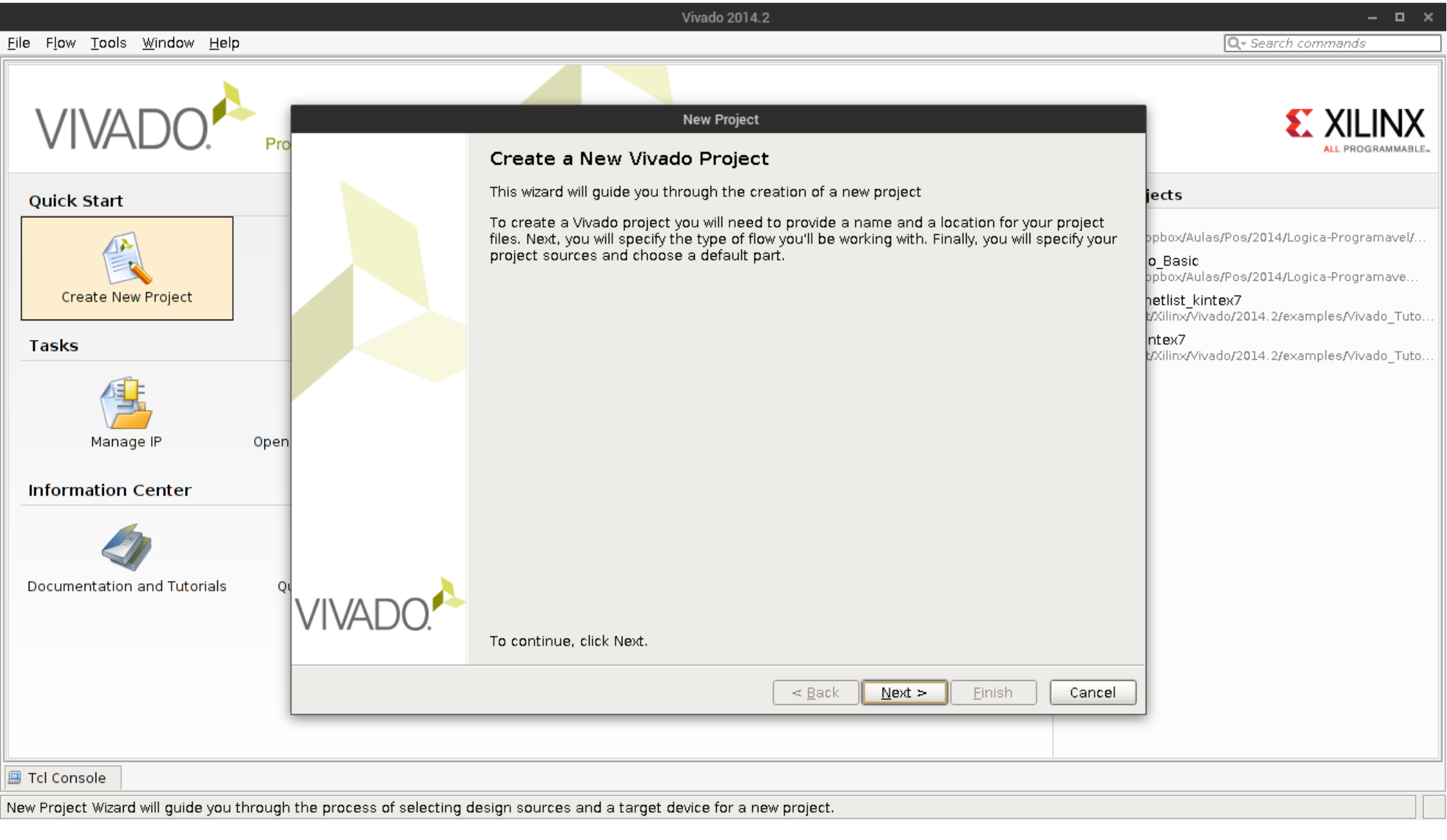
Tutorial Vivado

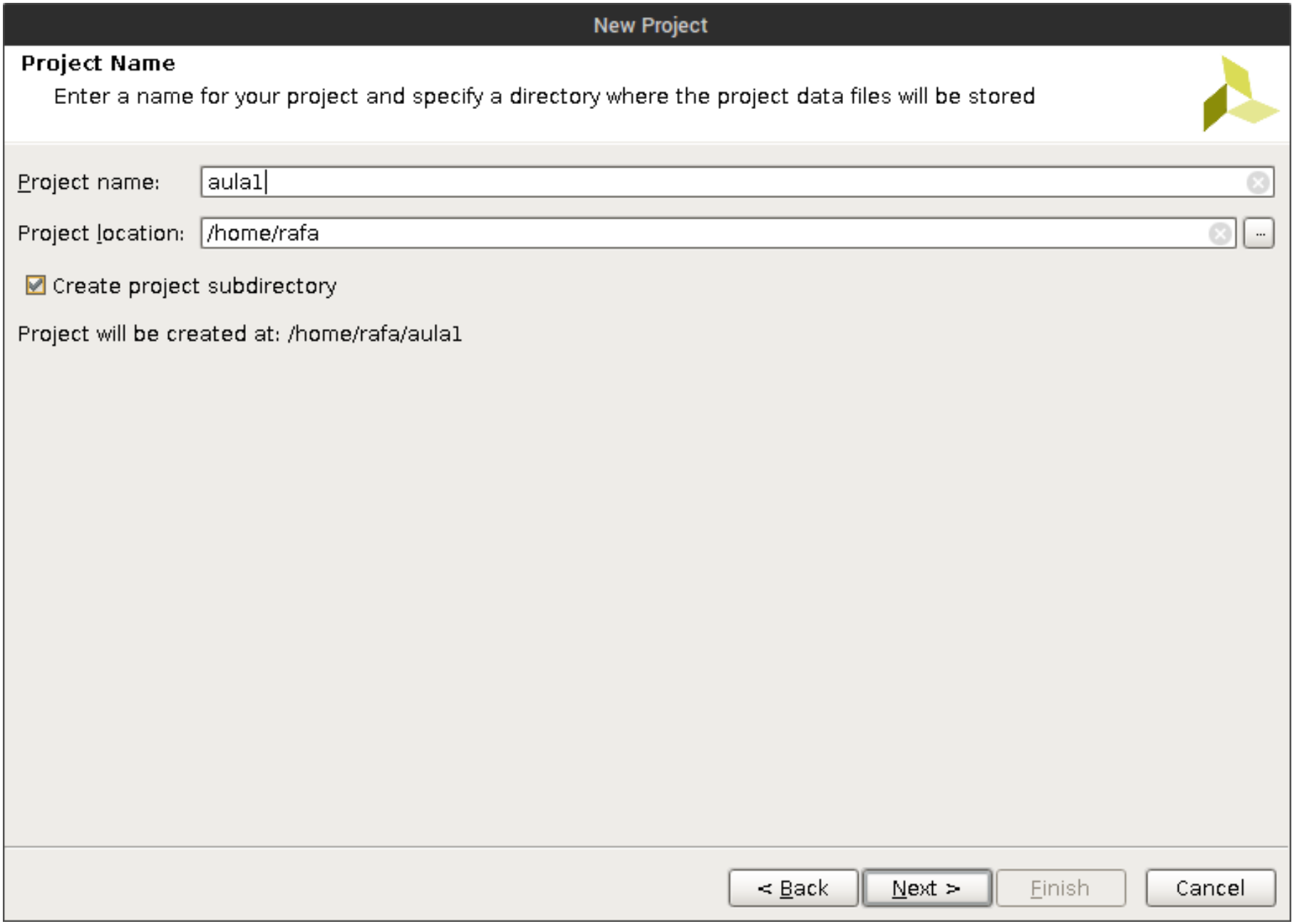
Introdução a ferramenta Xilinx Vivado

Rafael Corsi – Mauá

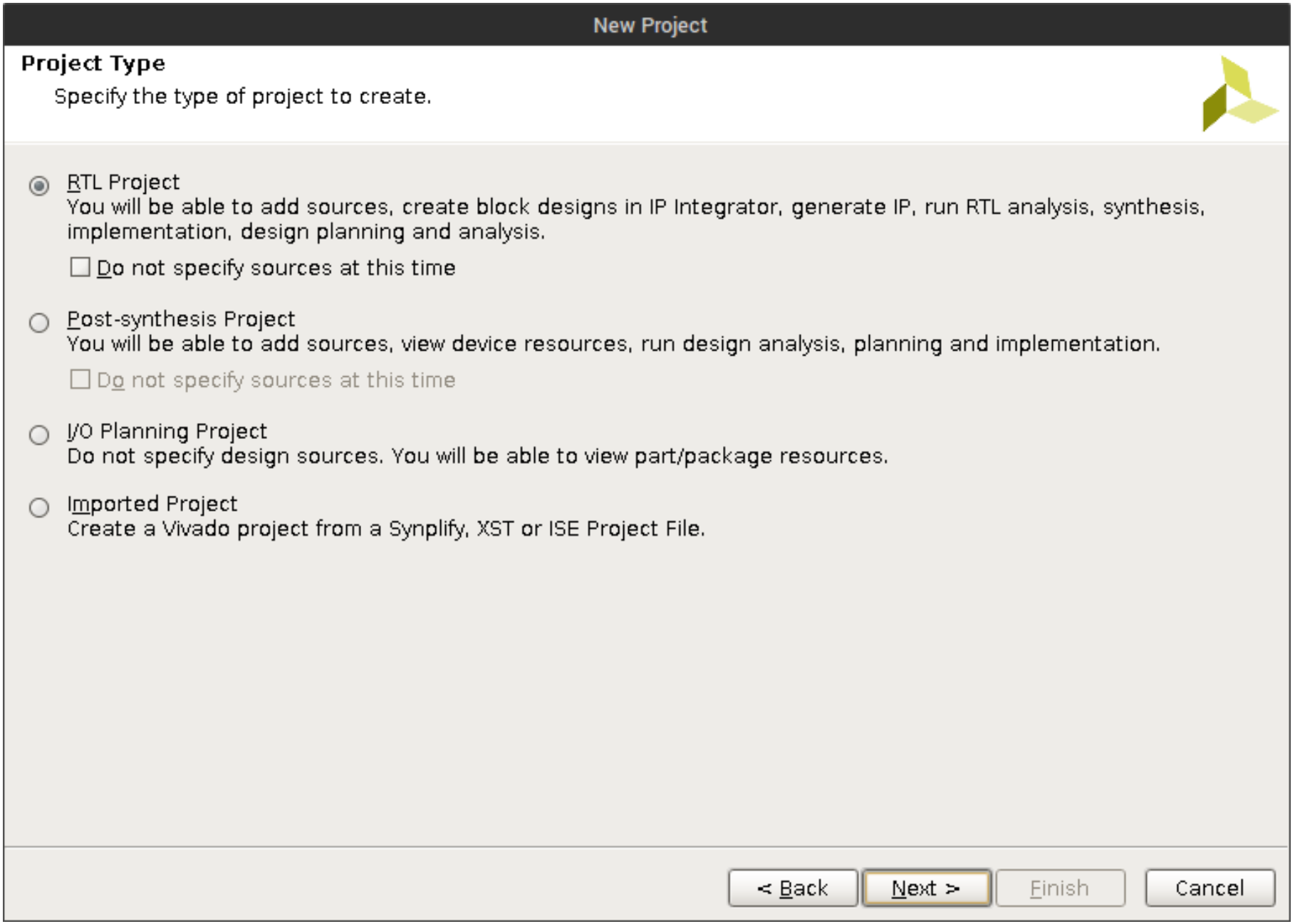
Criando um novo projeto



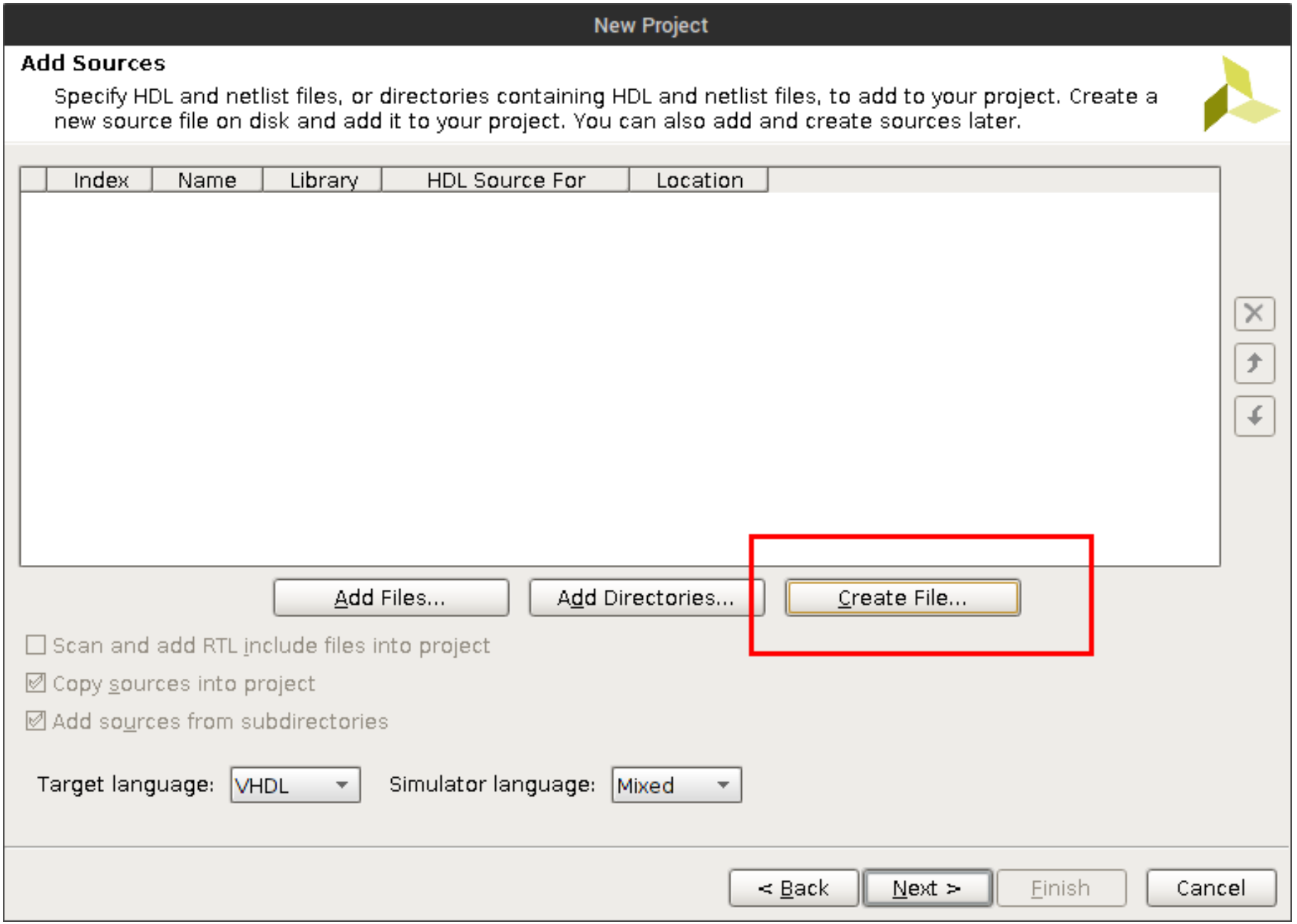
Nome do projeto



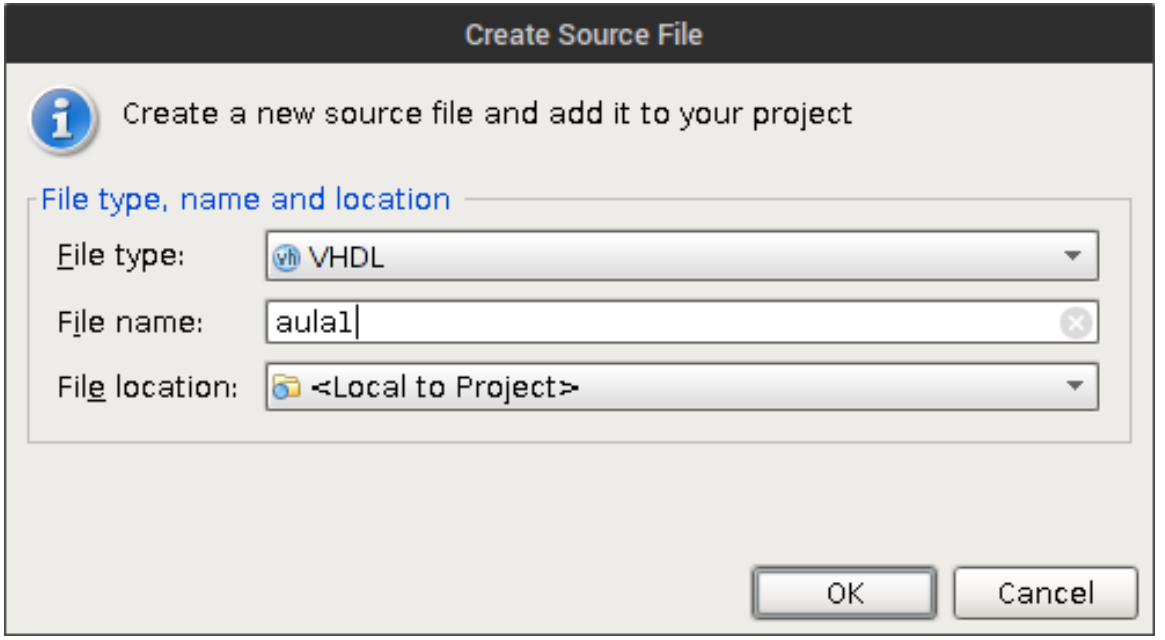
Tipo do projeto



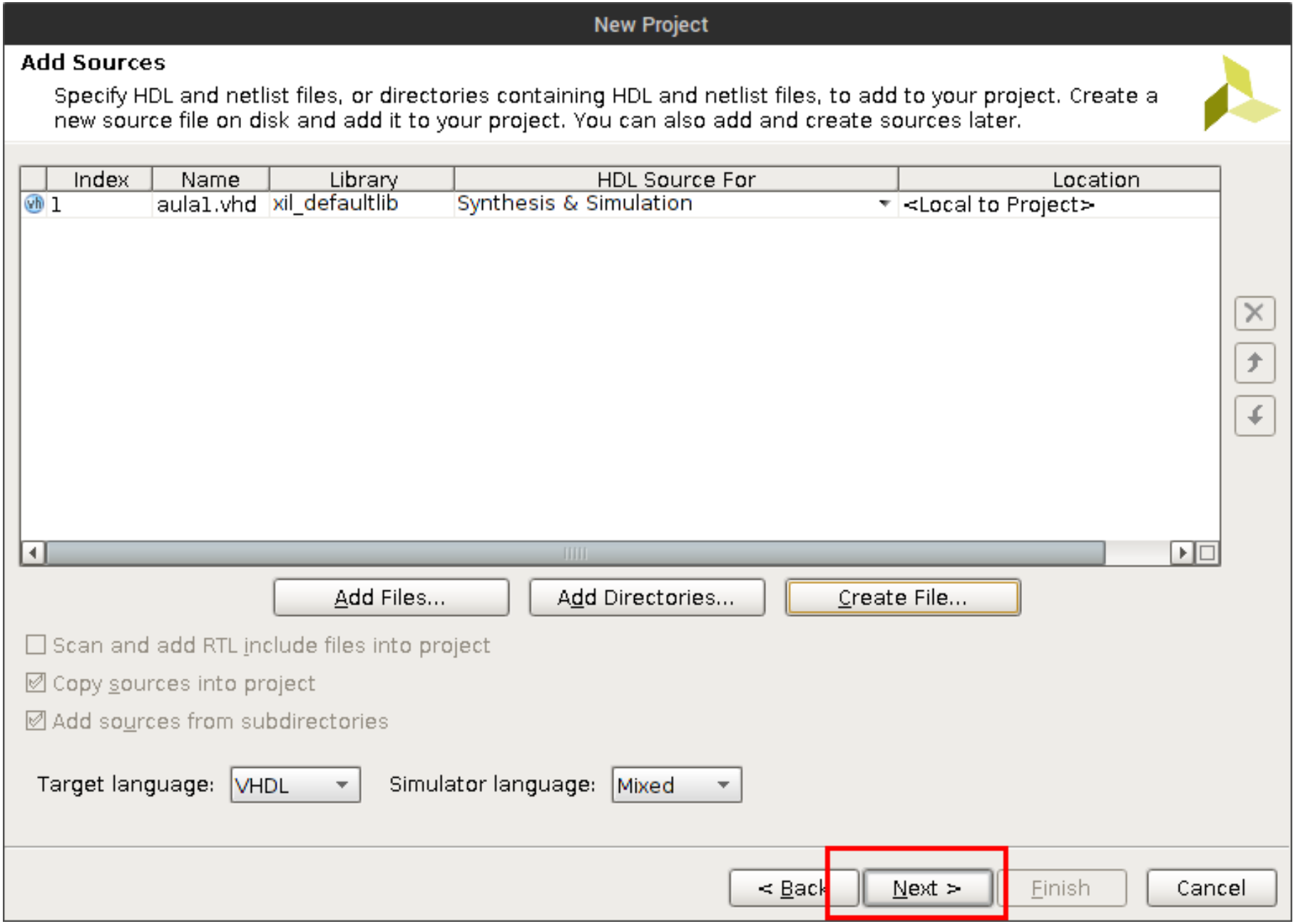
