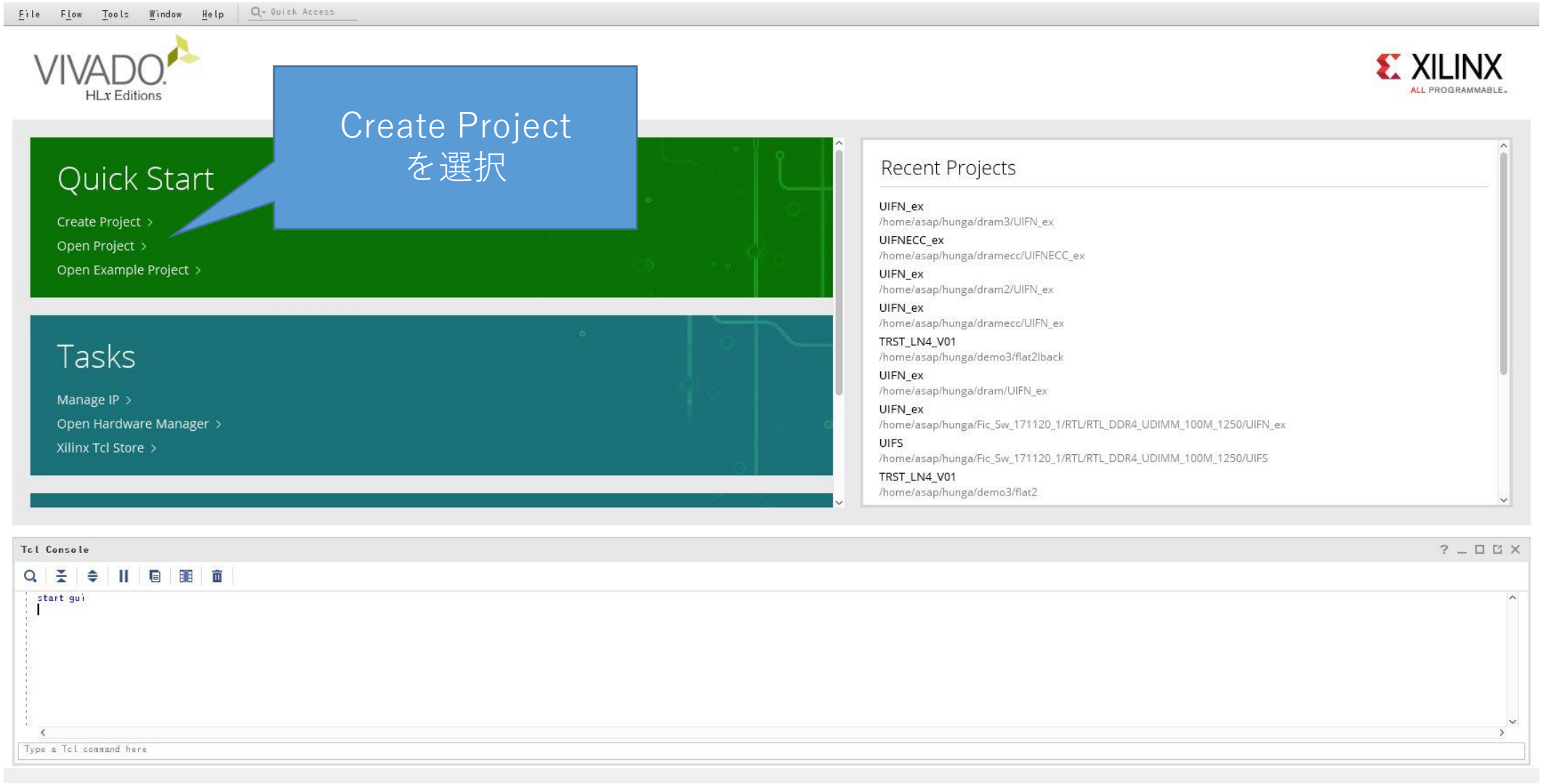
