



IMAGINATION大學計劃

RVfpga實驗1

建立Vivado專案

1. 簡介

要使用和修改RVfpga系統，需要先構建一個專案，其中包括定義系統的所有Verilog檔案、SystemVerilog檔案、頭檔案、配置檔和文字檔。本實驗將展示如何建立一個以Digilent的Nexys A7 FPGA開發板（100T版本，即RVfpgaNexys）作為本課程中所用SoC的目標的Vivado專案。（請記住，如果已有Nexys4 DDR開發板，也可以使用該開發板。）透過執行這些相同的步驟，即可修改RVfpgaNexys並重新加以綜合。

重要資訊：在開始RVfpga實驗之前，必須完成Imagination大學計劃（<https://university.imgtec.com/>）提供的「RVfpga入門指南」中所述的準備工作。

例如，必須依據「RVfpga入門指南」中的說明安裝Xilinx的Vivado和Verilator（如果尚未安裝）。此外，請確保已將從Imagination大學計劃下載的RVfpga資料夾複製到您的電腦上。我們將RVfpga資料夾所在目錄的絕對路徑稱為[RVfpgaPath]。[RVfpgaPath]/RVfpga/src資料夾包含RVfpga系統（即，我們將在所有實驗中使用和修改的RISC-V SoC）的Verilog和SystemVerilog來源檔案。[RVfpgaPath]/RVfpga/Labs資料夾包含實驗1至20中將會使用的一些程式。

2. 為RVfpgaNexys建立Vivado專案

您將在Xilinx的Vivado Design Suite¹中使用RTL（用於定義系統的Verilog檔案）編譯RVfpgaNexys系統。請按照以下的詳細步驟來編譯RVfpgaNexys系統，並將其指定為Nexys A7 FPGA開發板。

- 第1步. 開啟Vivado
- 第2步. 建立一個新的RTL專案
- 第3步. 新增RTL原始程式檔和限制檔
- 第4步. 選擇Nexys A7作為目標板
- 第5步. 將rvfpganexys設為頂層模組，common_defines.vh設為全域檔案
- 第6步. 產生位元串流

第1步. 開啟Vivado

如果尚未按照「RVfpga入門指南」的說明在電腦上安裝Vivado，請立即進行安裝。請確保同時安裝開發板檔案。

然後，執行Vivado（在Linux中，開啟一個終端機並輸入：vivado；在Windows中，通過「Start」（開始）功能表開啟Vivado）。Vivado歡迎介面隨即開啟。按一下「Create Project」（建立專案）（請參閱圖1）。