Adicionando arquivos fontes



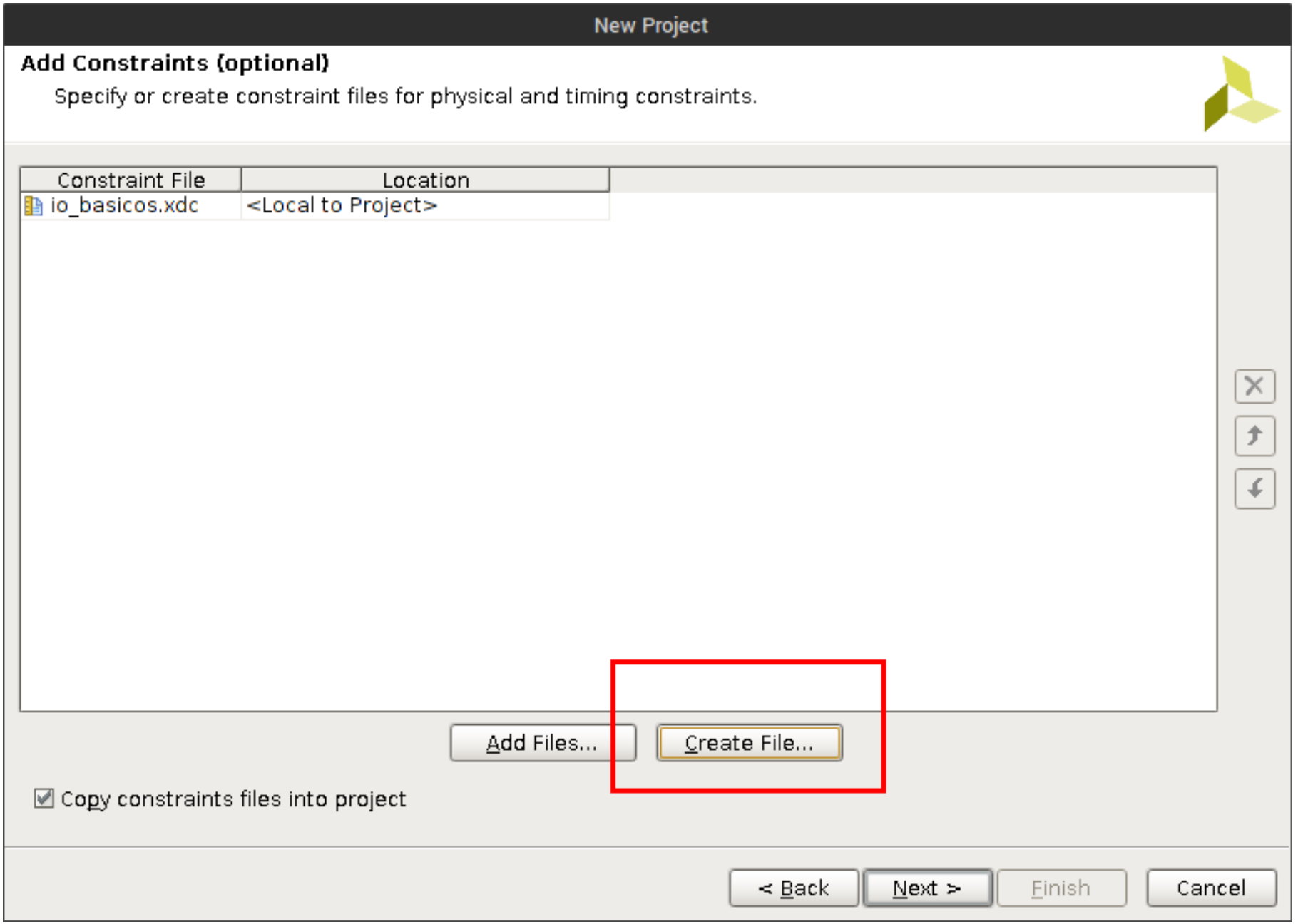
Nome e tipo do arquivo, repare nos diferentes tipos que podemos escolher.



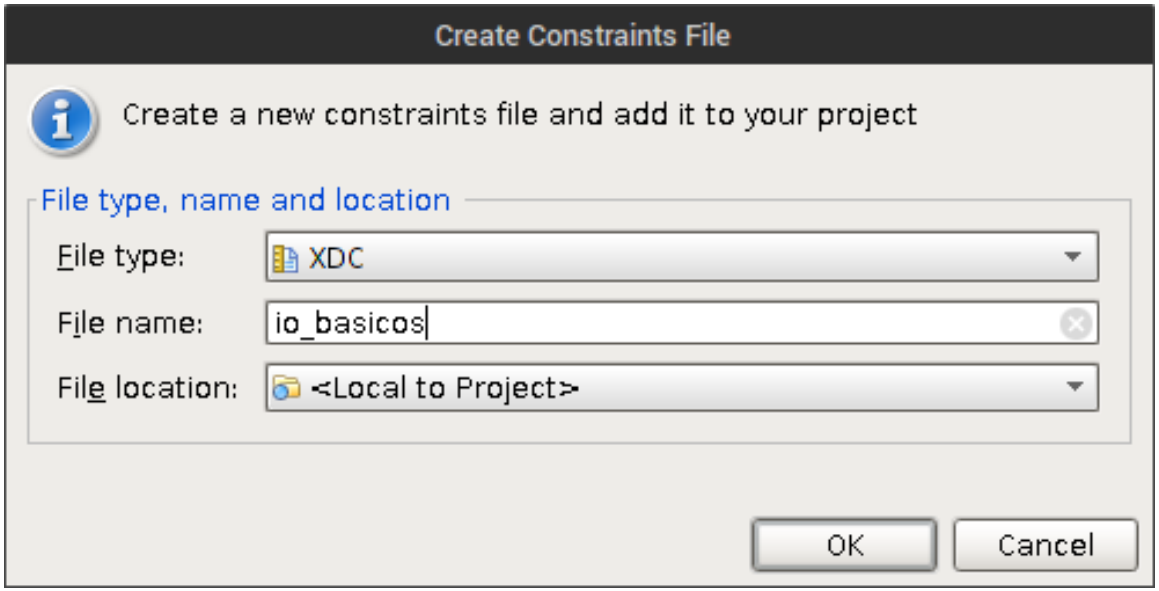
Verificamos que o novo arquivo foi automaticamente adicionado ao projeto



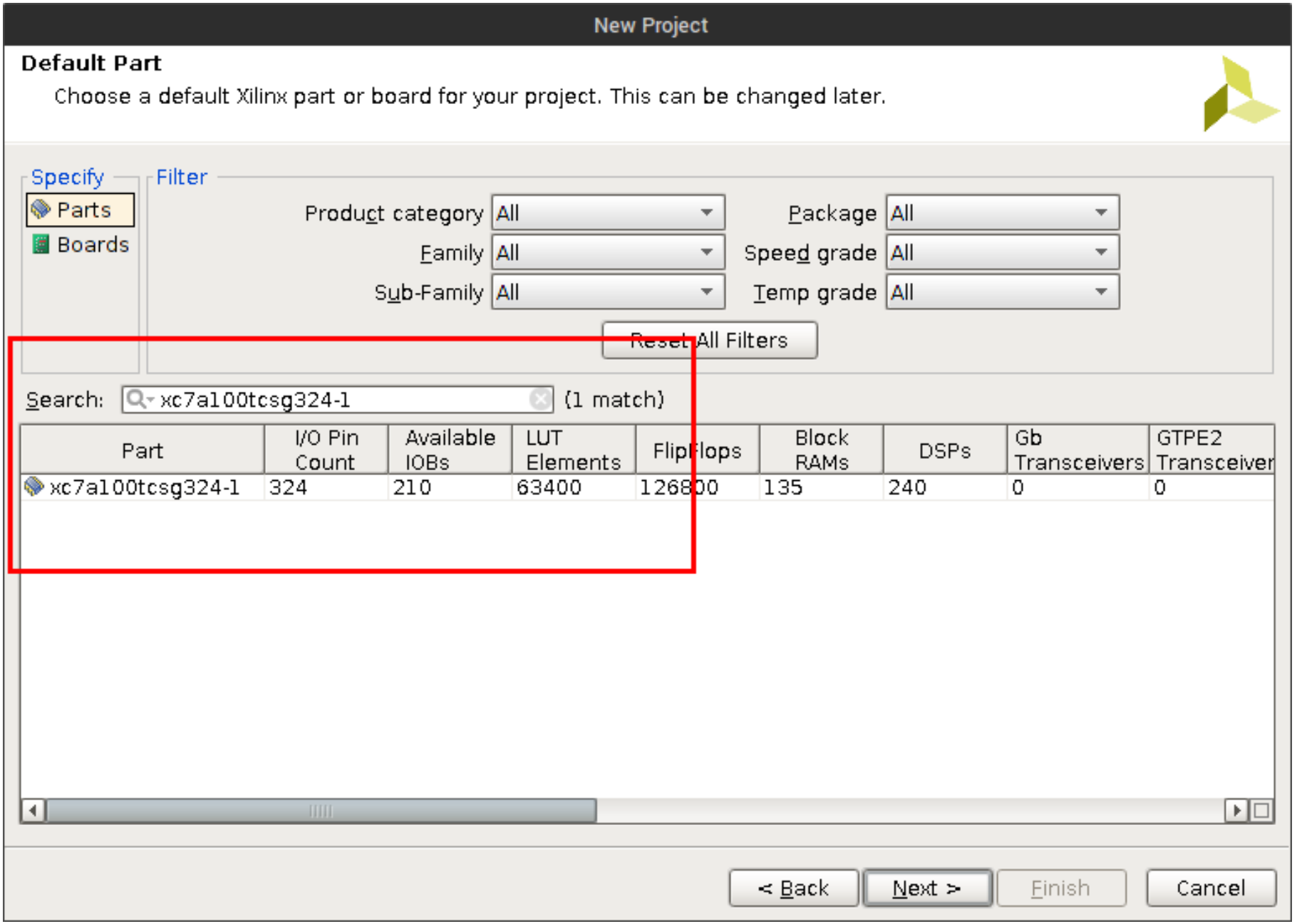
Adicionando arquivos de configuração (XCD)