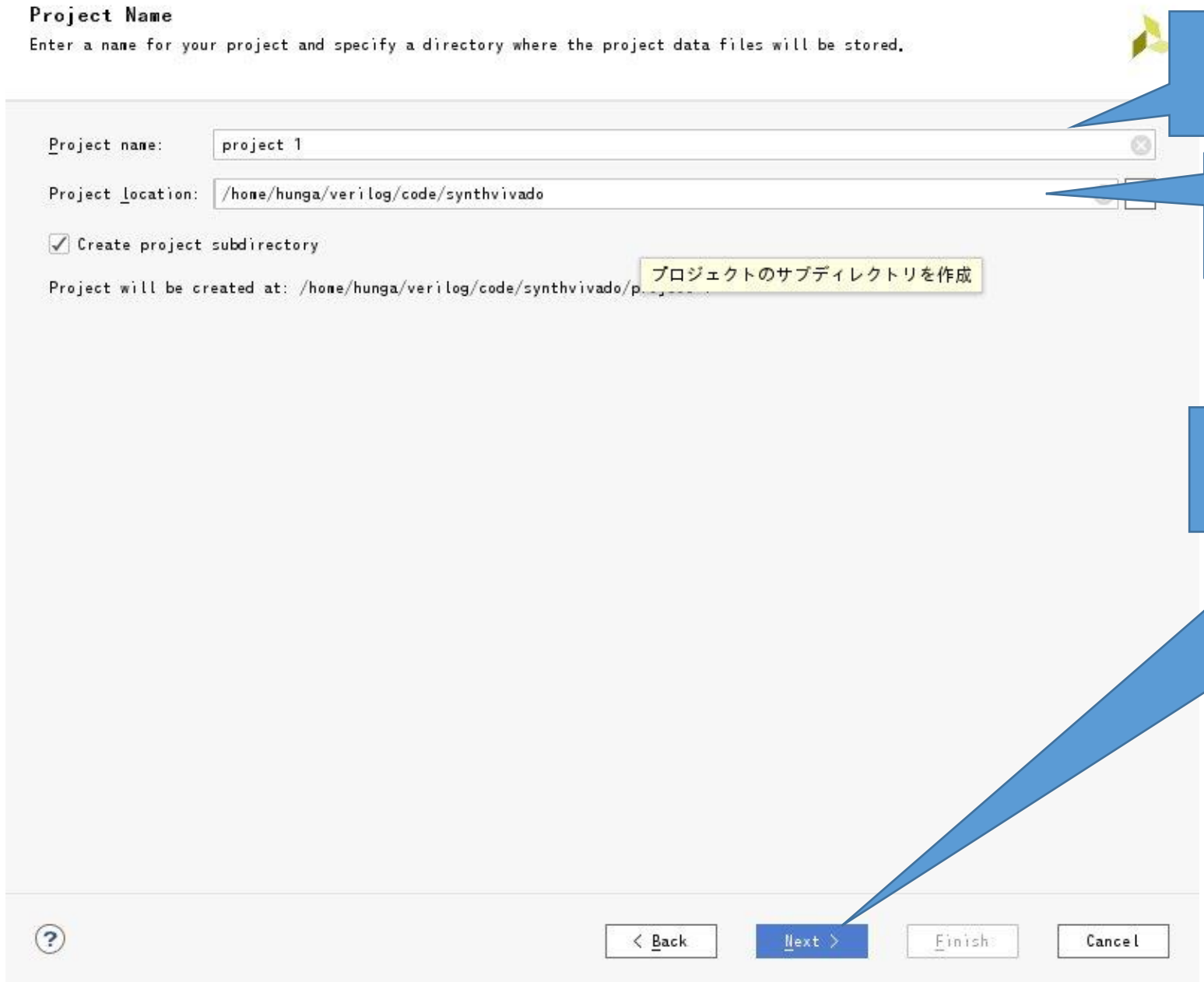
VivadoでPOCOを合成する

