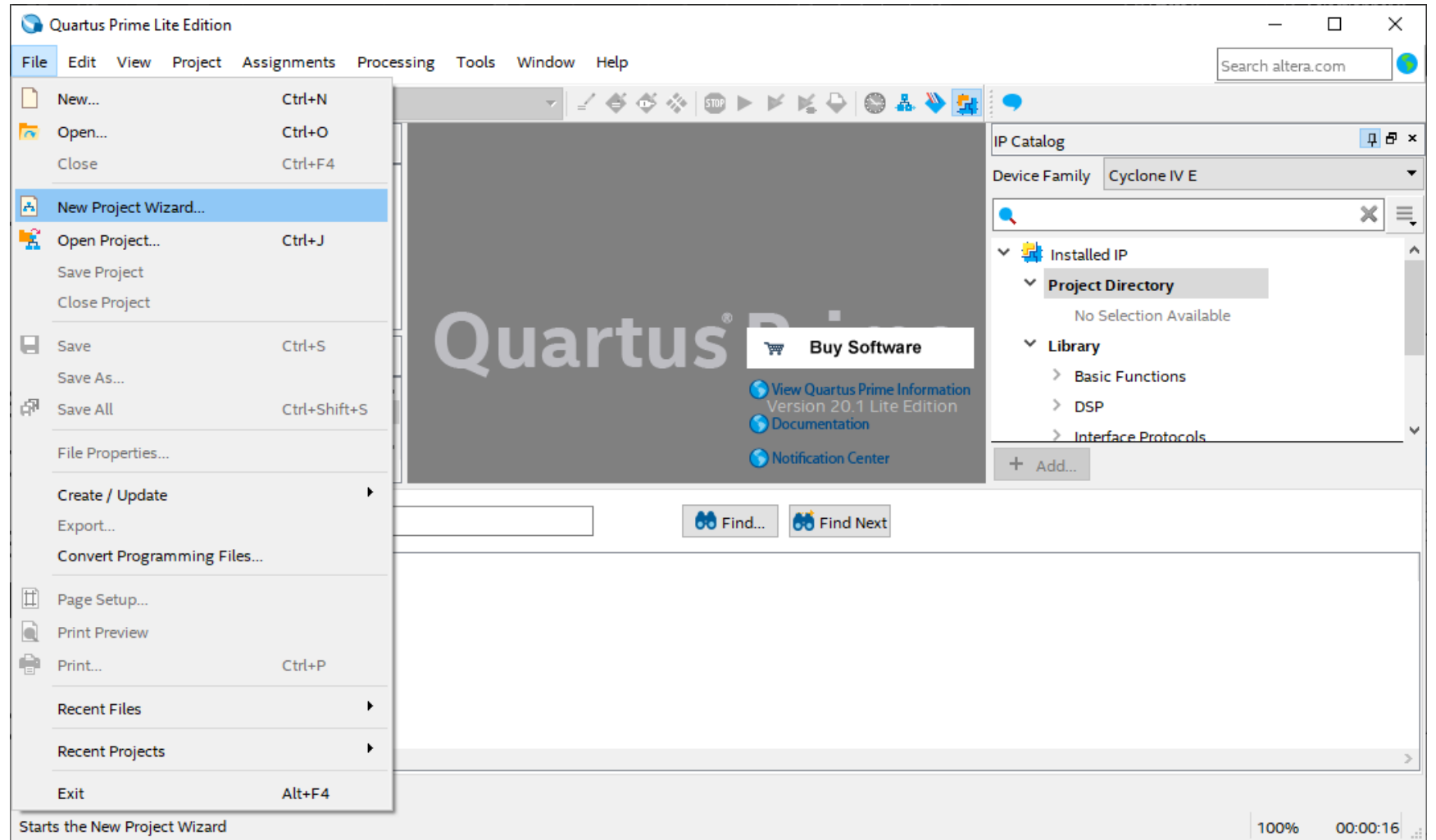




Guide d'utilisation de Quartus

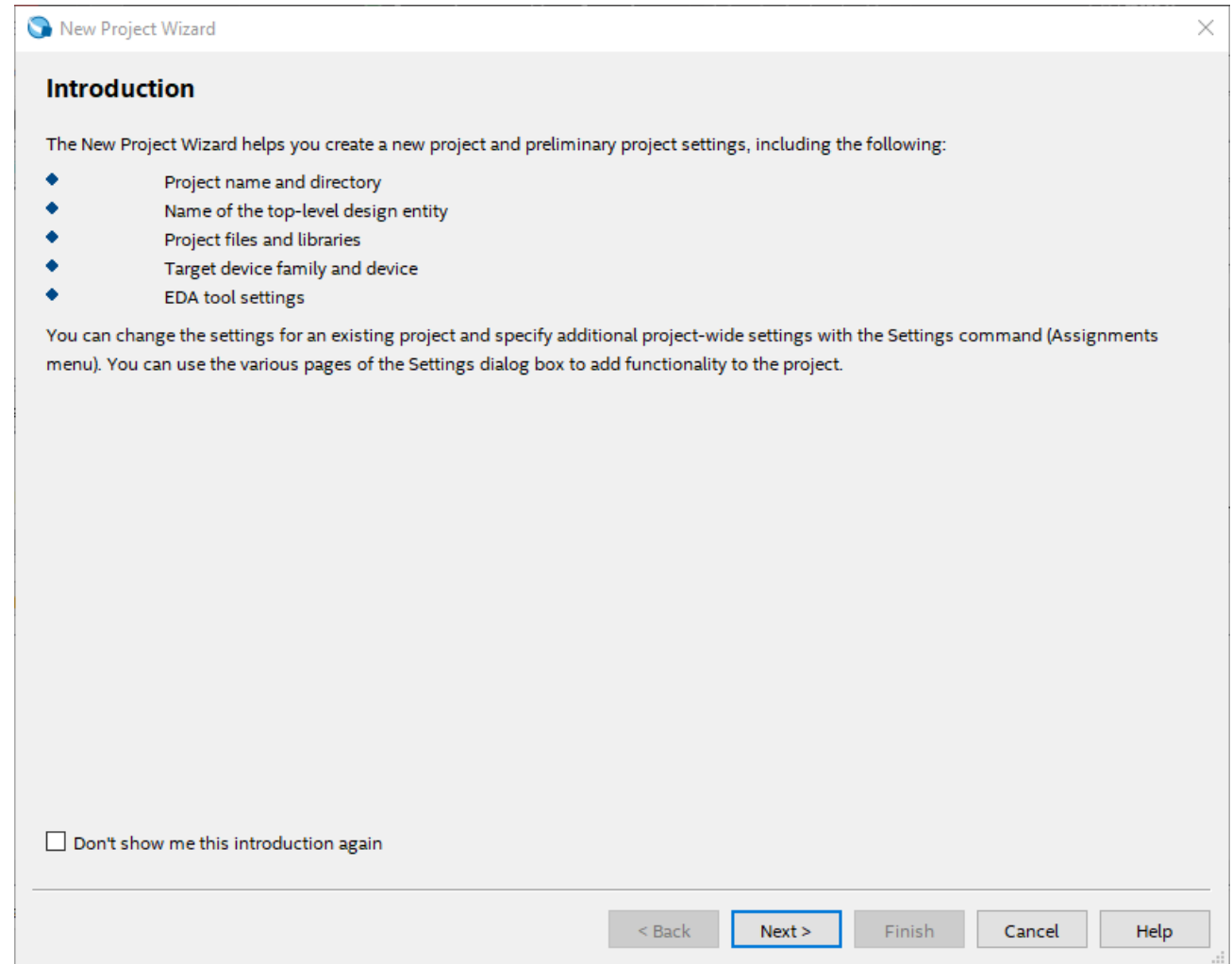
Création d'un projet

- Ouvrir Quartus
- Cliquer dans File → Open Project



Création d'un projet

- Cliquer sur Next



Création d'un projet

New Project Wizard

Directory, Name, Top-Level Entity

What is the working directory for this project?

PATH_WHERE_YOU_WANT_TO_SAVE_YOUR_PROJECT

What is the name of this project?

NAME_OF_YOUR_PROJECT

What is the name of the top-level design entity for this project? This name is case sensitive and must exactly match the entity name in the design file.

NAME_OF_YOUR_PROJECT

Use Existing Project Settings...

< Back Next > Finish Cancel Help

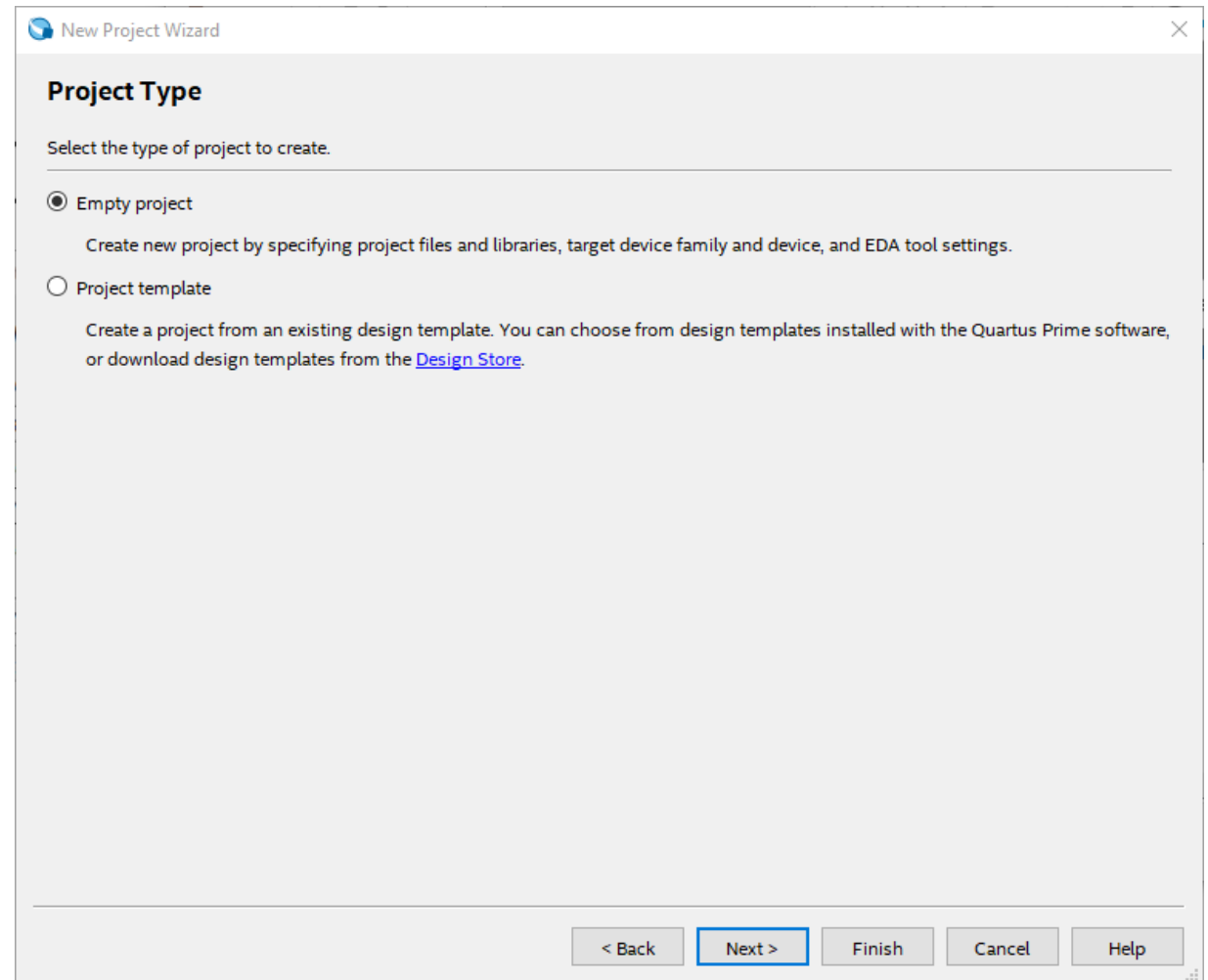
Chemin où le projet sera sauvegardé

Nom du projet

Cliquer sur Next

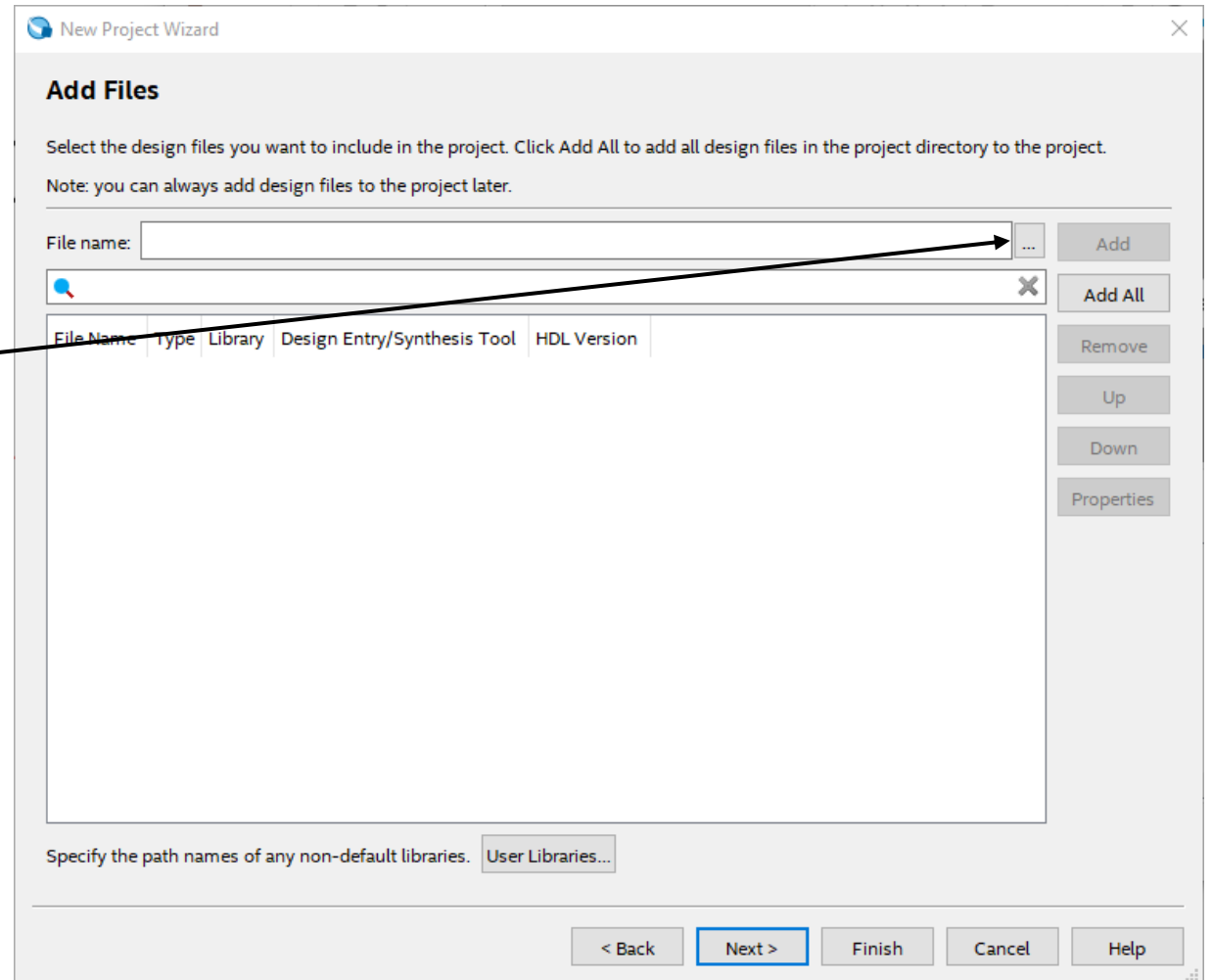
Création d'un projet

- Choisir « Empty project »
- Cliquer sur « Next »



Création d'un projet

- Ici on peut ajouter des sources VHDL existantes.
- Cliquer sur « ... » pour aller les chercher
- Ce n'est pas obligatoire d'ajouter des sources dès le début. On peut créer un projet vide.



Création d'un projet

- Sélectionner le FPGA que vous avez en regardant sur la puce de votre carte ou en regardant le manuel d'utilisation de votre carte.
- Cliquer sur « Next » jusqu'à arriver à la fin

New Project Wizard

Family, Device & Board Settings

Device Board

Select the family and device you want to target for compilation.
You can install additional device support with the Install Devices command on the Tools menu.

To determine the version of the Quartus Prime software in which your target device is supported, refer to the [Device Support List](#) webpage.

Device family

Family: Cyclone IV E

Device: All

Target device

☒ Auto device selected by the Fitter

☐ Specific device selected in 'Available devices' list

☐ Other: n/a

Show in 'Available devices' list

Package: FBGA

Pin count: 256

Core speed grade: 6

Name filter:

☒ Show advanced devices

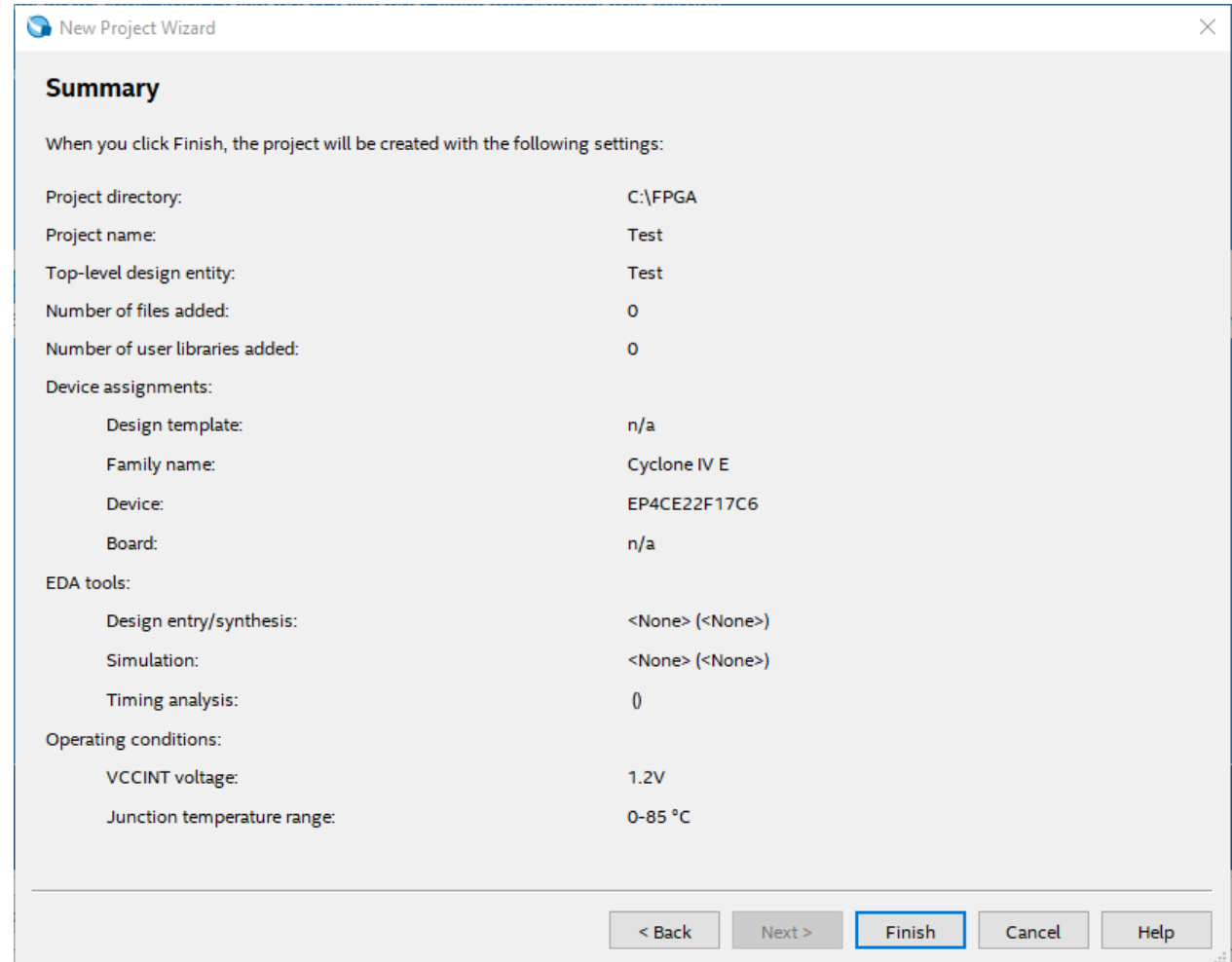
Available devices:

Name	Core Voltage	LEs	Total I/Os	GPIOs	Memory Bits	Embedded multiplier 9-bit elements	PLLs	Global Clocks
EP4CE6F17C6	1.2V	6272	180	180	276480	30	2	10
EP4CE10F17C6	1.2V	10320	180	180	423936	46	2	10
EP4CE15F17C6	1.2V	15408	166	166	516096	112	4	20
EP4CE22F17C6	1.2V	22320	154	154	608256	132	4	20

