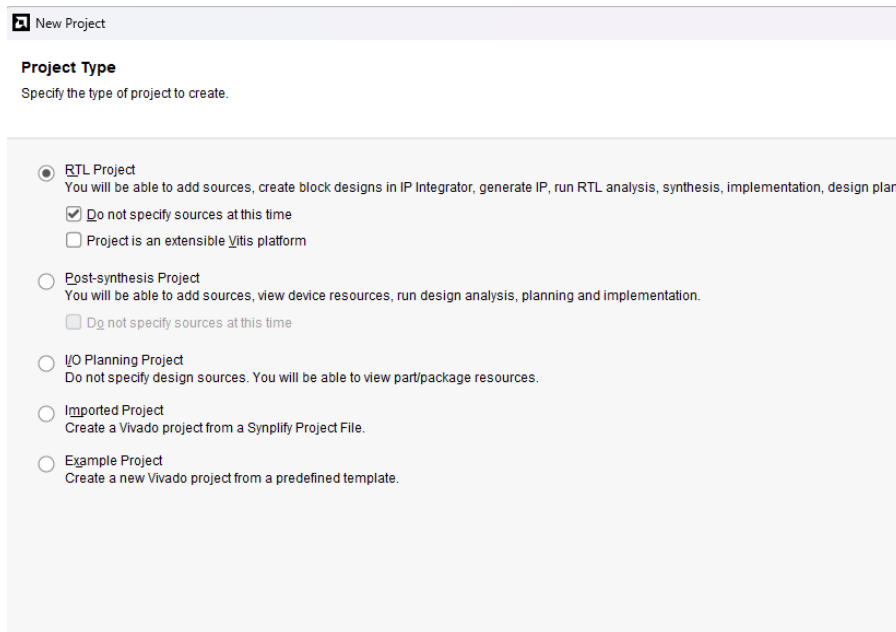


## Costruzione di un SoC con periferica personalizzata

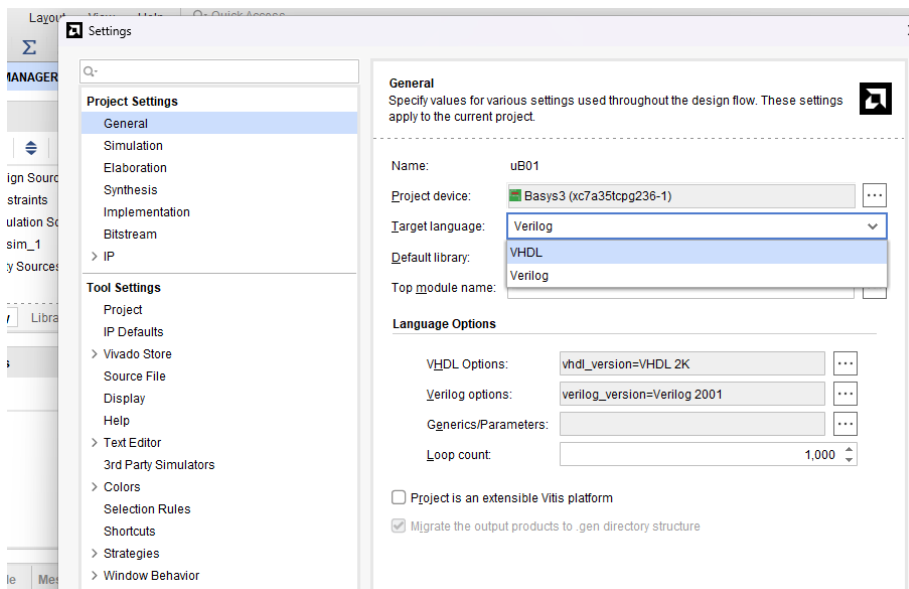
Creare un Progetto in vivado senza selezionare “Project is an extensible vitis platform”

(molte guide dicono di selezionare questa opzione, ma non è compatibile con l’ultima versione di VITIS)

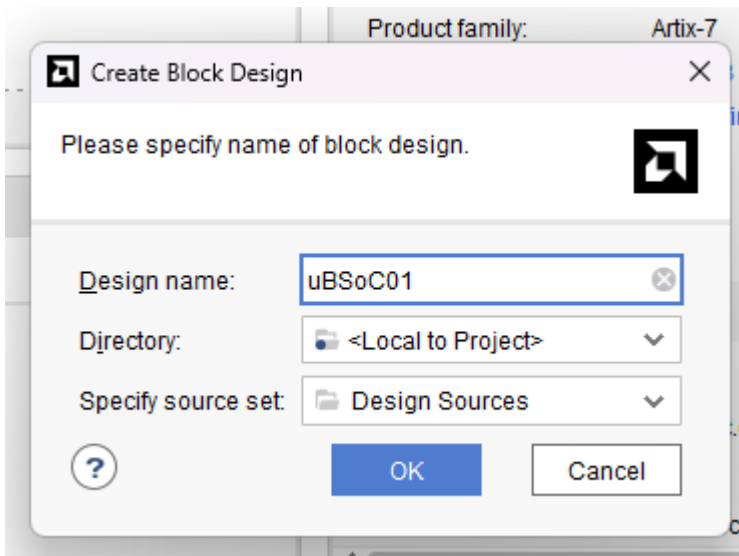


Selezionare la scheda di sviluppo (basys 3) e modificare nelle proprietà il default in “vhd”

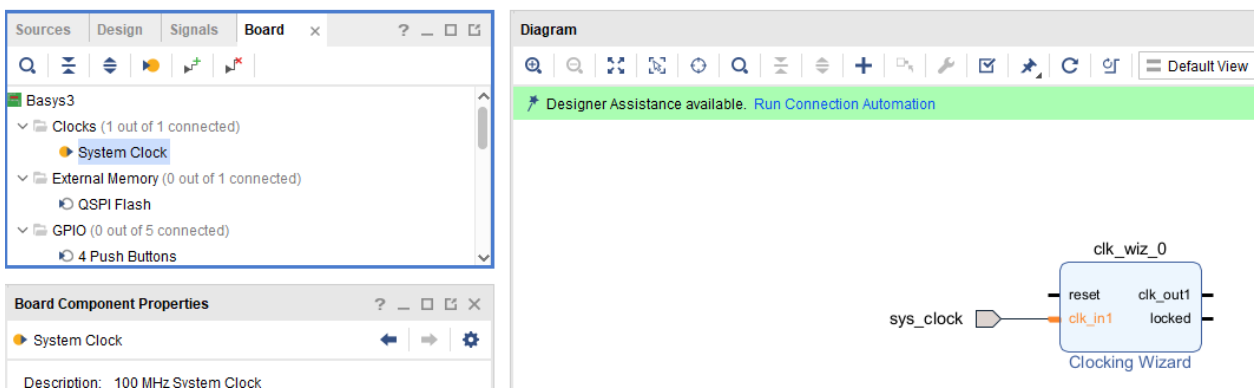
(Tools->settings->general)



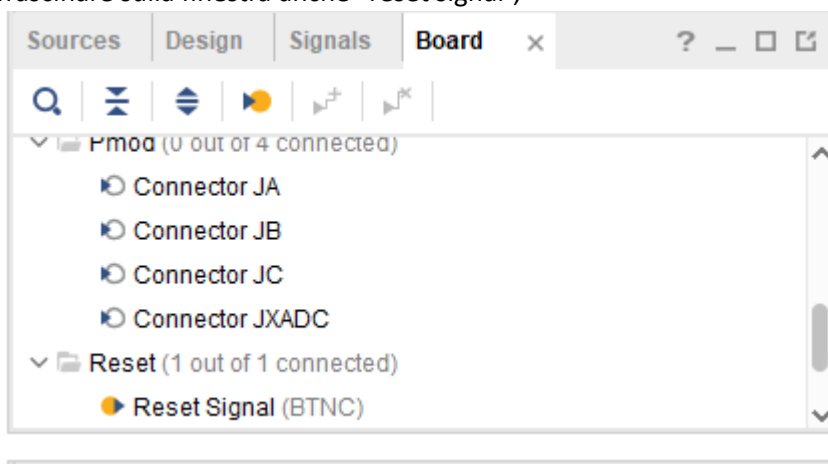
In “IP integrator” scegliere “create block design” e dare un nome.