¹ 在這些材料中，我們均使用Vivado 2019.2。儘管大多數內容在版本更高的Vivado中也適用，但我們強烈建議使用此版本。

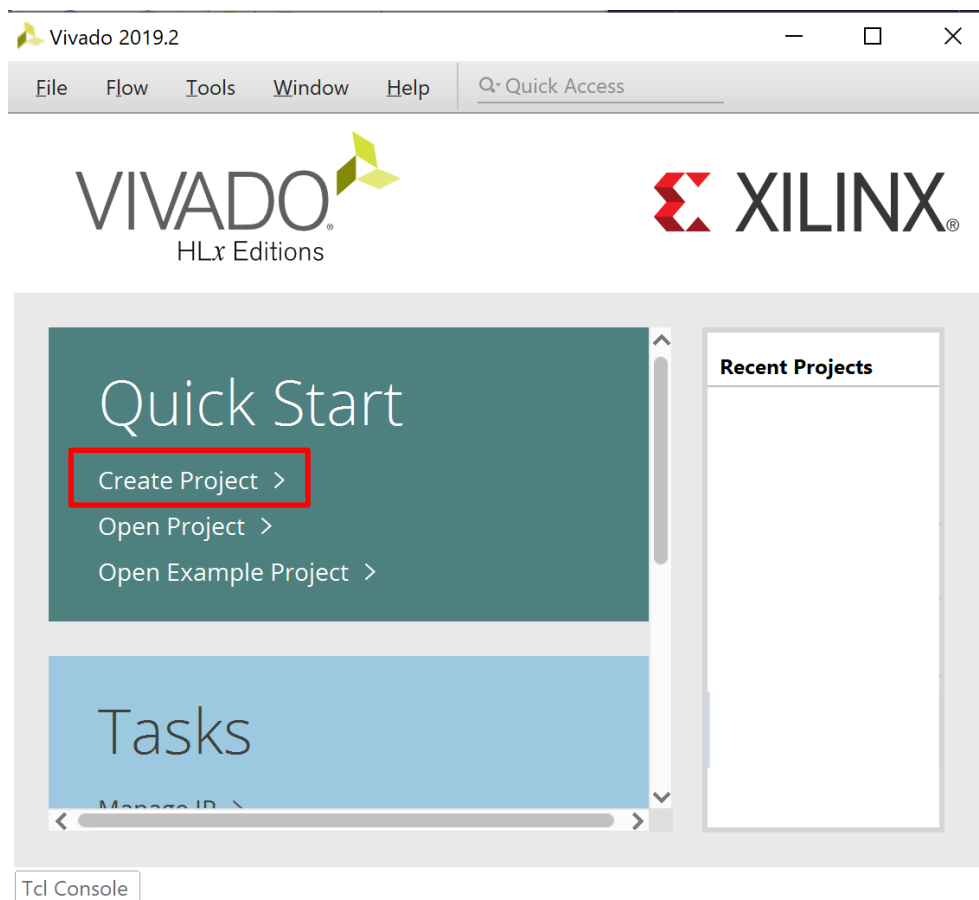


圖1. Vivado歡迎介面：建立專案

第2步. 建立一個新的RTL專案

「Create a New Vivado Project」（建立一個新的Vivado專案）精靈隨即開啟（請參閱圖2）。按「Next」（下一步）。

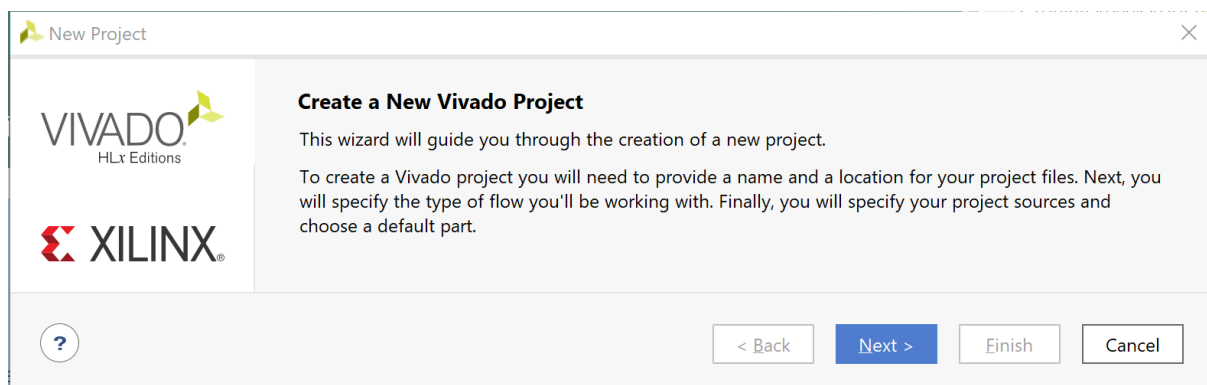


圖2. 「Create a New Vivado Project」（建立一個新的Vivado專案）精靈

呼叫專案名稱「Project1」並將其放到`[RVfpgaPath]/RVfpga/Labs/Lab1`資料夾中。選擇「Create project subdirectory」（建立專案子目錄）選項。然後按「Next」（下一步）（請參閱圖3）。

Project Name

Enter a name for your project and specify a directory where the project data files will be stored.

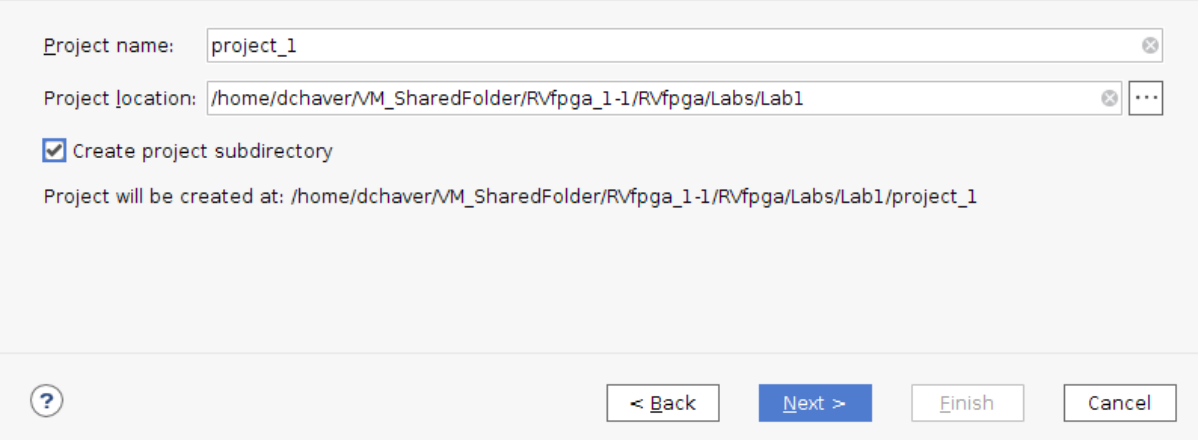



圖3. 專案名稱

選擇專案類型「RTL Project」（RTL專案），然後按「Next」（下一步）（請參閱圖4）。

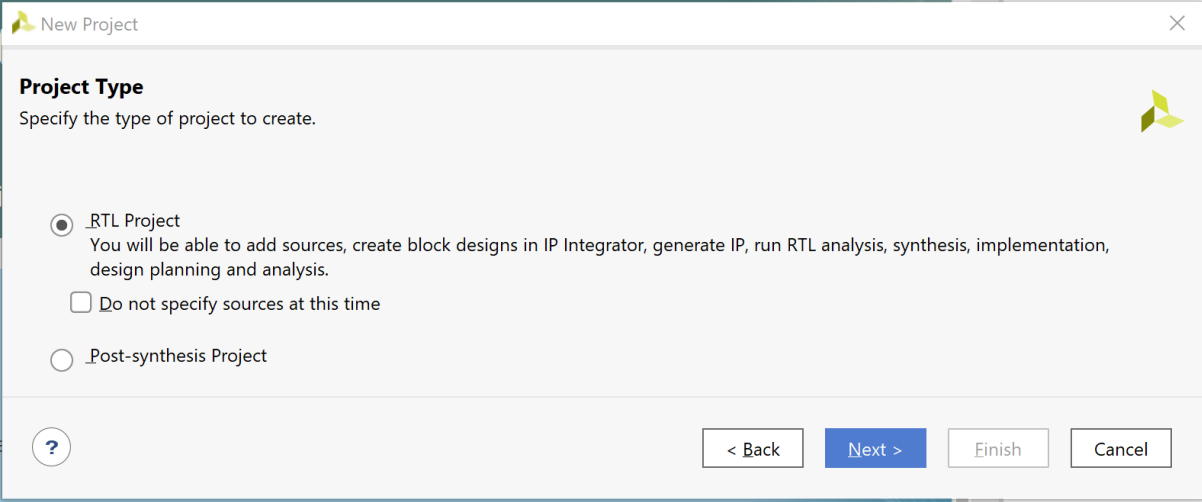


圖4. RTL專案

第3步. 新增RTL原始程式檔和限制檔

在「Add Sources」（新增原始程式碼）視窗中，按一下「Add Directories」（新增目錄），然後選擇`[RVfpgaPath]/RVfpga/src`（請參閱圖5）。確保同時選擇以下兩個選項（如圖5所示）：

- 「Scan and add RTL include files into project」（掃描RTL包含檔案並將其新增到專案中）
- 「Add sources from subdirectories」（從子目錄新增原始程式碼）