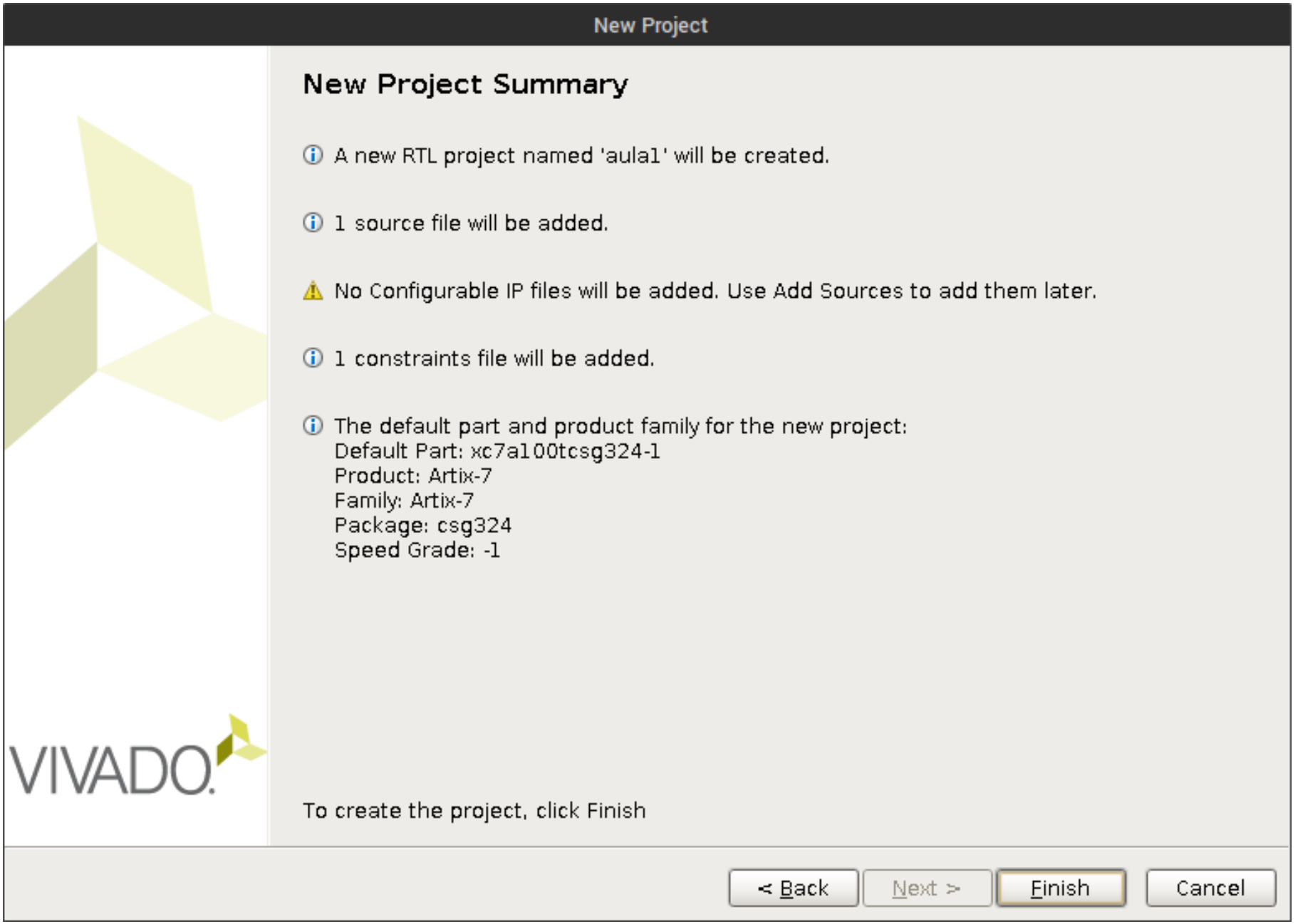
Nome do novo arquivo, repare nos tipos.