< Back Next > Finish Cancel Help

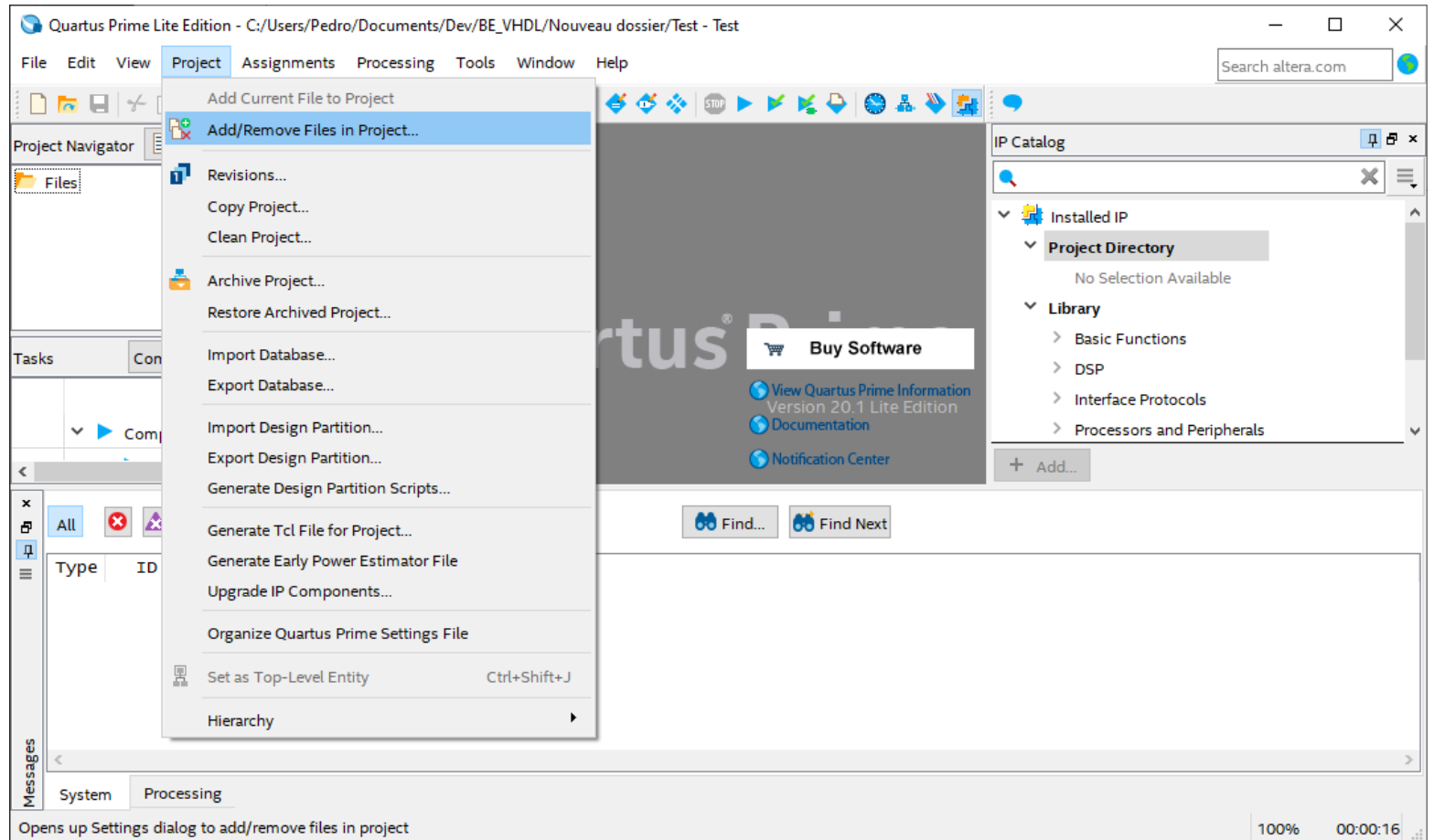
Création d'un projet

- Un résumé du projet est faite à la fin.
- Cliquer sur « Finish »

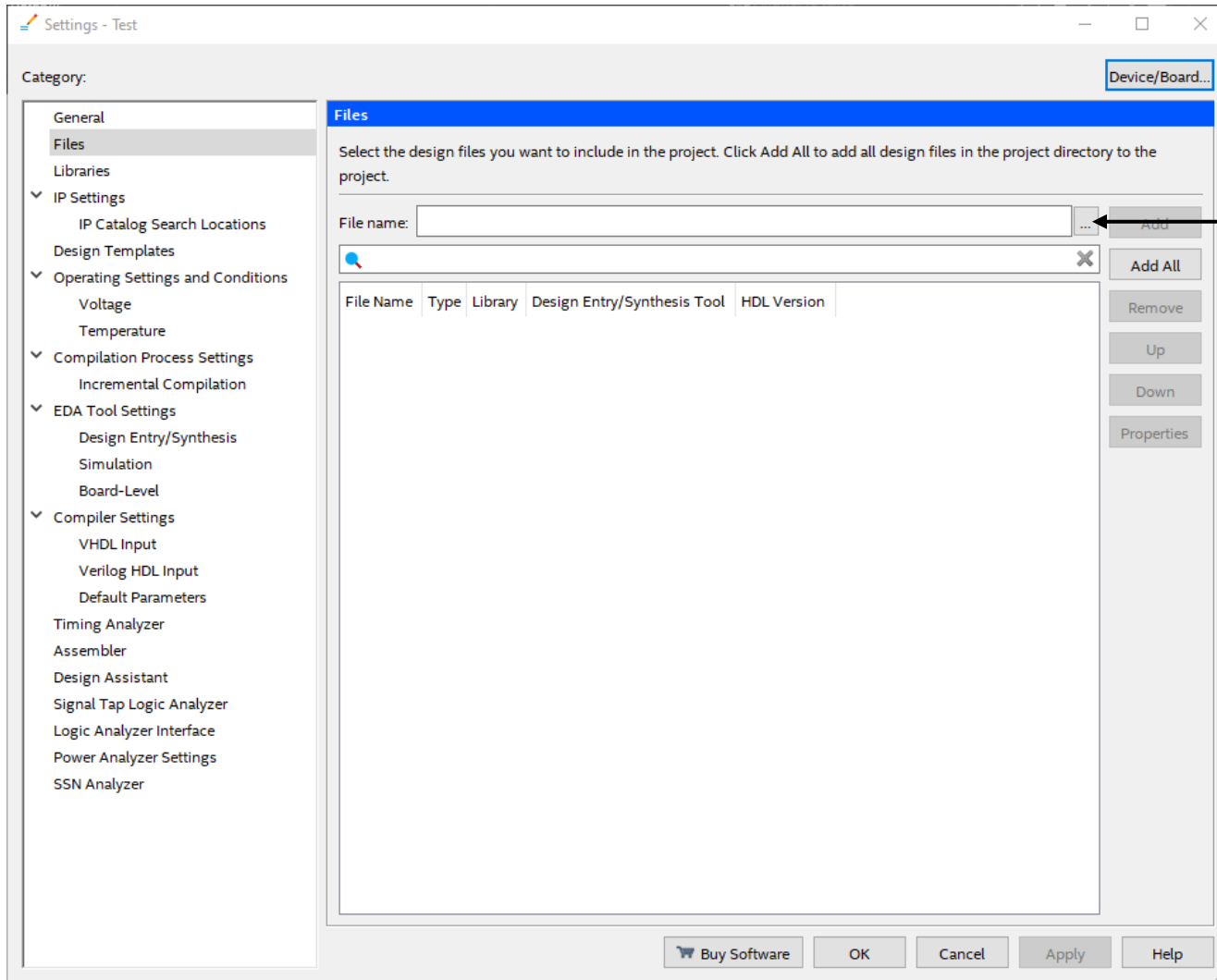


Ajouter des sources

- Cliquer dans
« Project →
Add/Remove Files in
Project... »



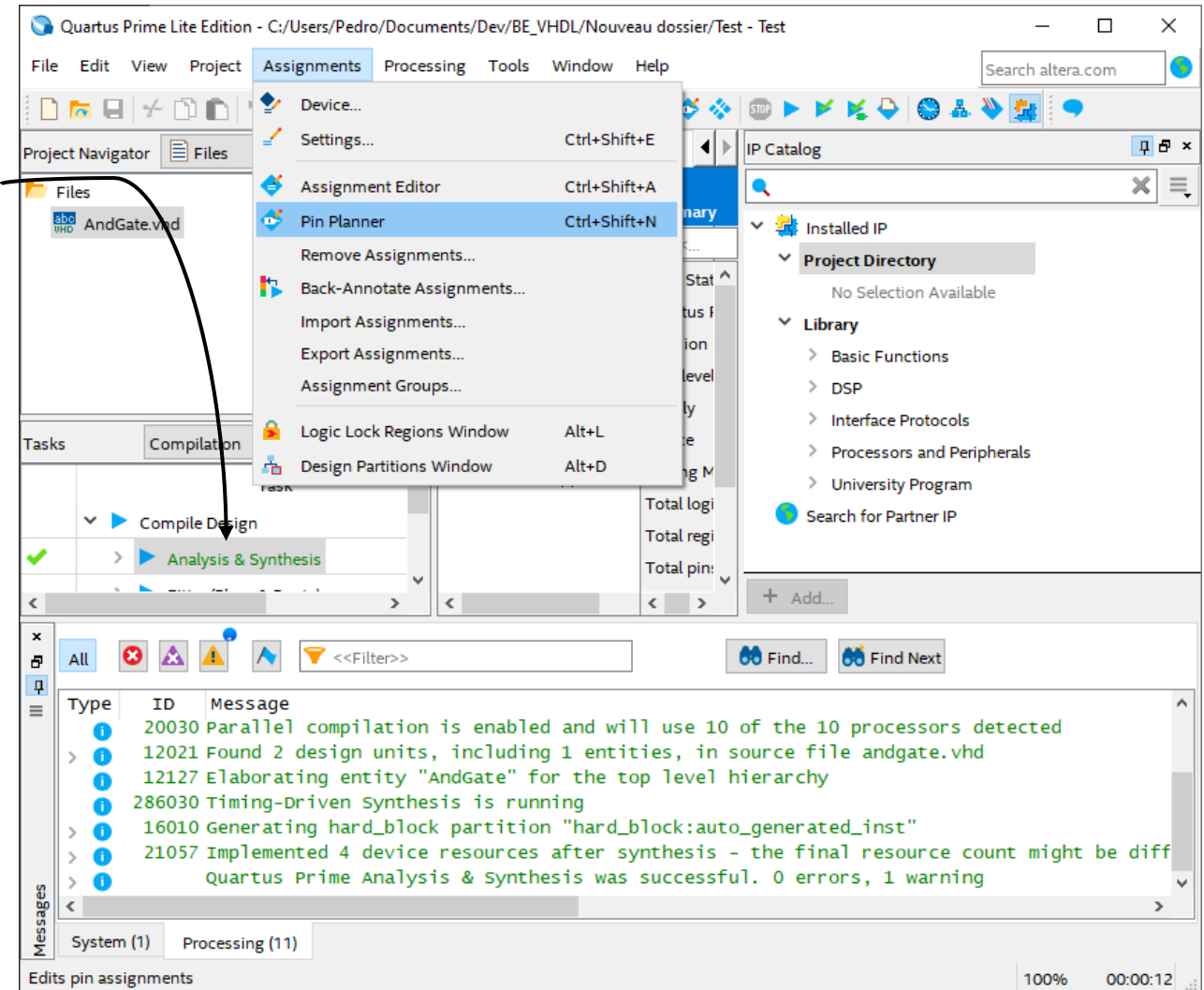
Ajouter des sources



Cliquer sur « ... »
pour ajouter des
sources existantes

Affectation des broches

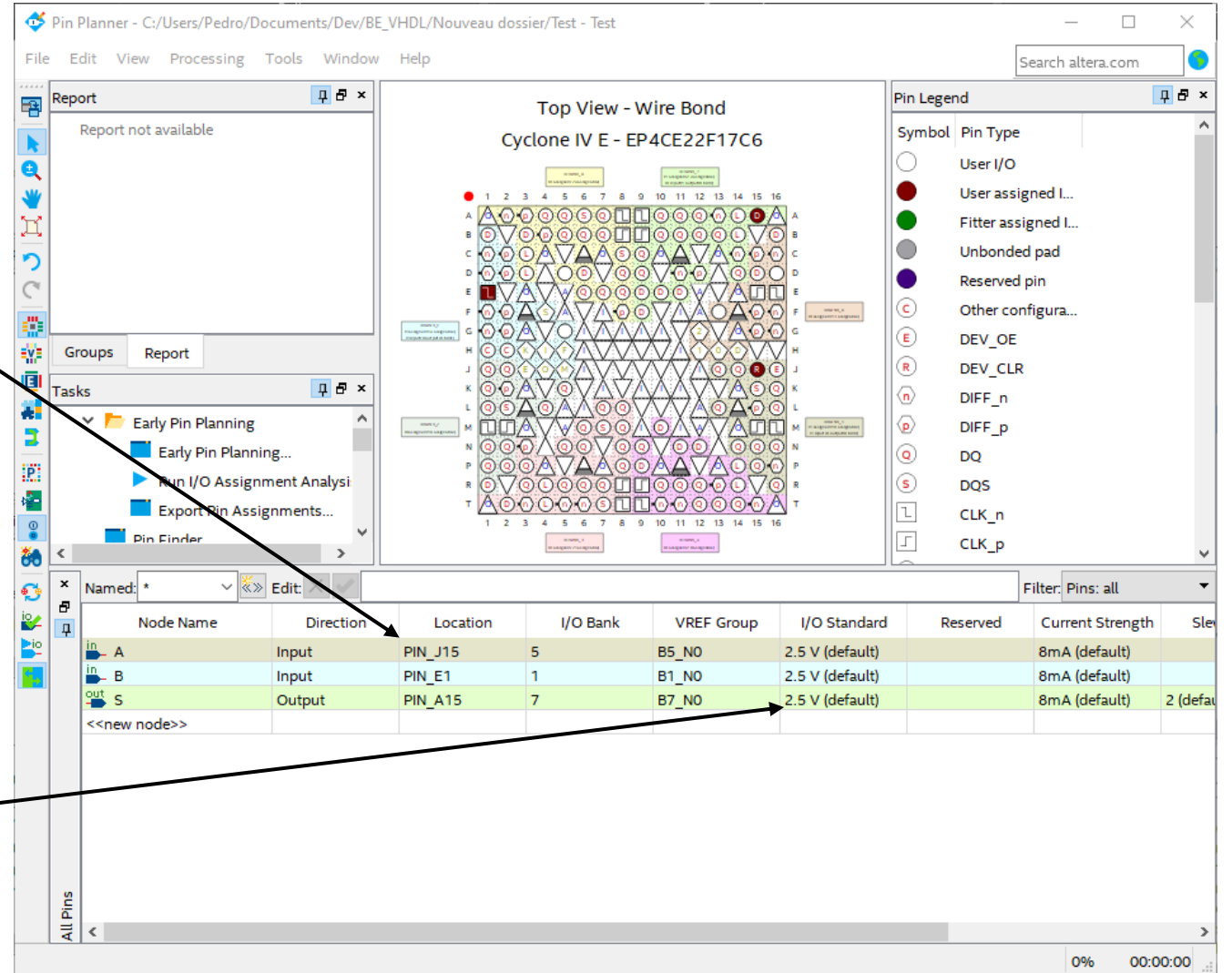
- Être sûr que la synthèse est faite « Analysis & Synthesis » avant d'affecter les broches
- Aller dans « Assignements → Pin Planner »



Affectation des broches

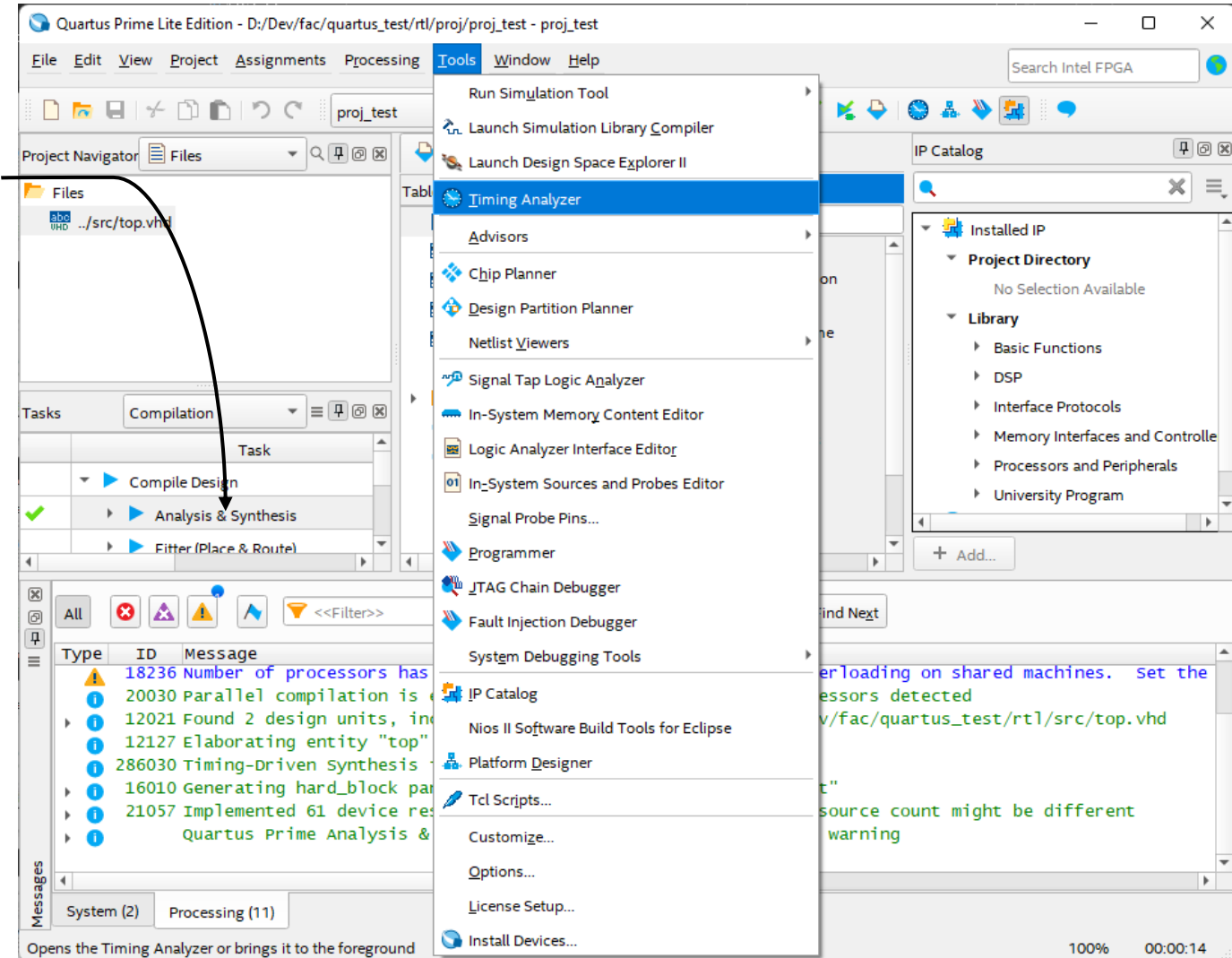
L'endroit où la broche sera affecté (voir le manuel d'utilisation de votre carte)

Configuration de la broche (LVCMOS33, LVCMOS25, LVDS...)