然後按「Next」（下一步）。

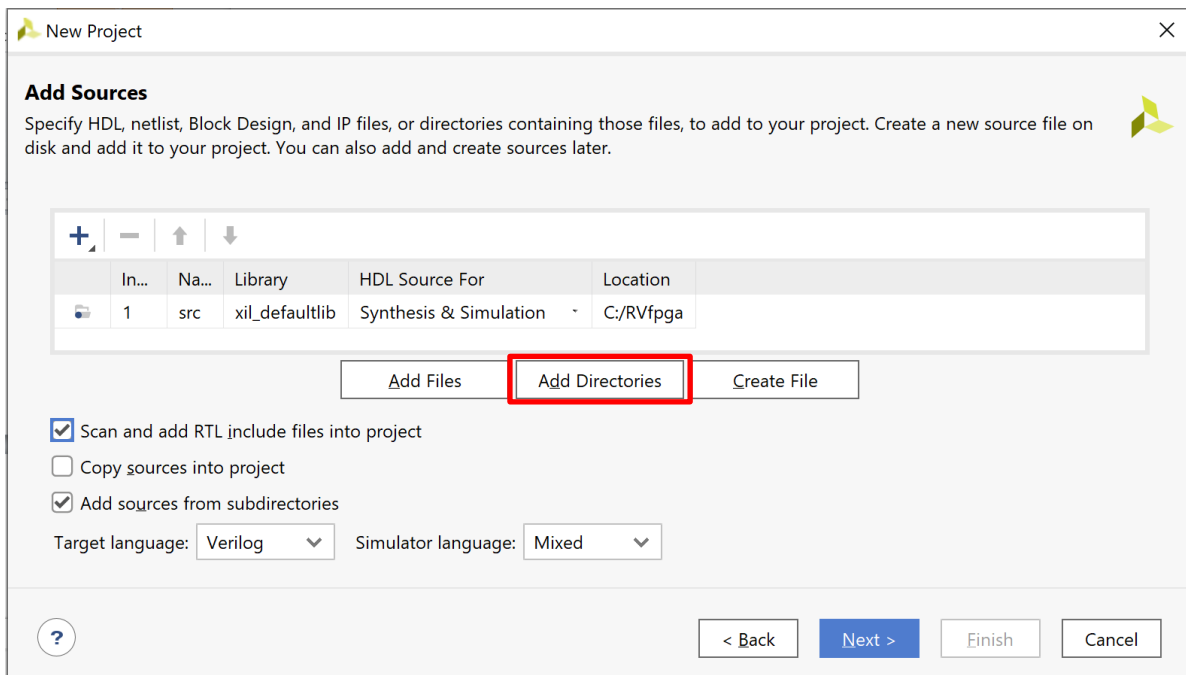


圖5. 新增原始程式碼

接下來將新增系統的限制檔。這些檔案會將訊號名稱映射到開發板上的引腳。例如，Nexys A7 FPGA開發板上的LED通過PCB中的走線與開發板上的FPGA引腳相連。Vivado必須知道這些資訊，以便將RTL中的訊號名稱正確地映射到相應的FPGA引腳。例如，*[RVfpgaPath]/RVfpga/src/rvfpganexys.xdc*檔（一種Xilinx的設計限制檔）中的下行表示FPGA引腳H17映射到LED的最低有效位元（o_led[0]），並使用VCMOS 3.3V訊號：

```
set_property -dict { PACKAGE_PIN H17 IOSTANDARD LVCMOS33 } [get_ports { o_led[0] }]
```

請注意，訊號名稱o_led是Verilog程式碼中用於驅動Nexys A7開發板LED的名稱。

在「Add Constraints」（新增限制檔）視窗中，按一下「Add Files」（新增檔案），然後選擇以下兩個檔案（請參閱圖6）：

```
[RVfpgaPath]/RVfpga/src/rvfpganexys.xdc  
[RVfpgaPath]/RVfpga/src/LiteDRAM/liteDRAM.xdc
```

然後按「Next」（下一步）。

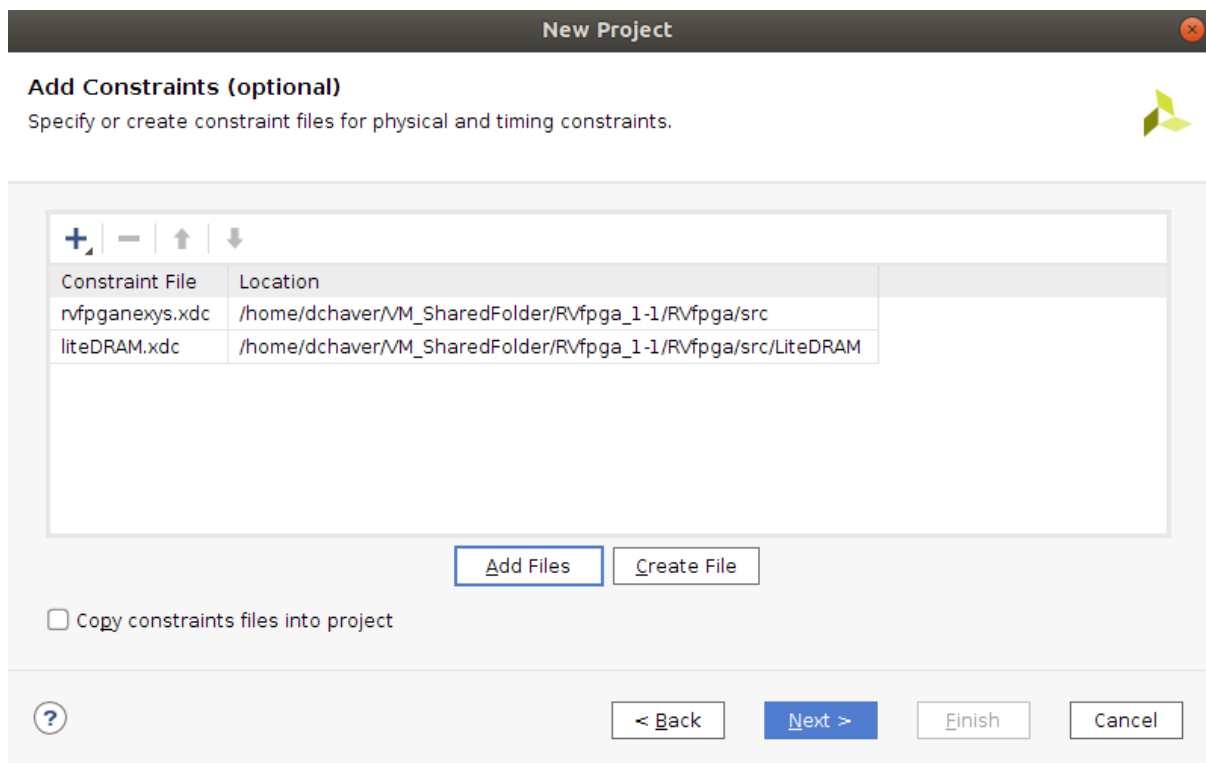


圖6. 新增限制檔

第4步. 選擇Nexys A7作為目標板

在「Default Part」（預設零件）視窗中，按一下「Boards」（開發板），然後選擇「Nexys A7-100T」（請參閱圖7）。可以使用「Search」（搜尋）方塊來縮小結果範圍。您還將注意到，「Part」（零件）欄中會列出實際目標FPGA的名稱：xc7a100tcsg324-1。這表明目標FPGA為Xilinx Artix-7 FPGA，具有100000個採用CSG（晶片級網格）套件的等效閘門和324個引腳。