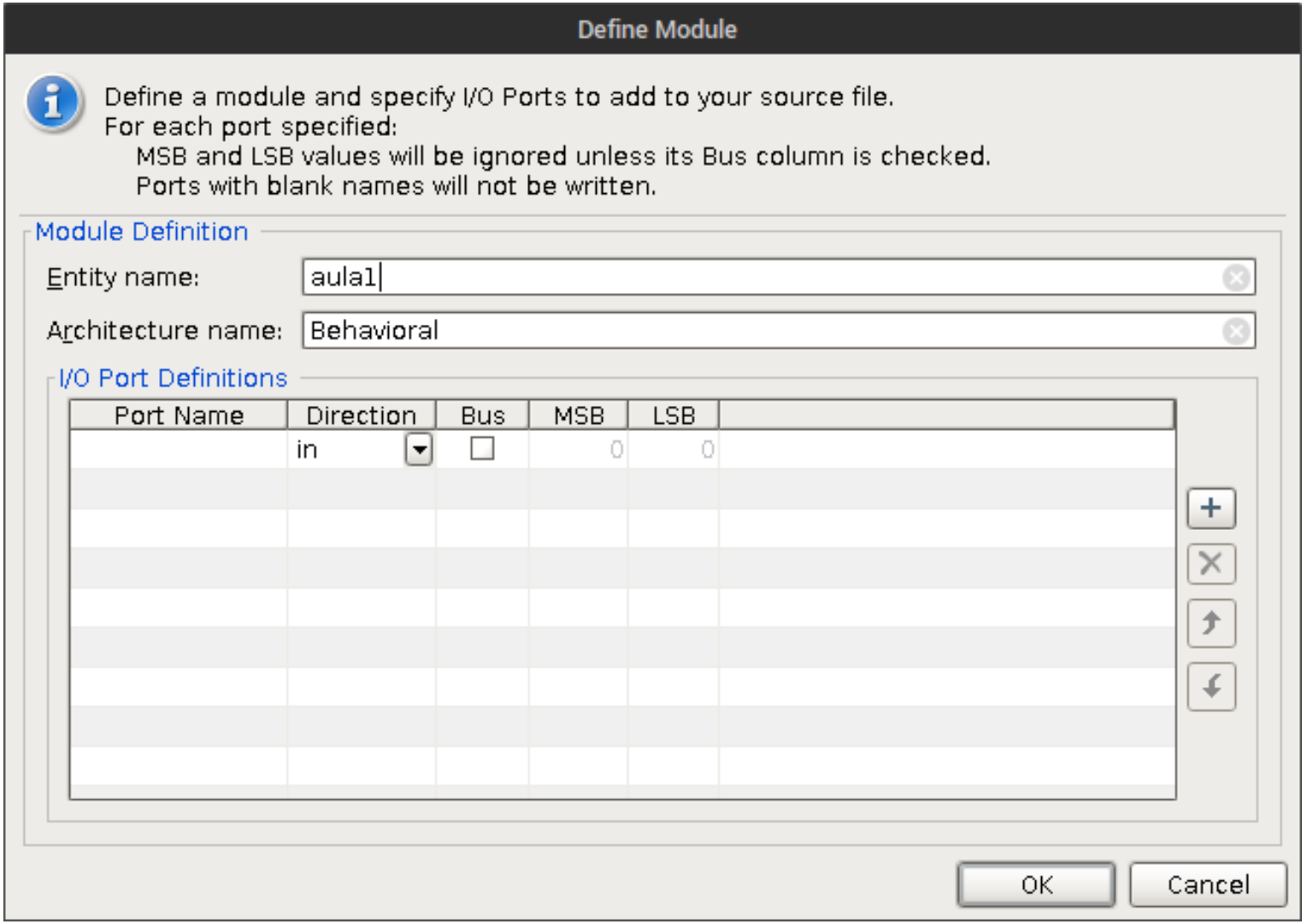
Selecionando a pastilha



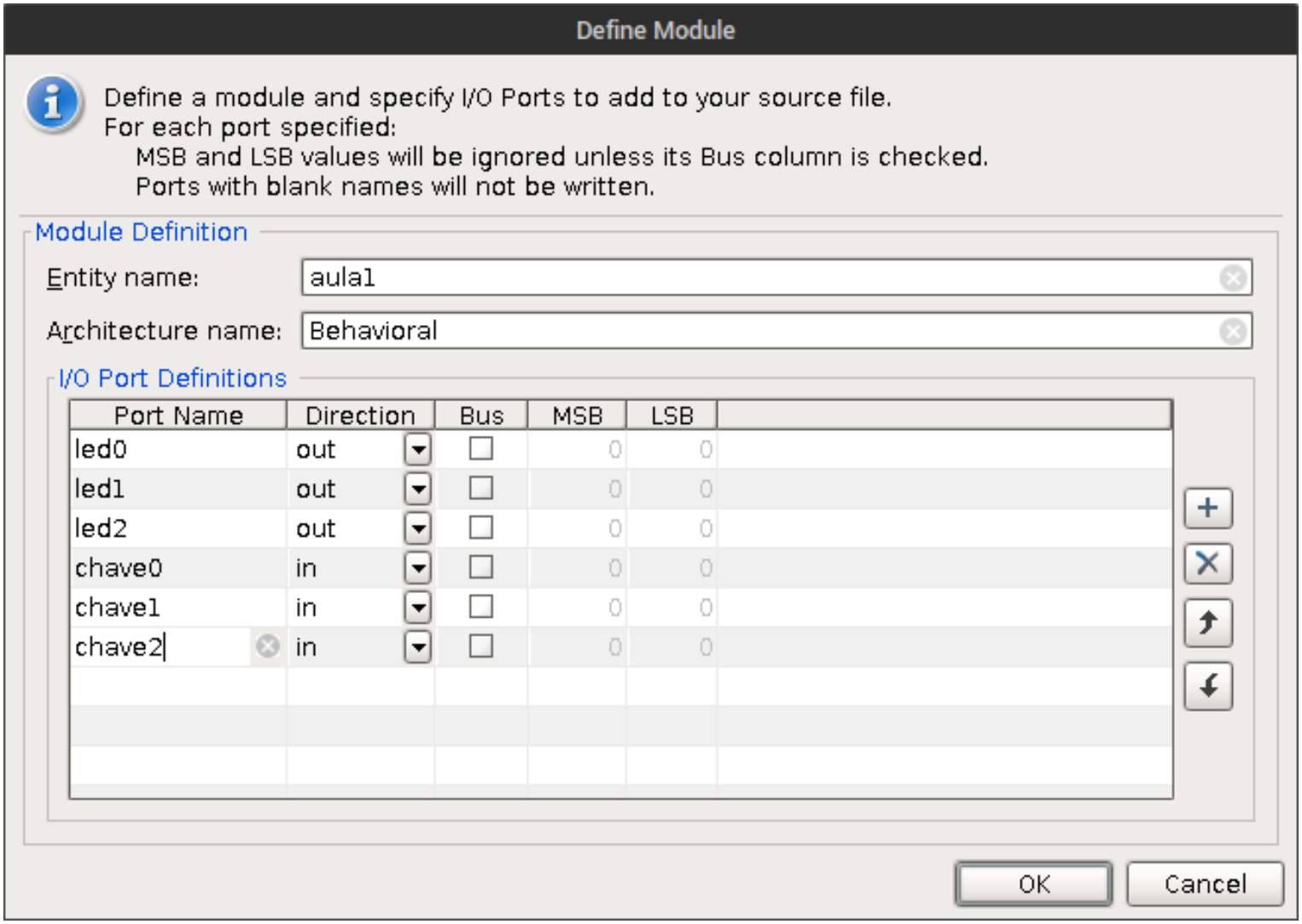
Resumo do projeto



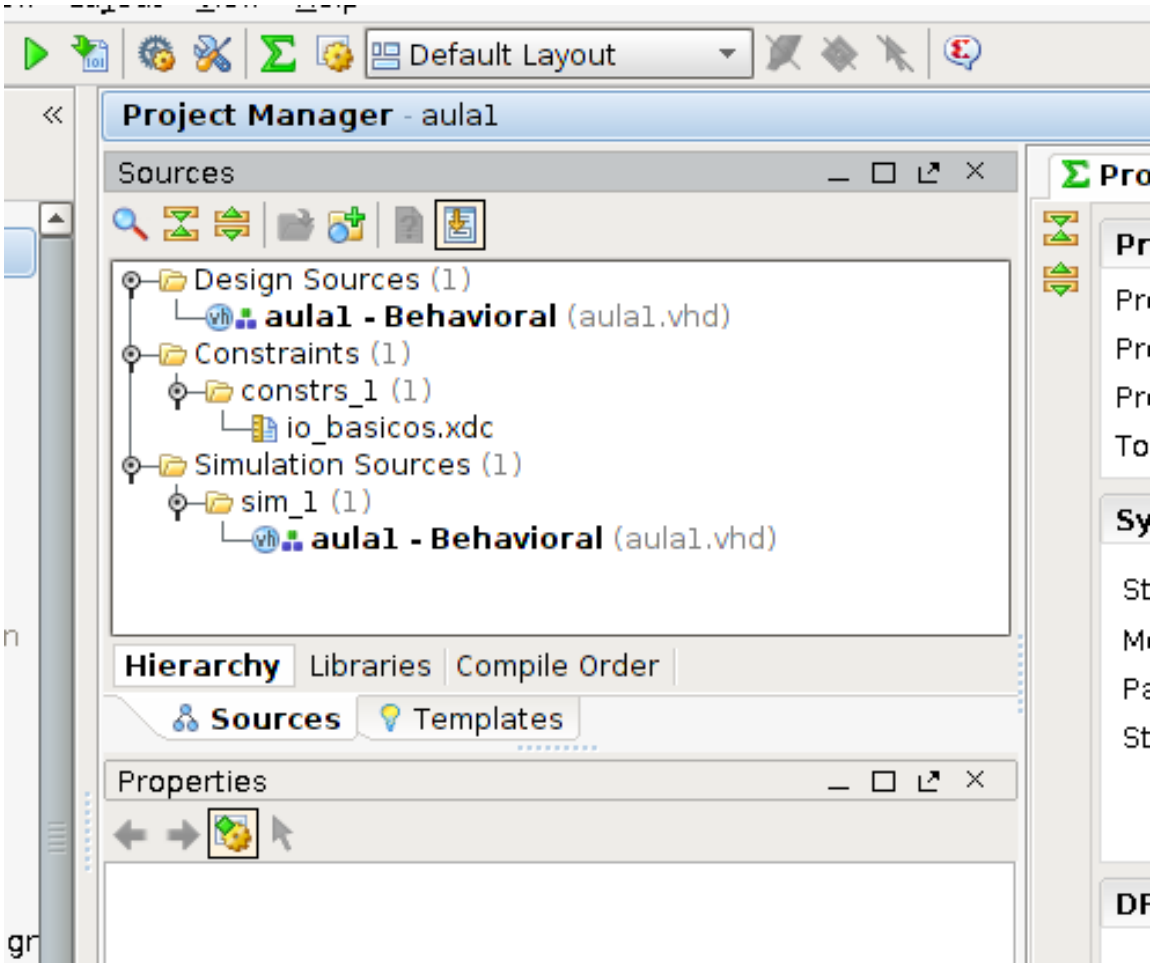
O Viado fornece uma ferramenta para inicio rápido de projetos, aqui podemos configurar as entradas e saídas.



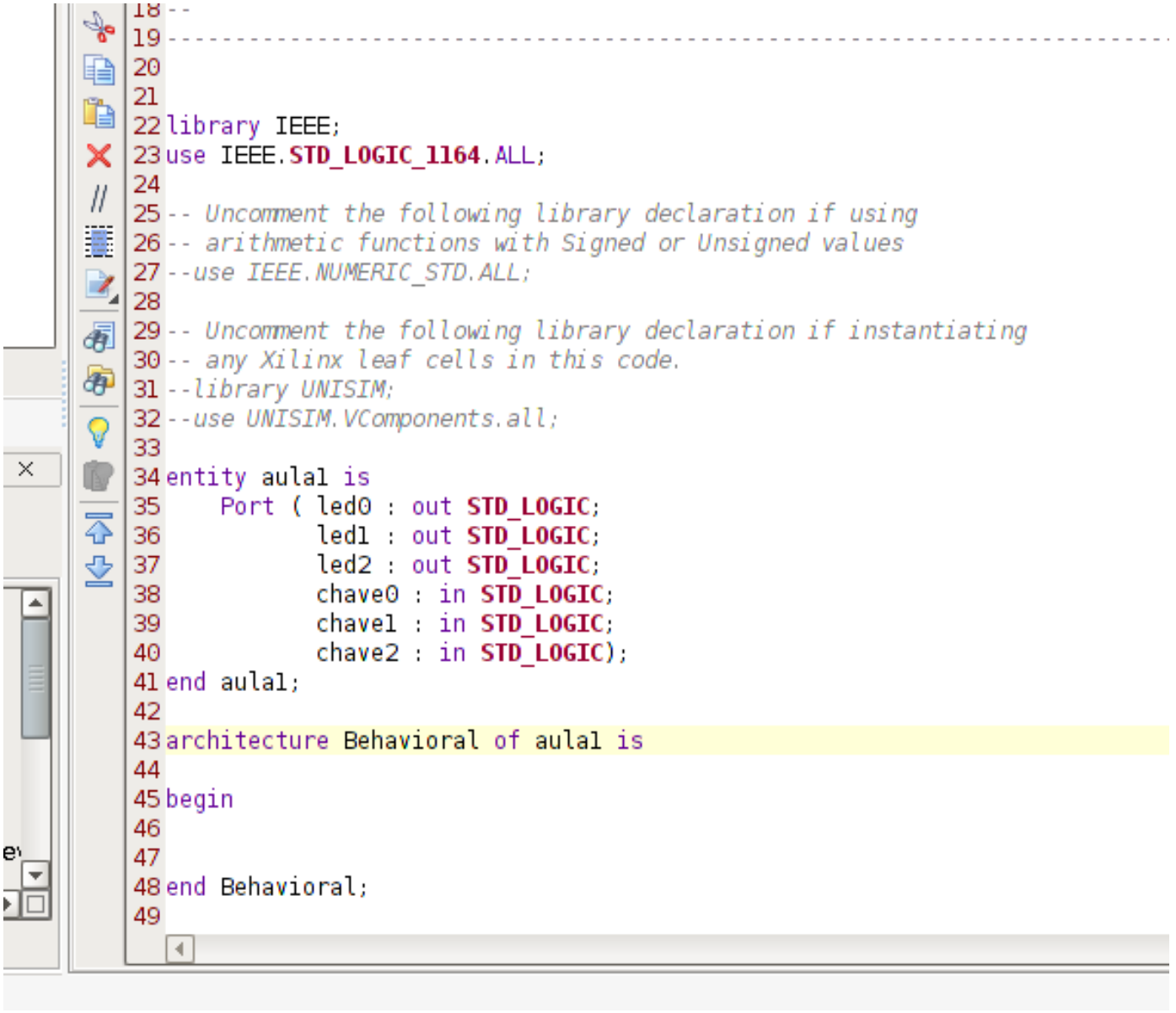
Definindo 3 entras e 3 saídas



Resumo dos arquivos adicionados ao projeto

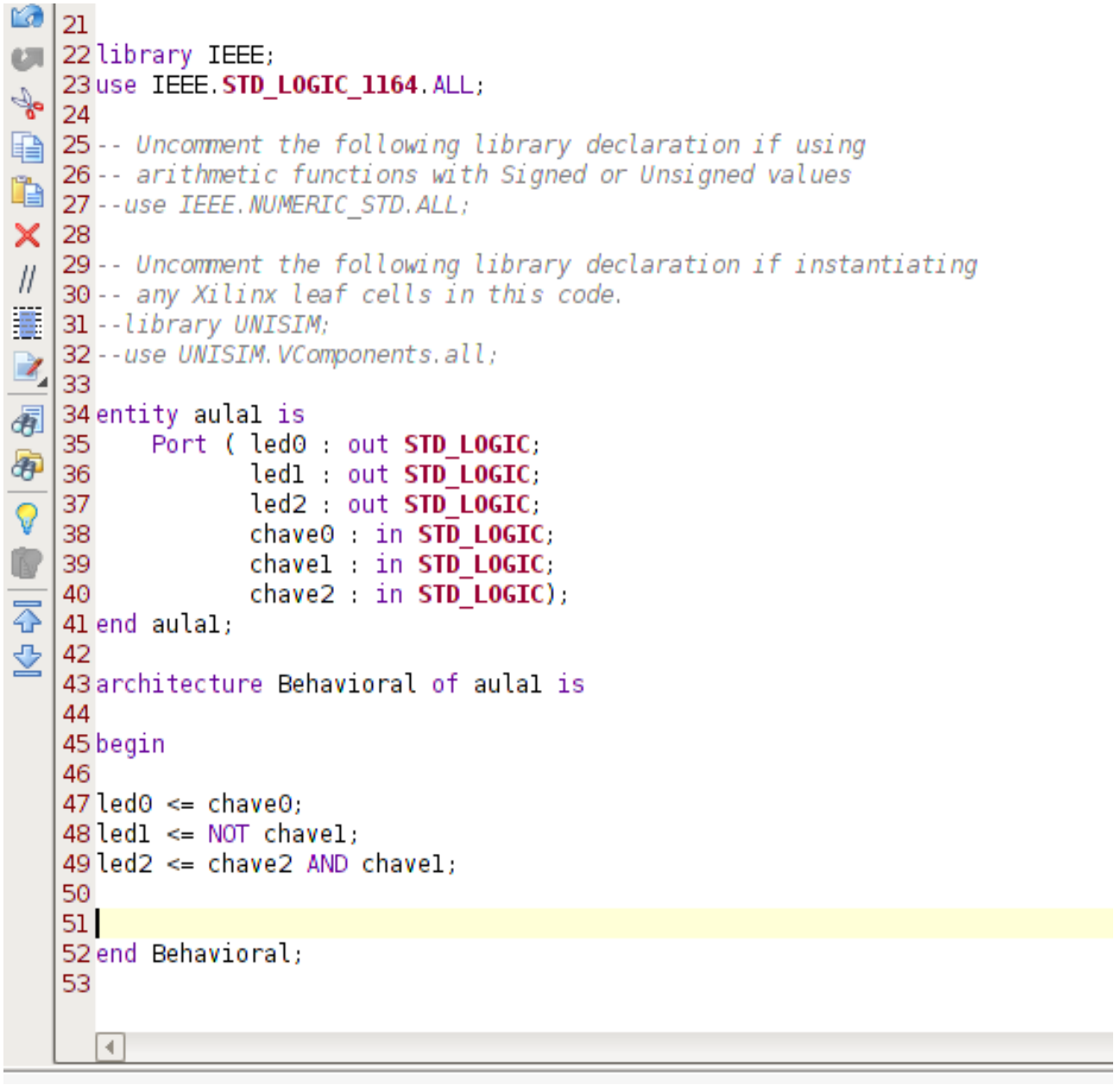


Vhdl criado pela ferramenta



```
18 --
19 ----------------------------------------------------------------------------------
20
21
22 library IEEE;
23 use IEEE.STD_LOGIC_1164.ALL;
24
25 -- Uncomment the following library declaration if using
26 -- arithmetic functions with Signed or Unsigned values
27 --use IEEE.NUMERIC_STD.ALL;
28
29 -- Uncomment the following library declaration if instantiating
30 -- any Xilinx leaf cells in this code.
31 --library UNISIM;
32 --use UNISIM.VComponents.all;
33
34 entity aula1 is
35     Port ( led0 : out STD_LOGIC;
36            led1 : out STD_LOGIC;
37            led2 : out STD_LOGIC;
38            chave0 : in STD_LOGIC;
39            chave1 : in STD_LOGIC;
40            chave2 : in STD_LOGIC);
41 end aula1;
42
43 architecture Behavioral of aula1 is
44
45 begin
46
47
48 end Behavioral;
49
```

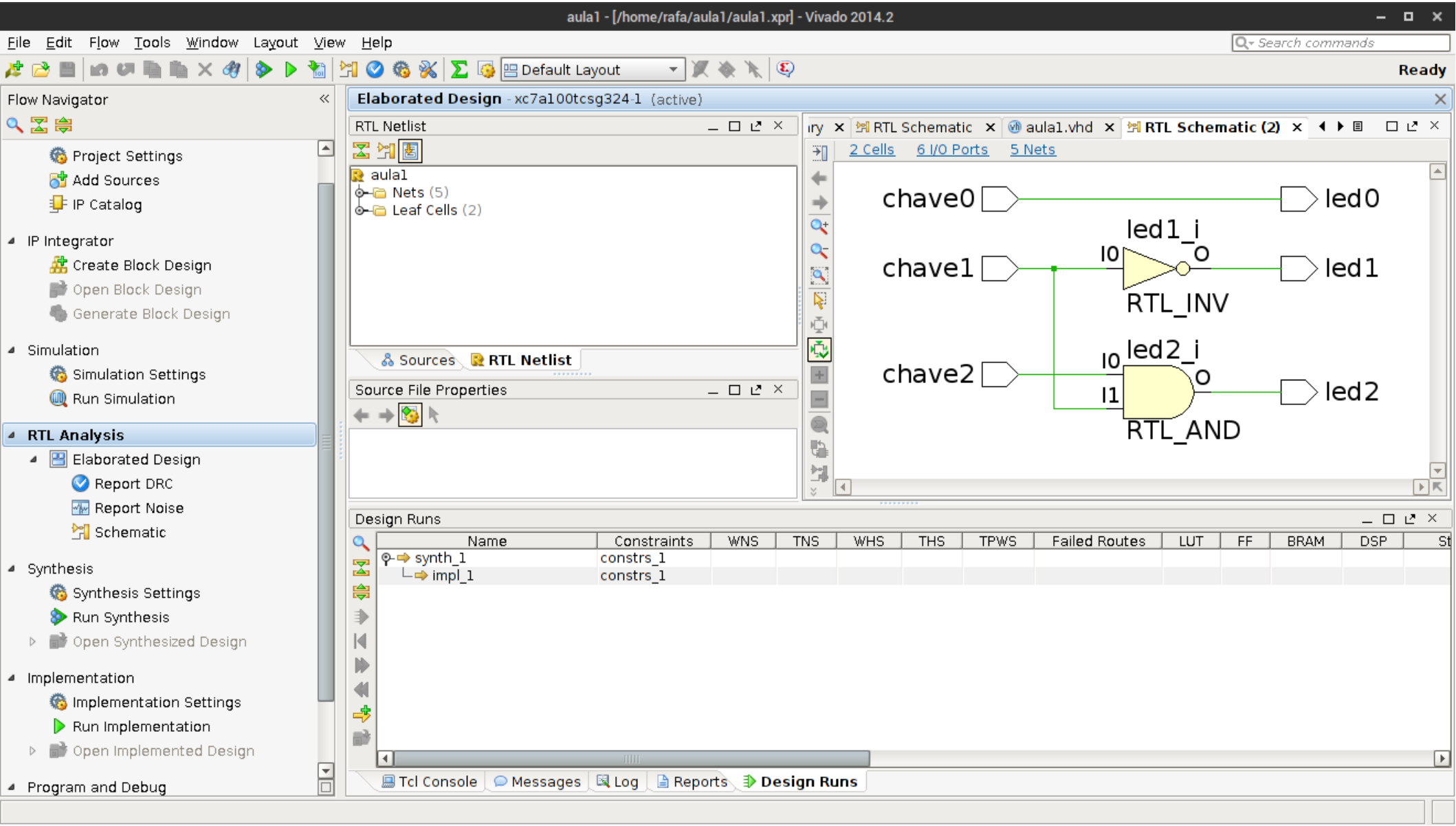
Modificação proposta



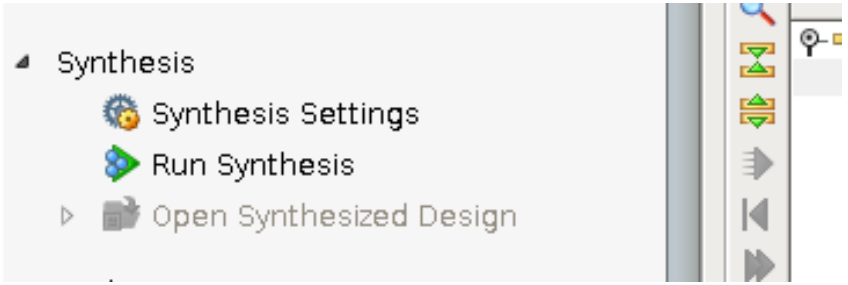
```
21
22 library IEEE;
23 use IEEE.STD_LOGIC_1164.ALL;
24
25 -- Uncomment the following library declaration if using
26 -- arithmetic functions with Signed or Unsigned values
27 --use IEEE.NUMERIC_STD.ALL;
28
29 -- Uncomment the following library declaration if instantiating
30 -- any Xilinx leaf cells in this code.
31 --library UNISIM;
32 --use UNISIM.VComponents.all;
33
34 entity aula1 is
35     Port ( led0 : out STD_LOGIC;
36            led1 : out STD_LOGIC;
37            led2 : out STD_LOGIC;
38            chave0 : in STD_LOGIC;
39            chave1 : in STD_LOGIC;
40            chave2 : in STD_LOGIC);
41 end aula1;
42
43 architecture Behavioral of aula1 is
44
45 begin
46
47 led0 <= chave0;
48 led1 <= NOT chave1;
49 led2 <= chave2 AND chave1;
50
51
52 end Behavioral;
53
```