マイクロプロセッサ特論

天野

vivadoと打ち込み立ち上げる





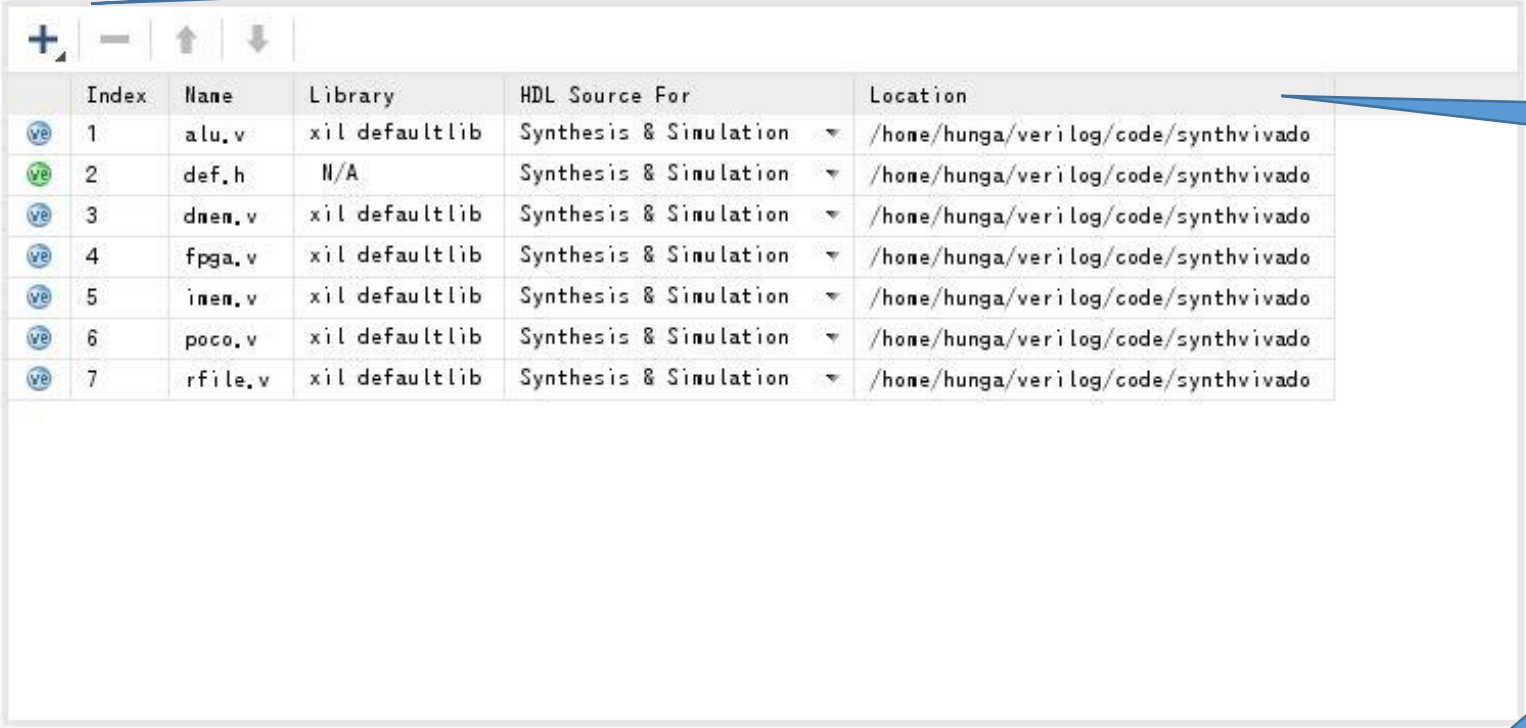
プロジェクト名はproject_1でOK

locationはsynthvivadoにしておく

Next

Add Sources

Specify HDL, netlist, Block Design, and IP files, or directories containing those files, to add to your project.
Create a new source file on disk and add it to your project. You can also add and create sources later.



	Index	Name	Library	HDL Source For	Location
	1	alu.v	xil_defaultlib	Synthesis & Simulation	/home/hunga/verilog/code/synthvivado
	2	def.h	N/A	Synthesis & Simulation	/home/hunga/verilog/code/synthvivado
	3	dmem.v	xil_defaultlib	Synthesis & Simulation	/home/hunga/verilog/code/synthvivado
	4	fpga.v	xil_defaultlib	Synthesis & Simulation	/home/hunga/verilog/code/synthvivado
	5	imem.v	xil_defaultlib	Synthesis & Simulation	/home/hunga/verilog/code/synthvivado
	6	poco.v	xil_defaultlib	Synthesis & Simulation	/home/hunga/verilog/code/synthvivado
	7	rfile.v	xil_defaultlib	Synthesis & Simulation	/home/hunga/verilog/code/synthvivado