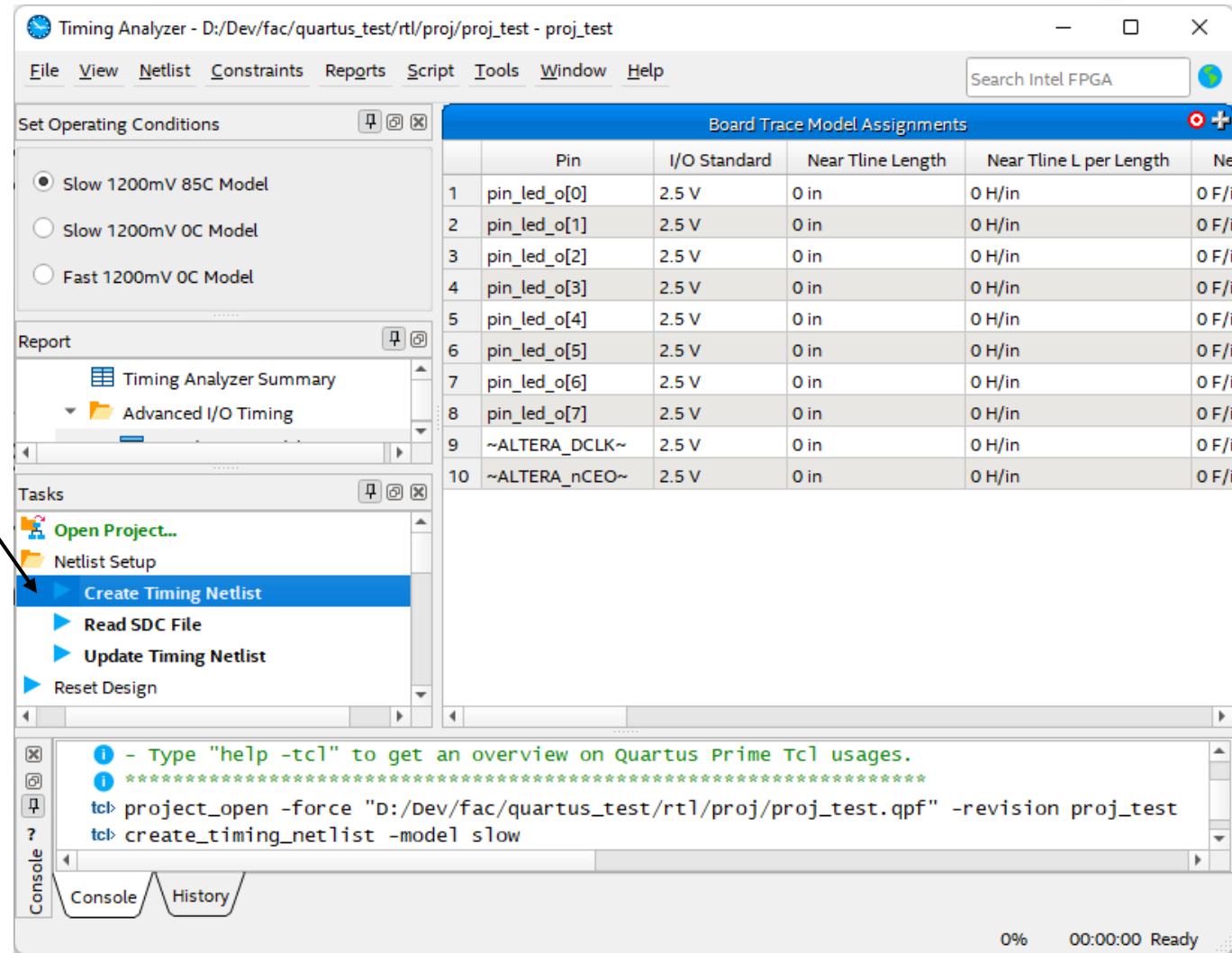
Ajout de contraintes de temps

- Être sûr que la synthèse est faite « Analysis & Synthesis » avant d'affecter les broches
- Aller dans « Tools → Pin Timing Analyzer »



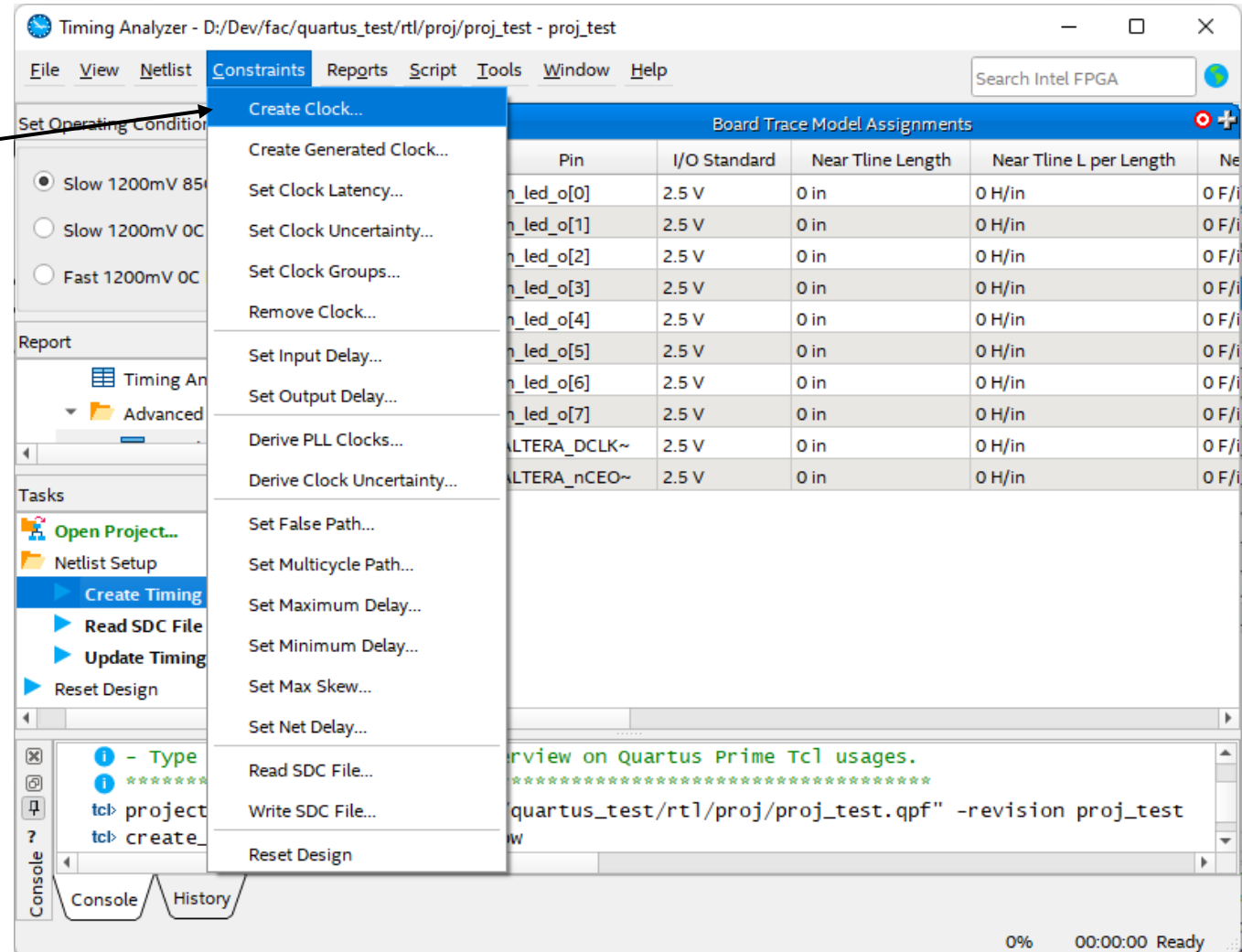
Ajout de contraintes de temps

Cliquer sur « Create Timing Netlist »



Ajout de contraintes de temps

Cliquer sur « Create Clock... »



Ajout de contraintes de temps

Clock name: pin_clk_i

Period: 20.000 ns

Waveform edges

Rising: ns

Falling: ns

Targets: ...

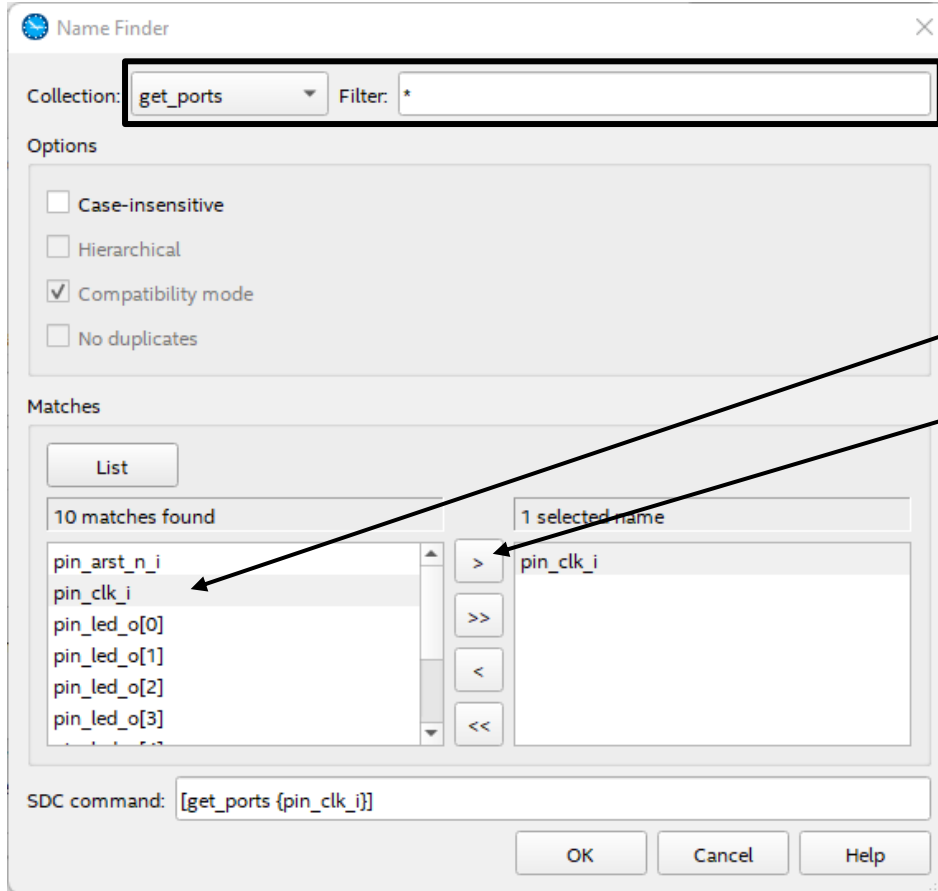
☐ Don't overwrite existing clocks on target nodes

SDC command: create_clock -name pin_clk_i -period 20.000

Run Cancel Help

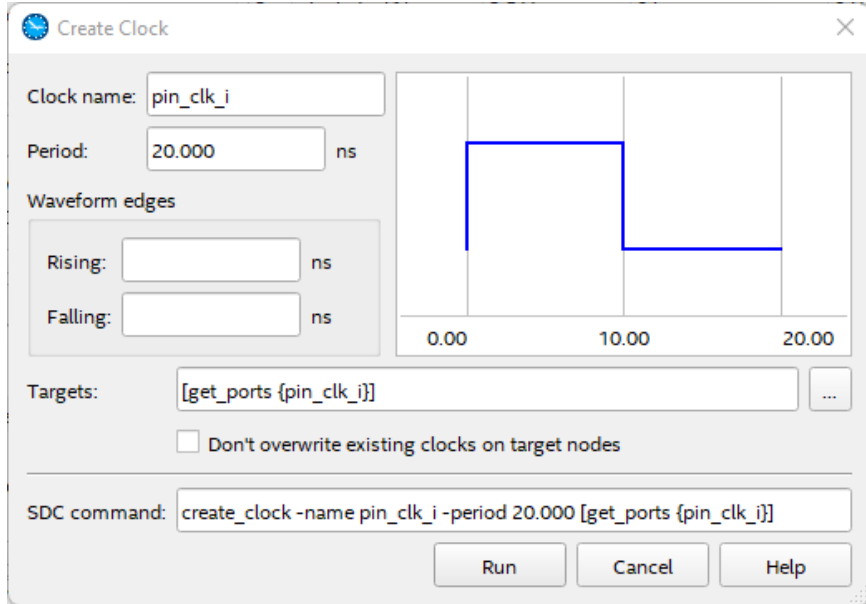
- Insérer un nom à l'horloge
- Insérer la période de l'horloge
- Cliquer ici pour choisir l'horloge dans le design

Ajout de contraintes de temps



- Choisir le bon filtre pour trouver l'horloge
- Sélectionner l'horloge
- Cliquer sur « > » pour l'ajouter
- Cliquer sur « OK »

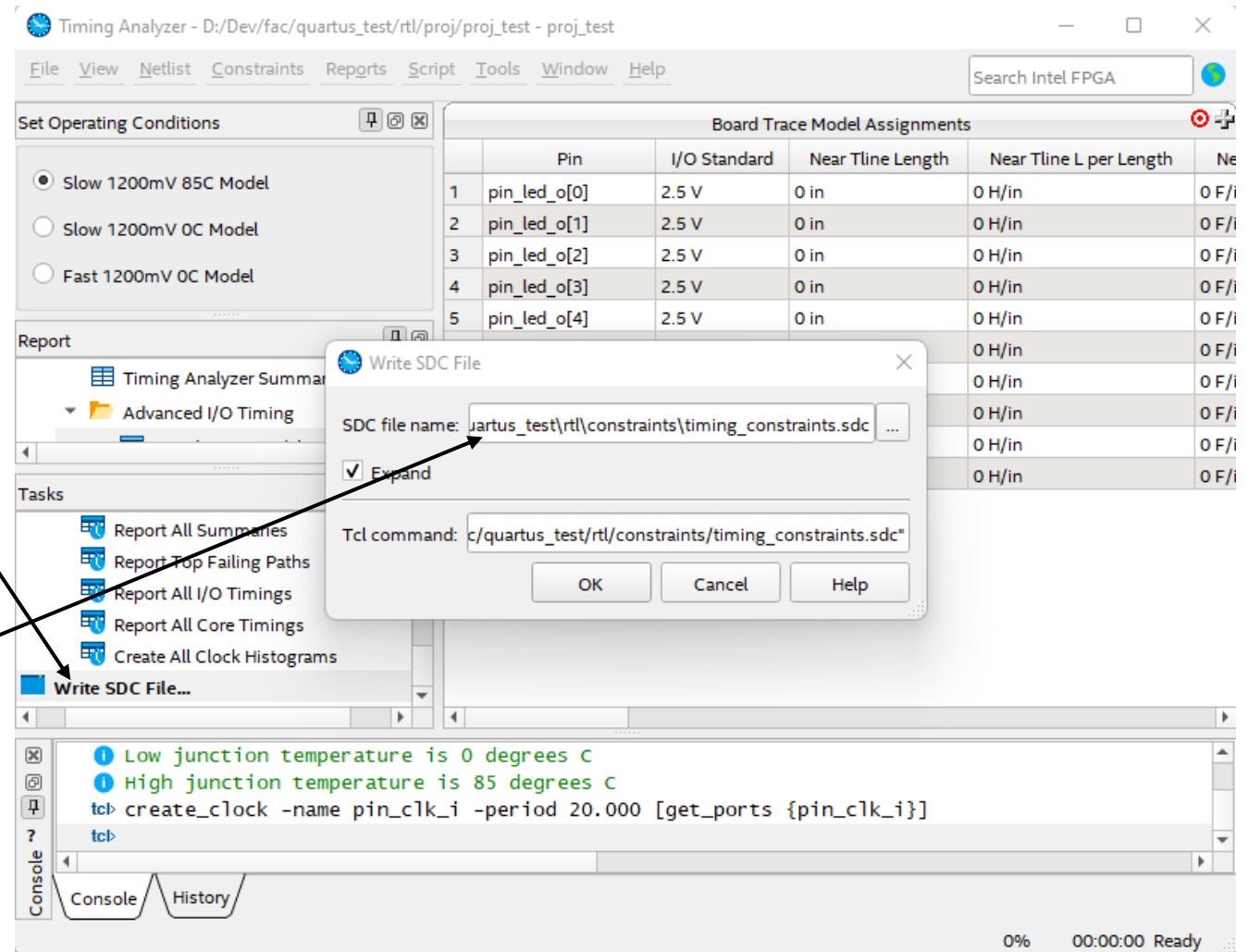
Ajout de contraintes de temps



- Cliquer sur « Run »

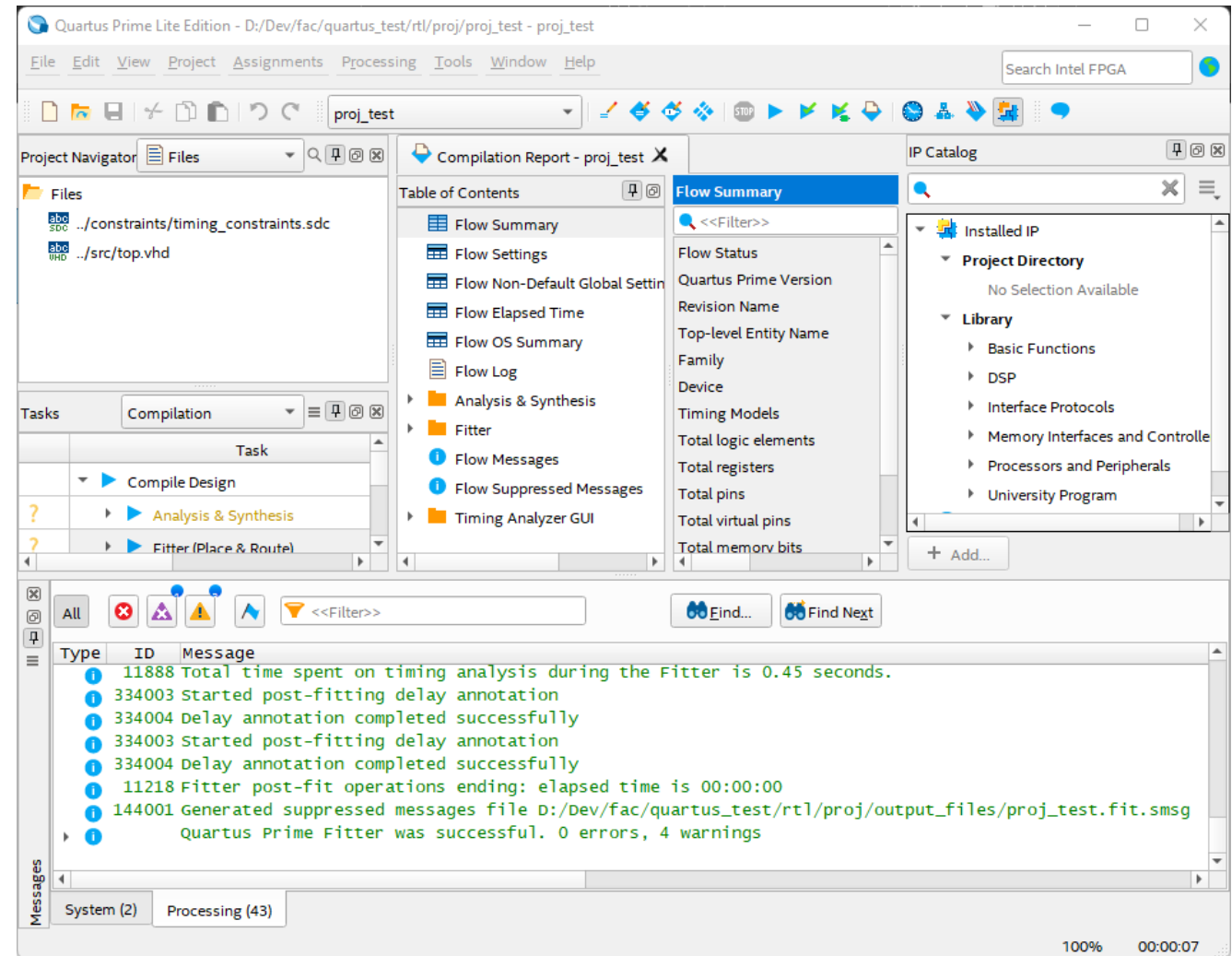
Ajout de contraintes de temps

- Cliquer sur « Write SDC File... » tout en bas
- Donner un nom au fichier de contraintes et l'endroit où sauvegarder le fichier de contraintes
- Cliquer sur « OK »
- Quitter « Timing Analyzer »