Vamos verificar agora o RTL desse projeto, rode o “schematic” no “RTL Analysis”

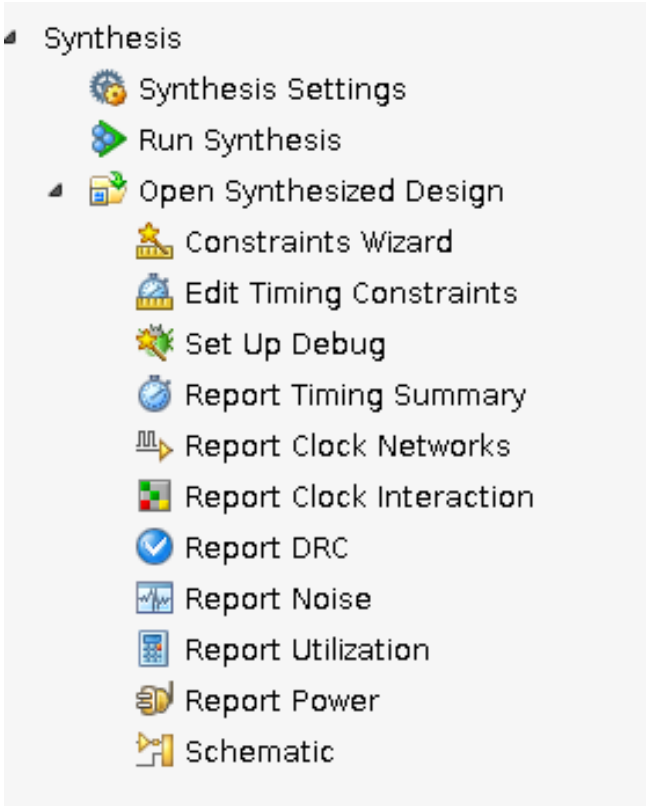
Verificamos a lógica interpretada pela ferramenta



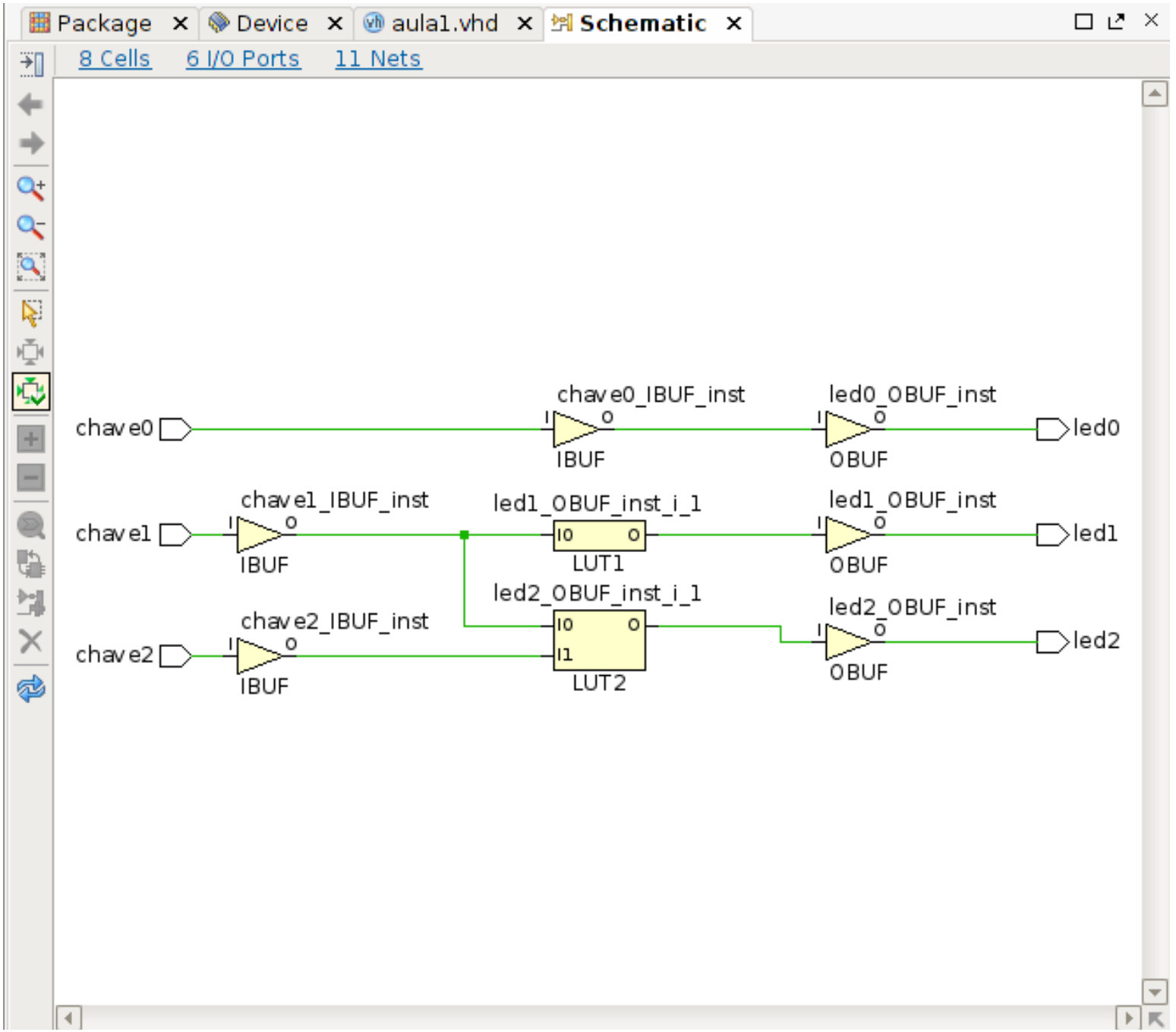
Agora podemos executar a “Synthesis” → run Synthesis.



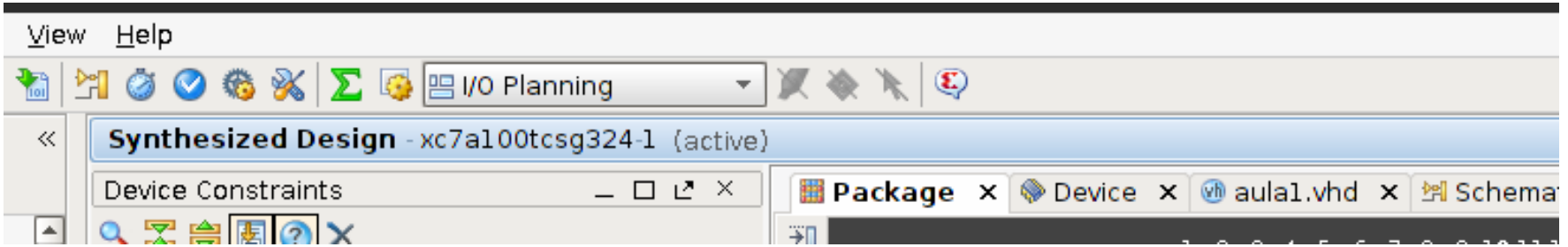
Verificamos agora o Schematic da synthesis



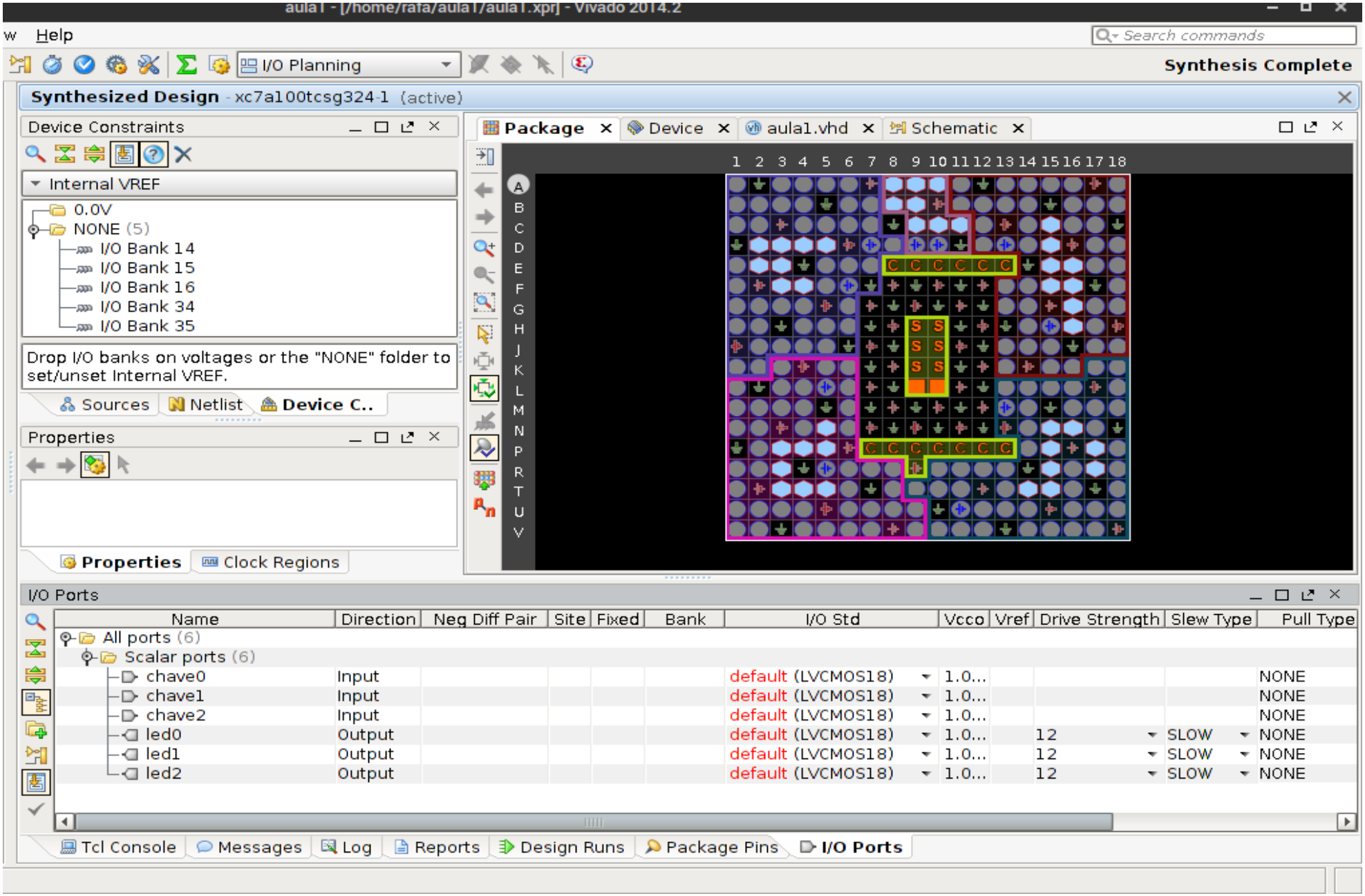
Aqui verificamos como a lógica RTL será implementada



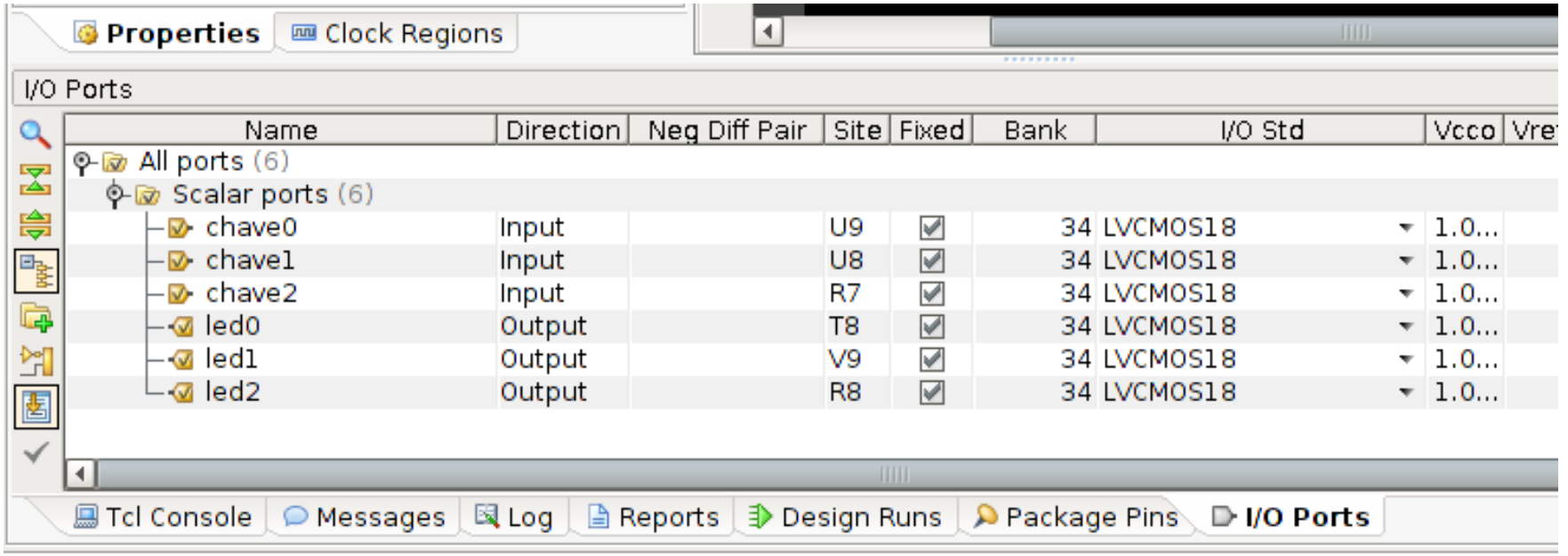
Precisamos definir os pinos e tipos de sinais de cada porta da entidade aula1, mude o layout da ferramenta para “I/O Planning”