Add Files | Add Directories | Create File

☐ Scan and add RTL include files into project

☐ Copy sources into project

☑ Add sources from subdirectories

Target language: Verilog Simulator language: Mixed

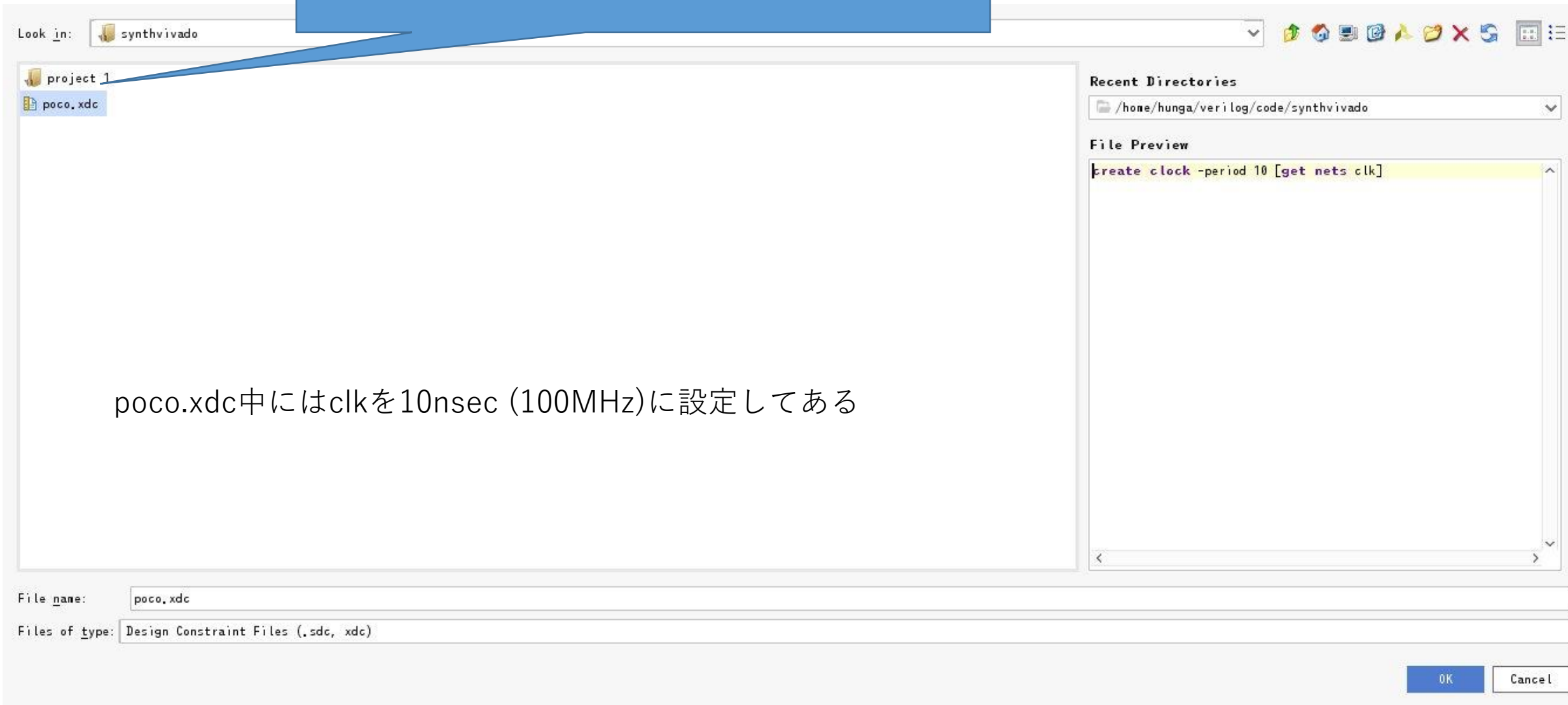
