



THE IMAGINATION UNIVERSITY PROGRAMME

RVfpga Deney 1

Bir Vivado Projesi Oluşturma

1. GİRİŞ

RVfpga'le çalışmak, RVfpga'i değiştirmek için sistemi tanımlayan Verilog, SystemVerilog, başlık, yapılandırma, metin dosyalarının hepsini içeren bir proje kurgulaman gerekecek. Bu deneyde RVfpga'i Nexys A7 Alanda-Programlanabilir Geçit Dizisi kartı, -100T sürümüne hedefleyen bir Vivado projesinin nasıl oluşturulacağını gösteriyoruz. (Nexys 4 DDR kartın varsa onu da kullanabileceğini unutma.) Bu adımları olduğu gibi izleyerek RVfpga çekirdeğini ya da sistemini değiştirip yeniden sentezleyebileceksin.

ÖNEMLİ: RVfpga Deneylerine başlamadan önce Imagination University Programme'in (<https://university.imgtec.com/>) sağladığı RVfpga İlk Kullanım Kılavuzunu bitirmiş olman gerekir.

Örneğin, daha yapmadıysan, Xilinx'in Vivado'sunu, PlatformIO'yu, Verilator'ü o kılavuzdaki yönergeleri izleyerek kur. Ayrıca, Imagination'ın University Programme'den indirdiğin **RVfpga** klasörünü kendi makinene kopyaladığının sağlamasını yap. RVfpga klasörünü koyduğun dizinin mutlak yolunu [*RVfpgaPath*] olarak adlandıracağız. RVfpga/src klasörü deneyler boyunca kullanacağımız, değiştireceğimiz RISC-V Yongadaki Sistem RVfpga için Verilog, SystemVerilog kaynaklarını içerir. RVfpga/Labs klasörü Deney 1'den 10'a değin kullanacağın kaynakları içerir.

2. RVfpga için bir Vivado Projesi Oluşturma

Yazmaç-Aktarım Düzeyliyi, sistemi tanımlayan Verilog dosyaları, kullanarak RVfpga sistemini kurgulamak için Xilinx'in Vivado Design Suite¹ini kullanacaksın. RVfpga sistemini kurgulayıp bir Nexys A7 Alanda-Programlanabilir Geçit Dizisi kartına hedeflemek için aşağıda detaylandırılmış adımları izle.

Adım 1. Vivado'yu aç

Adım 2. Yeni bir Yazmaç-Aktarım Düzeyli proje oluştur

Adım 3. Yazmaç-Aktarım Düzeyli kaynak dosyalarını, kısıtlandırma dosyalarını ekle