按「Next」（下一步）。

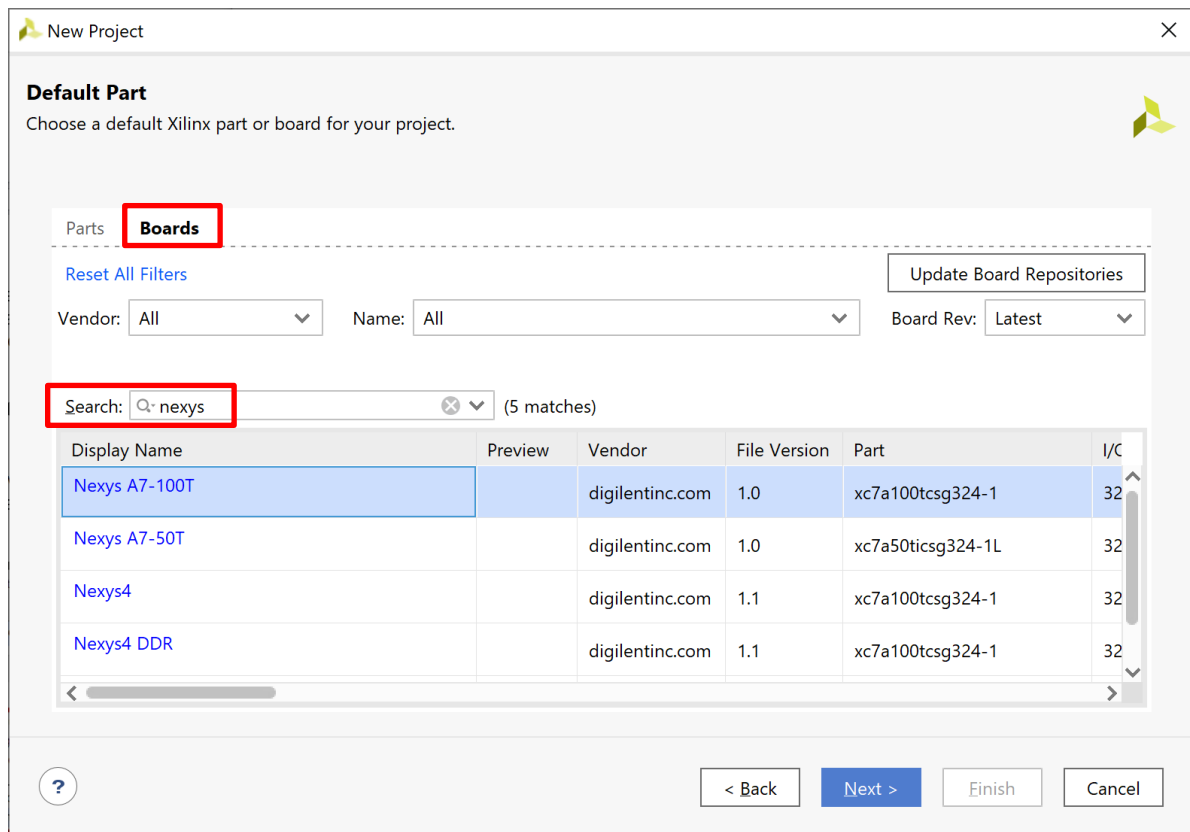


圖7. 選擇開發板：Nexys A7-100T

在「New Project Summary」（新增專案摘要）視窗中，按一下「Finish」（完成）（請參閱圖8）。

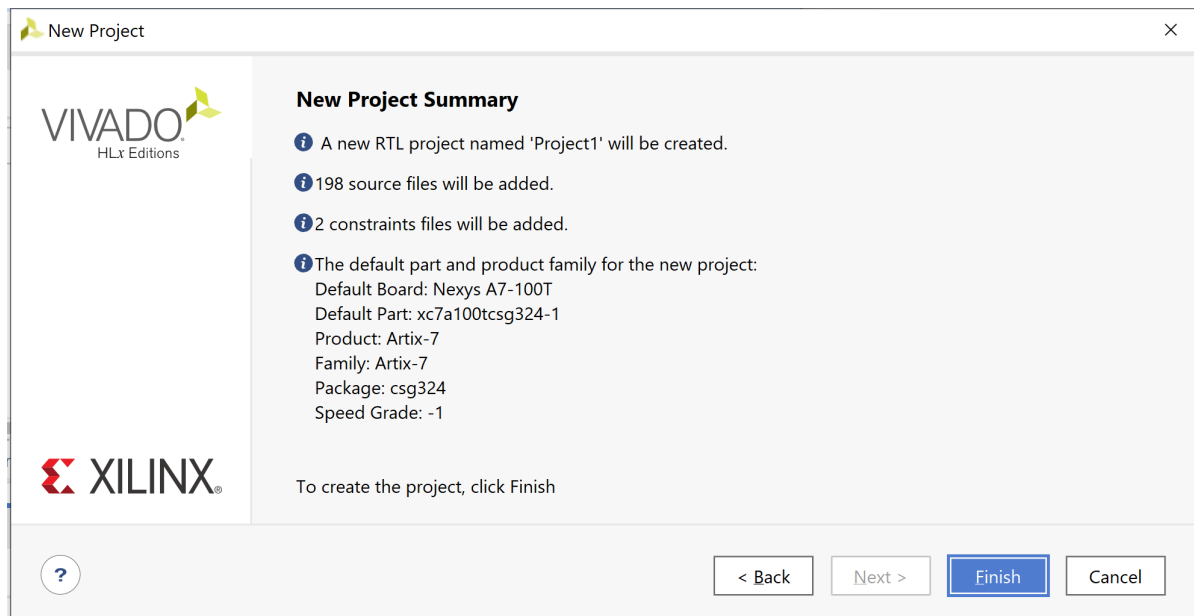


圖8. 「New Project Summary」（新增專案摘要）視窗

請注意，專案設定完成後，系統將指出檔案存在語法錯誤，下一步將修復此錯誤。

第5步. 專案組態：將rvfpganexys設為頂層模組，將檔案common_defines.vh設為全域檔案，將boot_main.mem新增到專案中，並包含Pulp平台資料夾