< Back | Next > | Finish | Cancel

＋をクリック

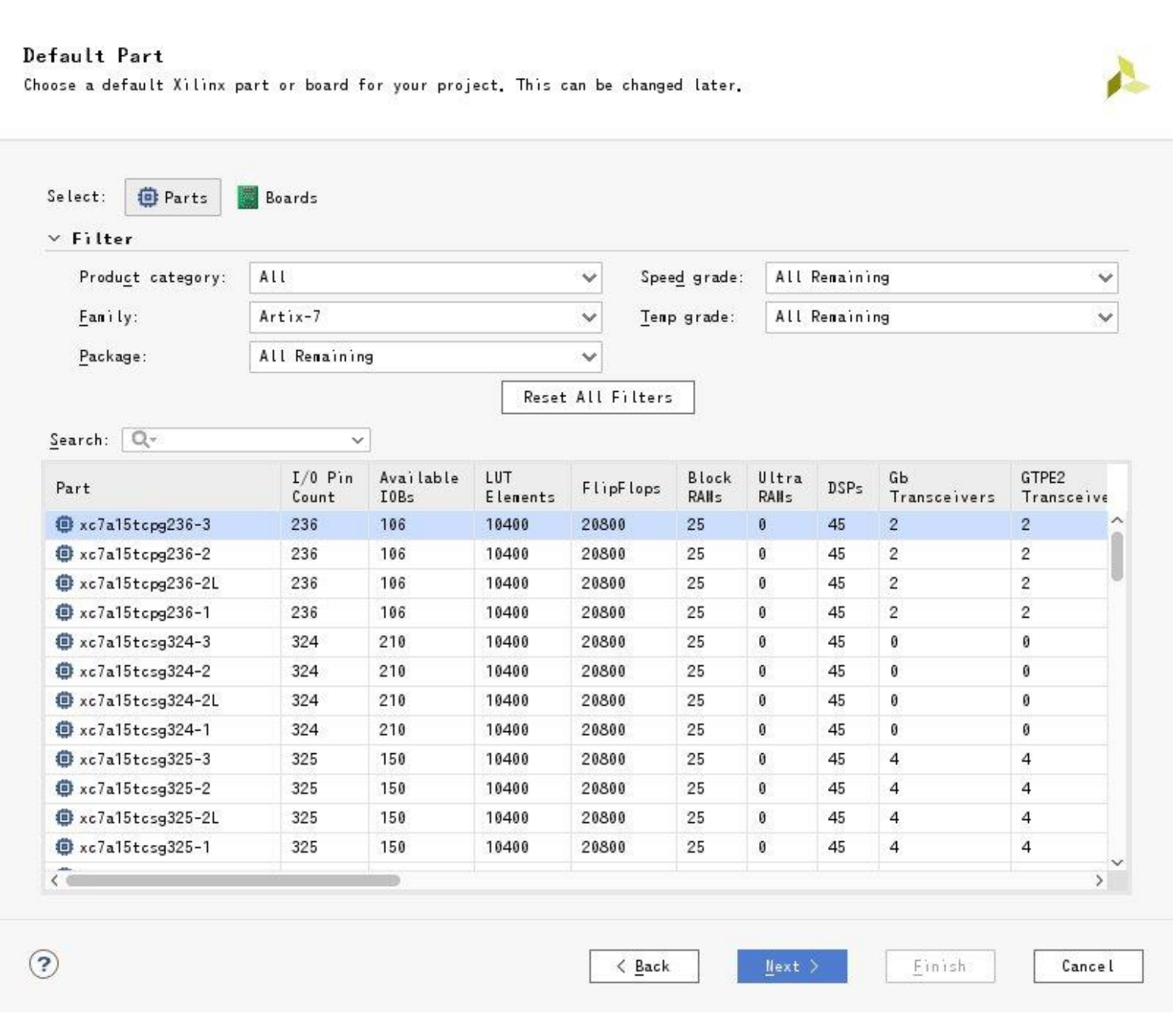
表れたファイルを
全部選択
→OK

NEXT

Constraint fileはpoco.xdcを選択
OK→Next

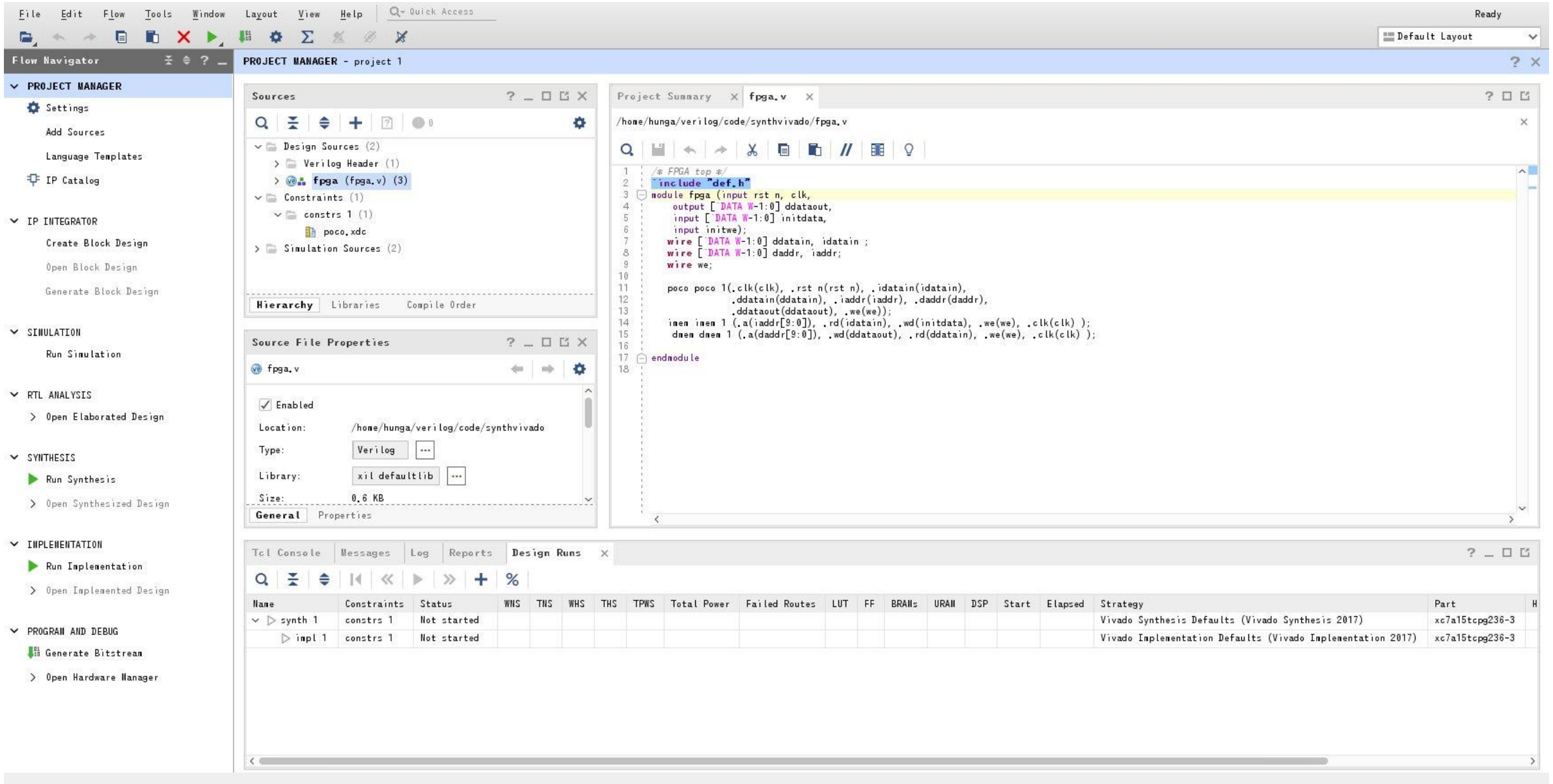


poco.xdc中にはclkを10nsec (100MHz)に設定してある

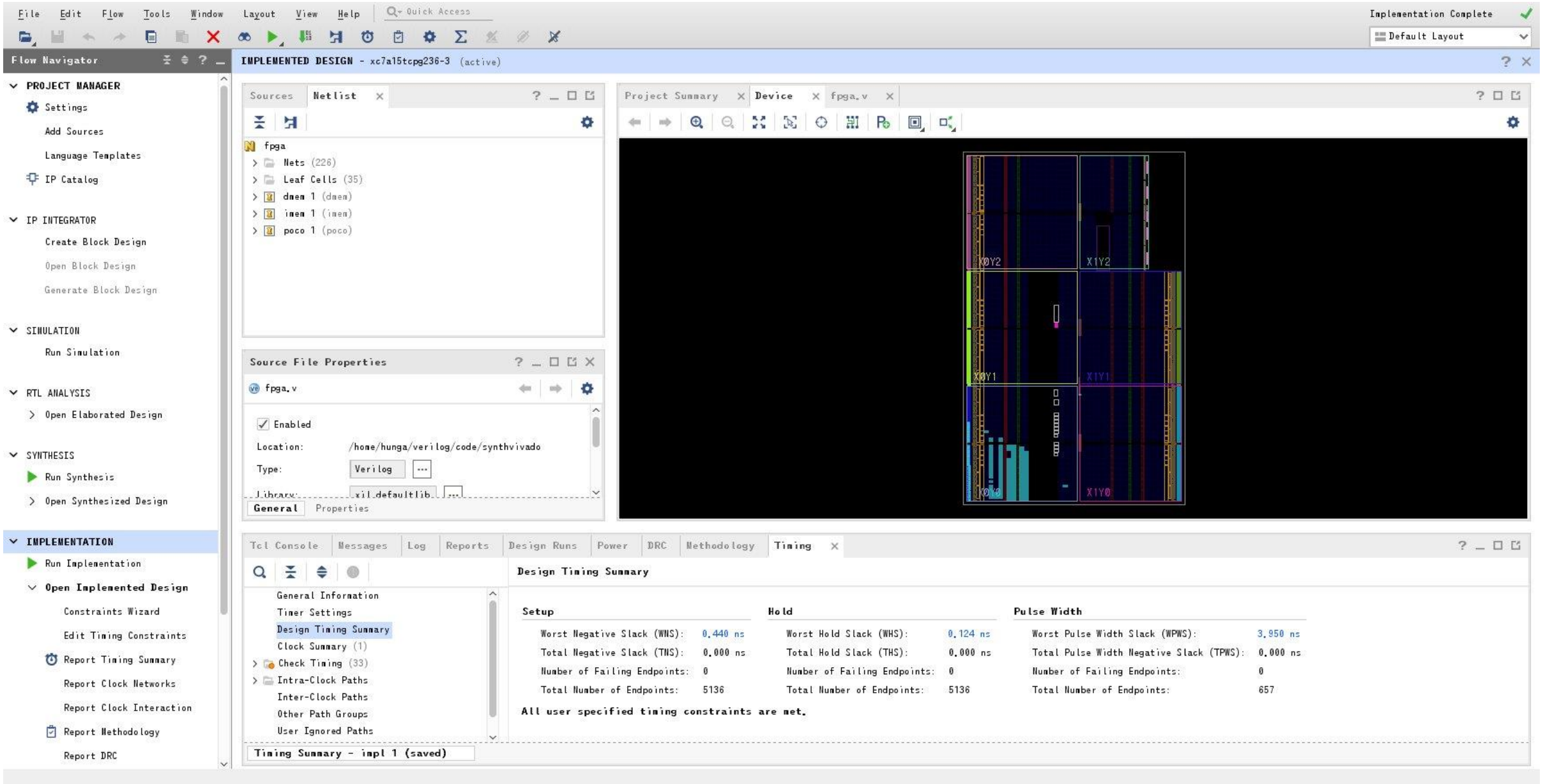