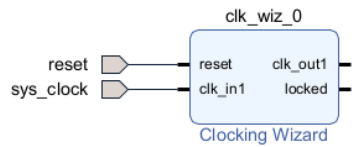
Trascinare System Clock nella finestra "Diagram".



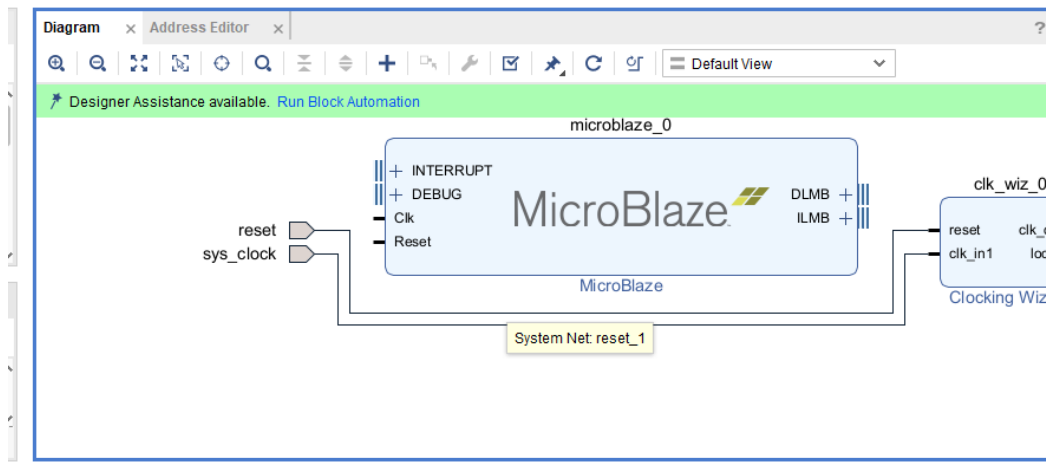
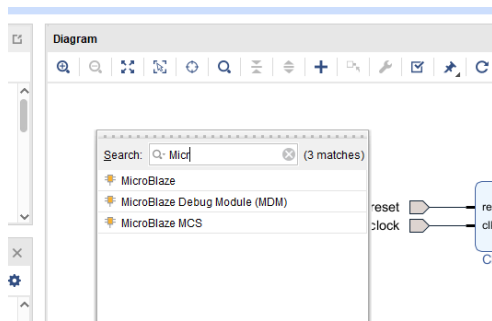
Trascinare sulla finestra anche "reset signal",



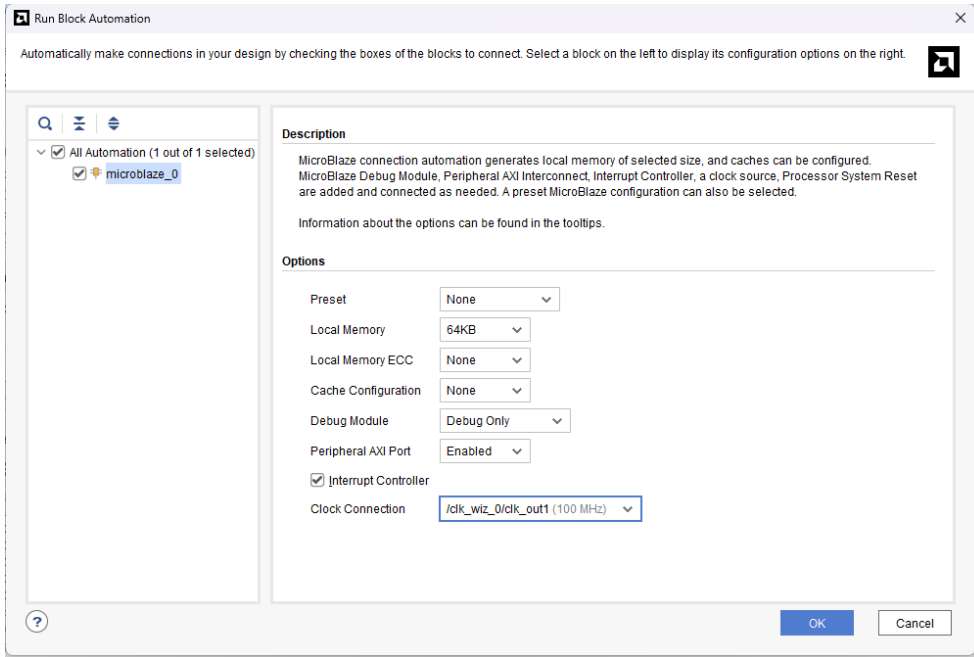
che automaticamente porterà a questa situazione:



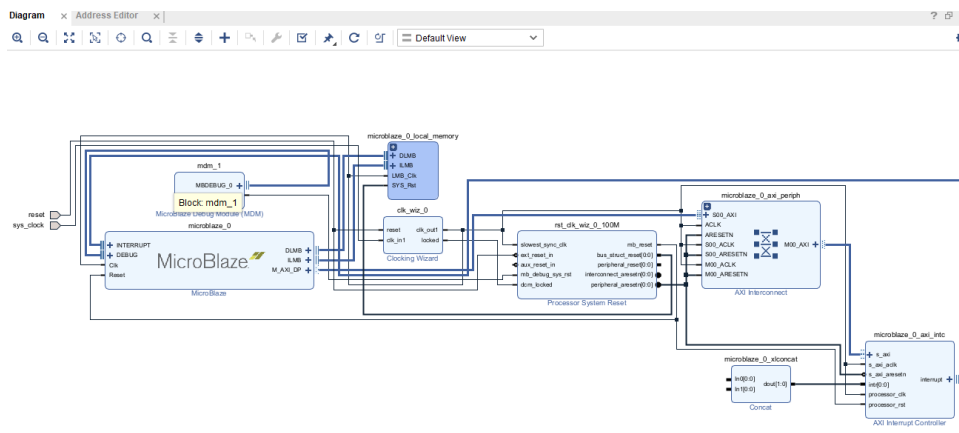
Aggiungiamo il la CPU micoblaze facendo attenzione a mettere una memoria adeguata (almeno 32k, meglio 64k). Per ottenere questo risultato, fare click sul segno “+” nelle icone della finestra “diagram”.



Dopo l’inserimento, per scegliere la memoria e ottenere la connessione automatica col clock, fare click su Run Connection Automation. Selezionare anche “Interrupt Controller” e verificare che Peripheral AXI port sia “enabled”.



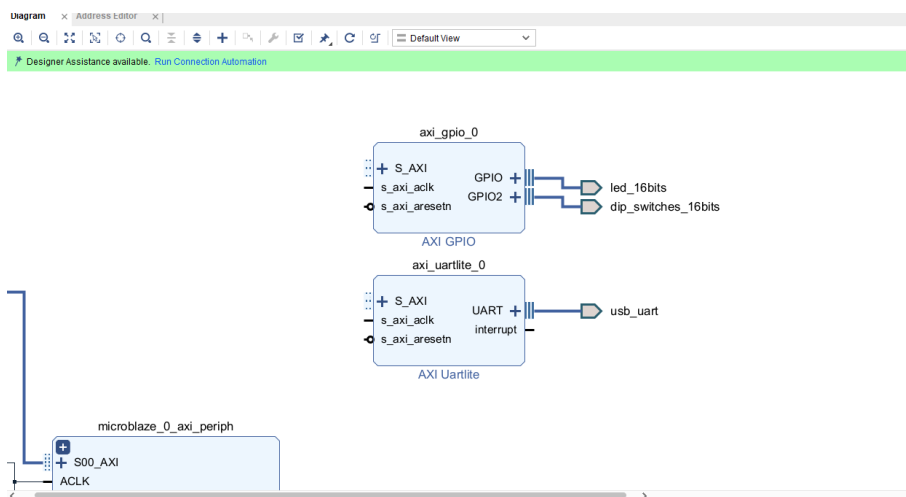
Al termine si ottiene:



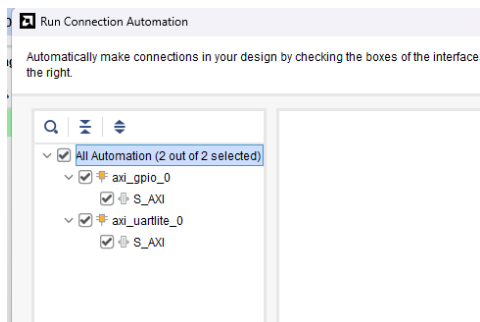
Inseriamo alcune periferiche predefinite dal menù “Board”.