將rvfpganexys設為頂層模組：接下來將rvfpganexys模組設為頂層模組。在「Design Sources」（設計原始程式碼）下的「Sources」（原始程式碼）窗格中向下捲動，用滑鼠右鍵按一下rvfpganexys模組，然後選擇「Set as Top」（設為頂層模組）（請參閱圖9）。也可以通過在搜尋方塊中輸入相應名稱來尋找rvfpganexys模組，如下圖所示。這會將rvfpganexys設為階層中級別最高的模組，並將其作為要在FPGA上合成並實作的目標。將rvfpganexys設為頂層模組後，階層將更新。

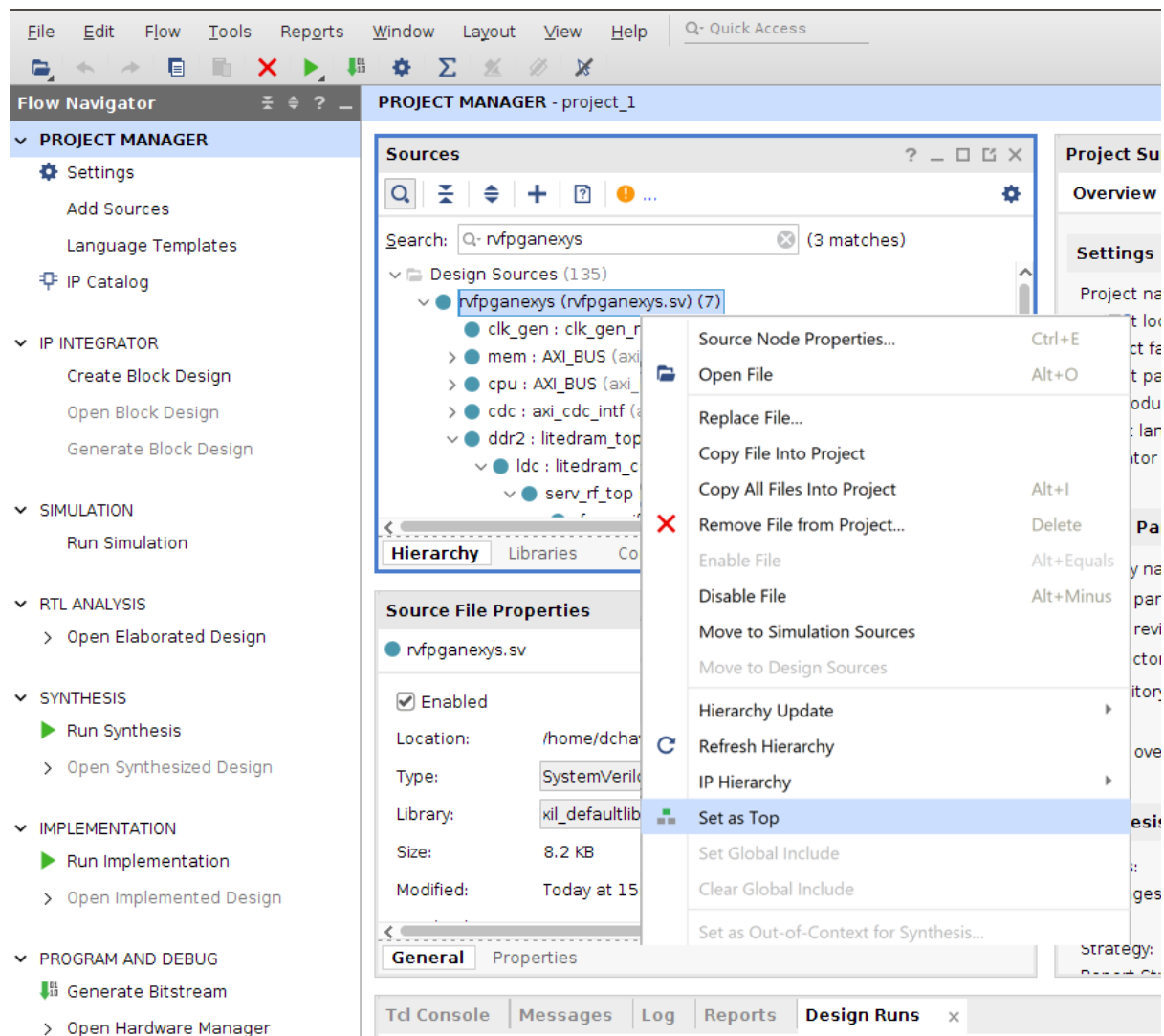


圖9. 將rvfpganexys設為頂層模組

將檔案common_defines.vh設為全域包含檔案：然後，還是在「Design Sources」（設計原始程式碼）下的「Sources」（原始程式碼）窗格中，展開「Non-modules」（非模組）檔案群組，然後按一下common_defines.vh。該檔案的屬性將顯示在「Sources」（原始程式碼）窗格下面的「Source File Properties」（原始程式檔屬性）窗格中。按一下「Global Include」（全域包含檔案）以勾選方塊（請參閱圖10）。階層隨即更新，並將該檔案包含在「Design Sources」（設計原始程式碼）/「Global Include」（全域包含檔案）中。請注意，語法錯誤檔將不會出現在後續步驟中。