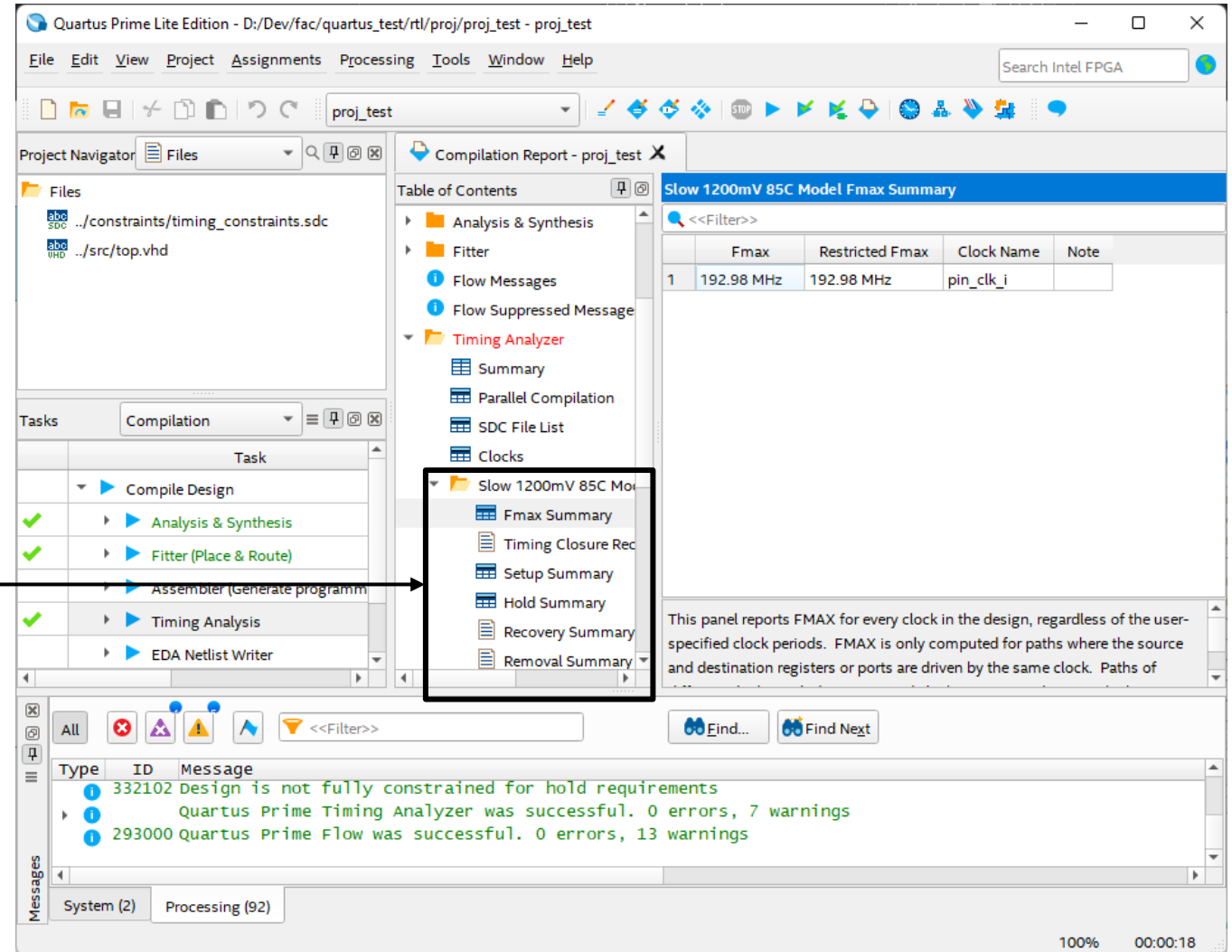
Ajout de contraintes de temps

- Ajouter le fichier de contrainte à votre projet en allant dans « Project → Add/Remove Files in Project... »



Ajout de contraintes de temps

- Cliquer sur « Timing Analysis » pour que le synthétiseur calcule la fréquence maximale du design
- Visualiser la fréquence maximale de votre design dans « Timing Analyser → Slow 1200mV 85C Model » ou « Timing Analyser → Slow 1200mV 0C Model »

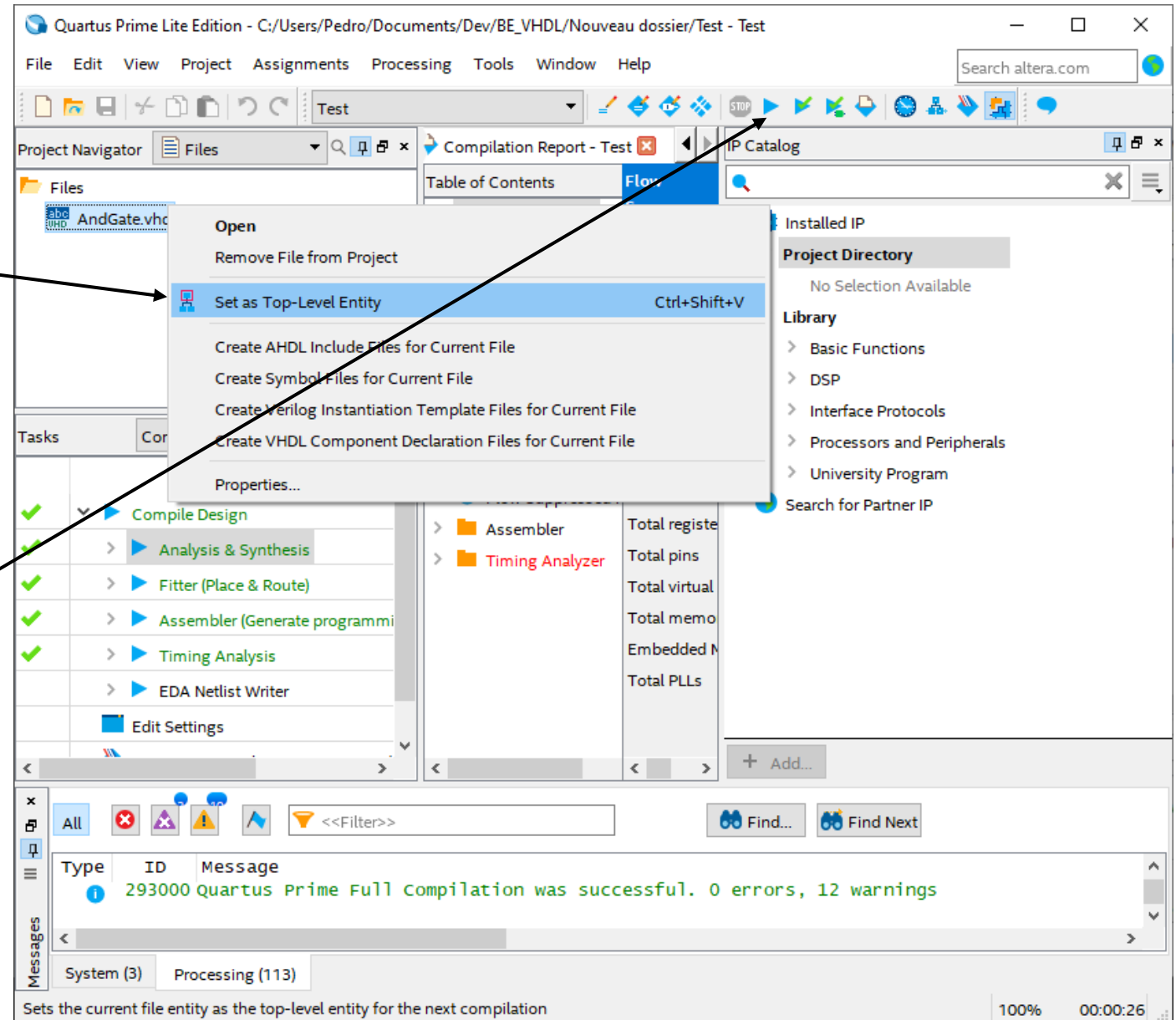


Compilation

Mettre d'abord en « Top Level » l'entité que vous voulez compiler

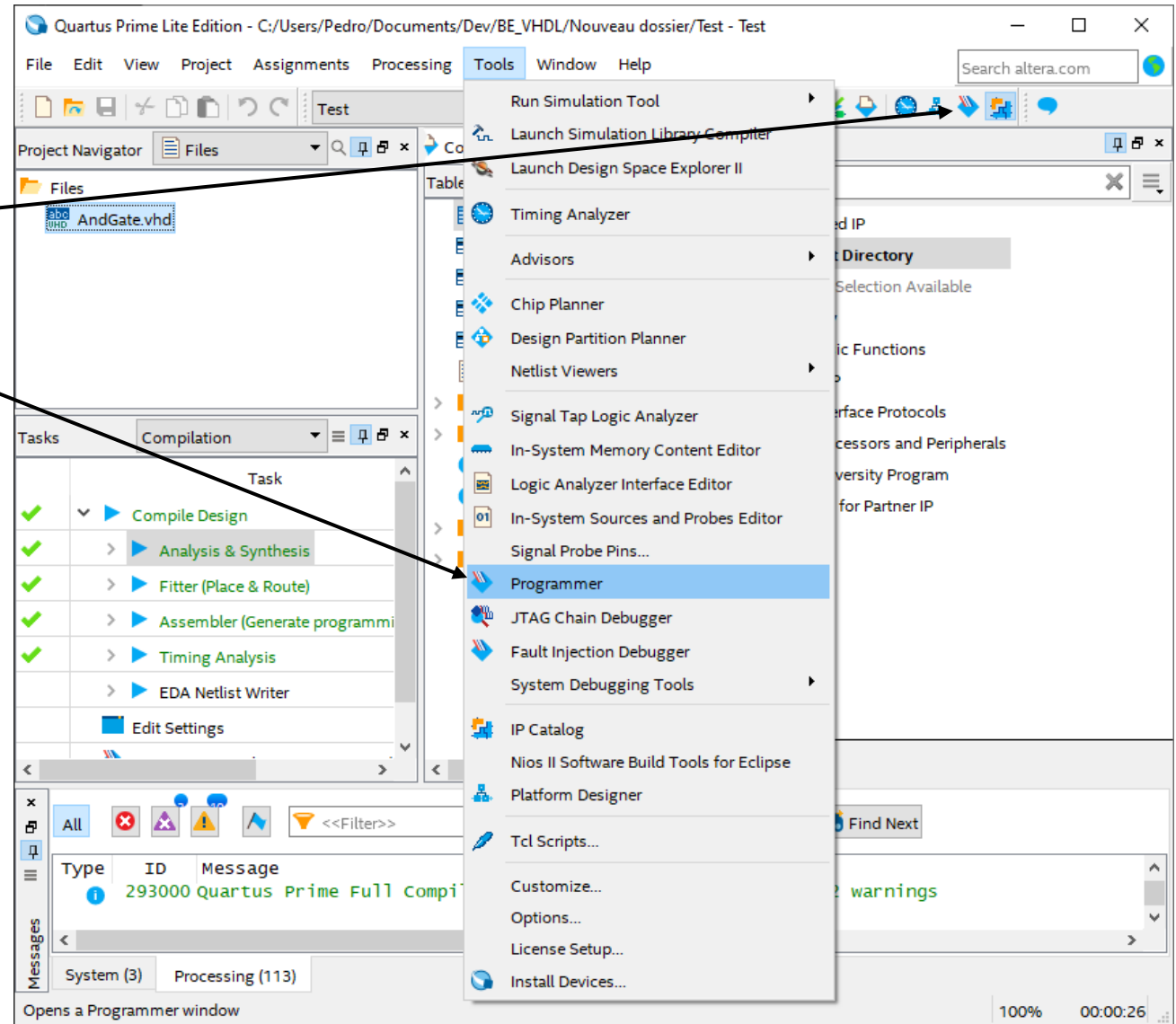
Compiler votre design. Il fera :

- La synthèse
- Le placement
- Le bitstream
- L'analyse temporelle



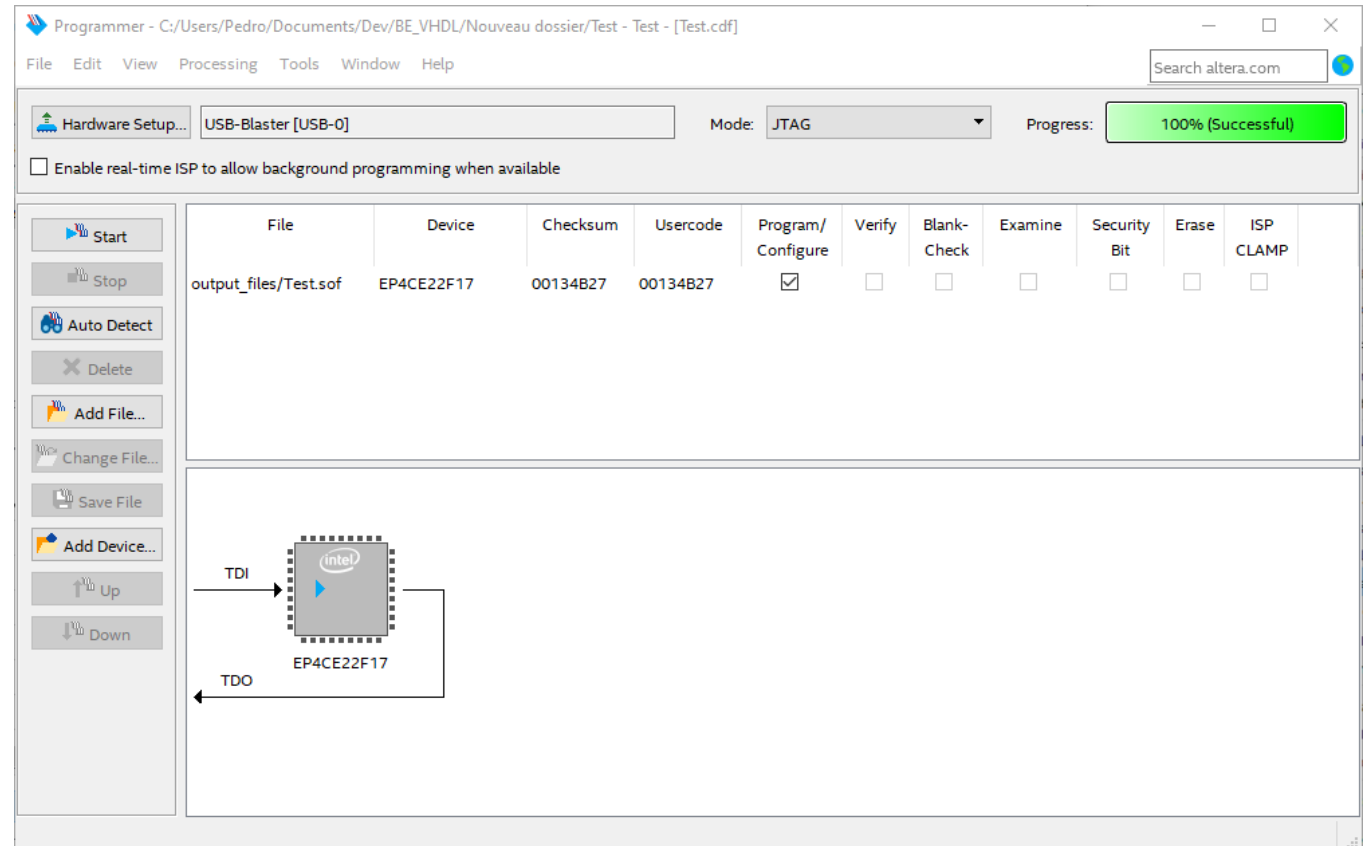
Programmation

Lancer le
programmeur ou



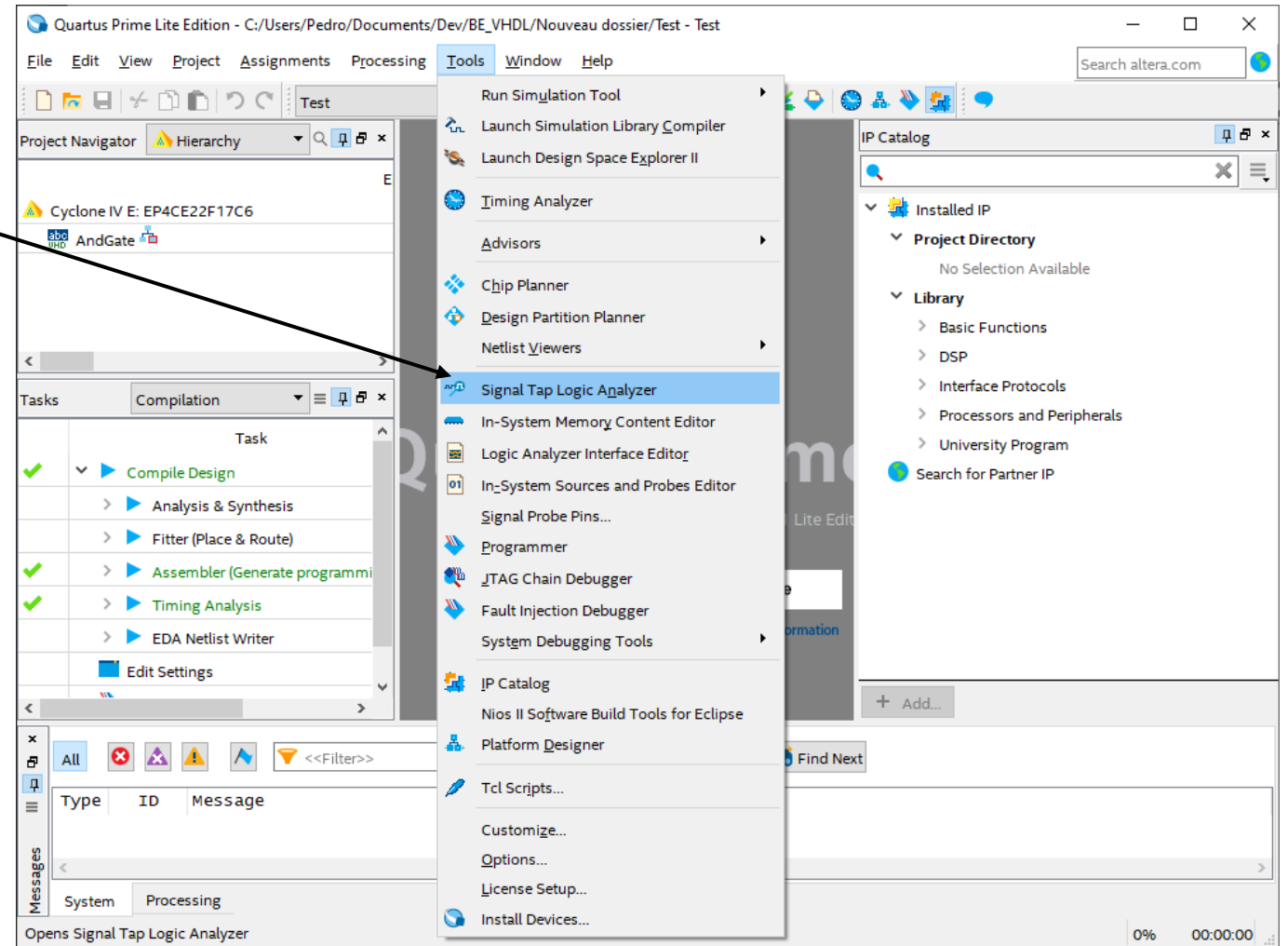
Programmation

- Par défaut le bon bitstream (.sof) sera sélectionné.
- Si ce n'est pas le cas, supprimer celui qui se trouve et ajouter le votre en cliquant sur « Add File... ».
- Cliquer sur « Start » pour programmer la carte.
- Si aucune carte est listé alors qu'elle est connecté cliquer sur « Hardware Setup » pour la sélectionner.



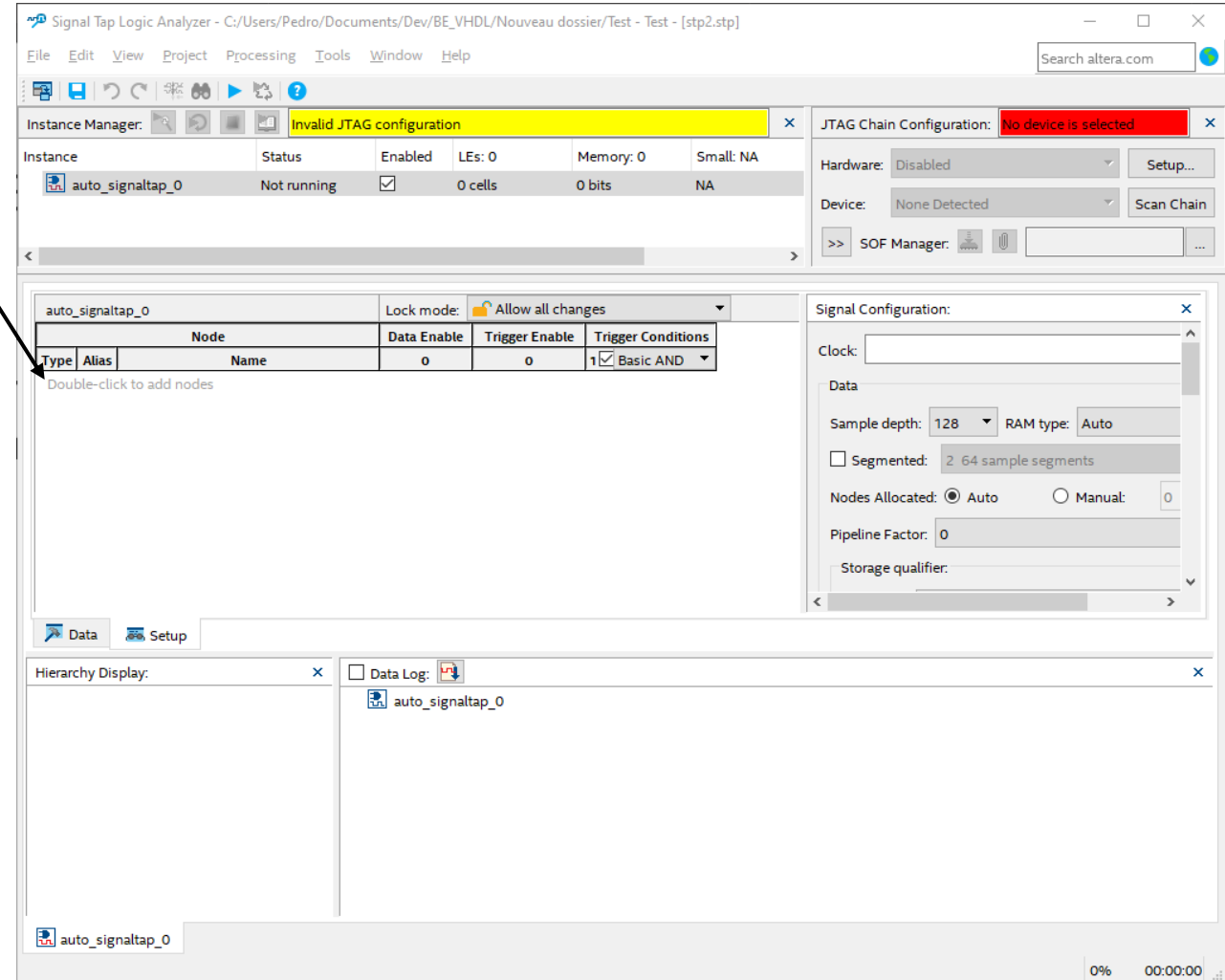
Analyseur logique (Signal Tap Logic Analyzer)

- Aller dans « Tools → Signal Tap Logic Analyzer »



■ Analyseur logique (Signal Tap Logic Analyzer)

- Double clic sur « Double-Click to add nodes » pour ajouter les signaux que vous voulez visualiser



■ Analyseur logique (Signal Tap Logic Analyzer)

- Dans « Signal Configuration »:
- Choisir l'horloge d'échantillonnage de l'analyseur logique
- La profondeur de l'acquisition (utilise des ressources de mémoire embarqué du FPGA)
- Laisser la configuration par défaut

Signal Configuration:

Clock: ...