Adım 4. Nexys A7'ı hedef kart olarak seç

Adım 5. rvfpga'i Üst Modül olarak, common_defines.vh'i genel olarak ayarla

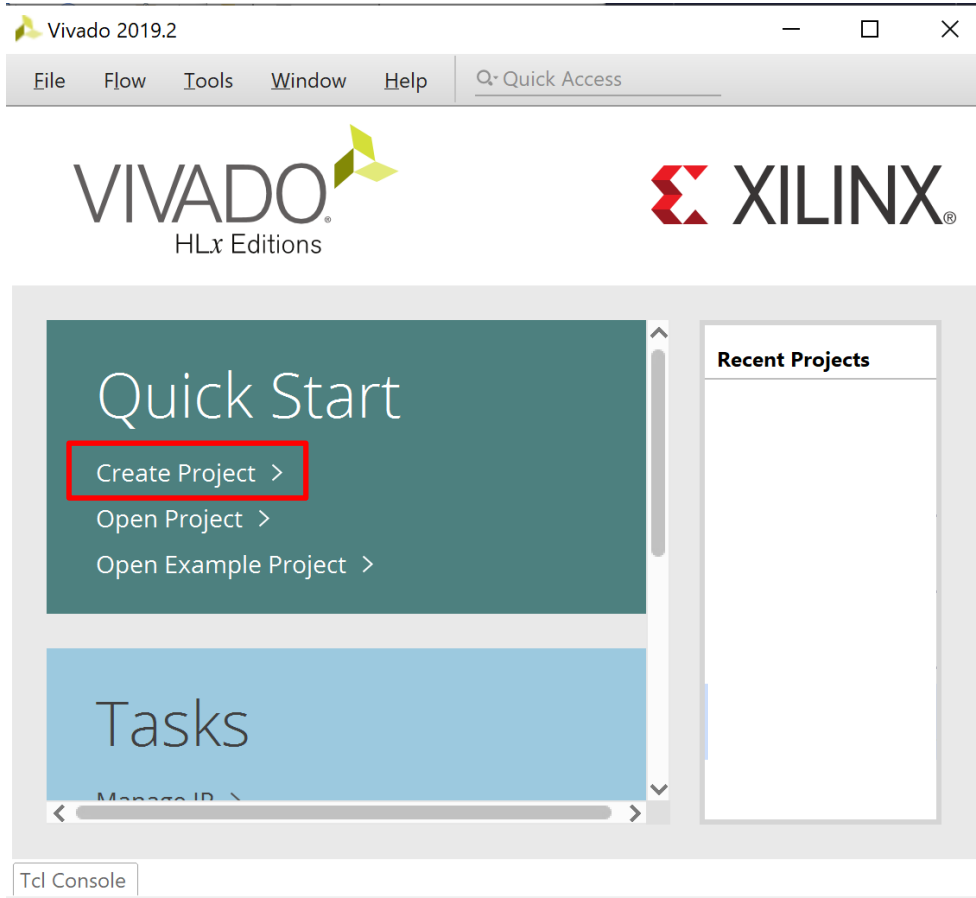
Adım 6. Veri Akışı oluştur

Adım 1. Vivado'yu aç

Vivado'yu RVfpga İlk Kullanım Kılavuzunda tanımlandığı gibi makinene kurmadıysan şimdi kur. Kart dosyalarını da kur.

Şimdi Vivado'yu çalıştır (**Linux**'da bir terminal açıp şunu yaz: vivado; **Windows**'ta Başlat menüsünden Vivado'yu aç). Vivado'nun hoş geldin ekranı açılacak. Create Project'e (Proje Oluştur) tıkla (Figür 1'e göz at).

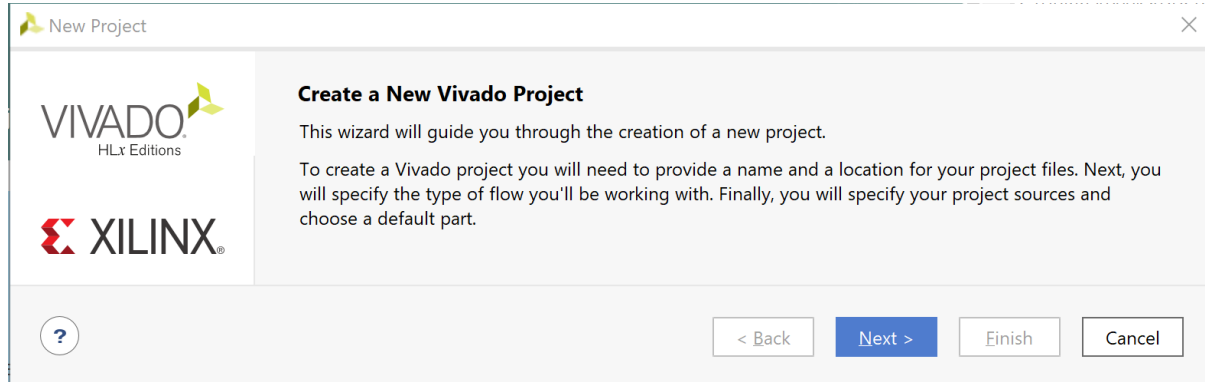
¹ Bu içeriklerde Vivado 2019.2'yu kullanıyoruz. Çoğu yeni Vivado dağıtımlarında çalışacak olsa da bu sürümün kullanımını öneriyoruz.