Inseriamo “uart lite”, Inseriamo la porta per i led (16 led) e quella per gli switch (16 switches).

Nota che queste ultime due convergono automaticamente in una singola porta GPIO come in figura:

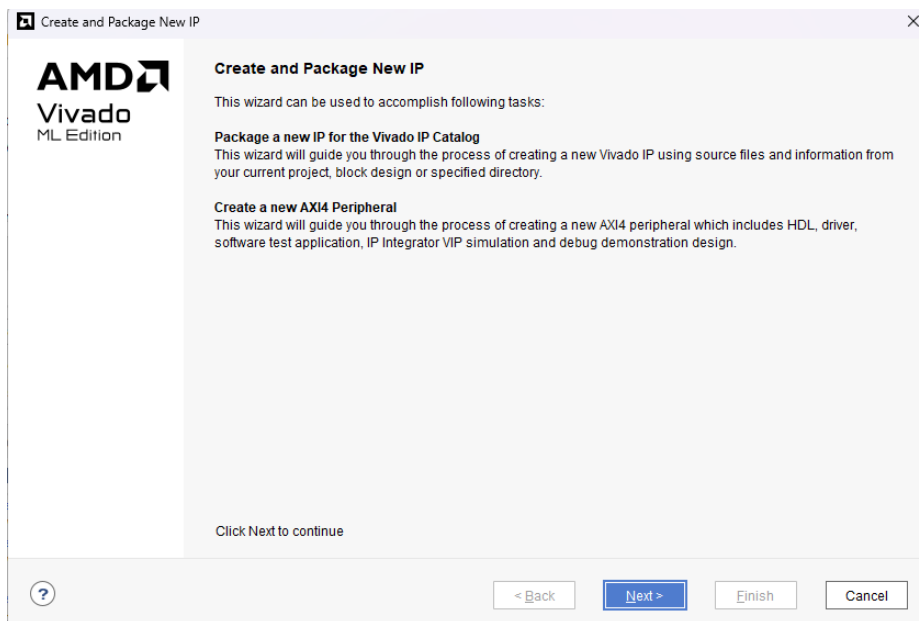
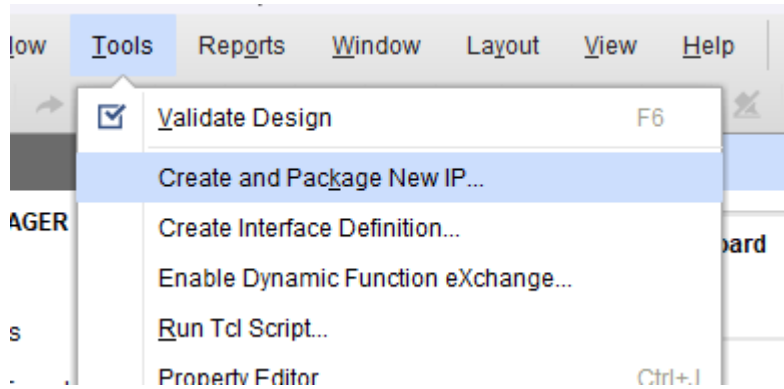


Facciamo click su “Run connection automation” e selezioniamo “all automation”, quindi next.



Ora aggiungiamo una periferica “vuota” che poi modificheremo. Definiamo per il momento due porte (di ingresso e di uscita). Clock e reset sono previsti automaticamente.

Per ottenere questo risultato selezionare "Create and Package New IP" in Tools.



Selezionare "Next".

Create and Package New IP

### Create Peripheral, Package IP or Package a Block Design

Please select one of the following tasks.

**Packaging Options**

- Package your current project  
Use the project as the source for creating a new IP Definition.
- Package a block design from the current project  
Choose a block design as the source for creating a new IP Definition.  
Select a block design:
- Package a specified directory  
Choose a directory as the source for creating a new IP Definition.

**Create AXI4 Peripheral**

- Create a new AXI4 peripheral  
Create an AXI4 IP, driver, software test application, IP Integrator AXI4 VIP simulation and debug demonstration design.

? < Back Next > Finish Cancel

Selezionare "Create a new AXI4 peripheral", quindi Next.

Create and Package New IP

### Peripheral Details

Specify name, version and description for the new peripheral

Name:

Version:

Display name:

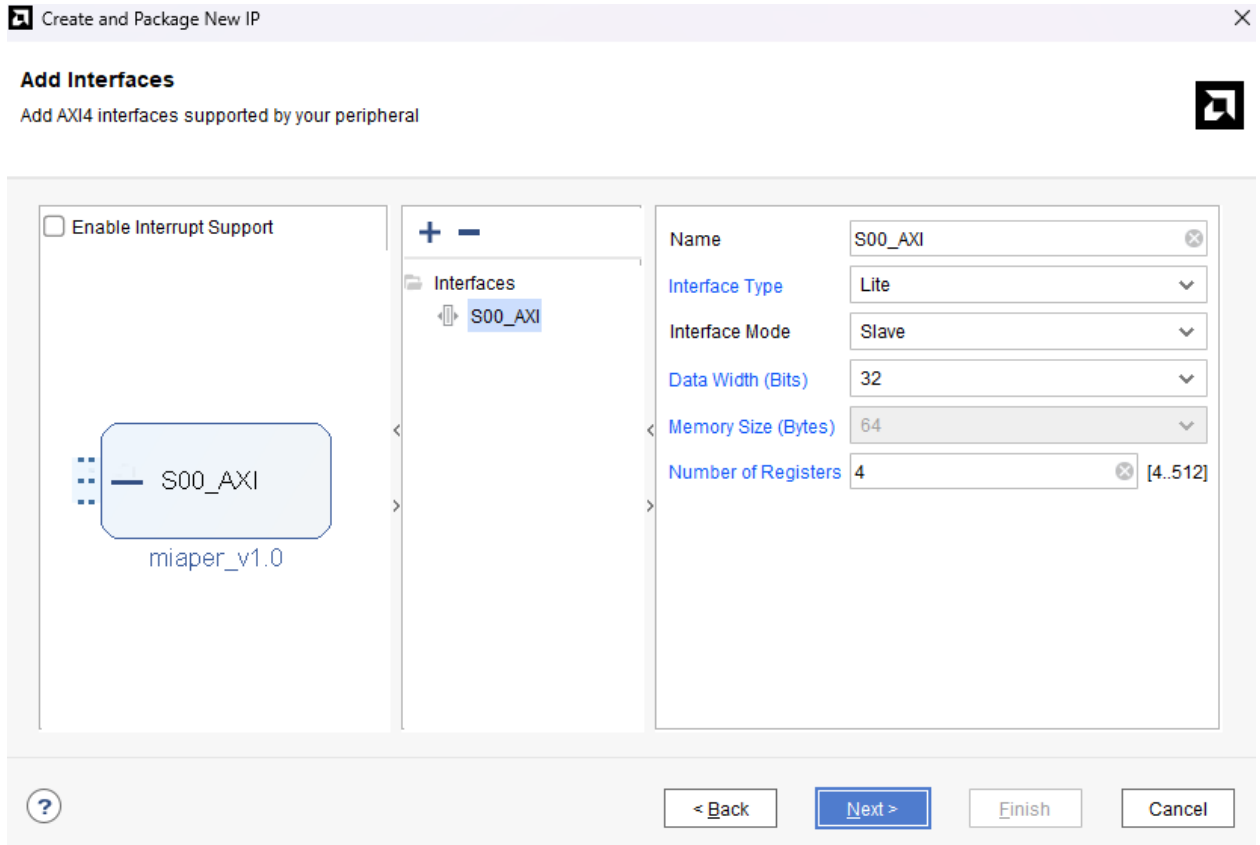
Description:

IP location:

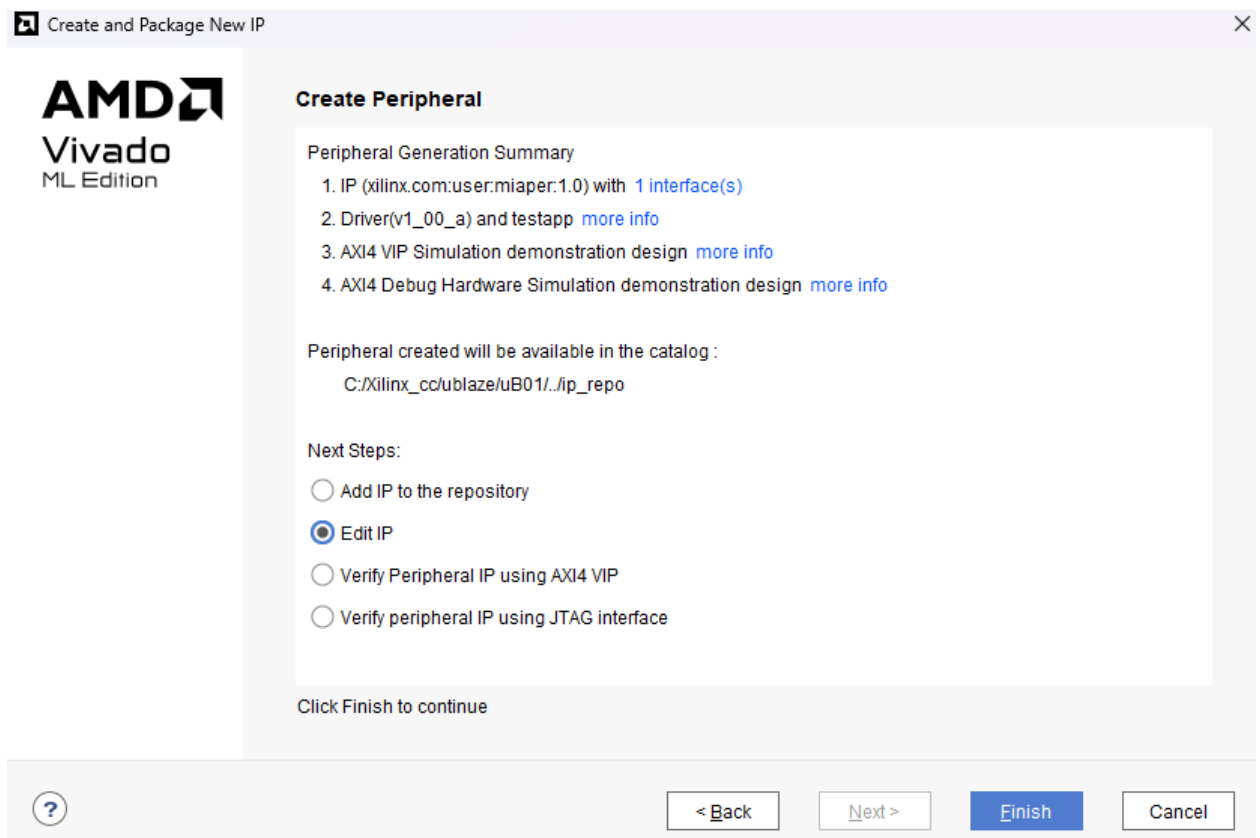
Overwrite existing

? < Back Next > Finish Cancel

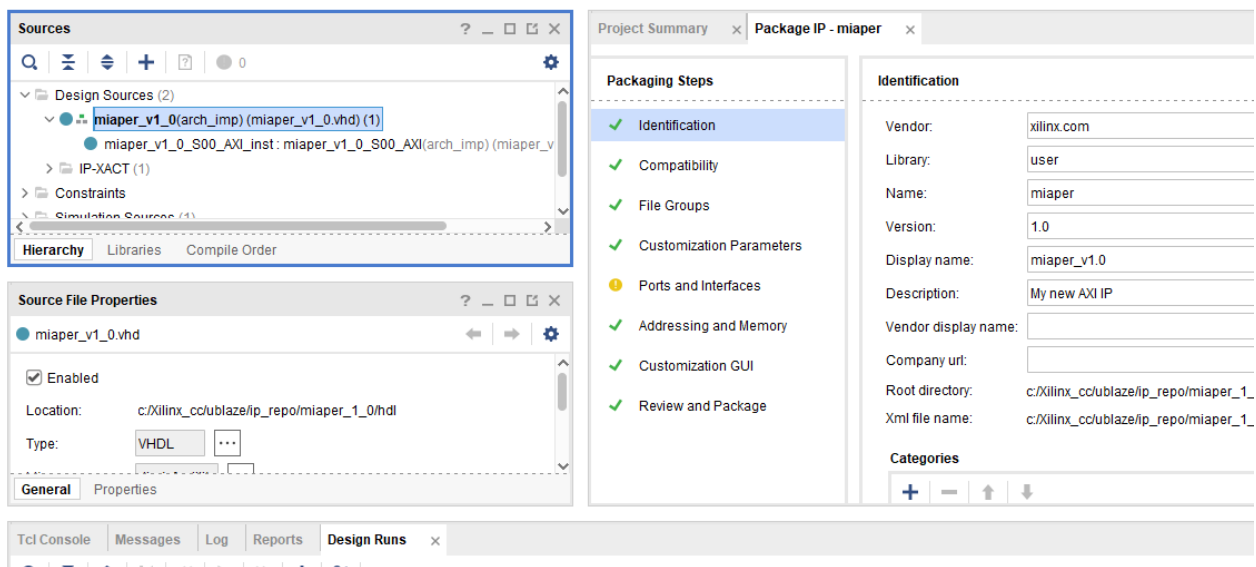
Dare un nome, quindi Next.



Verificare che le selezioni siano come nella figura sopra, quindi Next.



Scegliere "Edit IP", quindi Finish. A questo punto si apre una nuova finestra vivado.