Data

Sample depth: 128 RAM type: Auto

☐ Segmented: 2 64 sample segments

Nodes Allocated: ☒ Auto ☐ Manual: 0

Pipeline Factor: 0

Storage qualifier:

Type: Continuous

Input port: ...

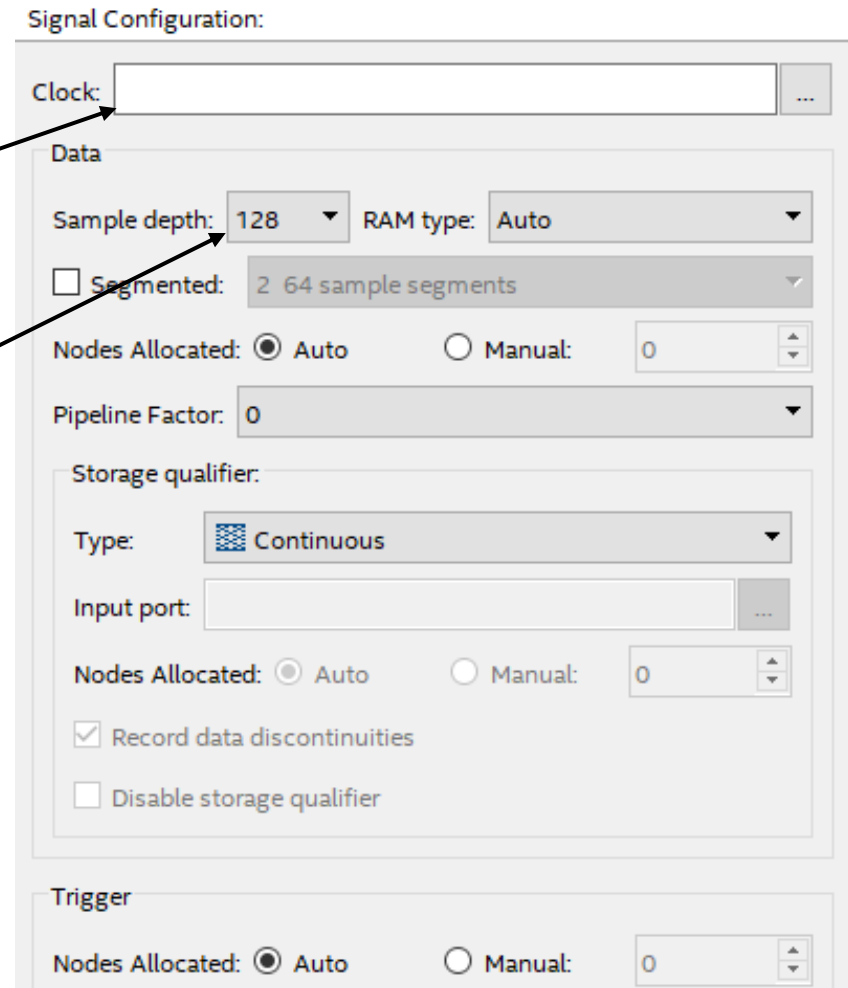
Nodes Allocated: ☒ Auto ☐ Manual: 0

☒ Record data discontinuities

☐ Disable storage qualifier

Trigger

Nodes Allocated: ☒ Auto ☐ Manual: 0

The image shows a screenshot of the 'Signal Configuration' dialog box. Two arrows originate from the text on the left. One arrow points to the 'Clock' field at the top of the dialog. The other arrow points to the 'Sample depth' dropdown menu, which is currently set to '128'.

■ Analyseur logique (Signal Tap Logic Analyzer)

- La position du trigger lorsque la condition d'acquisition est vraie
- Le nombre de trigger
- Laisser la configuration par défaut

Signal Configuration:

Trigger flow control: Sequential

Trigger position: Pre trigger position

Trigger conditions: 1

☐ Trigger in

☐ Pin:

☒ Node:

☐ Instance:

☐ Hard Processor System (HPS) trigger out

Pattern: 1 High

☐ Trigger out

☐ Pin:

☐ Instance:

☐ Hard Processor System (HPS) trigger in

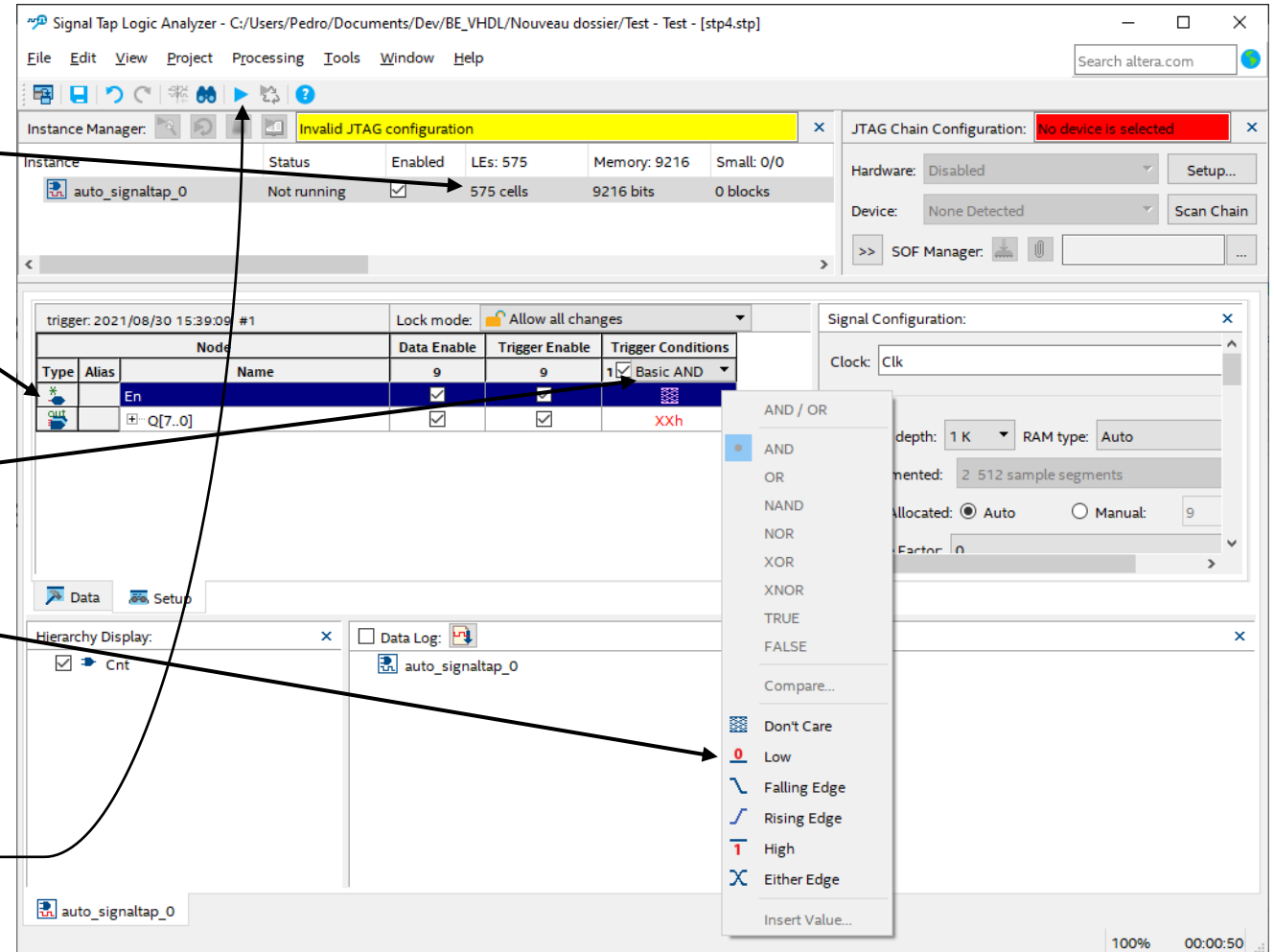
☐ Hard Processor System (HPS) event: 0

Level: Active High

Latency delay:

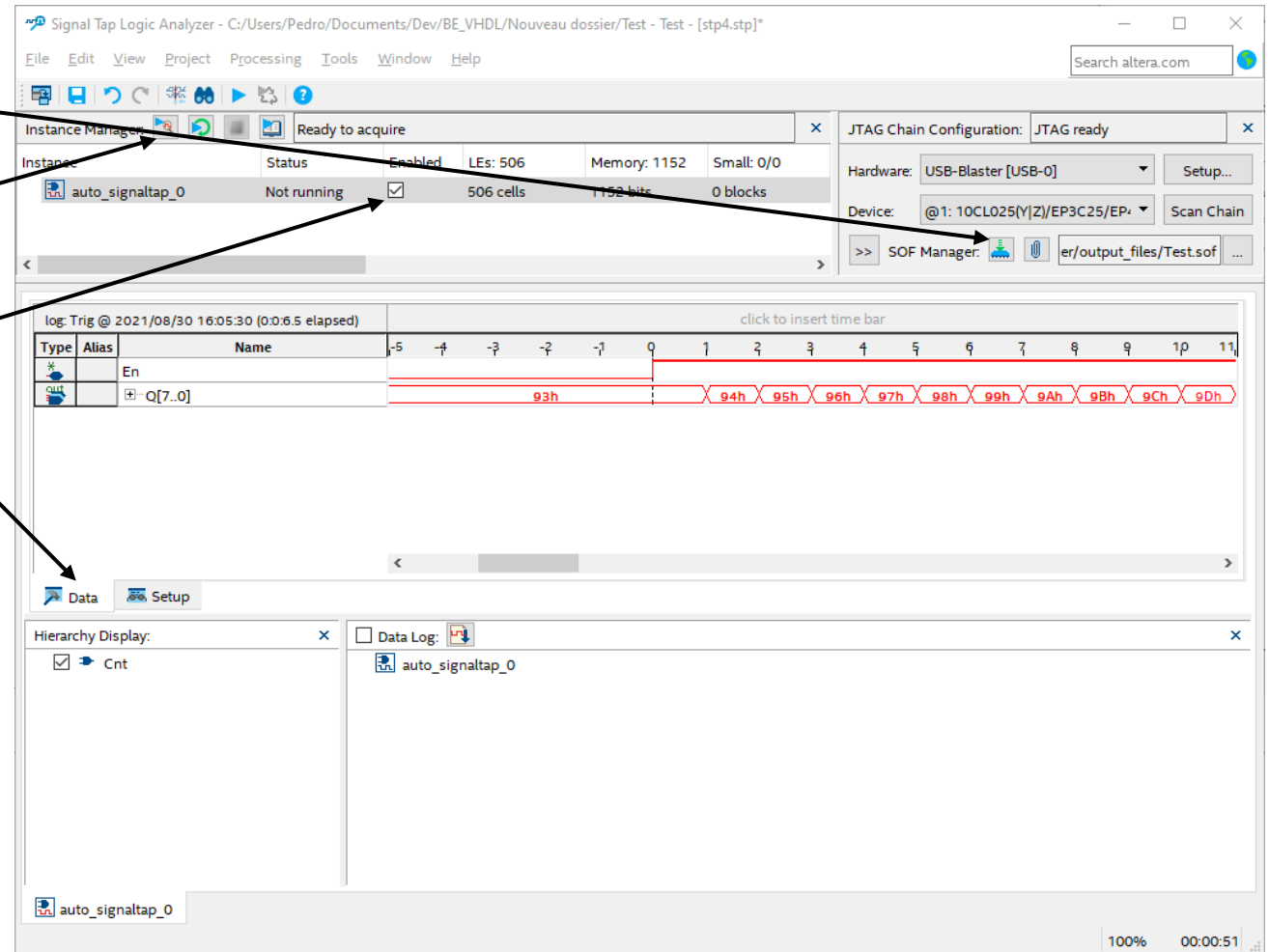
Analyseur logique (Signal Tap Logic Analyzer)

- Résumé du nombre de cellules et de mémoire que votre configuration de l'analyseur logique utilise
- Les signaux que vous avez décidé de visualiser
- Les conditions de trigger (AND, OR, ...) entre tous les signaux.
- Plusieurs types de conditions de trigger.
- La condition « Don't Care » désactive le signal du trigger
- Vous pouvez lancer la compilation ici



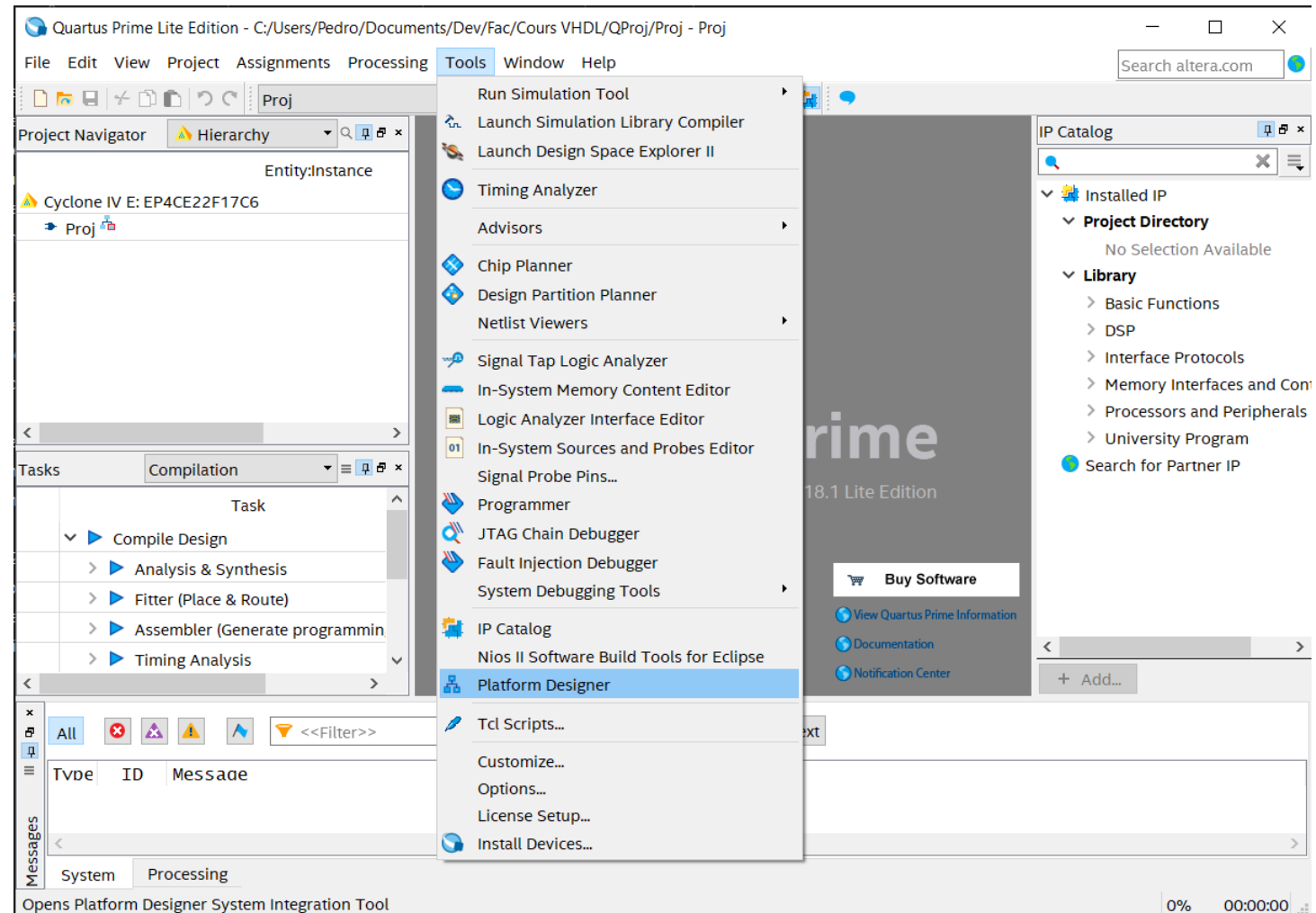
Analyseur logique (Signal Tap Logic Analyzer)

- Programmer votre carte
- Aller dans « Data »
- Lancer le trigger
- Désactive / active l'analyseur



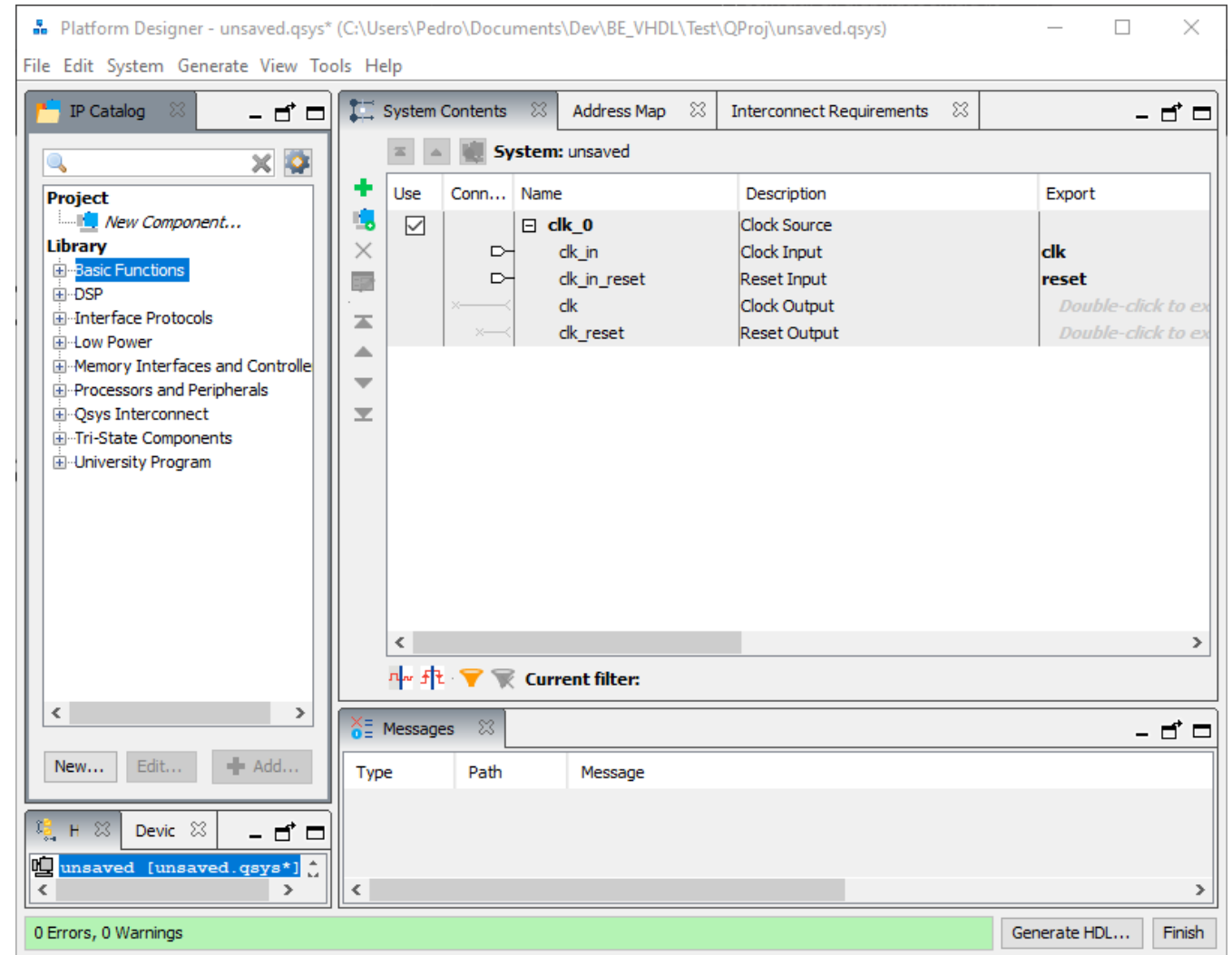
Platform Designer

- Aller dans « Tools → Designer »