デバイスの選択
Artix-7a15tcpg236-3にしておく
Artix-7aの中では小さいが、
16640Logic Cell
900KbのBRAMを持つ

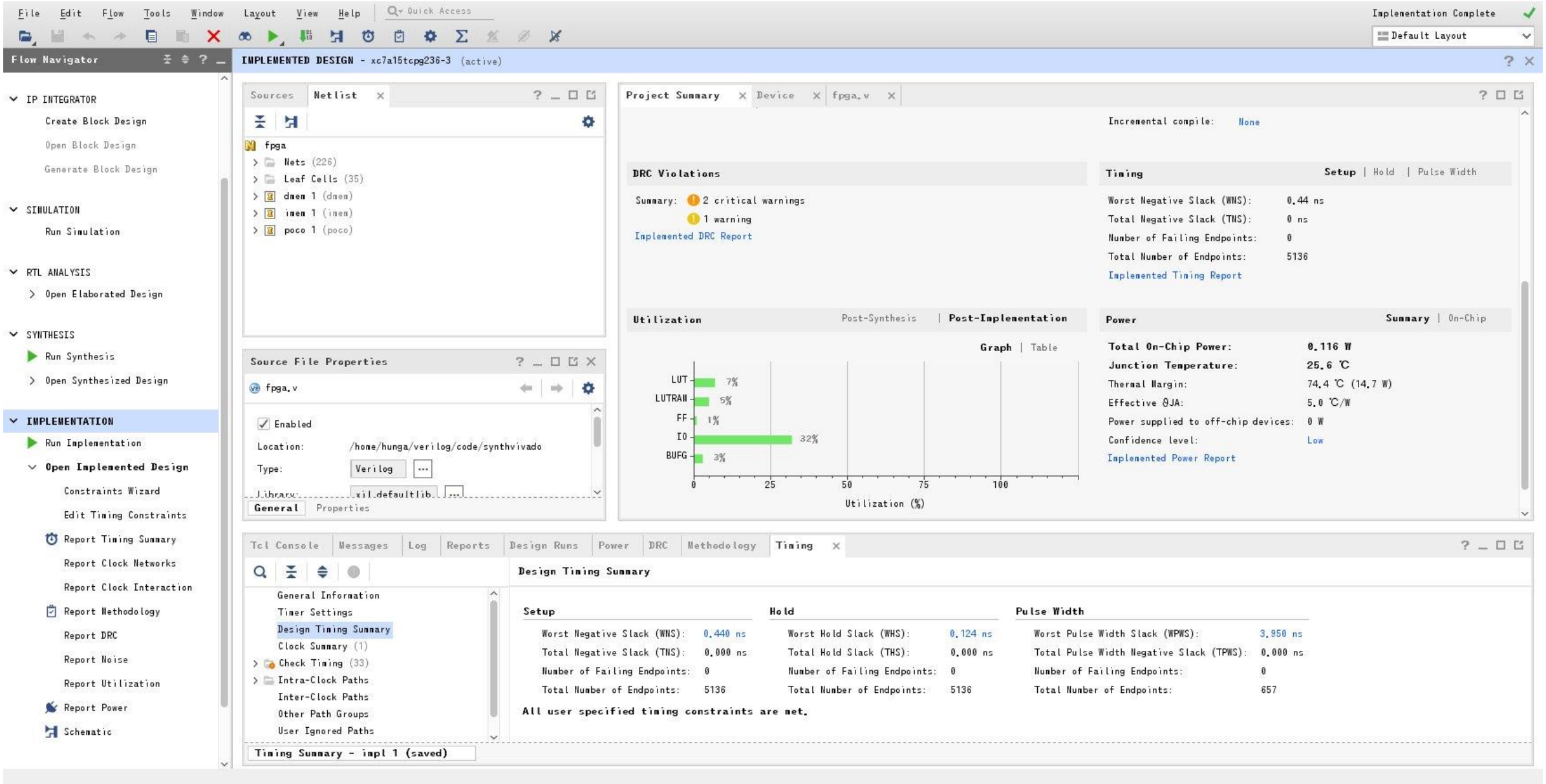
FamilyをArtix-7を選択
xc7a15tcpg236-3を選択
Next→Finish



プロジェクトの画面：　Run Synthesis（論理合成）→Run Implementation（配置配線）



Implementation終了後、Open Implementationを行った図、Deviceの様子が見れる



Project Summaryをクリック：Timing, Utilization, Powerが表示される。
詳細が見たい時はTools→Reportをクリック

レポートの見方

- Timing：　FPGAの動作タイミング
  - Total Negative Slack (TNS）が0nsecになっていれば目標達成、この場合は100MH z で動作する
  - 制約が満足できないと赤字で出る
- Utilization: FPGAのリソース利用率
  - LUT：　Look Up Tableでランダムロジック　７％
  - LUTRAM:　LUTで作ったRAM　５％
  - FF:　CLB中のFlip Flop　１％
  - I/O：　I/Oピン　３２％
  - BUFG:バッファ　３％
- Total On-Chip Power:　目安

演習

- 制約ファイルを書き換えて5nsにしてやってみよ。
- その時のTNSを報告せよ。
- LUTの％は変化したか？

メール中のテキストデータで報告せよ。