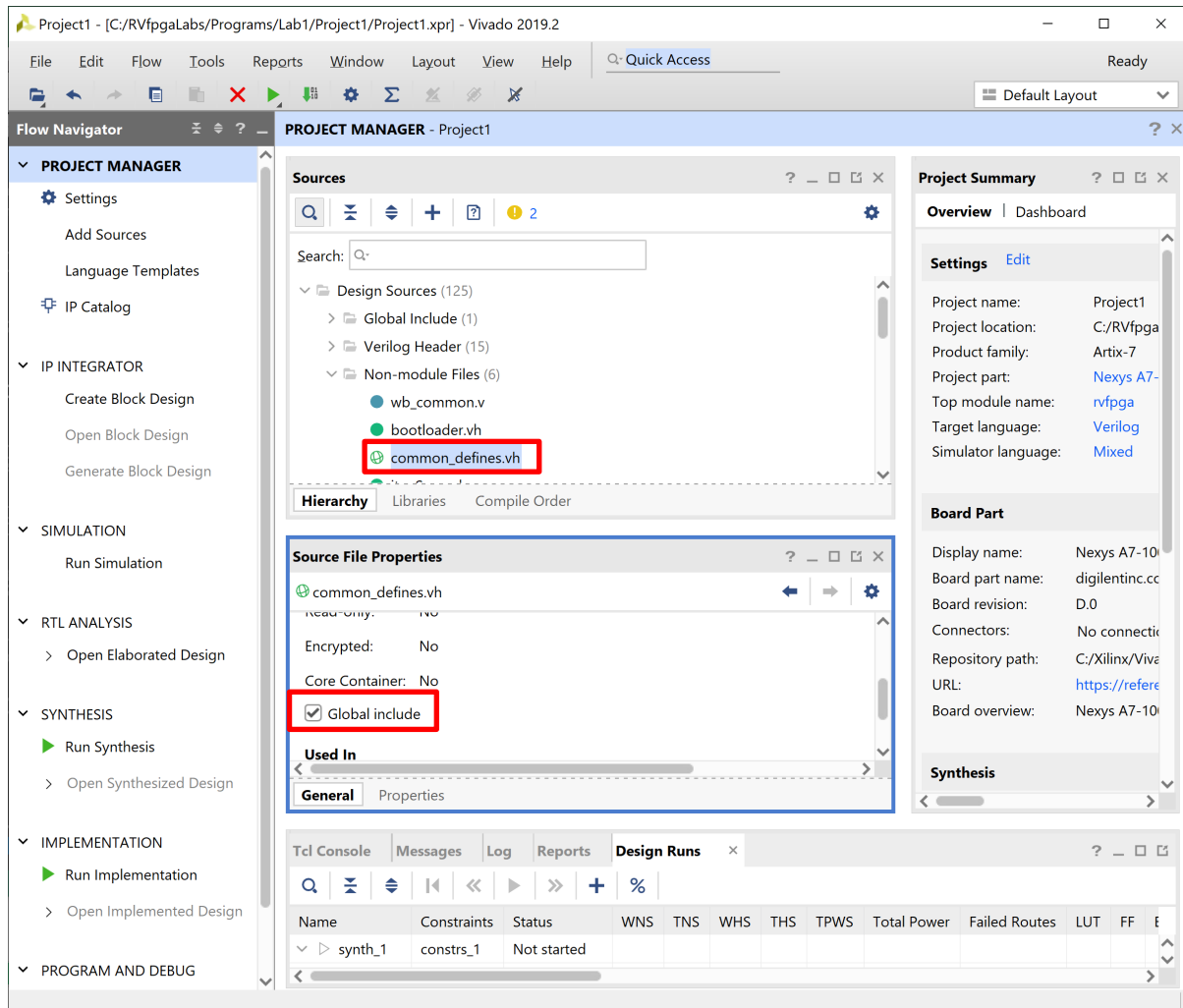


圖10. 將common_defines.vh設為全域包含檔

將**boot_main.mem**新增到專案中：在「Flow Navigator」（流程導導器）窗格中，按一下「Add Sources」（新增來源檔案），保持預設選項（「Add or create design sources」（新增或建立設計來源檔案）），然後按一下「Add Files」（新增檔案）（參見圖11）。導覽到 **[RVfpgaPath]/RVfpga/src/SweRVolfSoC/BootROM/sw** 並選擇 **boot_main.mem**（如圖11所示）。階層隨即更新，並將該檔案包含在「Design Sources」（設計來源檔案）/「Memory File」（記憶體檔案）中。