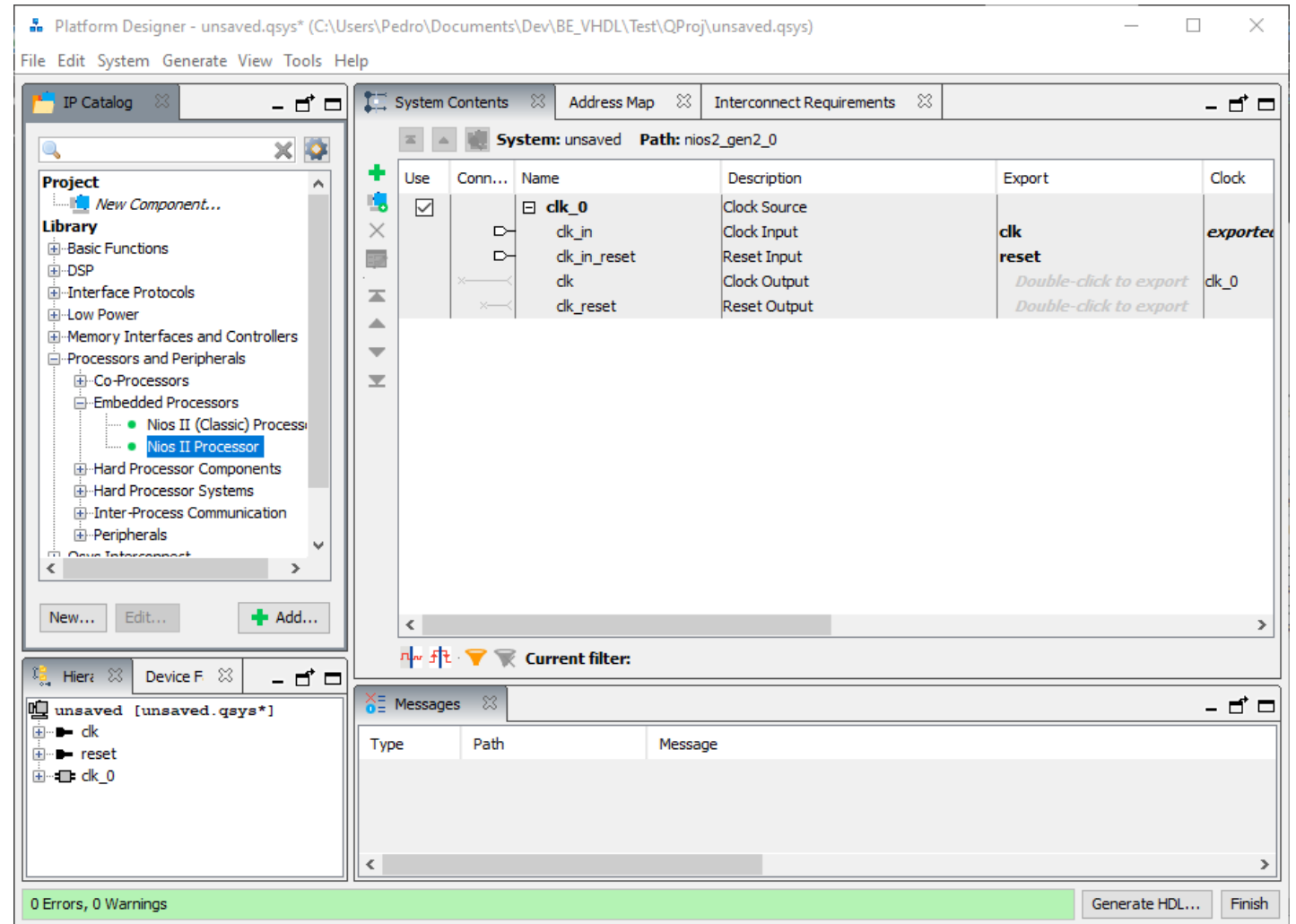
Platform Designer

- Platform Designer permet de construire un système complexe grâce à un GUI sans avoir à écrire du code
- Dans cet exemple on va apprendre comment construire un microcontrôleur basé sur un cœur Nios



Platform Designer (Nios)

- Ajouter le processeur Nios en double cliquant dans « Processors and Peripherals → Embedded Processors → Nios II (Classic) Processor → Nios II Processor »



Platform Designer (Nios)

- Dans Nios II Core, choisir un des deux.
- Nios II/e est une implémentation basique du Nios. Utile pour les FPGA qui ont pas beaucoup d'éléments logiques. Cette configuration est gratuite.
- Nios II/f est une implémentation plus performante du Nios. Utilise plus d'éléments logiques. Cette configuration est payante.
- Cliquer sur « Finish ».

Nios II Processor - nios2_gen2_0

Nios II Processor
altera_nios2_gen2

Block Diagram

Cette présentation a été enregistrée pour la dernière fois le : À l'instant

Show signals

Block Diagram showing connections for nios2_gen2_0:

- clk: clock
- reset: reset
- irq: interrupt
- debug_mem_slave: avalon
- nios2_gen2_0: nios2_gen2_0
- data_maste: data_maste
- instruction_maste: instruction_maste
- debug_reset_reques: debug_reset_reques
- custom_instruction_maste: custom_instruction_maste
- altera_nios2_gen2: altera_nios2_gen2

Select an Implementation

Nios II Core: ☒ Nios II/e ☐ Nios II/f

	Nios II/e	Nios II/f
Summary	Resource-optimized 32-bit RISC	Performance-optimized 32-bit RISC
Features	JTAG Debug ECC RAM Protection	JTAG Debug Hardware Multiply/Divide Instruction/Data Caches Tightly-Coupled Masters ECC RAM Protection External Interrupt Controller Shadow Register Sets MPU MMU
RAM Usage	2 + Options	2 + Options

Error: nios2_gen2_0: Reset slave is not specified. Please select the reset slave

Error: nios2_gen2_0: Exception slave is not specified. Please select the exception slave

Cancel Finish

Platform Designer (Nios)

- Ajouter la mémoire RAM en double cliquant dans « Basic Functions → On Chip Memory → On-Chip Memory (RAM or ROM) Intel FPGA IP »

The screenshot displays the Platform Designer (Nios) interface. The main window is titled "Platform Designer - unsaved.qsys* (C:\Users\Pedro\Documents\Dev\BE_VHDL\Test\QProj\unsaved.qsys)". The interface is divided into several panels:

- IP Catalog:** Shows a tree view of components. Under "On-Chip Memory", the "On-Chip Memory (RAM or ROM) Intel FPGA IP" is highlighted.
- System Contents:** Displays the system configuration. The "nios2_gen2_0" component is selected. The table below shows the connections and descriptions for this component.
- Messages:** Lists 4 errors related to the "nios2_gen2_0" component.

Use	Connections	Name	Description	Export
<input checked="" type="checkbox"/>		clk_0	Clock Source	
		clk_in	Clock Input	
		clk_in_reset	Reset Input	
		clk	Clock Output	
		clk_reset	Reset Output	
<input checked="" type="checkbox"/>		nios2_gen2_0	Nios II Processor	
		clk	Clock Input	Double-click to export
		reset	Reset Input	Double-click to export
		data_master	Avalon Memory Mapped Master	Double-click to export
		instruction_master	Avalon Memory Mapped Master	Double-click to export
		irq	Interrupt Receiver	Double-click to export
		debug_reset_request	Reset Output	Double-click to export
		debug_mem_slave	Avalon Memory Mapped Slave	Double-click to export
		custom_instruction_m...	Custom Instruction Master	Double-click to export

Messages:

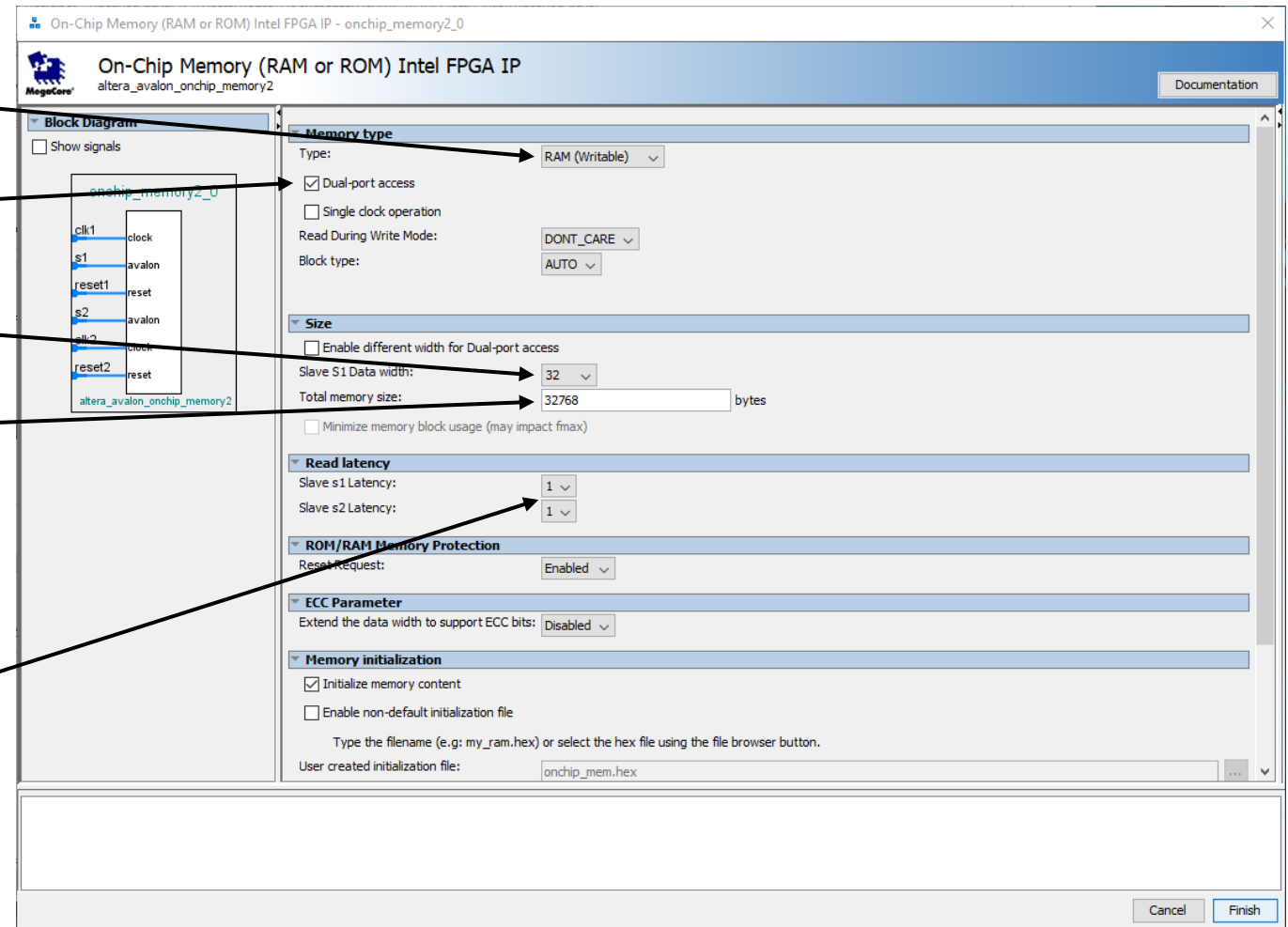
Type	Path	Message
4 Errors		
✗	unsaved.nios2_gen2_0	Reset slave is not specified. Please select the reset slave
✗	unsaved.nios2_gen2_0	Exception slave is not specified. Please select the exception slave
✗	unsaved.nios2_gen2_0	nios2_gen2_0.clk must be connected to a clock output
✗	unsaved.nios2_gen2_0	nios2_gen2_0.reset must be connected to a reset source

4 Errors, 0 Warnings

Generate HDL... Finish

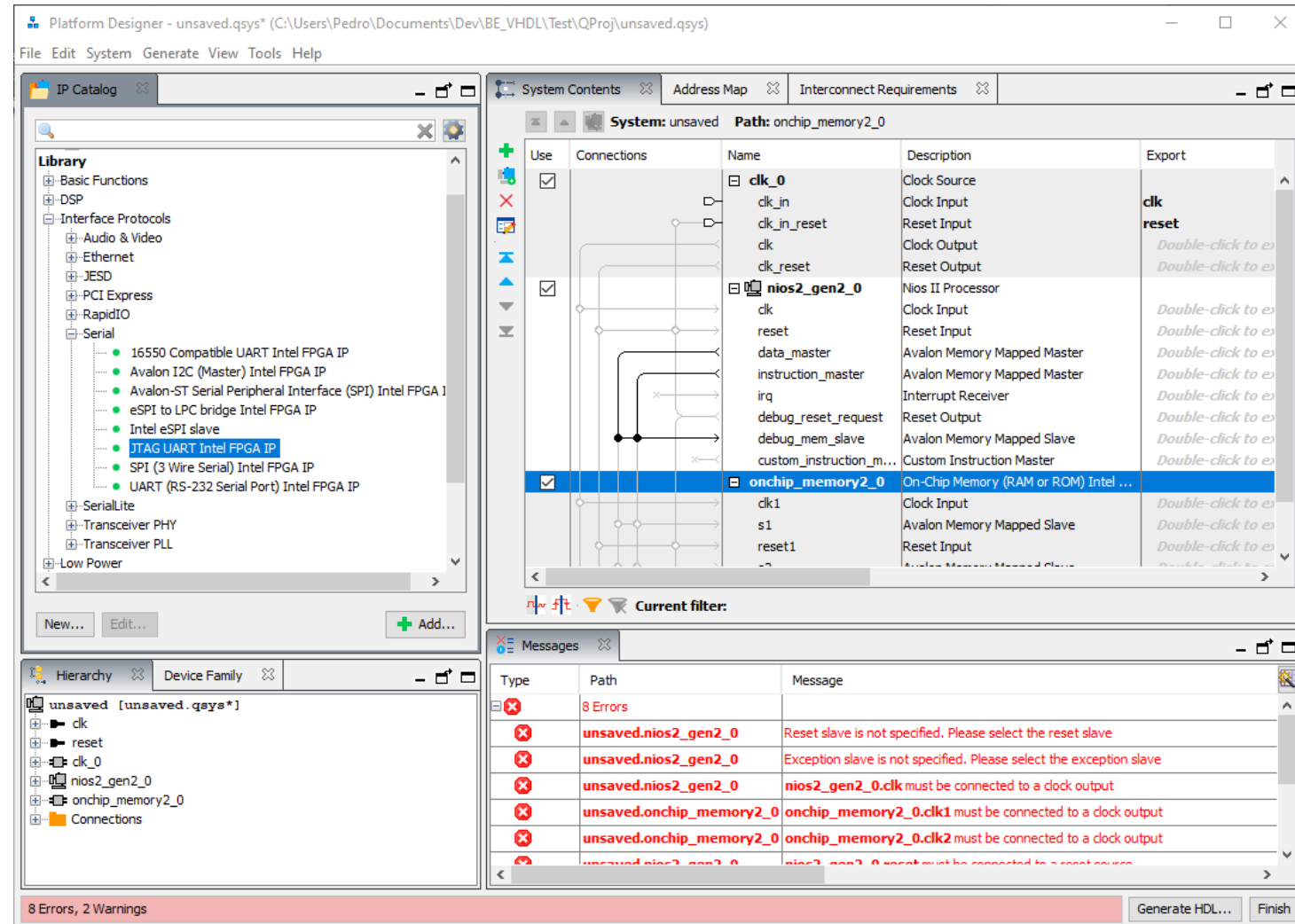
Platform Designer (Nios)

- Type : RAM
- Dual – port access (optionnel)
- Slave S1 Data width : 32
- Total memory size : 32768 bytes
 - Le EP4CE22F17C6 contient 608256 bits → 76032 bytes
- Read latency : 1
- Cliquer sur « Finish »



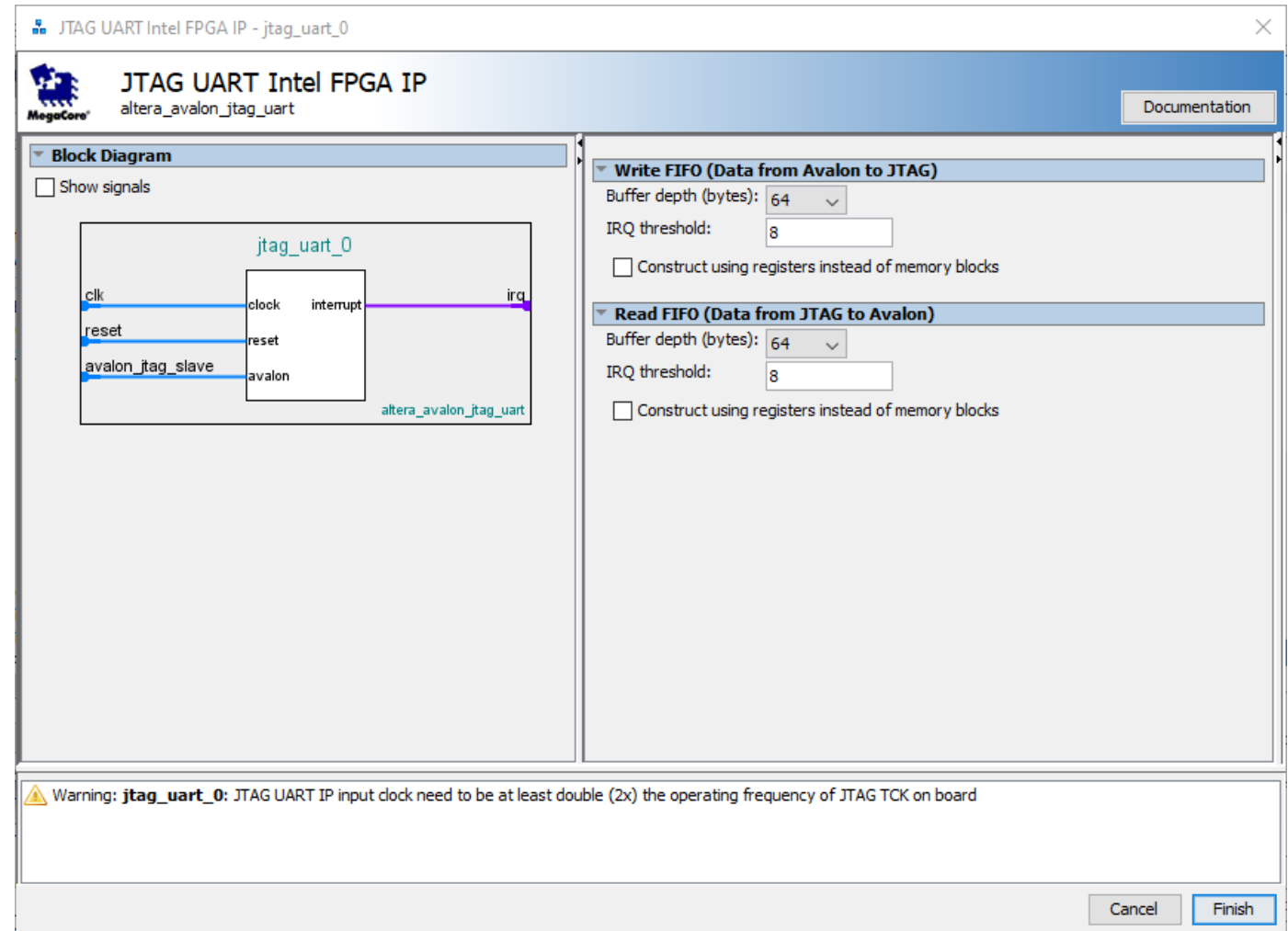
Platform Designer (Nios)

- Ajouter un JTAG UART en double cliquant dans « Interface Protocols → Serial → JTAG UART Intel FPGA IP »
- Cela permet d'ajouter :
 - UART (Interface série)
 - JTAG (Utiliser le debugger)