Figür 1. Vivado hoş geldin ekranı: Proje Oluştur

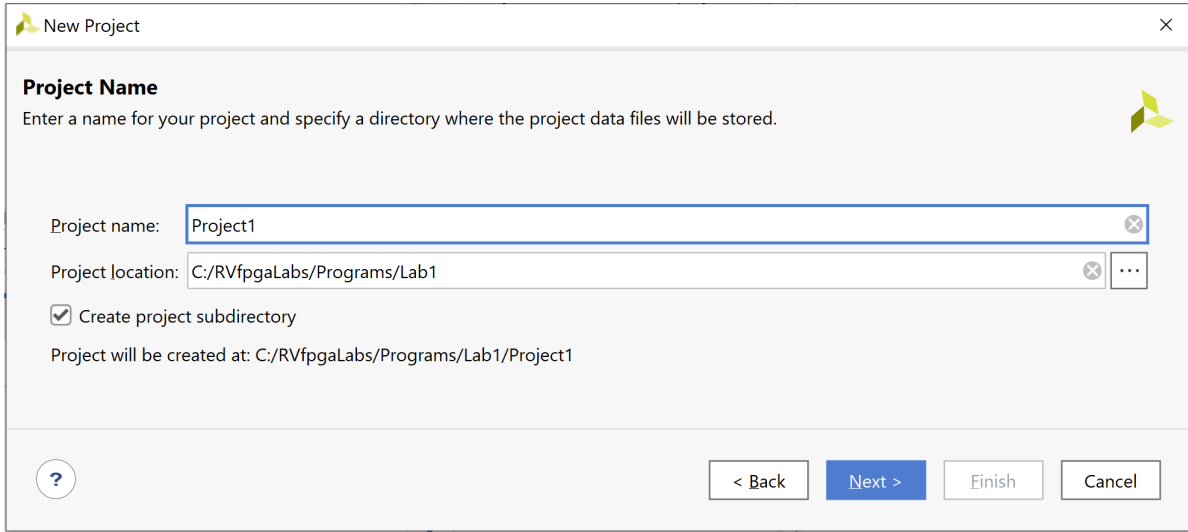
Adım 2. Yeni bir Yazmaç-Aktarım Düzeyli proje oluştur

Create a New Vivado Project (Yeni bir Vivado Projesi Oluştur) sihirbazı şimdi açılacak (Figür 2'ye göz at). Next'e (İleri) tıkla.



Figür 2. Yeni bir Vivado Projesi Oluştur Sihirbazı

Projenin adına Project1 diyip `[RVfpgaPath]/RVfpga/Labs/Lab1` klasörüne yerleştir. Ardından Next'e (İleri) (see Figür 3) tıkla.



New Project

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

Project name:

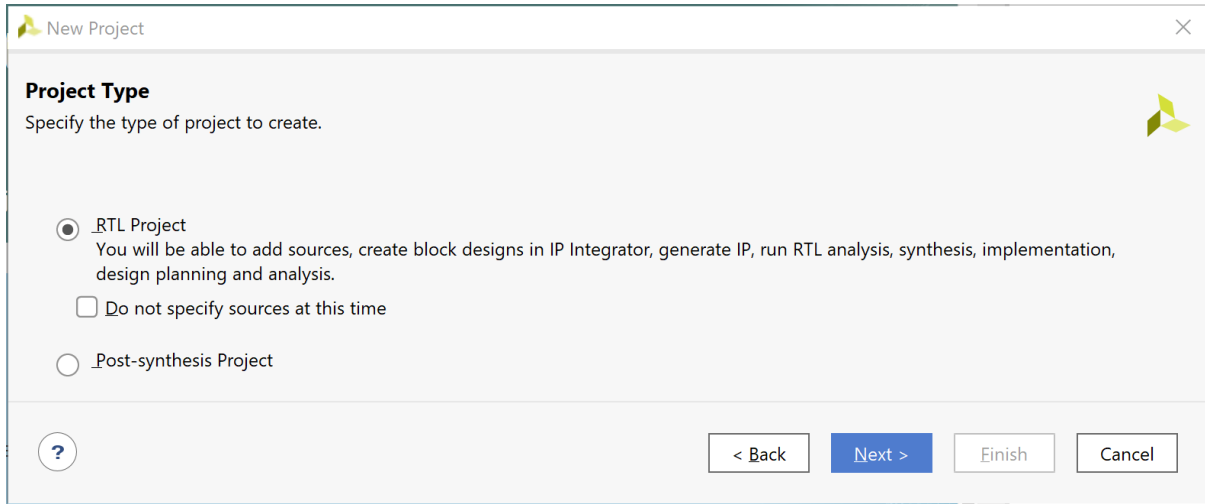
Project location:

☒ Create project subdirectory

Project will be created at: C:/RVfpgaLabs/Programs/Lab1/Project1

Figür 3. Proje Adı

Proje türünü RTL Project (Yazmaç-Aktarım Düzeyli Proje) olarak seç, Next'e (İleri) (Figür 4'e göz at) tıkla.



New Project

Project Type
Specify the type of project to create.

☒ **_RTL Project**
You will be able to add sources, create block designs in IP Integrator, generate IP, run RTL analysis, synthesis, implementation, design planning and analysis.