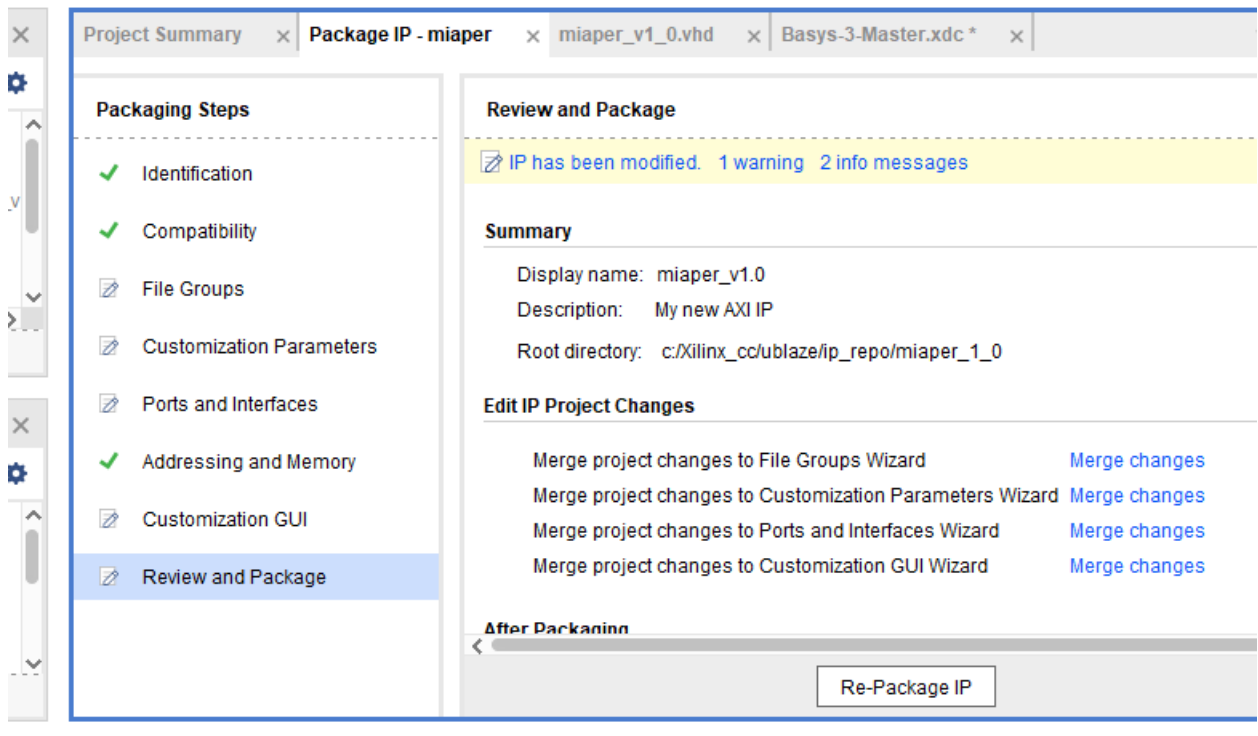
Aprire la top entity e modificarla aggiungendo le due righe relative a JA e JB.

```

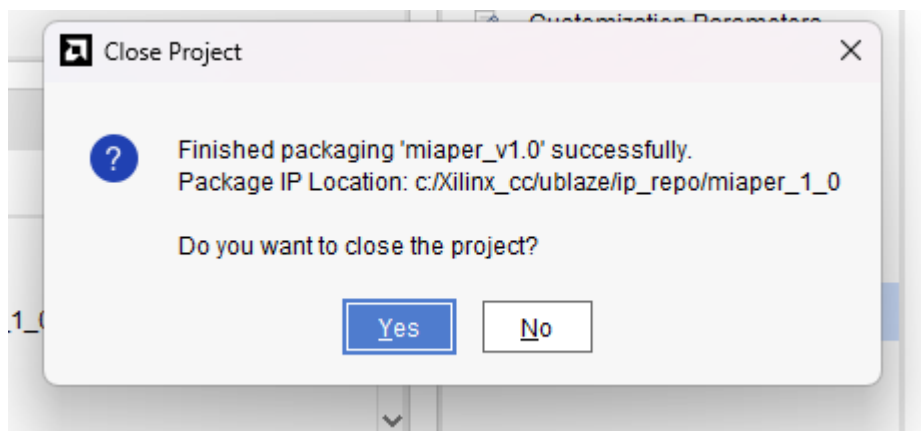
5  entity miaper_v1_0 is
6      generic (
7          -- Users to add parameters here
8
9          -- User parameters ends
10         -- Do not modify the parameters beyond this line
11
12
13         -- Parameters of Axi Slave Bus Interface S00_AXI
14         C_S00_AXI_DATA_WIDTH    : integer    := 32;
15         C_S00_AXI_ADDR_WIDTH   : integer    := 4
16     );
17     port (
18         -- Users to add ports here
19         JA:in std_logic_vector (7 downto 0);
20         JB:out std_logic_vector (7 downto 0);
21
22         -- User ports ends

```

Dopo aver salvato, selezionare Review and package. Quindi fare click su “Re-Package IP” dopo aver fatto il “Merge” dei cambiamenti.



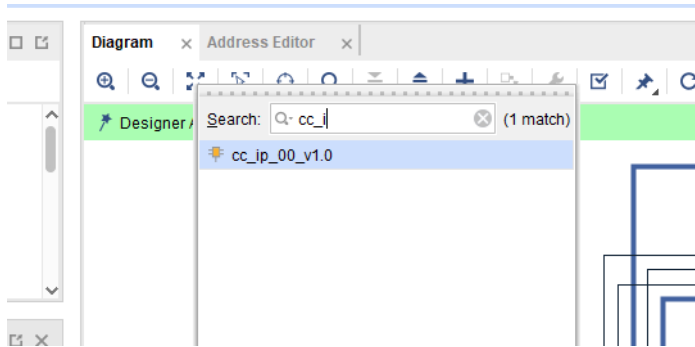
Completata l'operazione, rispondere "yes" quando viene chiesto di uscire:



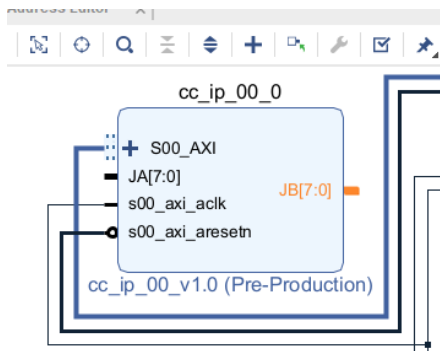
A questo punto si ritorna nella finestra vivado originale.

Aggiungiamo la periferica appena generata (che non fa nulla, perché non abbiamo specificato nulla) e inseriamola nella finestra.

**Attenzione! Il nome della periferica è cambiato nel frattempo: per un errore ho dovuto rifare tutto daccapo.**



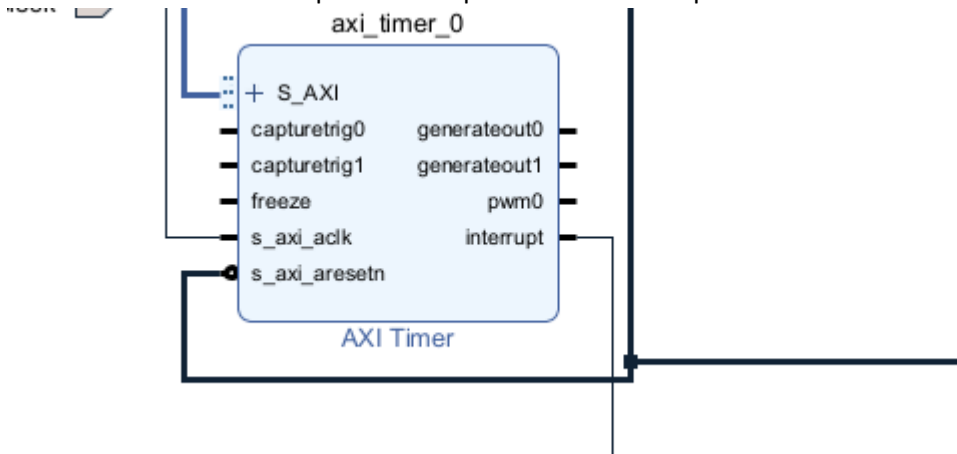
A questo punto “Run connection automation”.



Selezionare una per una le porte di uscita e di ingresso e scegliere “make external”.