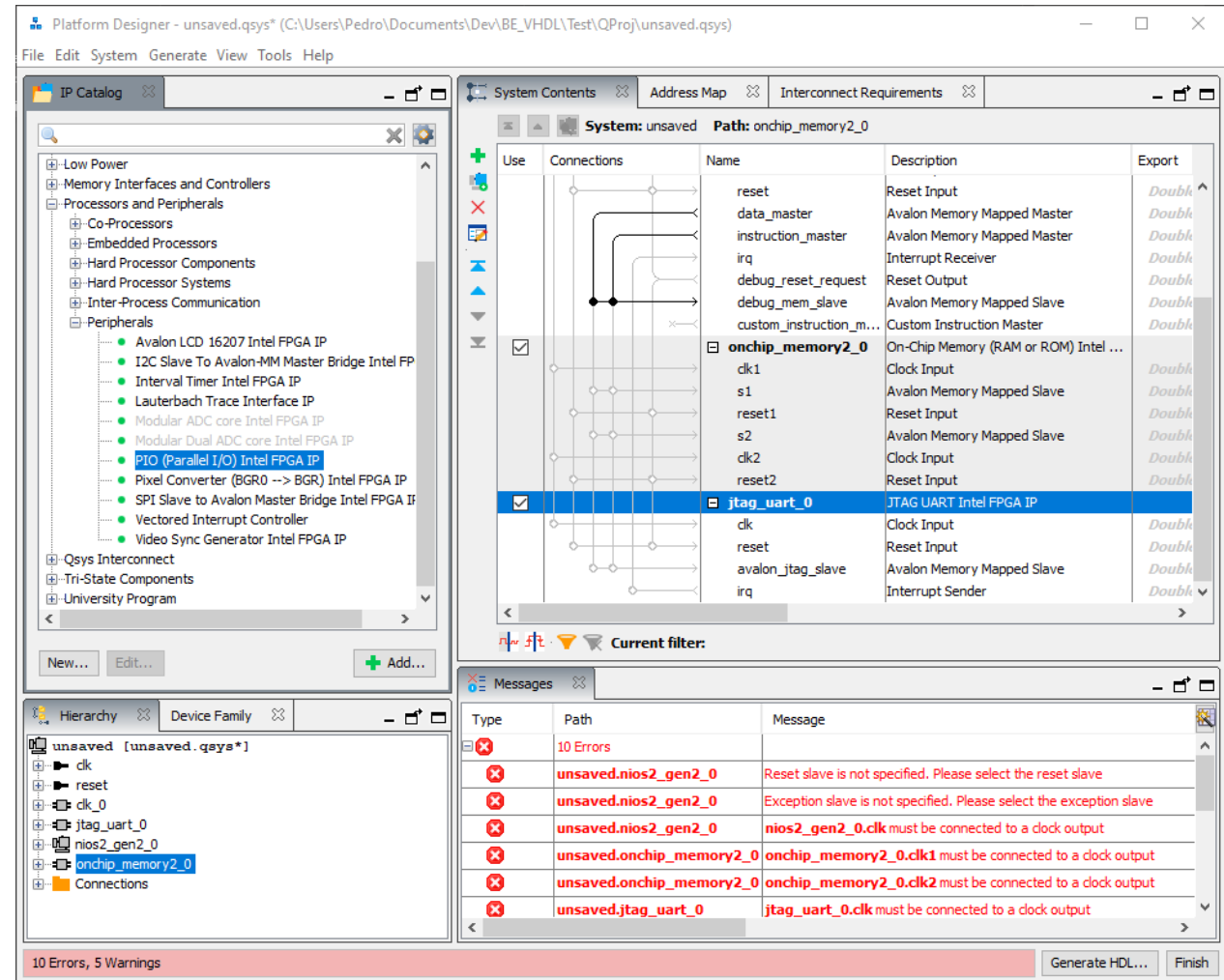
Platform Designer (Nios)

- Cliquer sur « Finish »



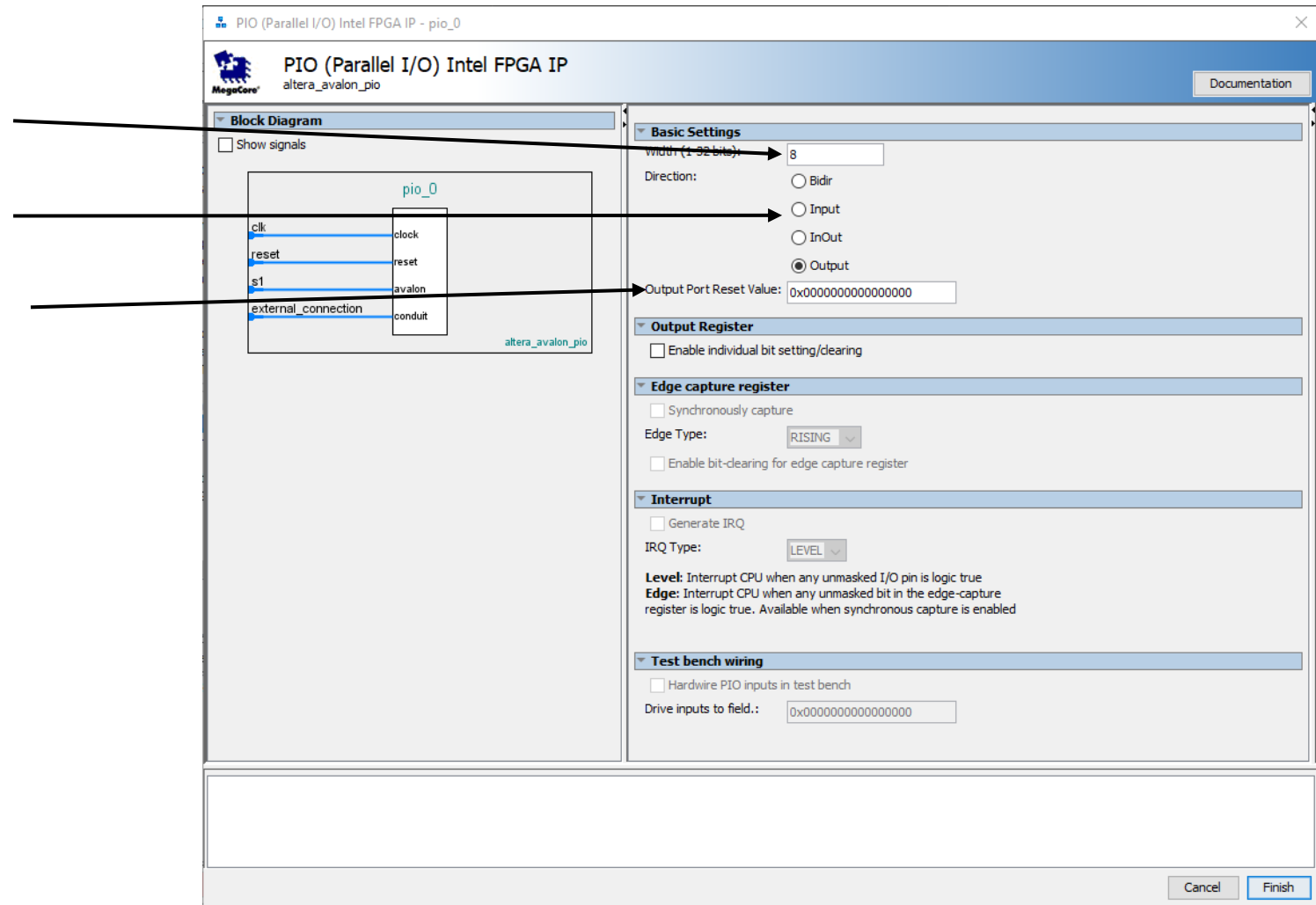
Platform Designer (Nios)(Optionnel)

- Ajouter des PIO (Parallel Input/Output) en double cliquant dans « Processors and Peripherals → Peripherals → PIO (Parallel I/O) Intel FPGA IP
- Cela permet d'ajouter des simples entrées/sorties pour connecter des boutons, leds, etc...



Platform Designer (Nios)(Optionnel)

- Le nombre d'IO
- Configuration des IO
- Valeur par défaut lors du Reset



Platform Designer (Nios)

- Faire la connexion des composants en cliquant sur les boules.
- Adresse attribué aux composant
- Numéro de l'interruption
- Nom personnalisé du composant
- Double cliquer ici pour exporter les I/O dans l'entité.

Use	Connections	Name	Description	Export	Clock	Base	End	IRQ
<input checked="" type="checkbox"/>		Clk	Clock Source					
		clk_in	Clock Input	clk	exported			
		clk_in_reset	Reset Input	reset				
		clk	Clock Output	Double-click	Clk			
		clk_reset	Reset Output	Double-click				
<input checked="" type="checkbox"/>		CPU	Nios II Processor					
		clk	Clock Input	Double-click	Clk			
		reset	Reset Input	Double-click	[clk]			
		data_master	Avalon Memory Mapped Master	Double-click	[clk]			
		instruction_master	Avalon Memory Mapped Master	Double-click	[clk]			
		irq	Interrupt Receiver	Double-click	[clk]			IRQ 0 IRQ 31
		debug_reset_request	Reset Output	Double-click	[clk]			
		debug_mem_slave	Avalon Memory Mapped Slave	Double-click	[clk]	0x0800	0x0fff	
		custom_instruction_master	Custom Instruction Master	Double-click				
<input checked="" type="checkbox"/>		RAM	On-Chip Memory (RAM or RO...					
		clk1	Clock Input	Double-click	Clk			
		s1	Avalon Memory Mapped Slave	Double-click	[clk1]	0x0000	0x7fff	
		reset1	Reset Input	Double-click	[clk1]			
		s2	Avalon Memory Mapped Slave	Double-click	[clk2]	0x0000	0x7fff	
		clk2	Clock Input	Double-click	Clk			
		reset2	Reset Input	Double-click	[clk2]			
<input checked="" type="checkbox"/>		Jtag	JTAG UART Intel FPGA IP					
		clk	Clock Input	Double-click	Clk			
		reset	Reset Input	Double-click	[clk]			
		avalon_jtag_slave	Avalon Memory Mapped Slave	Double-click	[clk]	0x0000	0x0007	
		irq	Interrupt Sender	Double-click	[clk]			
<input checked="" type="checkbox"/>		Leds	PIO (Parallel I/O) Intel FPGA IP					
		clk	Clock Input	Double-click	Clk			
		reset	Reset Input	Double-click	[clk]			
		s1	Avalon Memory Mapped Slave	Double-click	[clk]	0x0000	0x000f	
		external_connection	Conduit	leds				

Platform Designer (Nios)

- Doubler cliquer sur le composant « CPU/Nios II Processor » ou clique droit sur « CPU/Nios II Processor → Edit... »

The screenshot displays the Platform Designer (Nios) interface. On the left, a list of components is shown with their connections. The 'CPU' component is highlighted, and a context menu is open over it, showing options like 'Edit...', 'Add...', 'Rename', 'Duplicate', 'Remove', 'Details', 'Move Up', 'Drill into subsystem', 'Add new subsystem to current system', 'Push down...', 'Pull up', 'Set Design Environment', 'Show Arbitration Shares', 'Allow Connection Editing', 'Lock Base Address', and 'Expand All'.

Use	Connections	Name	Description	Export	Clock	Base	End	IRQ
<input checked="" type="checkbox"/>		Clk	Clock Source					
		clk_in	Clock Input	clk				
		clk_in_reset	Reset Input	reset	exported			
		clk	Clock Output		Clk			
		clk_reset	Reset Output					
<input checked="" type="checkbox"/>		CPU	Nios II Processor					
		clk						
		reset						
		data_master						
		instruction_master						
		irq						
		debug_reset_request						
		debug_mem_slave						
		custom_instruction_mas						
<input checked="" type="checkbox"/>		RAM						
		clk1						
		s1						
		reset1						
		s2						
		clk2						
		reset2						
<input checked="" type="checkbox"/>		Jtag						
		clk						
		reset						
		avalon_jtag_slave						
		irq						
<input checked="" type="checkbox"/>		Leds						
		clk						
		reset						
		s1						
		external_connection						

Context Menu for CPU/Nios II Processor:

- Connections
- Filter
- Edit... (Ctrl+E)
- Add...
- Rename (Ctrl+R)
- Duplicate (Ctrl+D)
- Remove
- Details
- Move Up (Ctrl+Maj+U)
- Drill into subsystem (Ctrl+Maj+D)
- Add new subsystem to current system (Ctrl+Maj+N)
- Push down...
- Pull up
- Set Design Environment
- Show Arbitration Shares
- Allow Connection Editing
- Lock Base Address
- Expand All

IRQ 0 IRQ 31

0x0800 0x0fff

0x0000 0x7fff

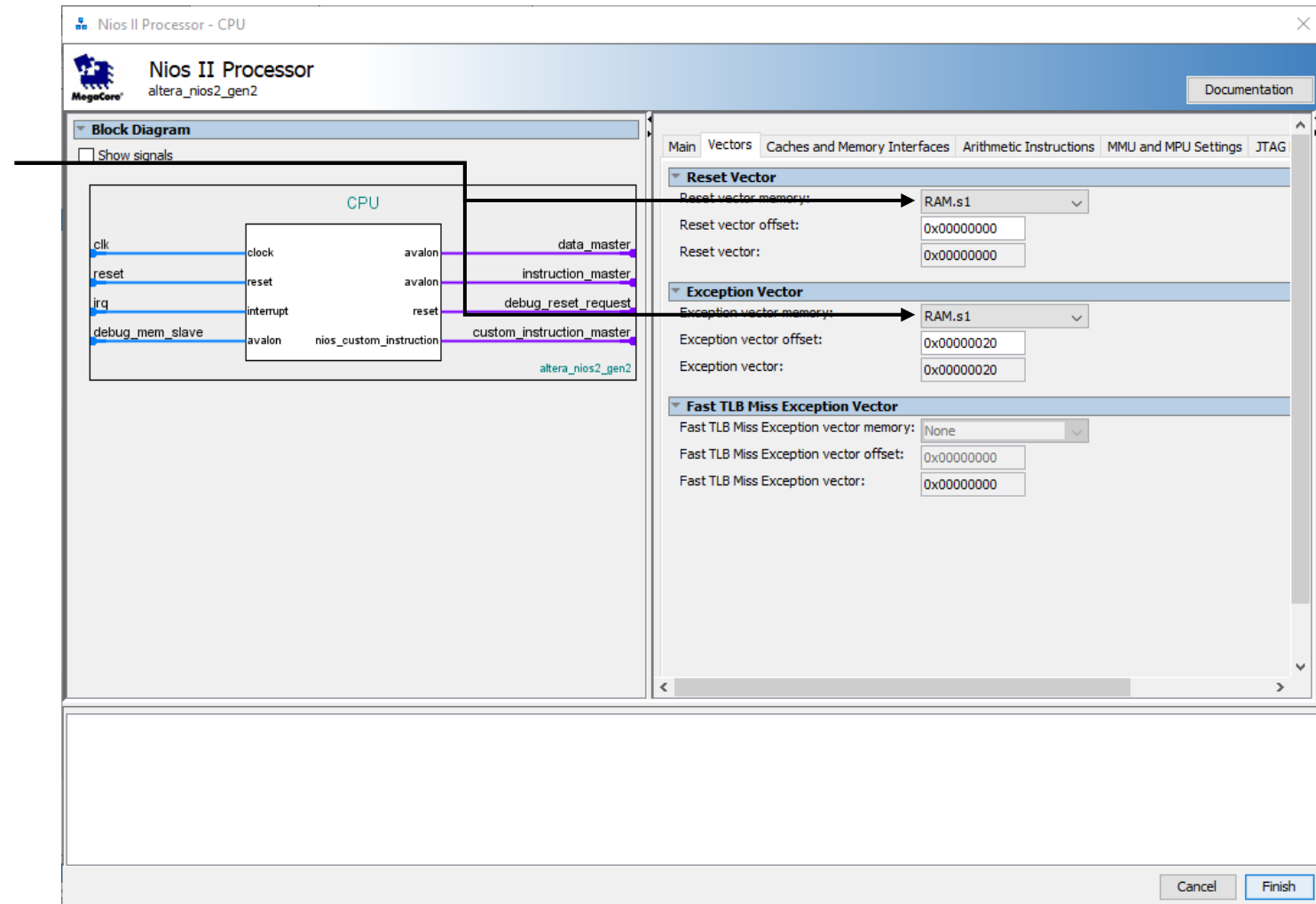
0x0000 0x7fff

0x0000 0x0007

0x0000 0x000f

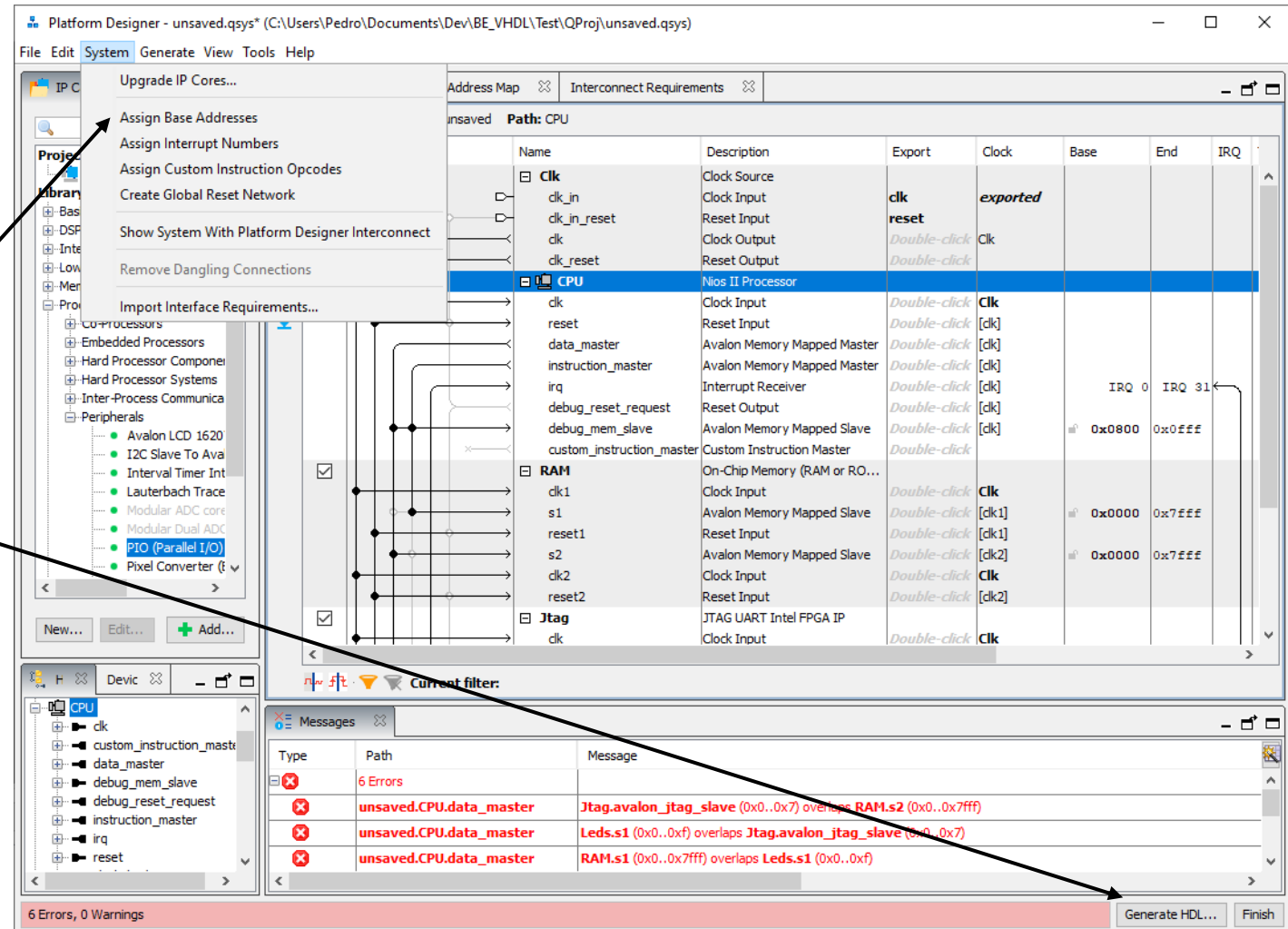
Platform Designer (Nios)

- Cliquer sur l'onglet « Vectors »
- Dans Reset vector memory et Exception vector memory choisir :
 - <NOM_RAM>.s1