☐ Do not specify sources at this time

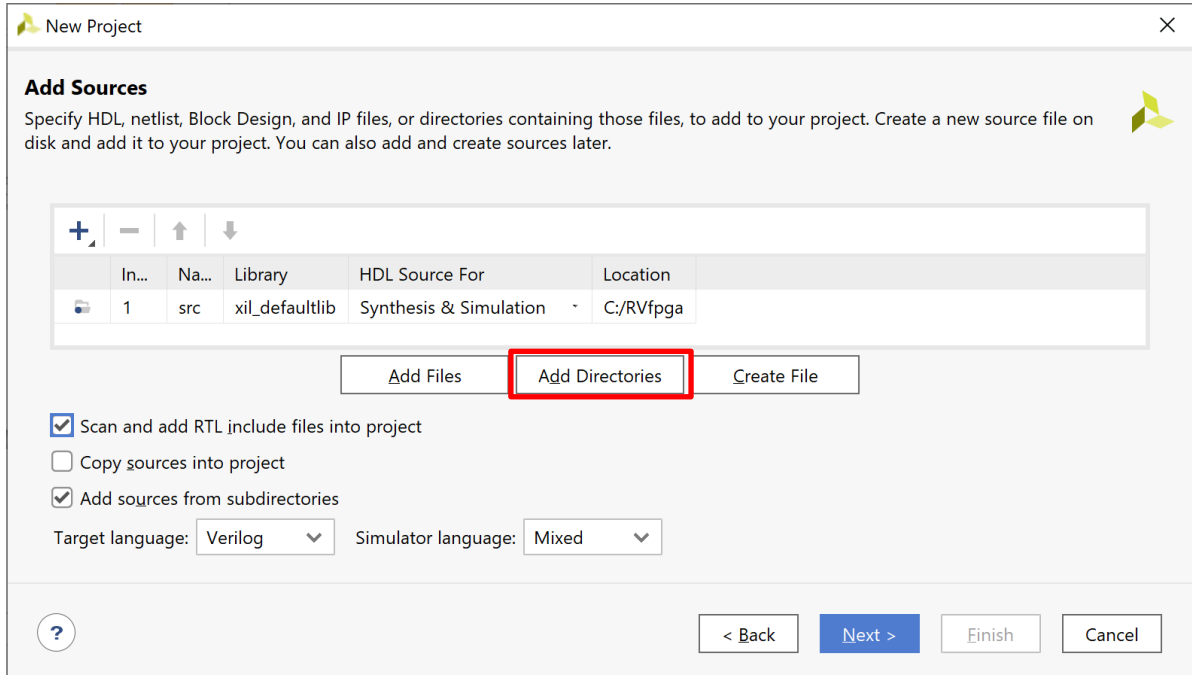
☐ **_Post-synthesis Project**

Figür 4. Yazmaç-Aktarım Düzeyli Proje

Adım 3. Yazmaç-Aktarım Düzeyli kaynak dosyalarını, kısıtlandırma dosyalarını ekle
Add Sources (Kaynaklar Ekle) penceresinde Add Directories'e (Dizinler Ekle) tıkla, *[RVfpgaPath]/RVfpga/src*'i (Figür 5'e göz at) seç. Şu iki seçeneğin seçildiğinin sağlamlasını yap (Figür 5'de görüldüğü gibi):

- Scan and add RTL include files into project (Yazmaç-Aktarım Düzeyli içerilecek dosyaları projeye tarayıp ekle)
- Add sources from subdirectories (altdizinlerden kaynaklar ekle)

Ardından Next'e (İleri) tıkla.



Figür 5. Kaynaklar Ekle

Şimdi system için kısıtlandırmaları ekleyeceksin. Bu dosyalar sinyal adlarını karttaki uçlarla eşler. Örneğin, Nexys A7 Alanda-Programlanabilir Geçit Dizisi kartındaki LEDler karttaki Alanda-Programlanabilir Geçit Dizisi uçlarına Baskılı Devre Kartındaki izler üzerinden bağlıdır. Yazmaç-Aktarım Düzeyli doğru sinyali doğru Alanda-Programlanabilir Geçit Dizisi ucuna bağlamak için Vivado'nun bunu bilmesi gerekir. Örneğin, *[RVfpgaPath]/RVfpga/src/rvfpga.xdc* dosyasındaki, bir Xilinx tasarım kısıtlandırmaları dosyası, şu satır Alanda-Programlanabilir Geçit Dizisi ucu H17'in düşük önemli LED'e (*o_led[0]*) eşlenip LVCMOS 3.3V sinyalleşmesini kullandığını gösterir:

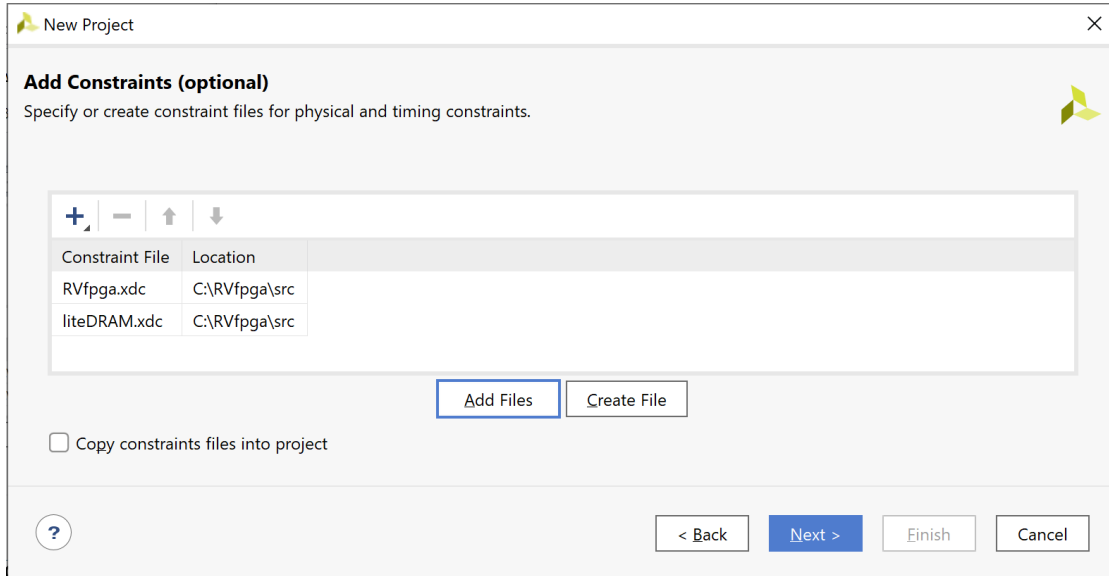
```
set_property -dict { PACKAGE_PIN H17 IOSTANDARD LVCMOS33 } [get_ports { o_led[0] }]
```