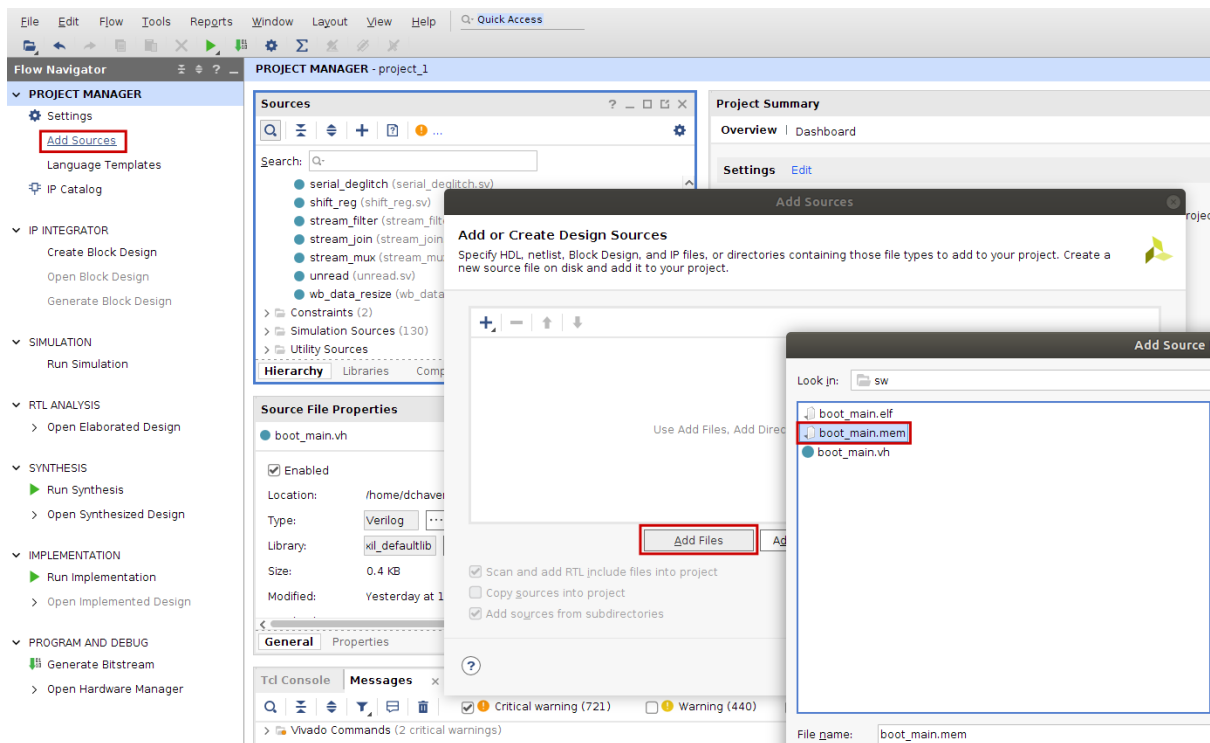




圖11. 新增記憶體檔案boot_main.mem

該檔案（*boot_main.mem*）用於初始化我們的SoC的啟動ROM，方法是在檔案 *[RVfpgaPath]/RVfpga/src/rvfpganexys.rv* 中將其作為參數叫用：

```
25 module rvfpga
26   #(parameter bootrom_file = "boot_main.mem")
```

入門指南中的第6.A節包含該檔案的更多資訊。

包含資料夾：最後，包含Pulp平台的兩個資料夾（參見圖12）。在「Flow Navigator」（流程導導器）窗格中按一下「**Settings**」（設定），然後在開啟的視窗中依次按一下「**General**」（一般）和「**Verilog options**」（Verilog選項）（）。在新視窗中，按一下  並導覽到相應目錄，新增以下兩個包含目錄：

```
[RVfpgaPath]/RVfpga/src/SweRVolfSoC/Interconnect/AxiInterconnect/pulp-platform.org__axi_0.25.0/include
[RVfpgaPath]/RVfpga/src/OtherSources/pulp-platform.org__common_cells_1.20.0/include
```

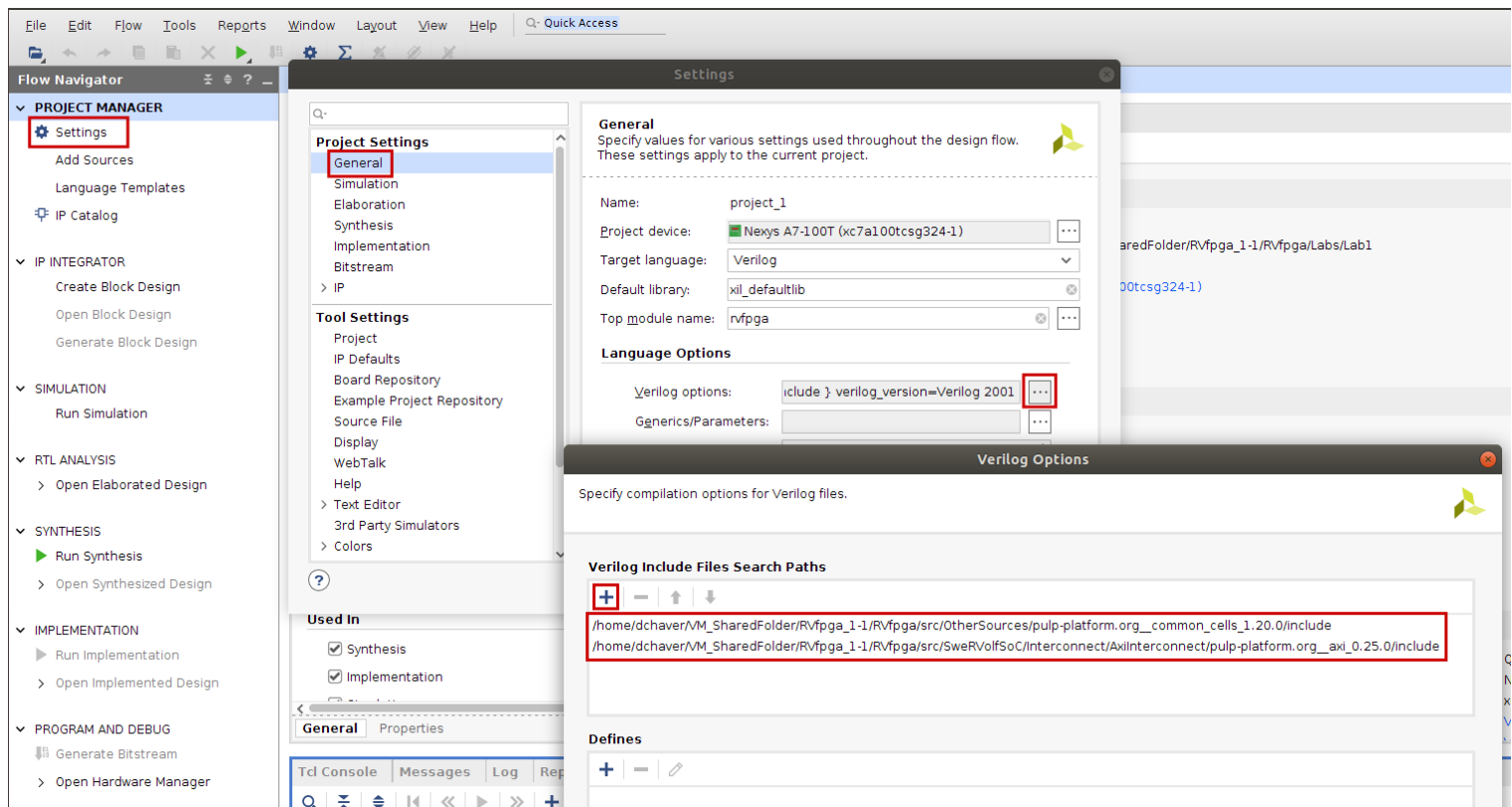


圖12. 包含Pulp平台的資料夾

第6步. 產生位元串流

接下來按一下「Flow」（流動）→「Generate Bitstream」（產生位元串流），如圖13所示。此時可能會彈出一個視窗，提示沒有可用的實作結果，並要求啟動綜合和實作（請參閱圖14）。按一下「Yes」（是）。然後在「Launch Runs」（啟動執行）視窗中按一下「OK」（確定）（請參閱圖15）。此步驟會對RVfpgaNexys進行綜合（根據專案中Verilog和SystemVerilog檔案的定義）、將其映射到FPGA並建立位元串流。此程序通常需要20-50分鐘，具體取決於電腦的速度。