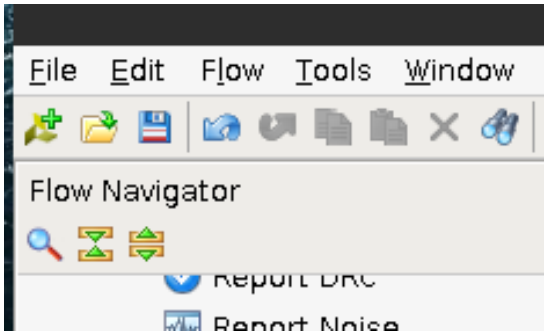
Devemos verificar essa janela no I/O Planning. Note que as portas não estão mapeadas



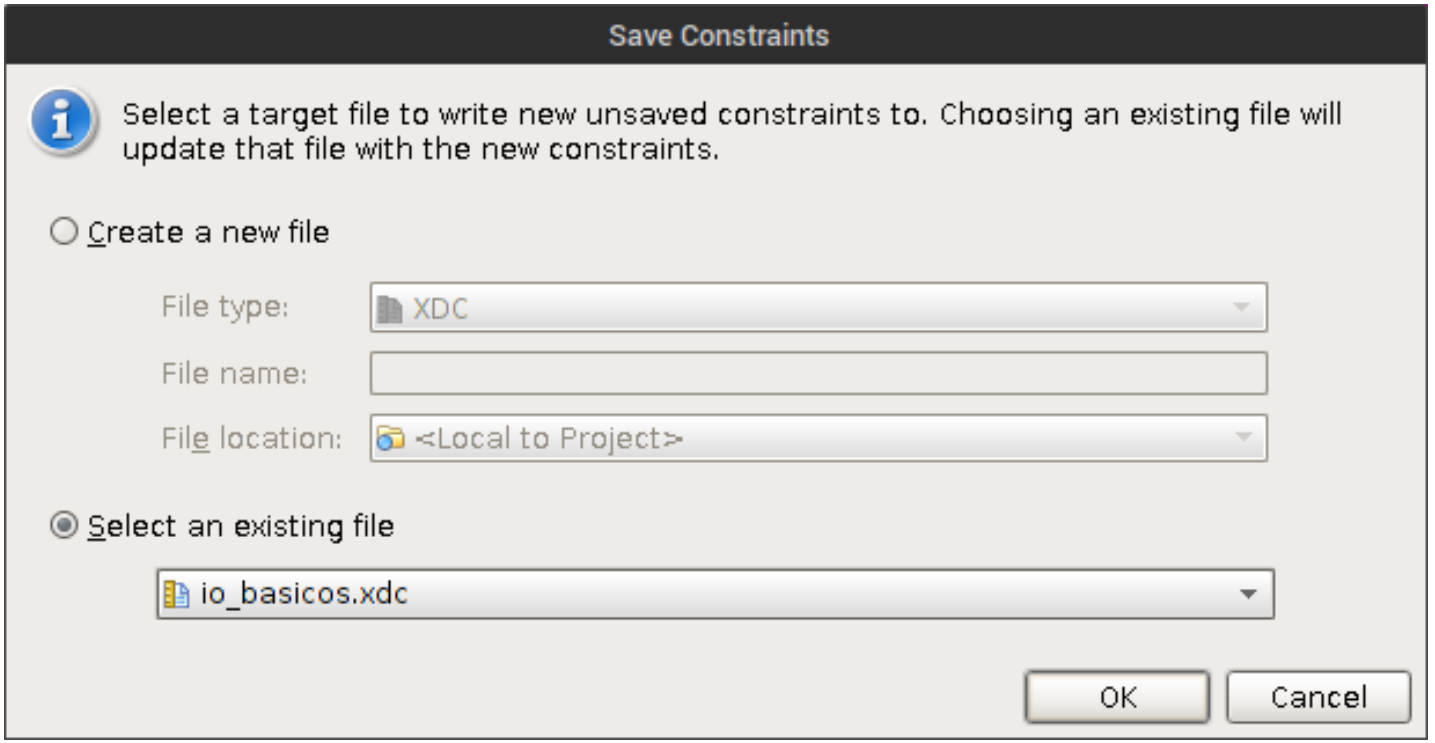
Modifique conforme exemplo abaixo:



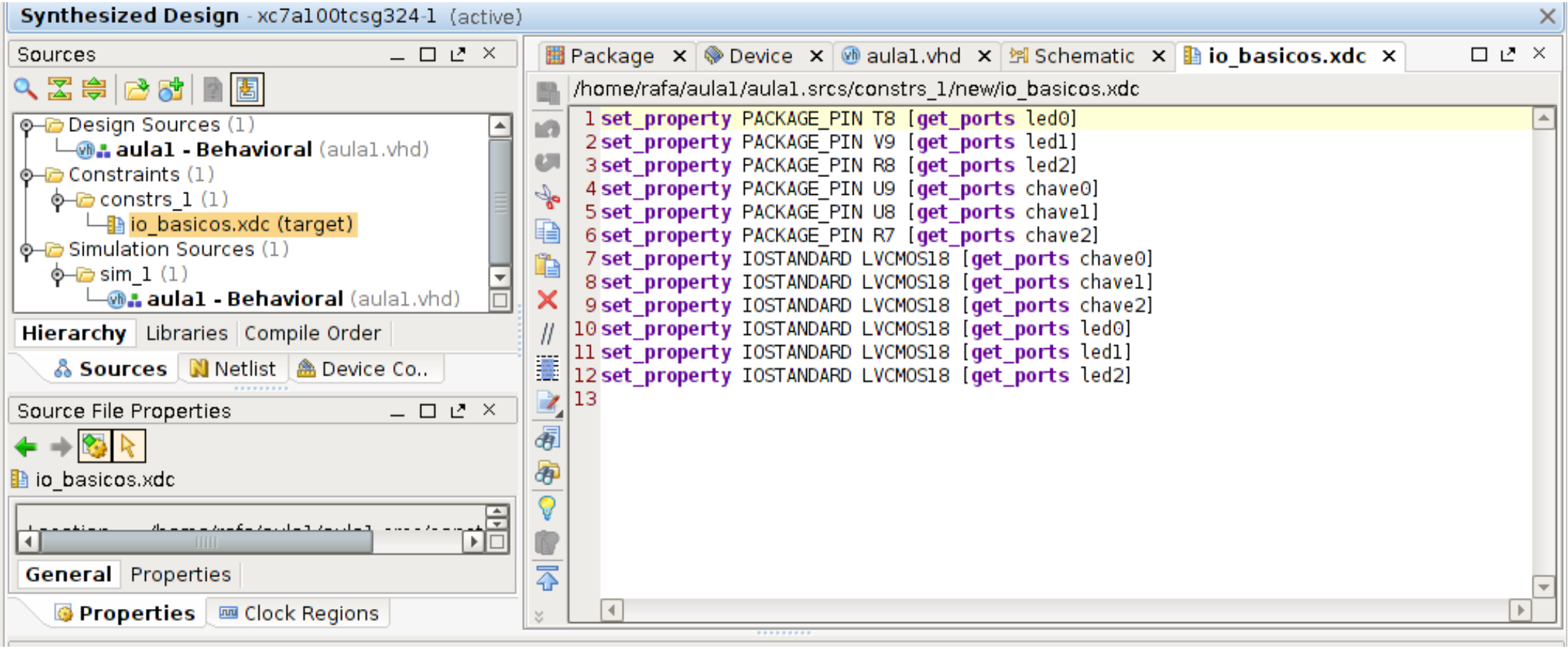
Após a modificação, salve as alterações



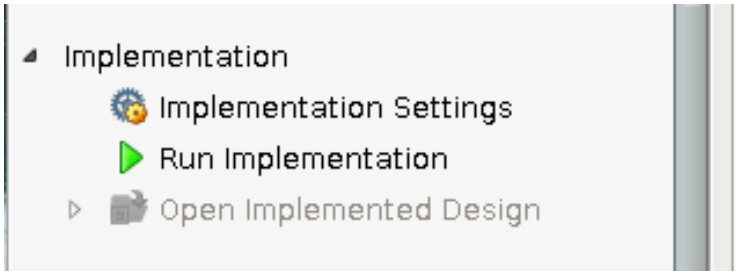
Deve aparecer essa janela, selecione o .xdc criado no inicio do projeto



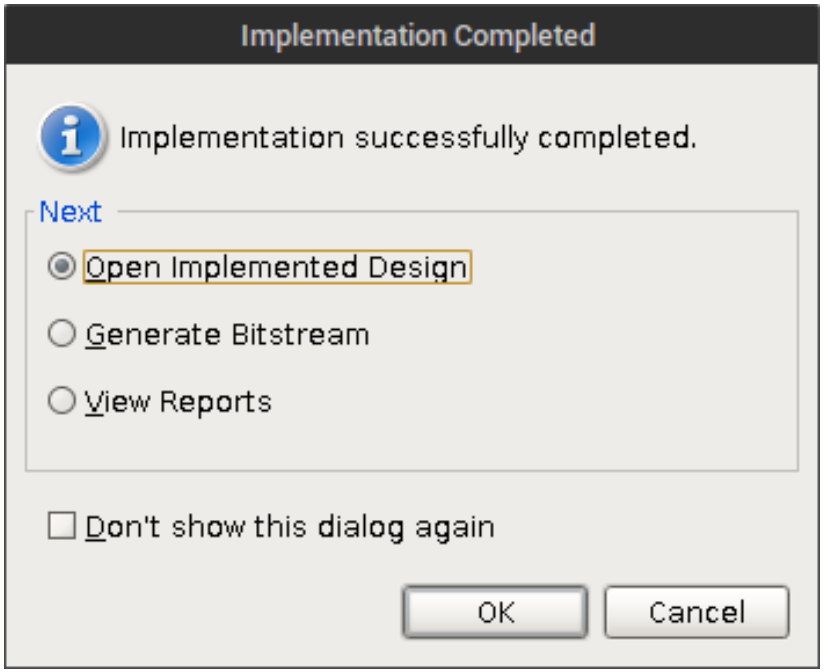
O .xdc foi alterado pela ferramenta, note que poderíamos ter editado o .XDC sem usar a ferramenta gráfica.