Önemli olarak sinyal adı *o_led* RVfpga'in Verilog kodunda Nexys A7 kartının LEDlerini sürmek için kullanılan addır.

Add Constraints (Kısıtlandırmalar Ekle) penceresinde Add Files'a (Dosyalar Ekle) tıklayıp şu iki dosyayı seç (Figür 6'ya göz at):

```
[RVfpgaPath]/RVfpga/src/rvfpga.xdc  
[RVfpgaPath]/RVfpga/src/litedram.xdc
```

Ardından Next'e (İleri) tıkla.

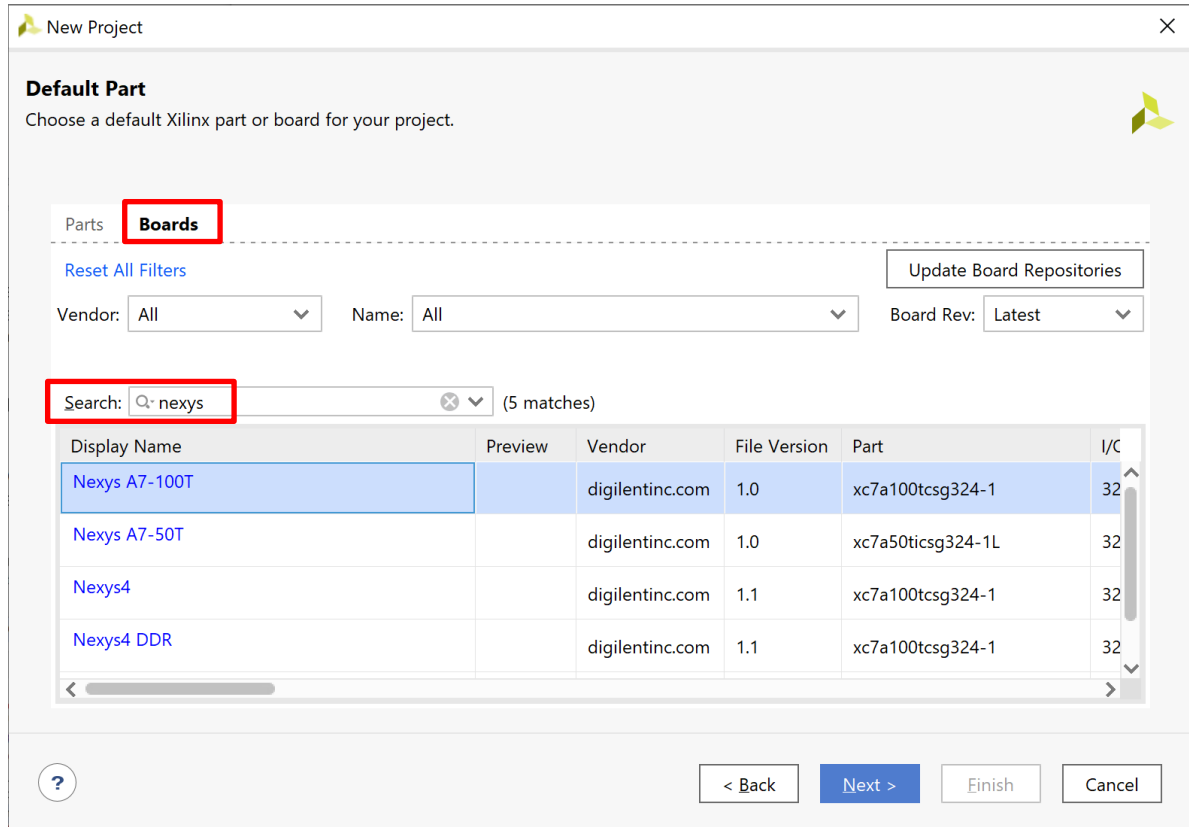


Figür 6. Kısıtlandırmalar Ekle

Adım 4. Nexys A7'ı hedef kart olarak seç