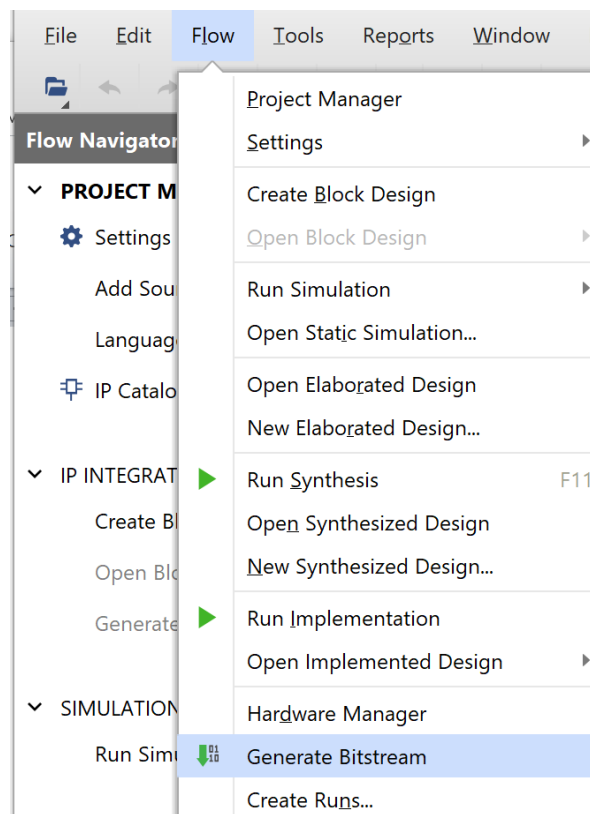


圖13. 產生位元串流

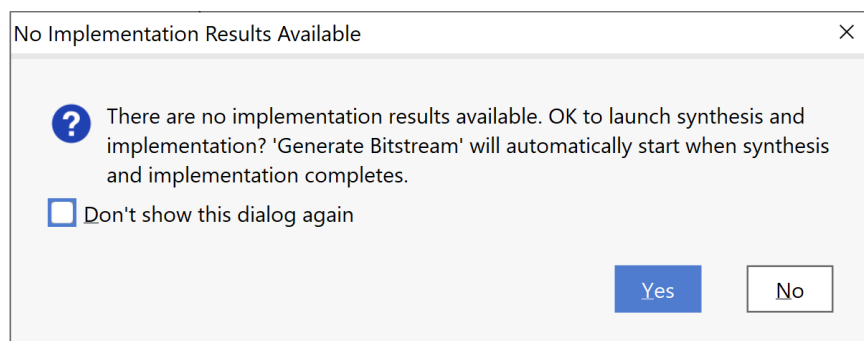


圖14. 啟動綜合和實作視窗

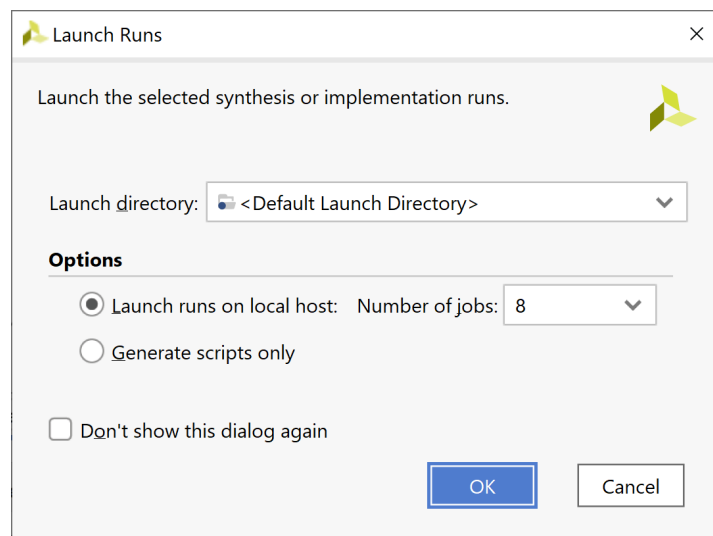


圖15. 啟動執行

產生位元串流後，將彈出一個視窗，如圖16所示。按一下右上角的 **X** 按鈕關閉該視窗。

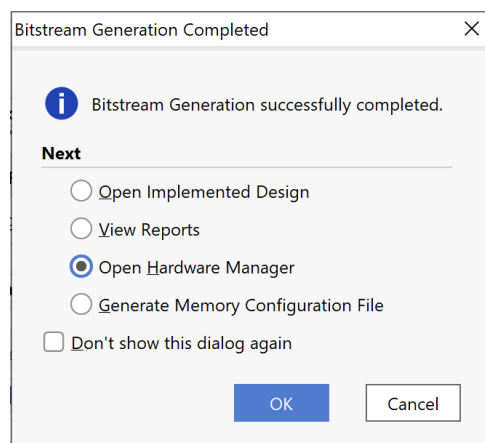


圖16. 位元串流產生完成

現在您已自行編譯了RVfpgaNexys系統，在實驗6-10中對其進行修改之後，可以重新編譯RVfpgaNexys。隨後需要使用剛剛編譯的RVfpgaNexys系統，利用PlatformIO在系統中執行下載並執行程式。

建議使用PlatformIO將RVfpgaNexys下載到Nexys A7開發板。有關此方法的詳細說明，請參閱「RVfpga入門指南（GSG）」的第6.A節。GSG中的其他範例（6.B至6.H）也顯示，將RVfpgaNexys系統下載到Nexys A7開發板上的FPGA之後，將使用PlatformIO在RVfpgaNexys上下載和執行/偵錯程式。

也可使用Verilator（一種HDL模擬器）來模擬RVfpgaSim上執行的程式，如「RVfpga入門指南」第7節所述。借助這些RTL級模擬，可以在軟體程式執行時檢視低階硬體訊號。對於實驗6-10，在延伸RVfpga系統以及測試和除錯變更時，我們將很大程度依賴Verilator。