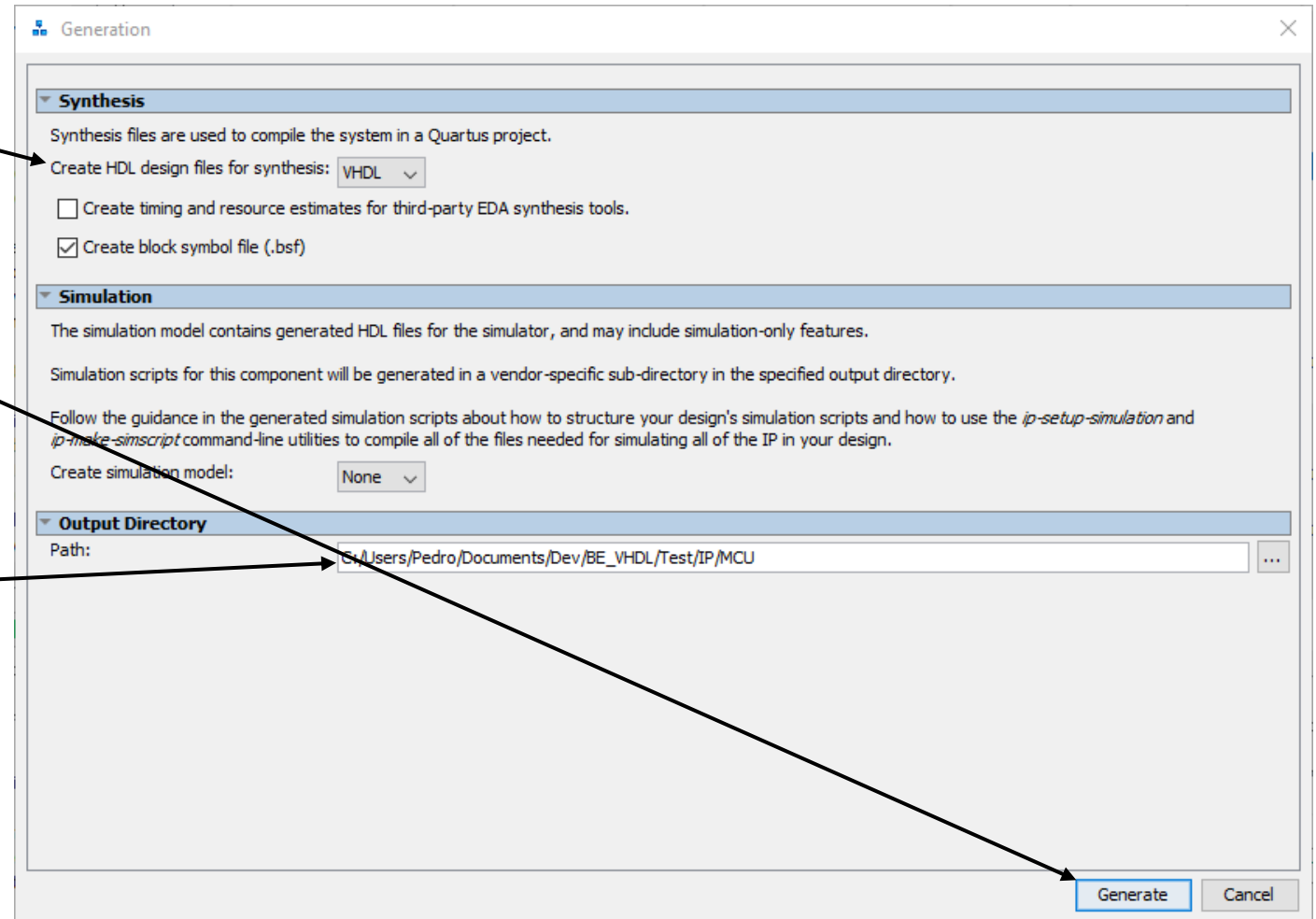
Platform Designer (Nios)

- Faire l'affectation automatique des adresses des composants en allant dans « System → Assign Base Addresses »
- Sauvegarder et cliquer sur « Generate HDL »



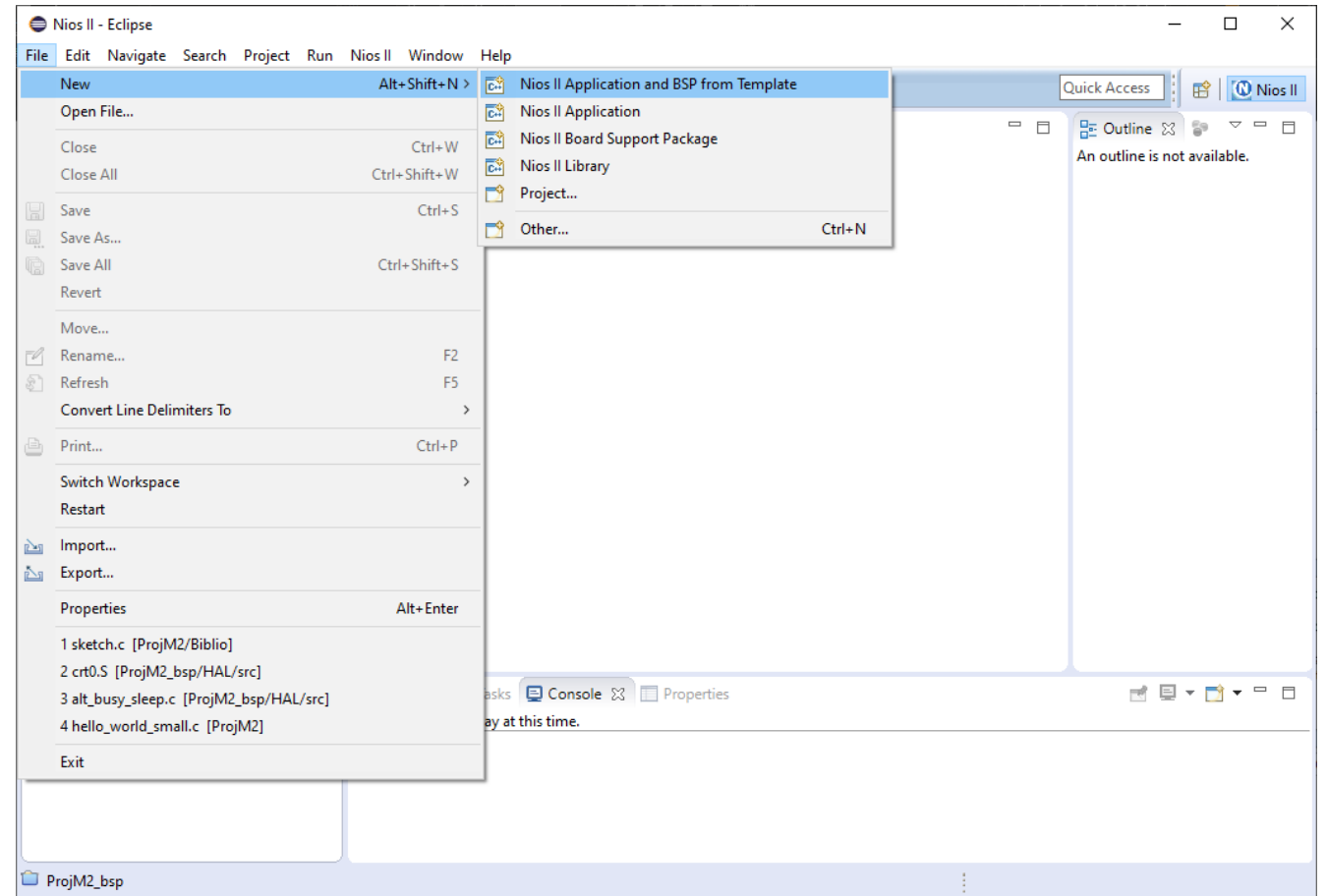
Platform Designer (Nios)

- Dans « Create HDL design files for synthesis » choisir « VHDL »
- Cliquer sur « Generate »
- Ajouter ensuite le .qip qui se trouve où vous avez sauvegardé le projet de « Platform Designer / Qsys » sous le dossier « synthesis ».



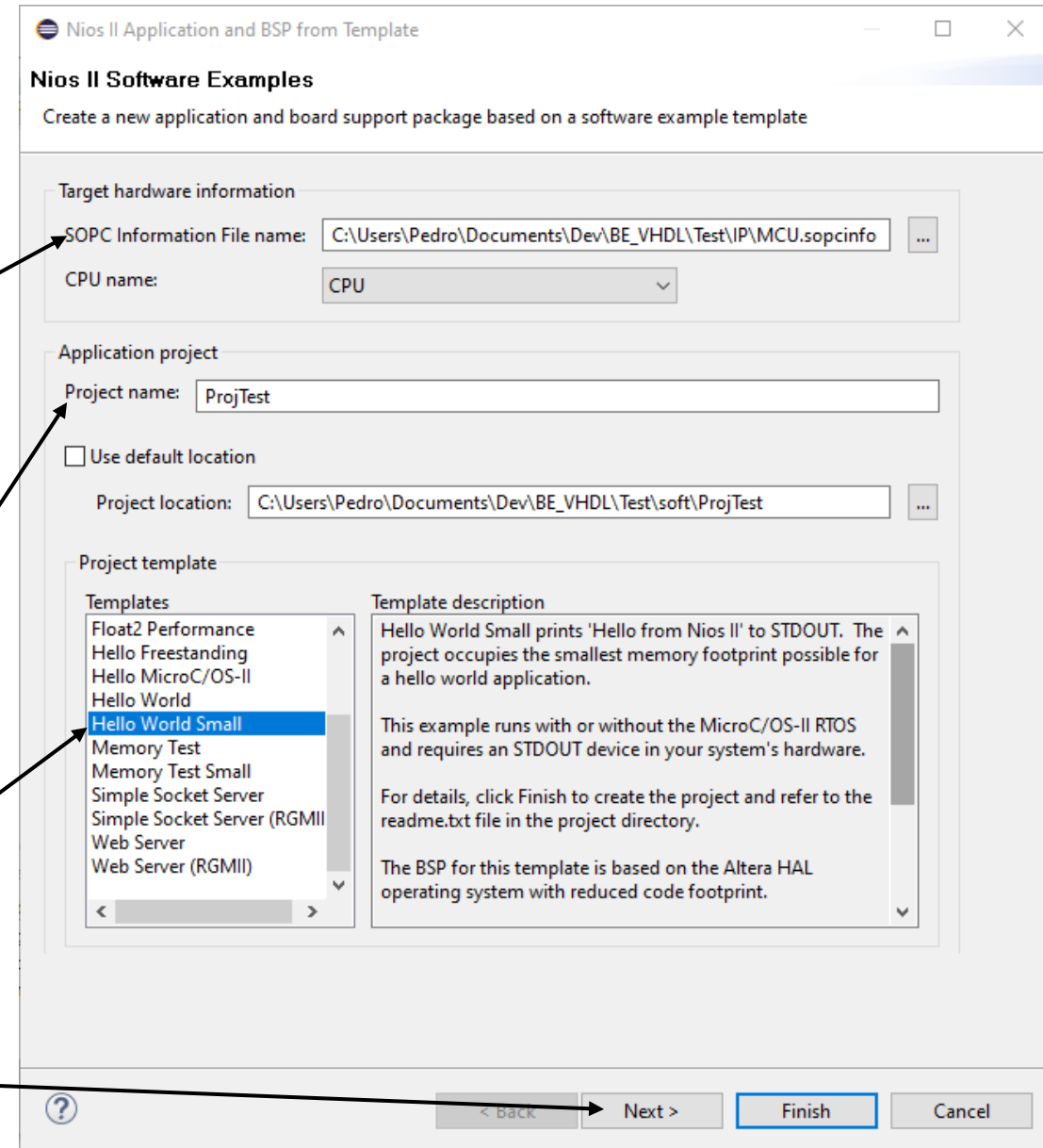
Nios Software Build Tools

- Créer un nouveau projet en allant dans « File → New → Nios II Application and BSP from Template »



Nios Software Build Tools

- Ajouter le fichier .sopc qui contient la configuration du microcontrôleur que vous avez créé avec « Platform Designer ». Ce fichier se trouve où vous avez sauvegardé le projet de « Platform Designer / Qsys »
- Saisir le nom de votre projet
- Choisir « Hello World Small »
- Cliquer sur « Next »



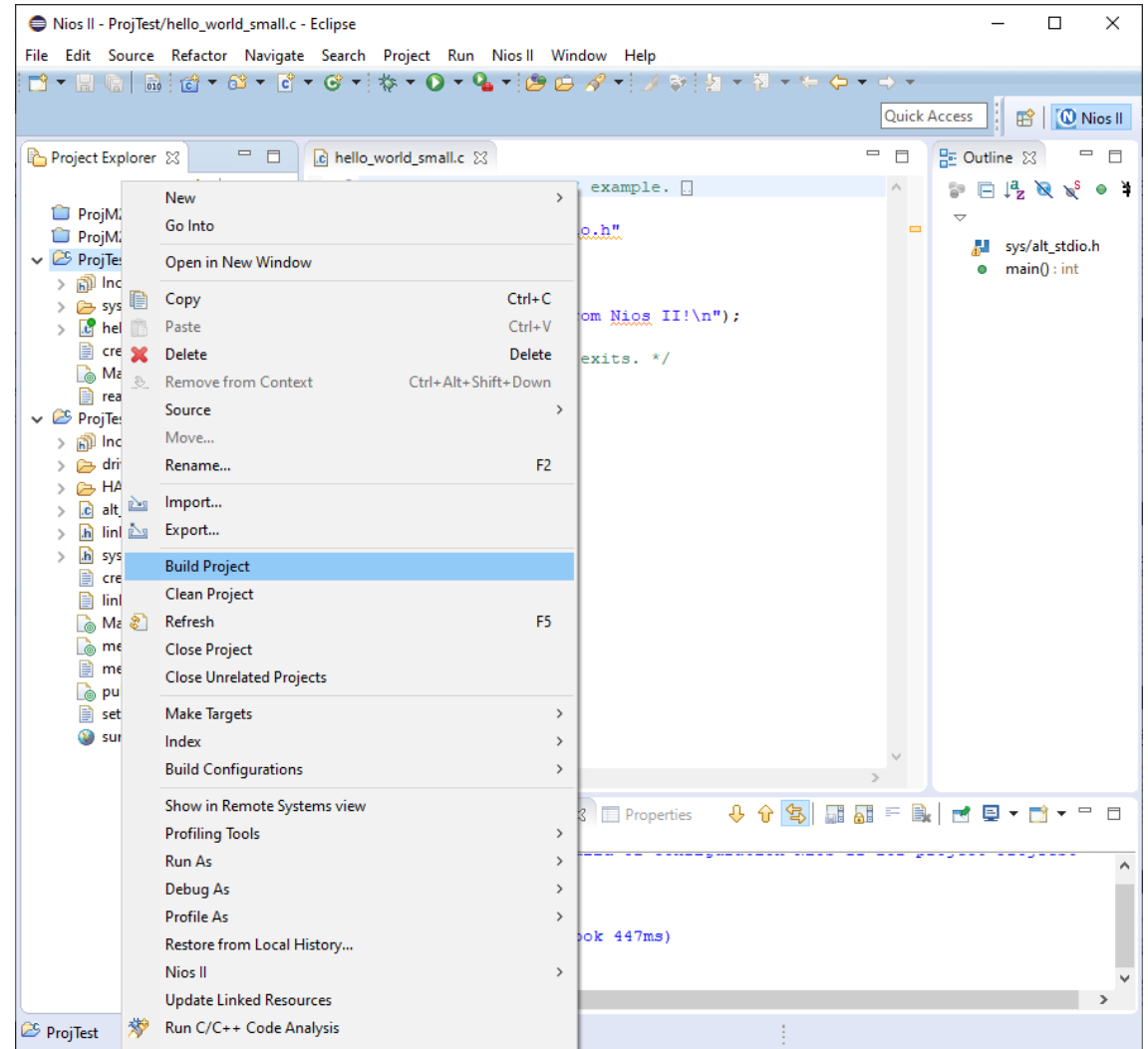
Nios Software Build Tools

- Ici vous pouvez un autre bsp soit laisser le logiciel créer un en fonction de la configuration du .sopc
- Le bsp est un projet qui contient toutes les définitions système ainsi que les adresses « base » de vos composants que vous avez ajouté sur « Platform Designer »
- Cliquer sur « Finish »

The screenshot shows a Windows-style dialog box titled "Nios II Application and BSP from Template". Below the title bar, it says "Nios II Software Examples" and "Select a board support package for your application". There are two radio buttons: the first is selected and labeled "Create a new BSP project based on the application project template", and the second is labeled "Select an existing BSP project from your workspace". Under the first option, there is a "Project name:" text box containing "ProjTest_bsp", a checked "Use default location" checkbox, and a "Project location:" text box containing a file path. To the right of the "Project location" box is a small "..." button. Under the second option, there is a large empty rectangular box, and to its right are "Create..." and "Import..." buttons. At the bottom of the dialog, there is a row of four buttons: a help button with a question mark, "< Back", "Next >", and "Finish" (which is highlighted with a blue border), and a "Cancel" button.

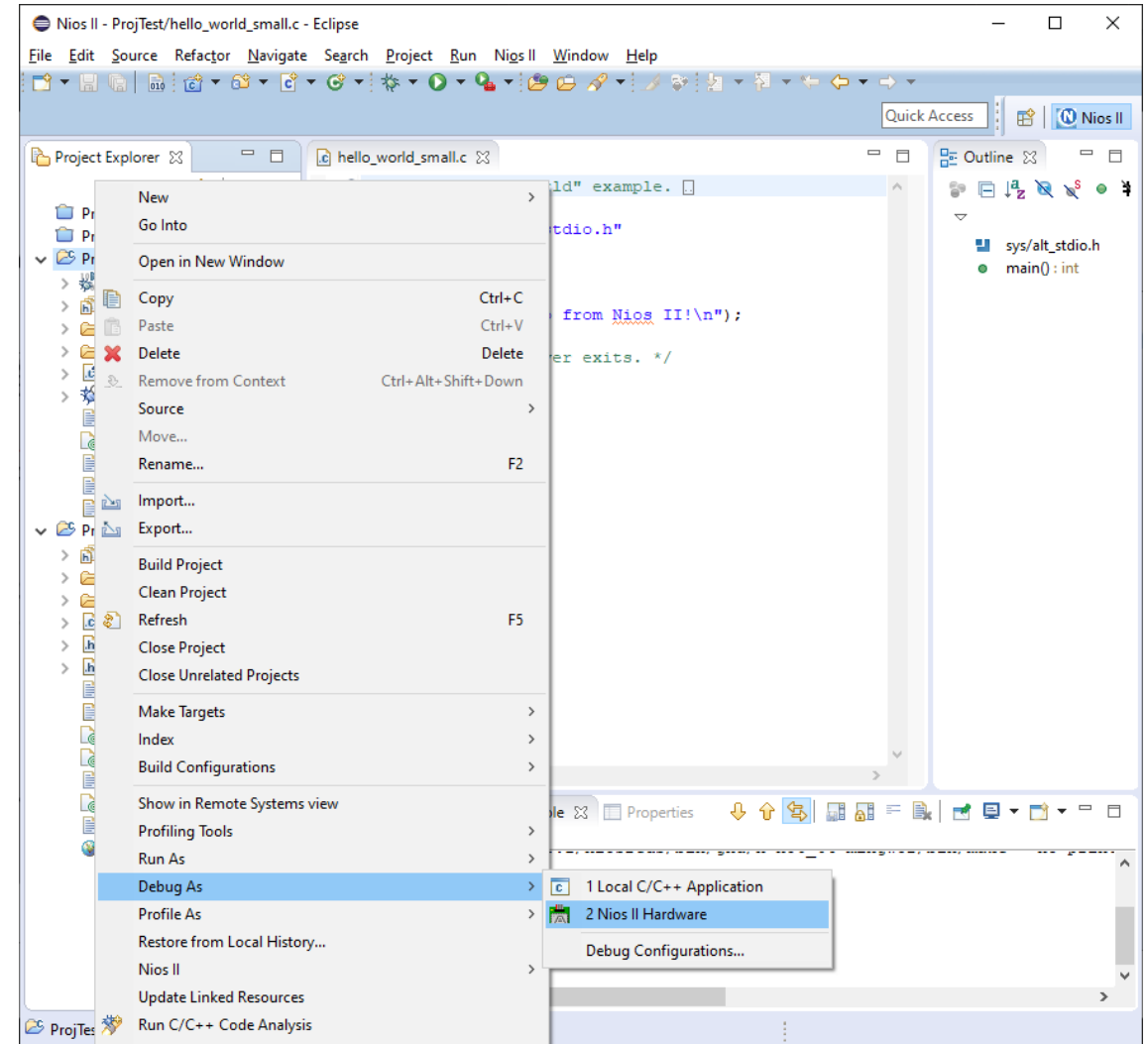
Nios Software Build Tools

- Après avoir créé le projet, un code template vous est donné qui affiche « Hello from Nios II! » sur la console UART si vous avez ajouté un UART sur platform designer.
- Pour compiler le projet faites clic droit sur « votre projet → Build Project »



Nios Software Build Tools

- Pour télécharger votre code vers le Nios et démarrer une session de debug faites un clic droit sur « votre projet → Debug As → Nios II Hardware ».



Nios Software Build Tools

- Si c'est la première fois que vous téléversez votre code vers la cible, il faut scanner le FPGA
- Assurez-vous que le FPGA est programmé avec la configuration du Nios (.sof)
- Assurez-vous que ces deux cases sont cochées si vous n'avez pas ajouté le composant « System ID Peripheral Intel FPGA IP ».
- Aller dans « Target Connection → Refresh Connections » pour scanner le FPGA
- Choisir le FPGA et cliquer sur « Apply » et ensuite « Debug »