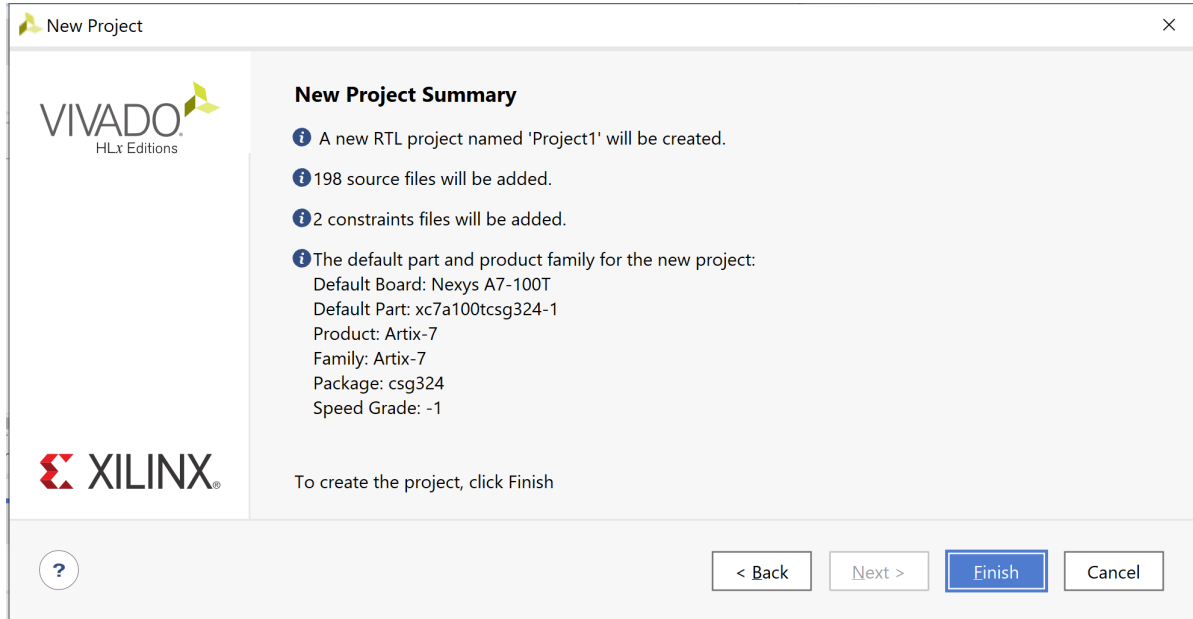
Default Part (Varsayılan Parça) penceresinde, click on Boards'a (Kartlar) tıkla, Nexys A7-100T'yi (Figür 7'ye göz at) seç. Sonuçları daraltmak için Search (Arama) kutusunu kullanabilirsin. Ayrıca güncel hedeflenen Alanda-Programlanabilir Geçit Dizisinin adının Part (Parça) sütununda listelendiğini göreceksin: xc7a100tcsg324-1. Bu onun 100k eşdeğer geçitli, CSG (yonga-ölçekli ızgara) paketli, 324 uçlu bir Xilinx Artix-7 FPGA olduğunu gösterir.

Next'e (İleri) tıkla.