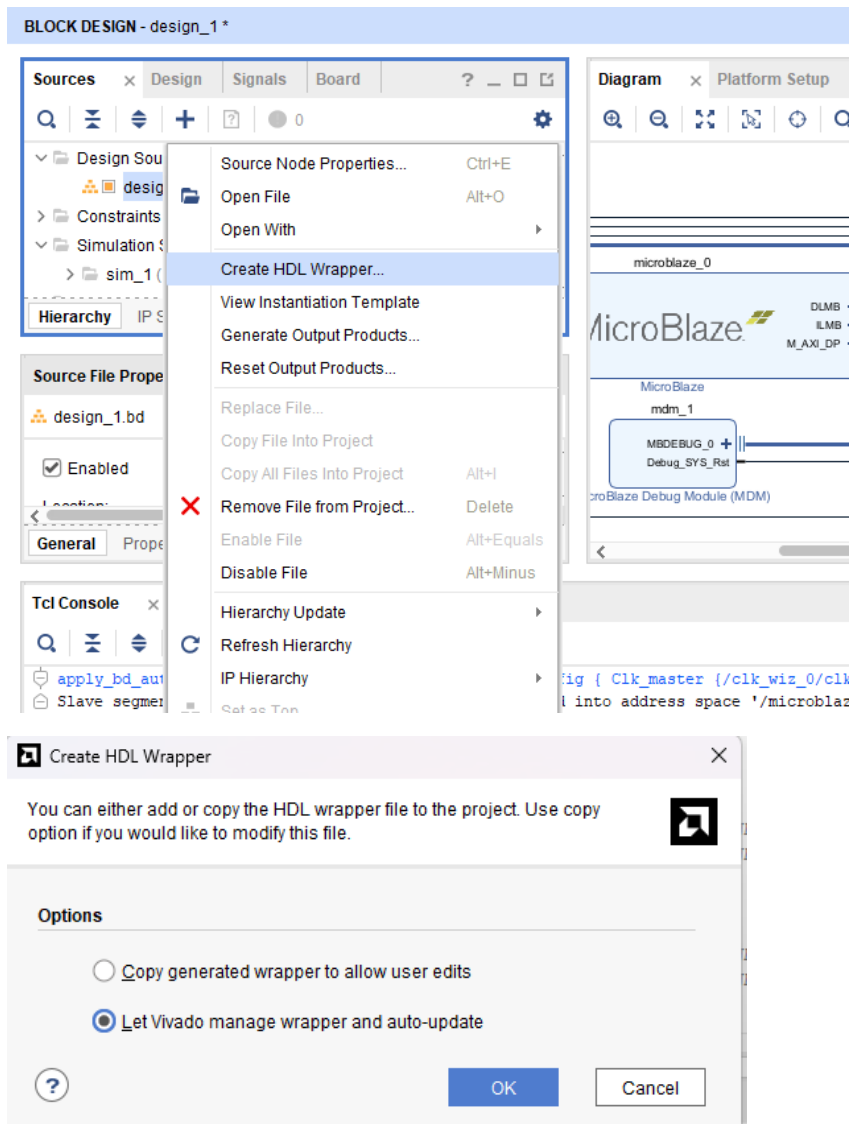
A questo punto bisogna considerare che periferiche inserite dalla sezione “Board” contengono la corrispondenza fra le porte e i piedini della FPGA. Però questo non vale per l’ultima interfaccia. A questo punto bisogna includere il file di constraints, limitatamente alle righe relativa a JA e JB.

Inseriamo infine un timer per rendere possibile un interrupt cadenzato. Non usiamo i pin addizionali:



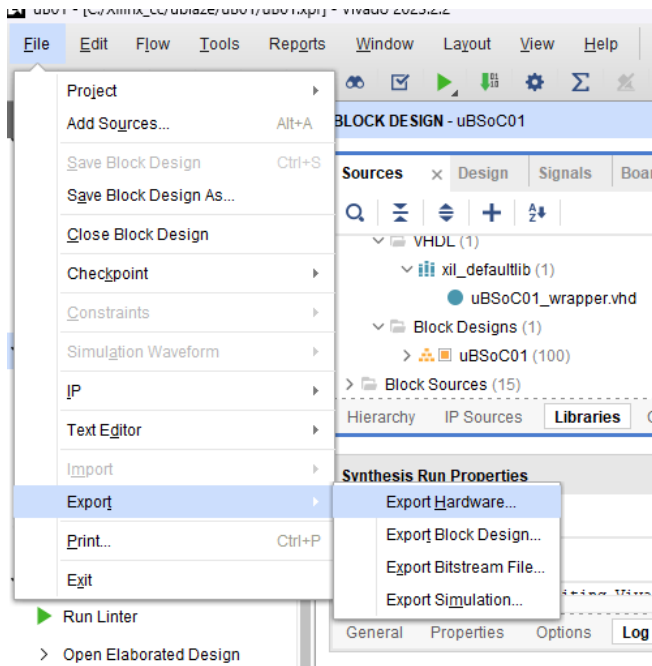
Collegare l’interrupt della serial uart e del timer all’interrupt controller.

A questo punto bisogna creare un HDL wrapper come in figura>

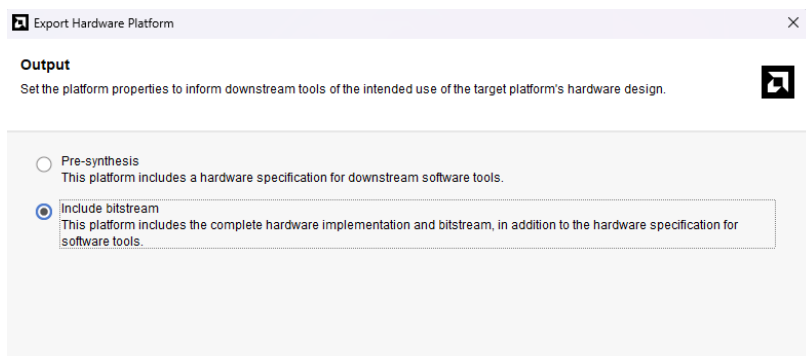


Infine bisogna generare il bitstream.

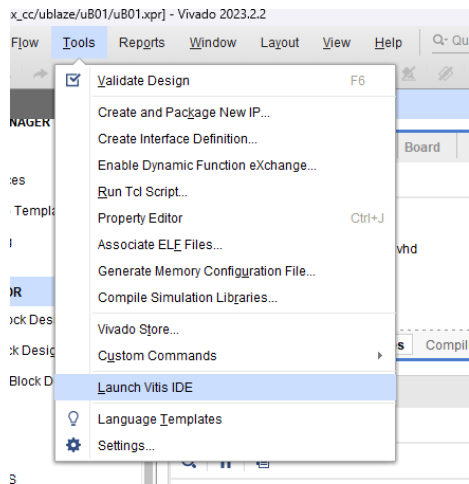
Al termine esportare l'hardware:



Si faccia attenzione a includere il bitstream:

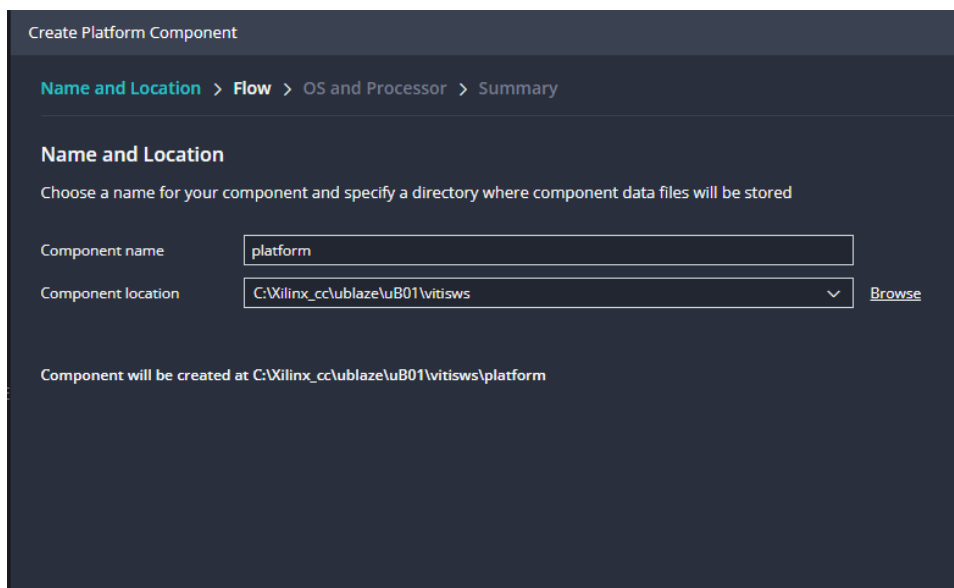
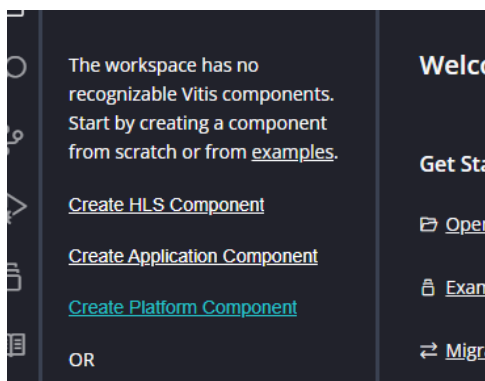


A questo punto si può aprire vitis.