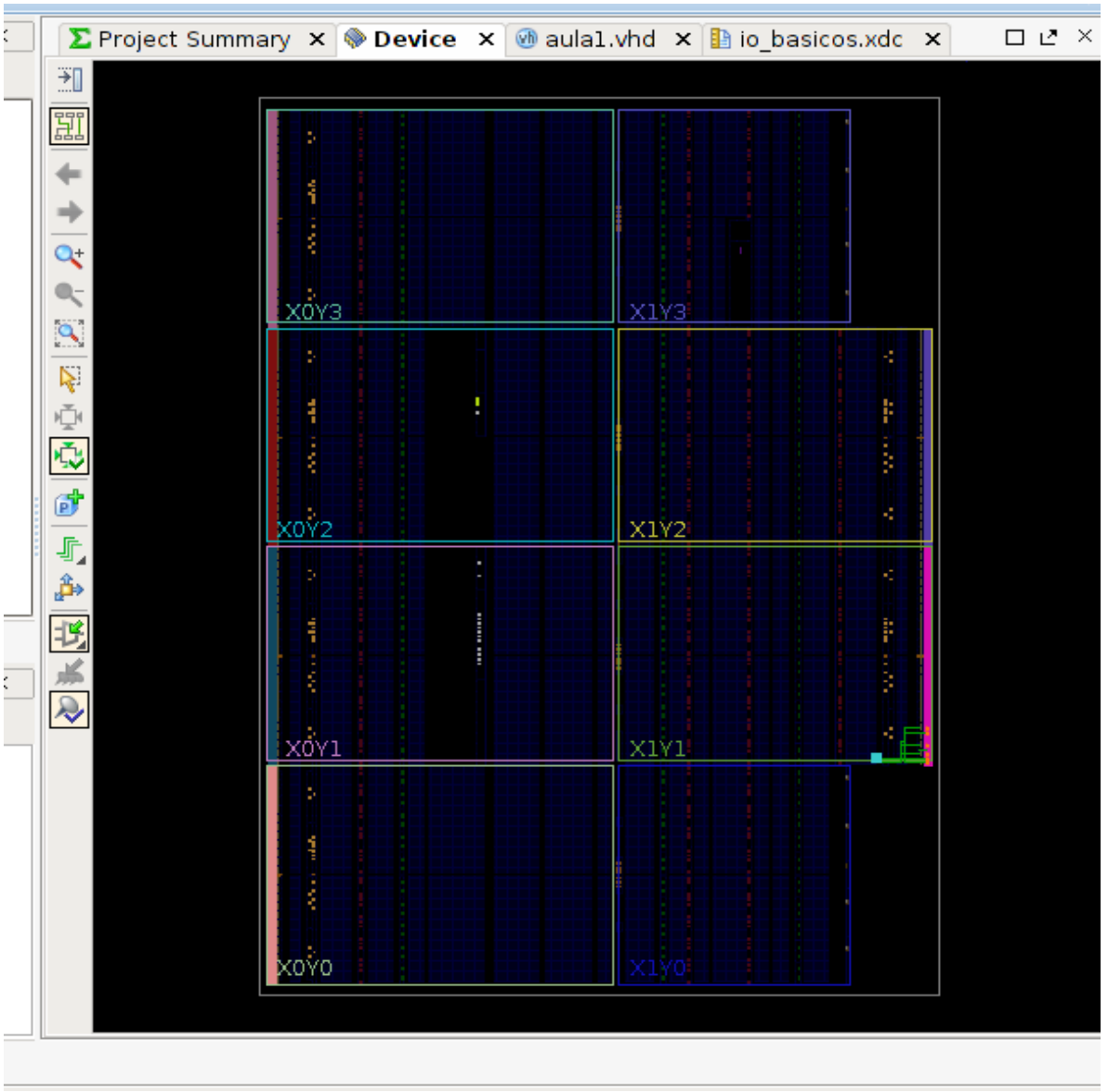
Execute a implementação



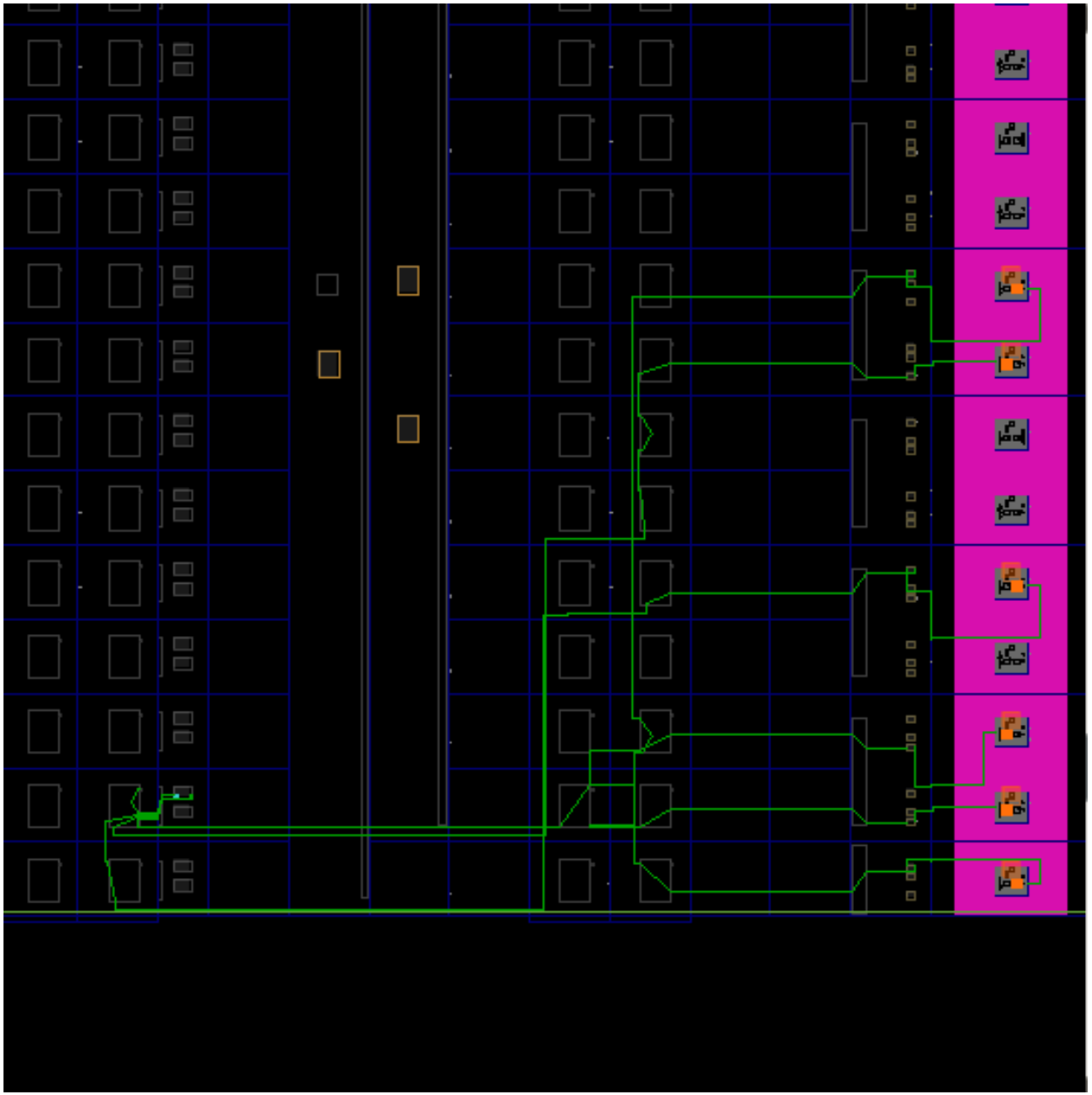
E abra o seu resultado



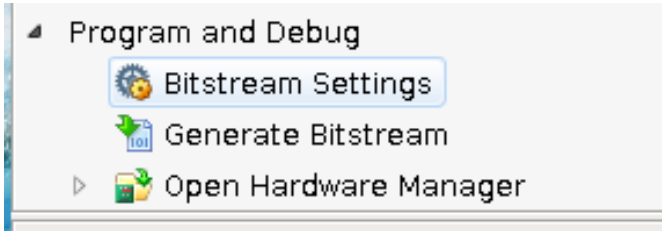
Aqui verificamos todas as ligações internas realizadas para implementar o projeto.



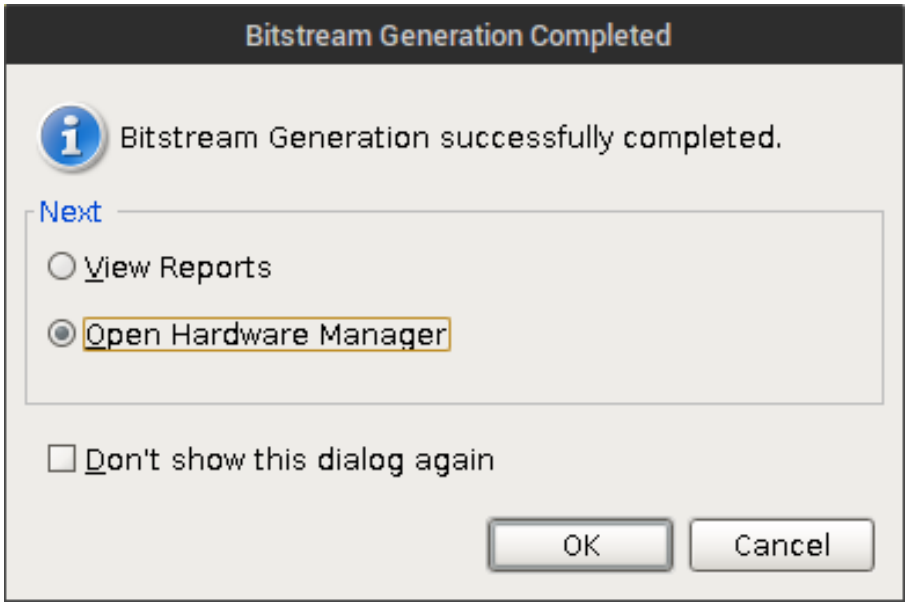
Podemos verificar as ligações e os I/Os



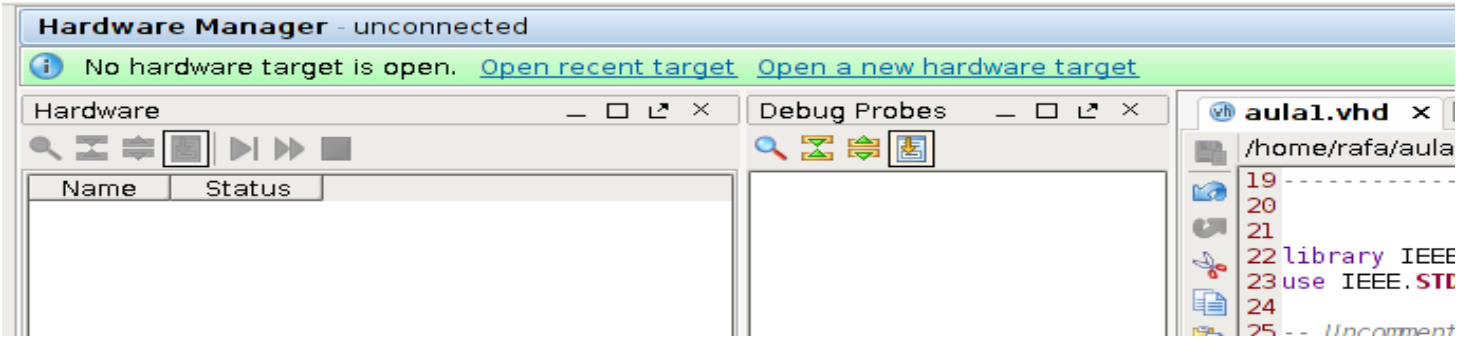
Agora vamos gravar o projeto na FPGA, antes é preciso gerar o arquivo de gravação, chamado de Bitstream:

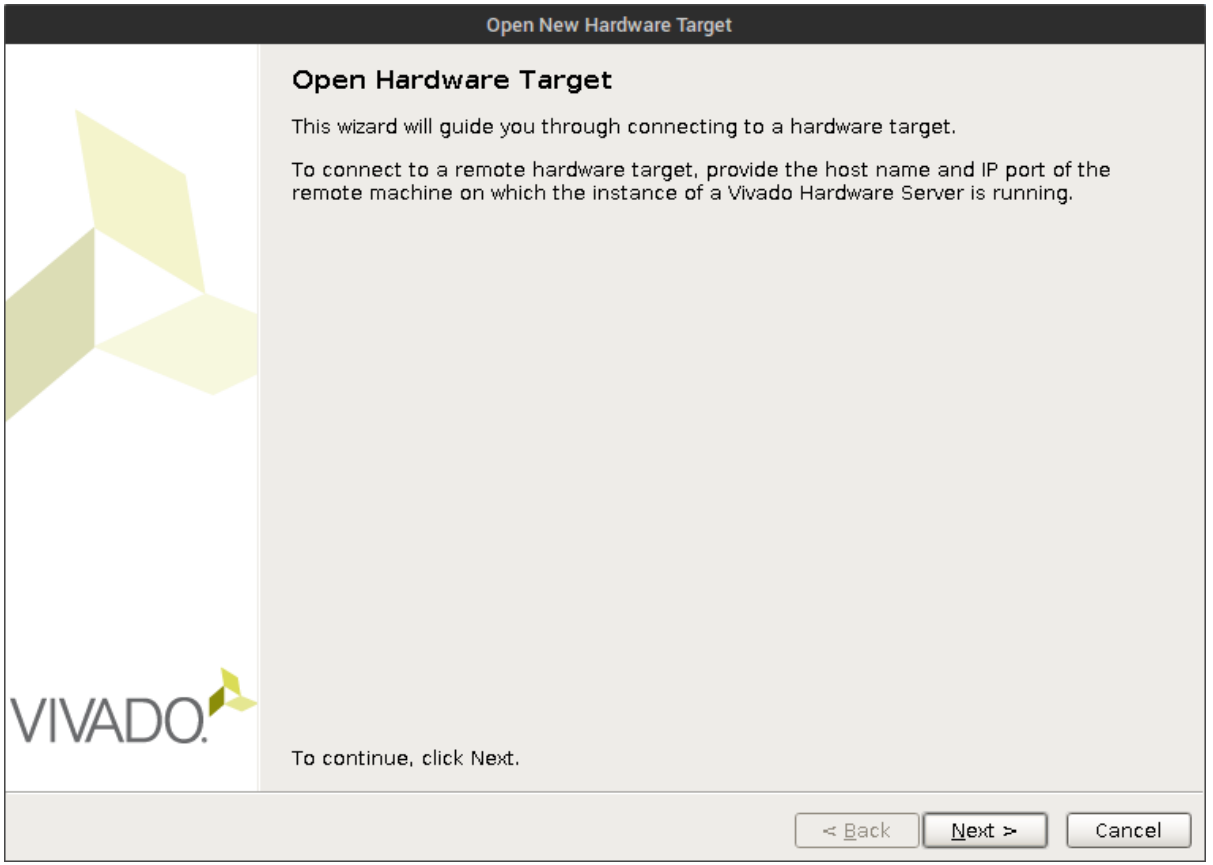


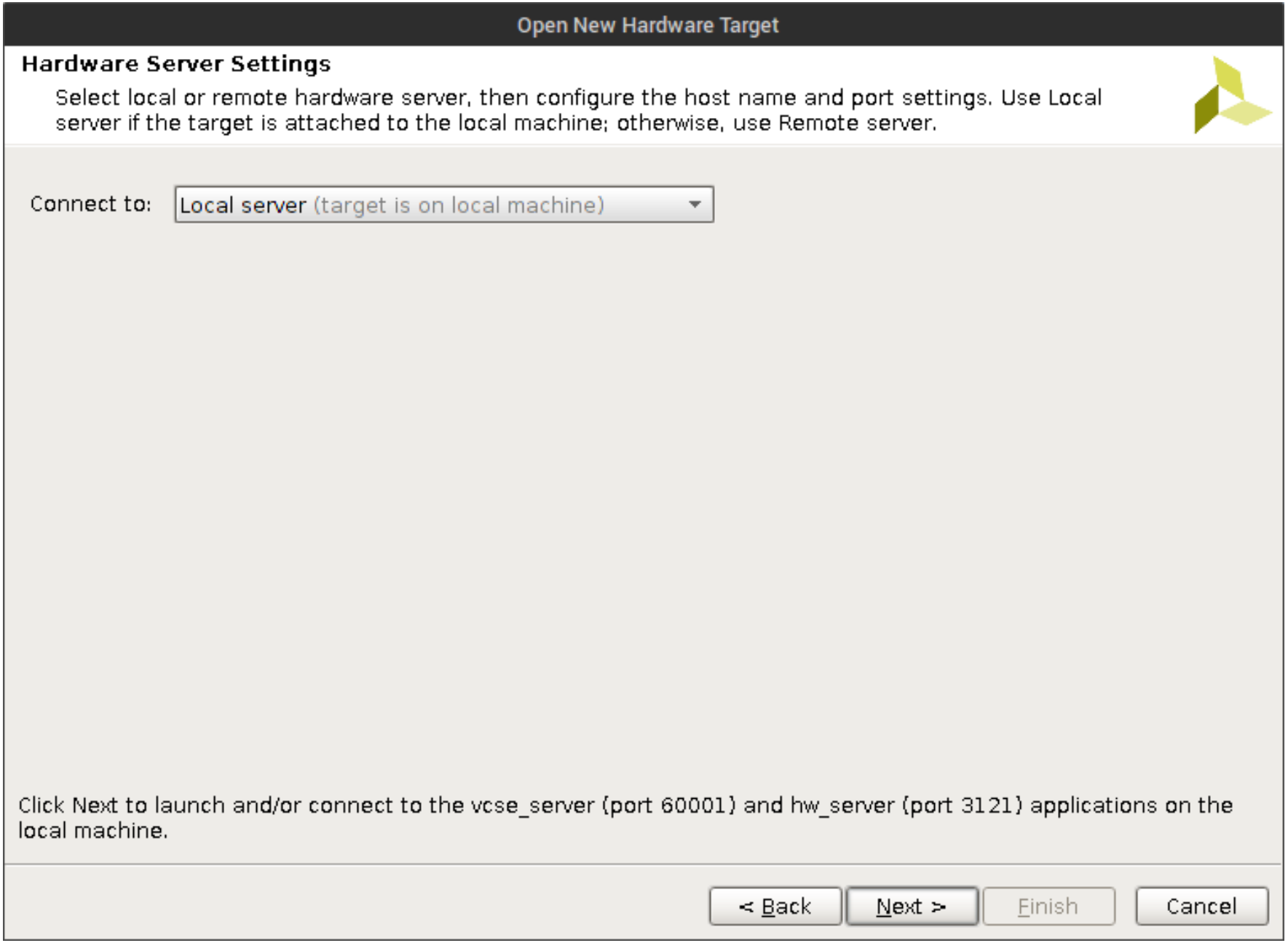
E abriremos a interface de gravação

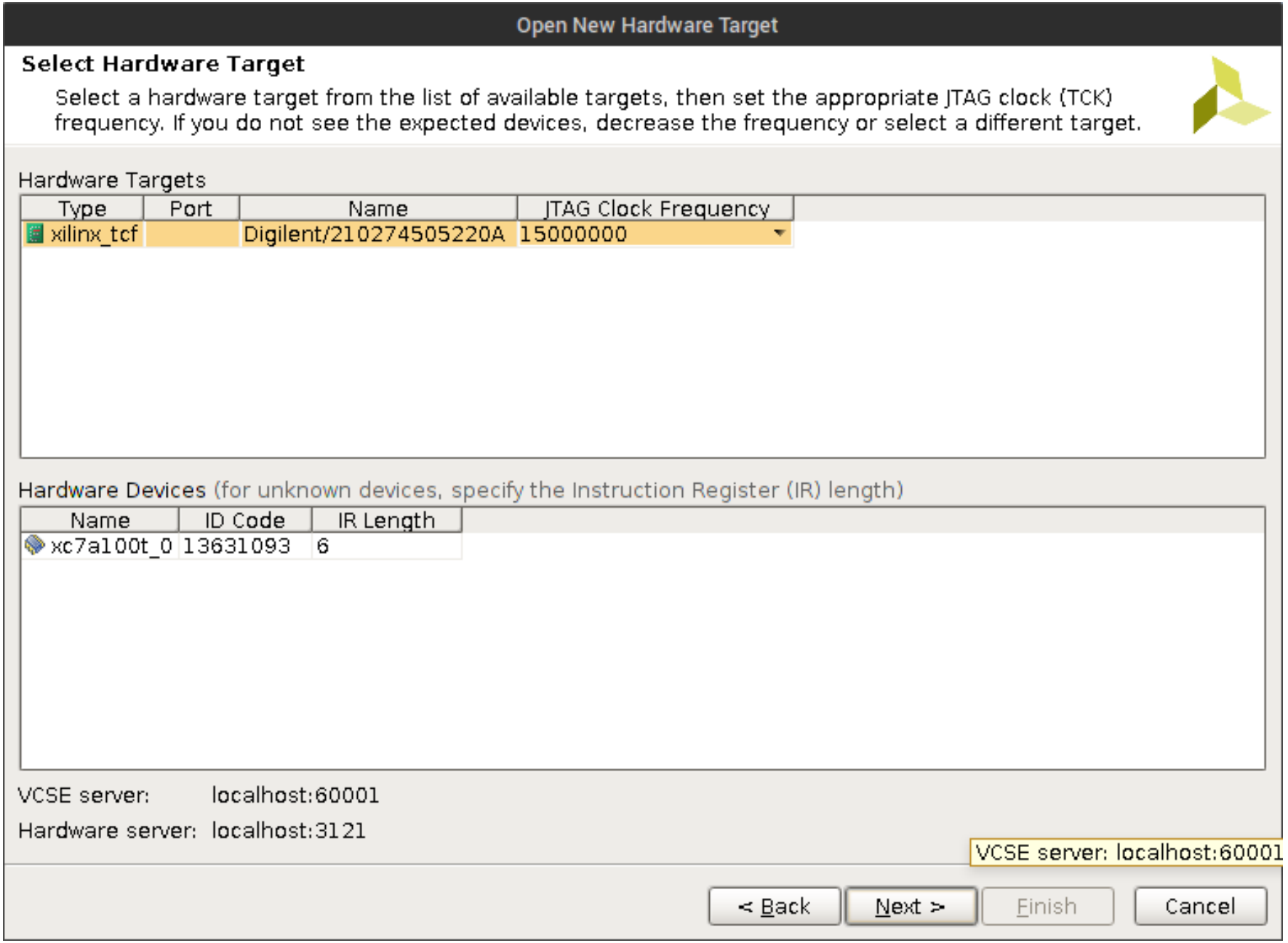


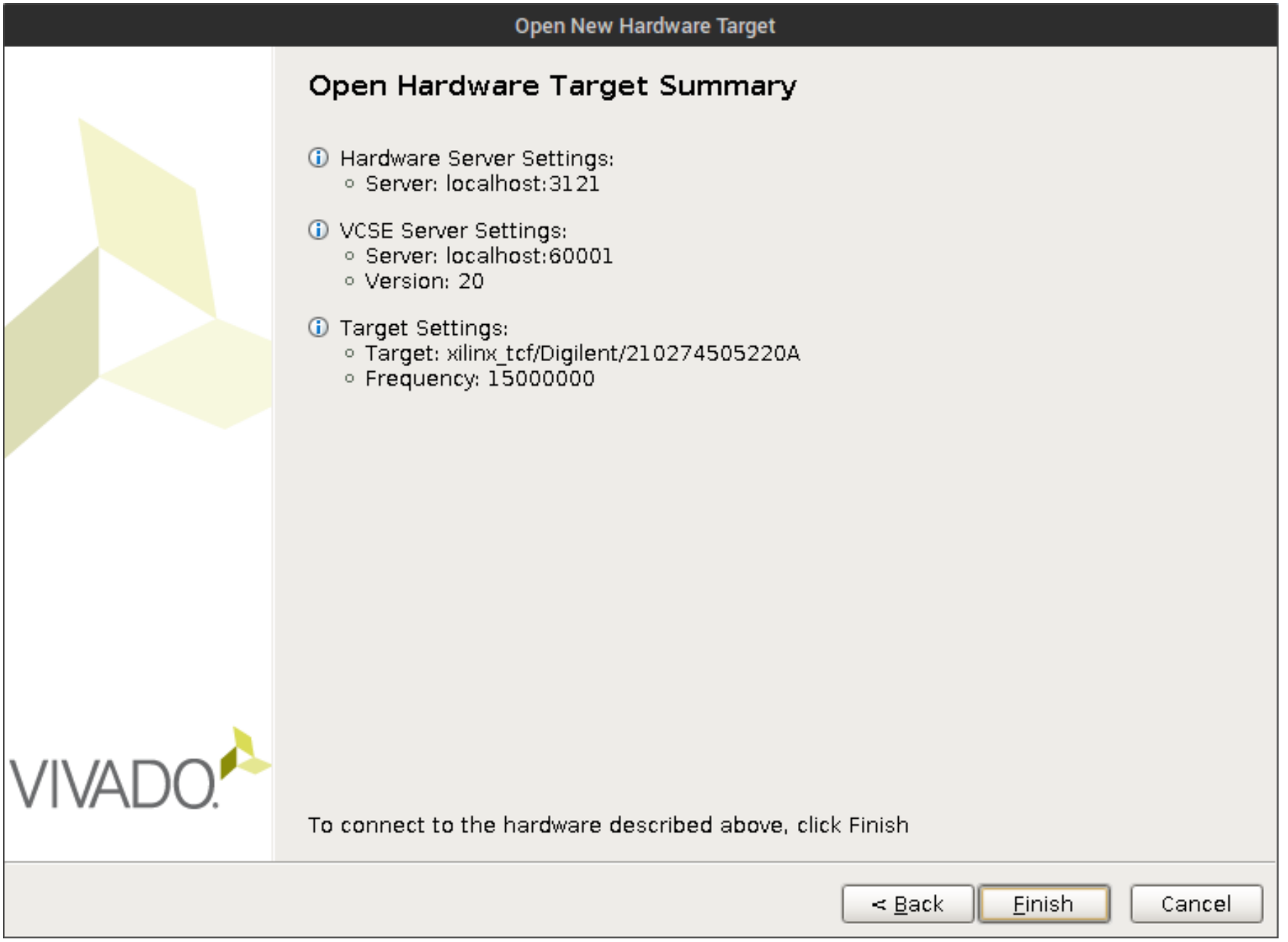
Na primeira vez é preciso configurar o hardware, → Open a new hardware target



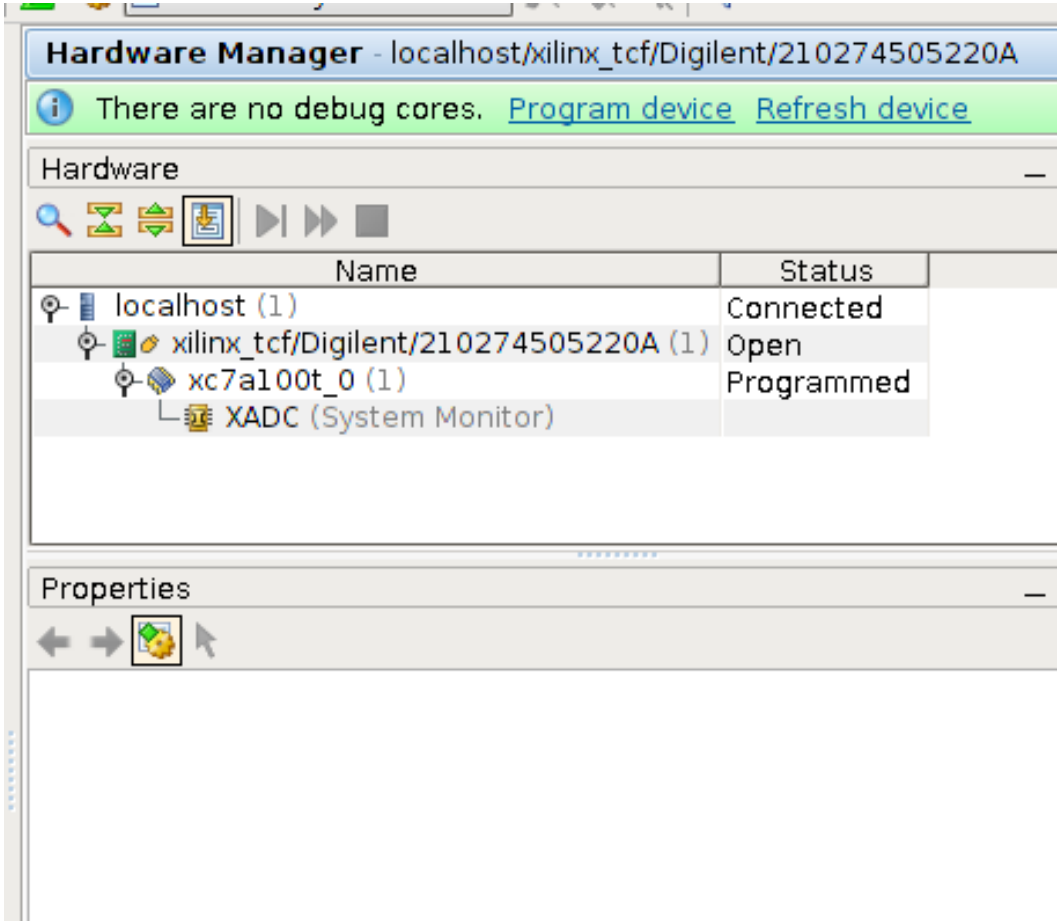



Open New Hardware Target
Hardware Server Settings
Select local or remote hardware server, then configure the host name and port settings. Use Local server if the target is attached to the local machine; otherwise, use Remote server.
Connect to:
Local server (target is on local machine)
Click Next to launch and/or connect to the vcse_server (port 60001) and hw_server (port 3121) applications on the local machine.
< Back
Next >
Finish
Cancel

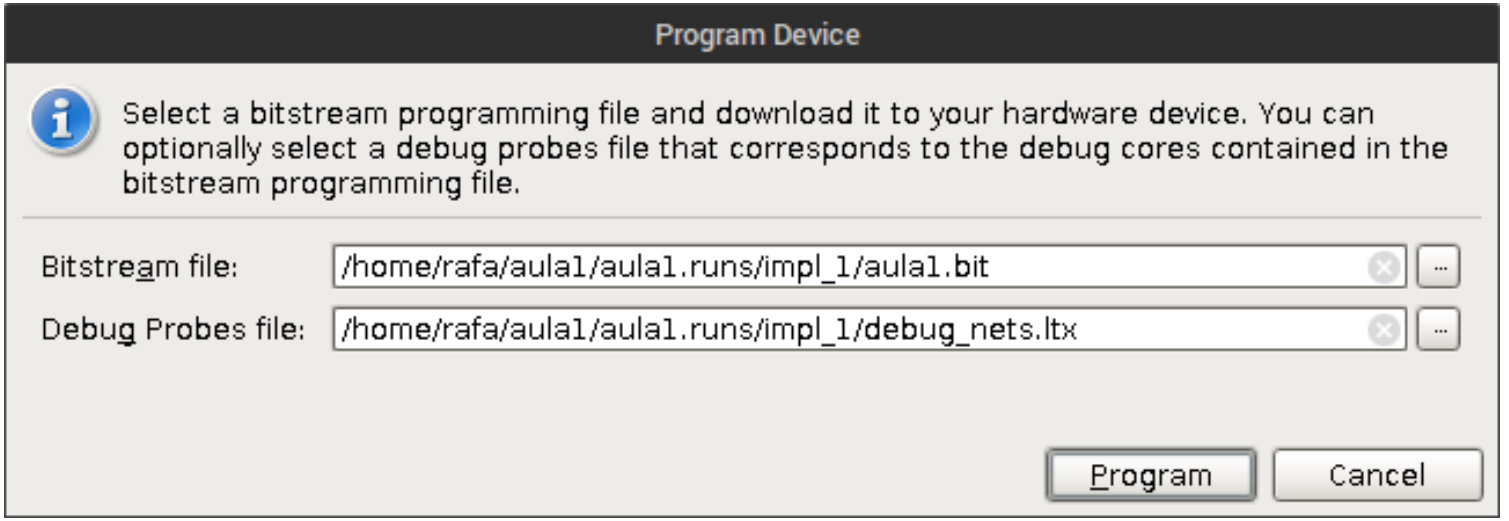

Open New Hardware Target
Select Hardware Target
Select a hardware target from the list of available targets, then set the appropriate JTAG clock (TCK) frequency. If you do not see the expected devices, decrease the frequency or select a different target.
Hardware Targets
Type
Port
Name
JTAG Clock Frequency
xilinx_tcf
Digilent/210274505220A
15000000
Hardware Devices (for unknown devices, specify the Instruction Register (IR) length)
Name
ID Code
IR Length
xc7a100t_0
13631093
6
VCSE server: localhost:60001
Hardware server: localhost:3121
VCSE server: localhost:60001
< Back
Next >
Finish
Cancel


Open New Hardware Target
Open Hardware Target Summary
Hardware Server Settings:
Server: localhost:3121
VCSE Server Settings:
Server: localhost:60001
Version: 20
Target Settings:
Target: xilinx_tcf/Digilent/210274505220A
Frequency: 15000000
VIVADO
To connect to the hardware described above, click Finish
< Back
Finish
Cancel

Com o HW configurado, basta clicarmos em “Program device” para gravarmos o programa na FPGA



Escolha o Bitstream e clique em program



FIM