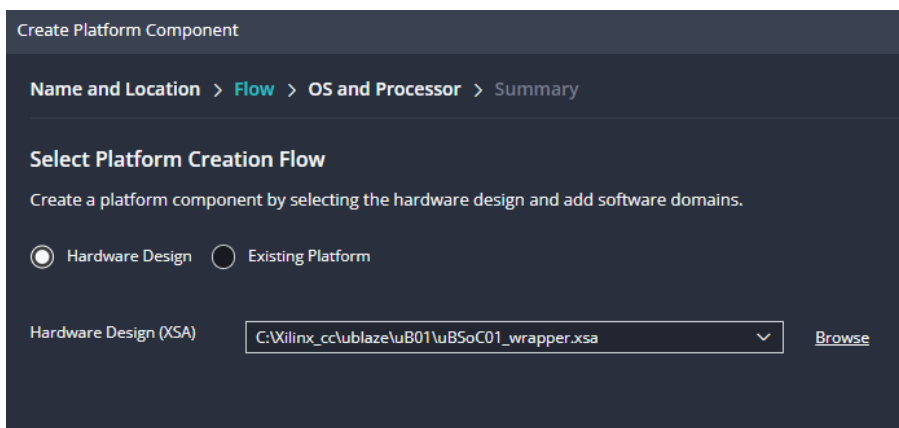
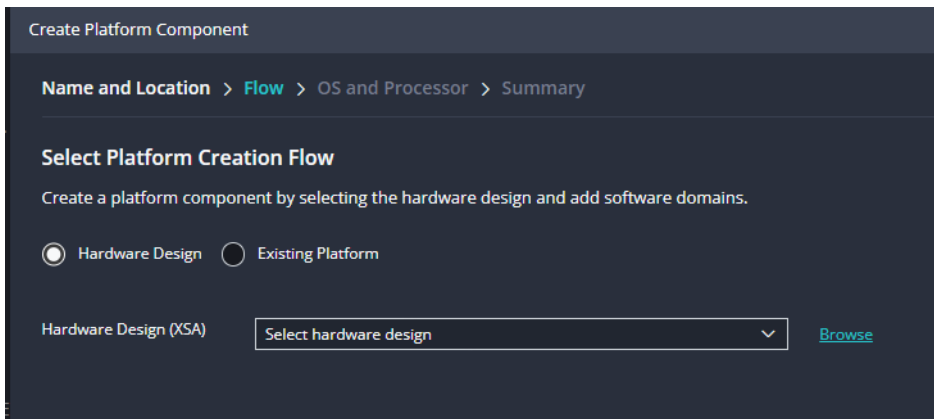


Dentro vitis, per prima cosa aprire un workspace.

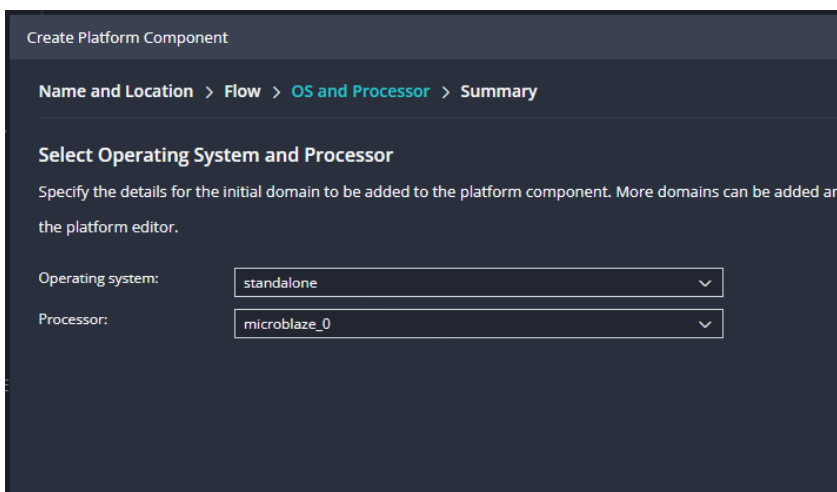
Quindi creare un platform component



Fare next e poi scegliere l'hardware appena esportato:

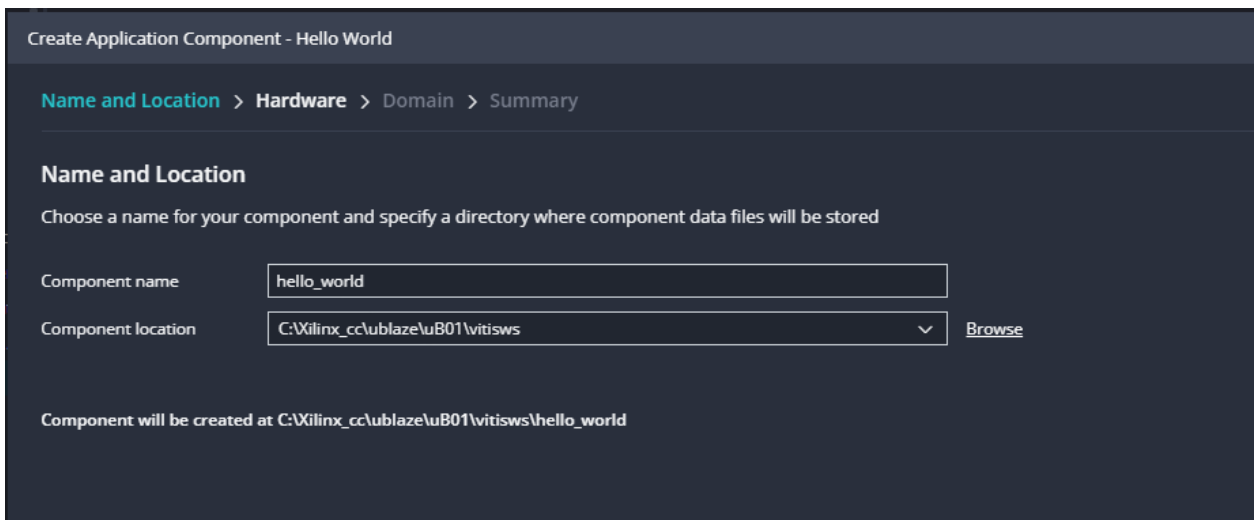
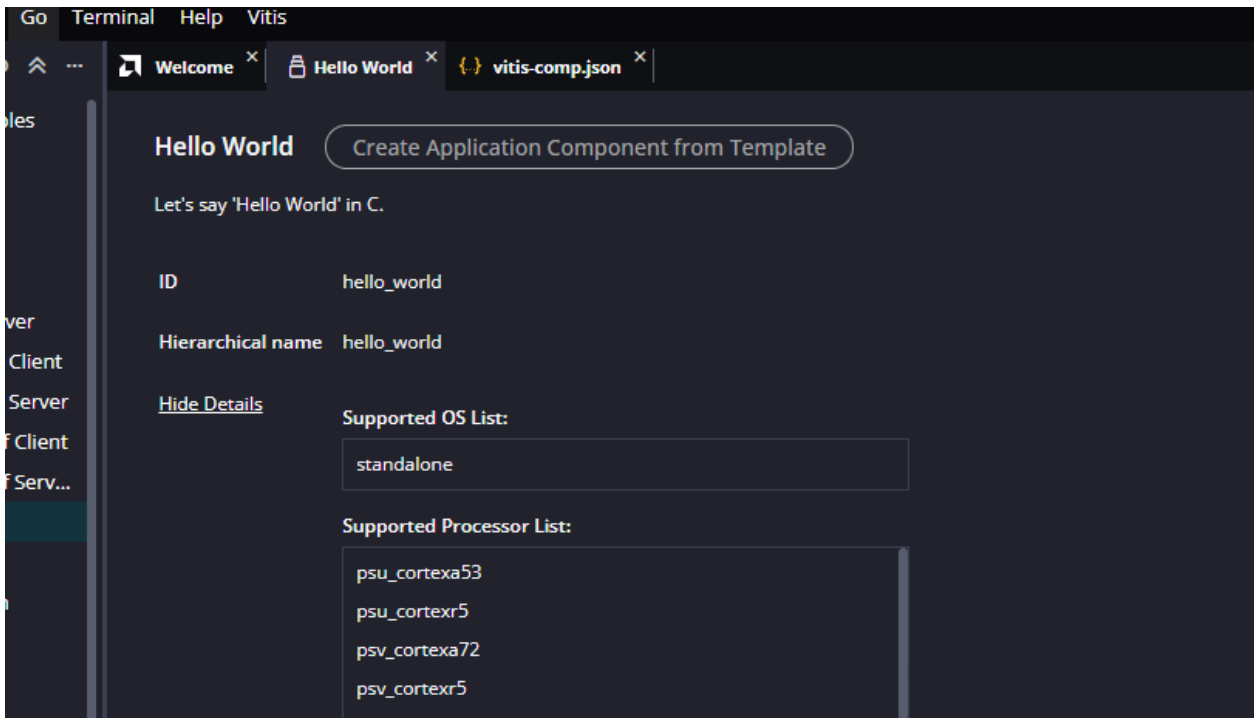


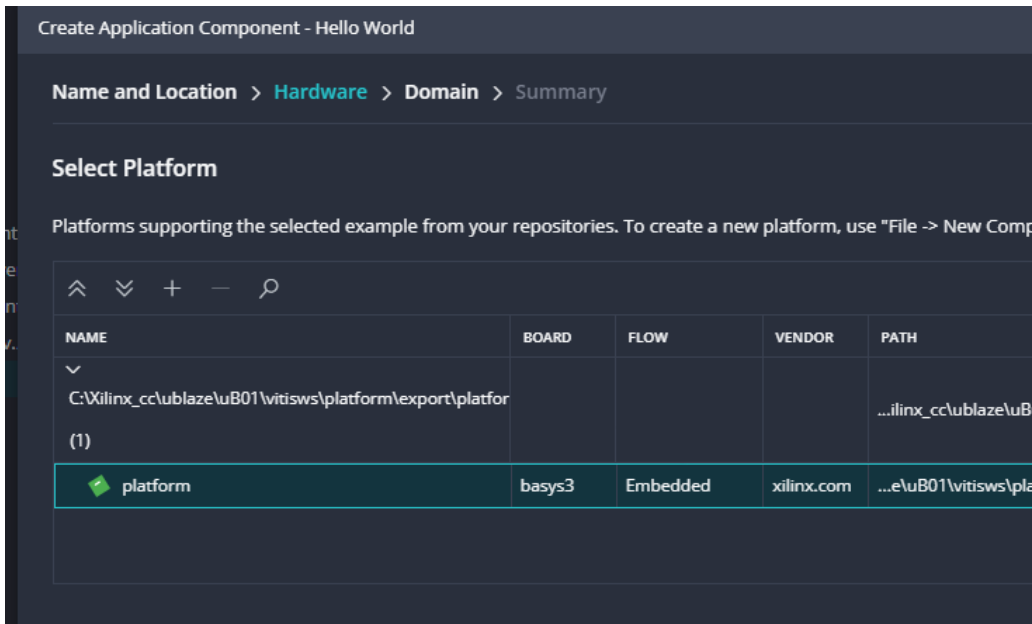
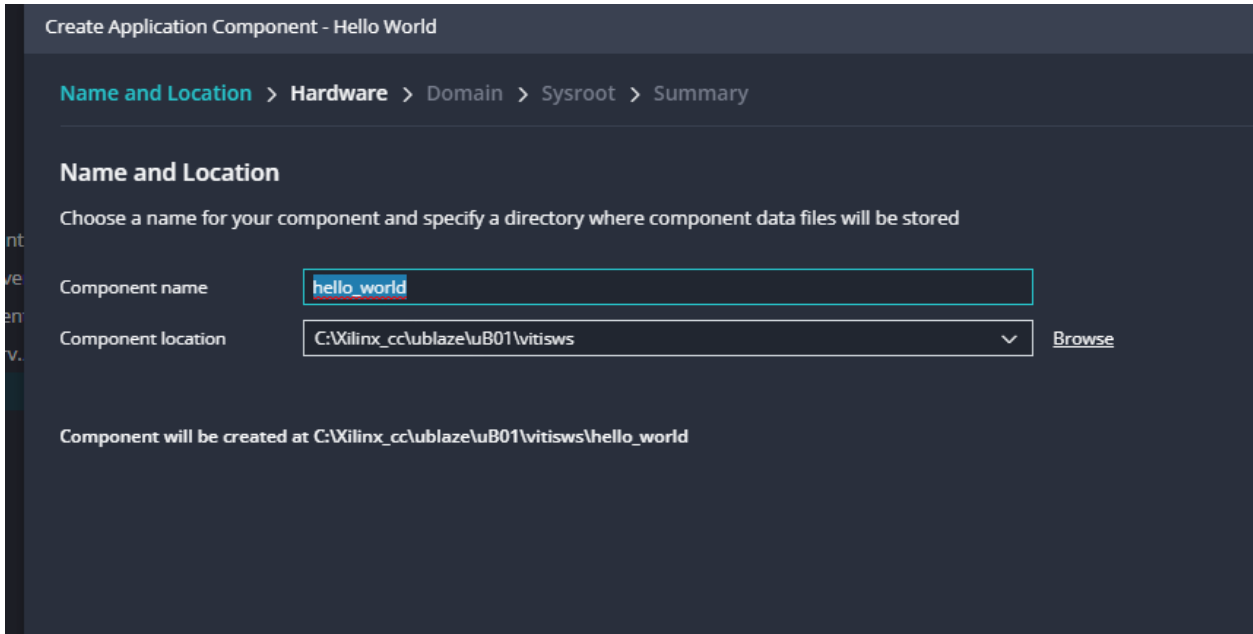
A questo punto vitis carica la piattaforma e ricava informazioni sul tipo di piattaforma e sul processore:



Dopo la creazione della piattaforma (processo lento), fare build della stessa.

Al termine si crea una applicazione partendo da helloworld negli esempi.





Create Application Component - Hello World

Name and Location > Hardware > Domain > Summary

### Select Domain

Choose a domain from the available domains in the platform

Name	Details
standalone_microblaze_0	
+ create new...	
	Name standalone_microblaze_0
	Display Name standalone_microblaze_0
	OS standalone
	Processor microblaze_0

Create Application Component - Hello World

Name and Location > Hardware > Domain > Summary

### Summary

The following new application component will be created

Name	hello_world
Workspace	C:\Xilinx_cc\ublaze\B01\vtisws
Component directory	C:\Xilinx_cc\ublaze\B01\vtisws\hello_world
Platform	platform
Domain	standalone_microblaze_0 (OS:standalone, Processor:microblaze_0)

Dopo aver fatto il build di "hello world", si può fare run con la scheda basys collegata. Con un programma terminale aperto sulla com utilizzata dalla BASYS 3 si deve leggere:

"Hello World

Successfully ran Hello World application".

Ogni volta che si preme il pulsante di reset, sul terminale vengono nuovamente scritte le due frasi.

A questo punto si può usare hello world per la scrittura del proprio software. Ovviamente occorre anche modificare la periferica per aggiungere quanto necessario a farle svolgere la funzione che desideriamo.