Figür 7. Hedef kartı seç: Nexys A7-100T

New Project Summary (Yeni Proje Özeti) penceresinde Finish'e (Bitir) tıkla (Figür 8'e göz at).

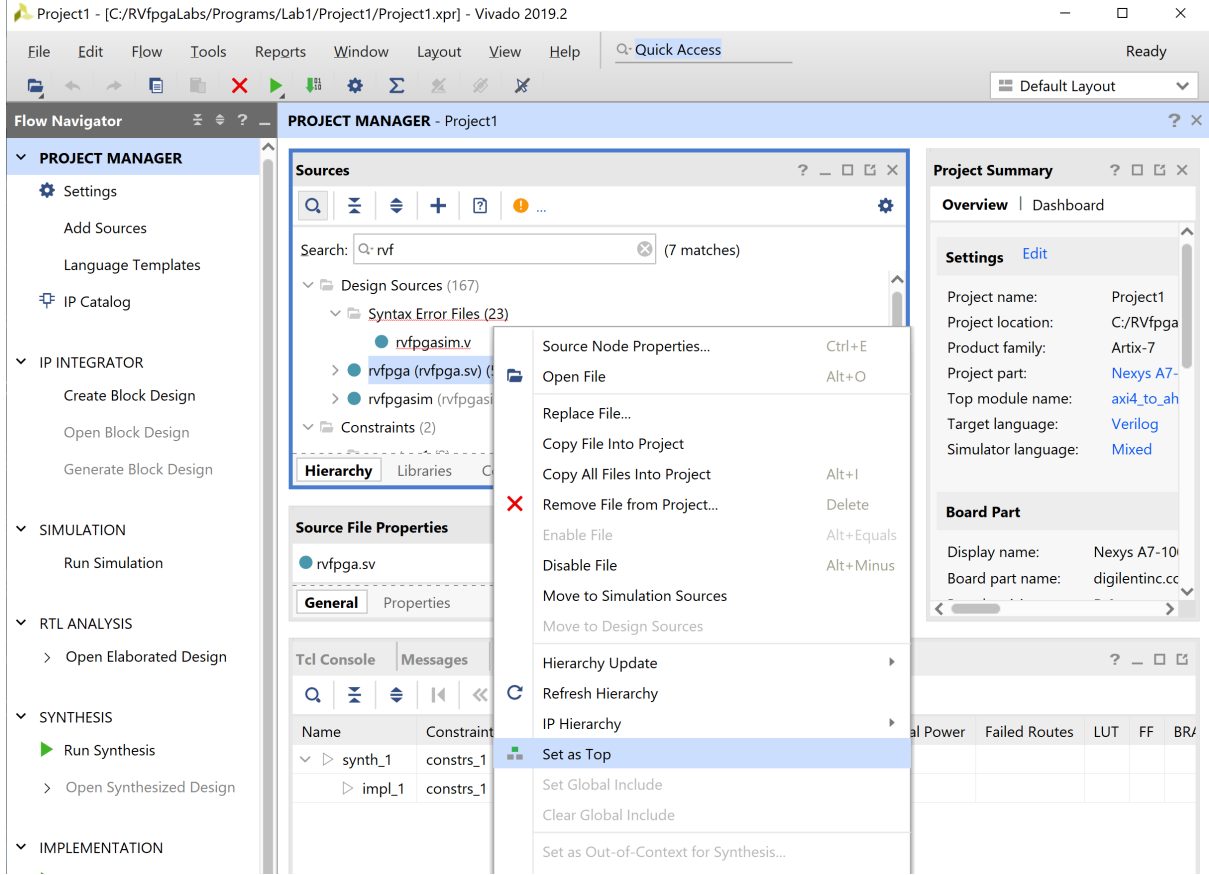


Figür 8. Yeni Proje Özeti Penceresi

Önemli olarak proje kurulumu bitince dosyalarda Syntax Errors (Sözdizim Hataları) bulunduğunu gösterecek – bu sonraki adımda onarılacak.

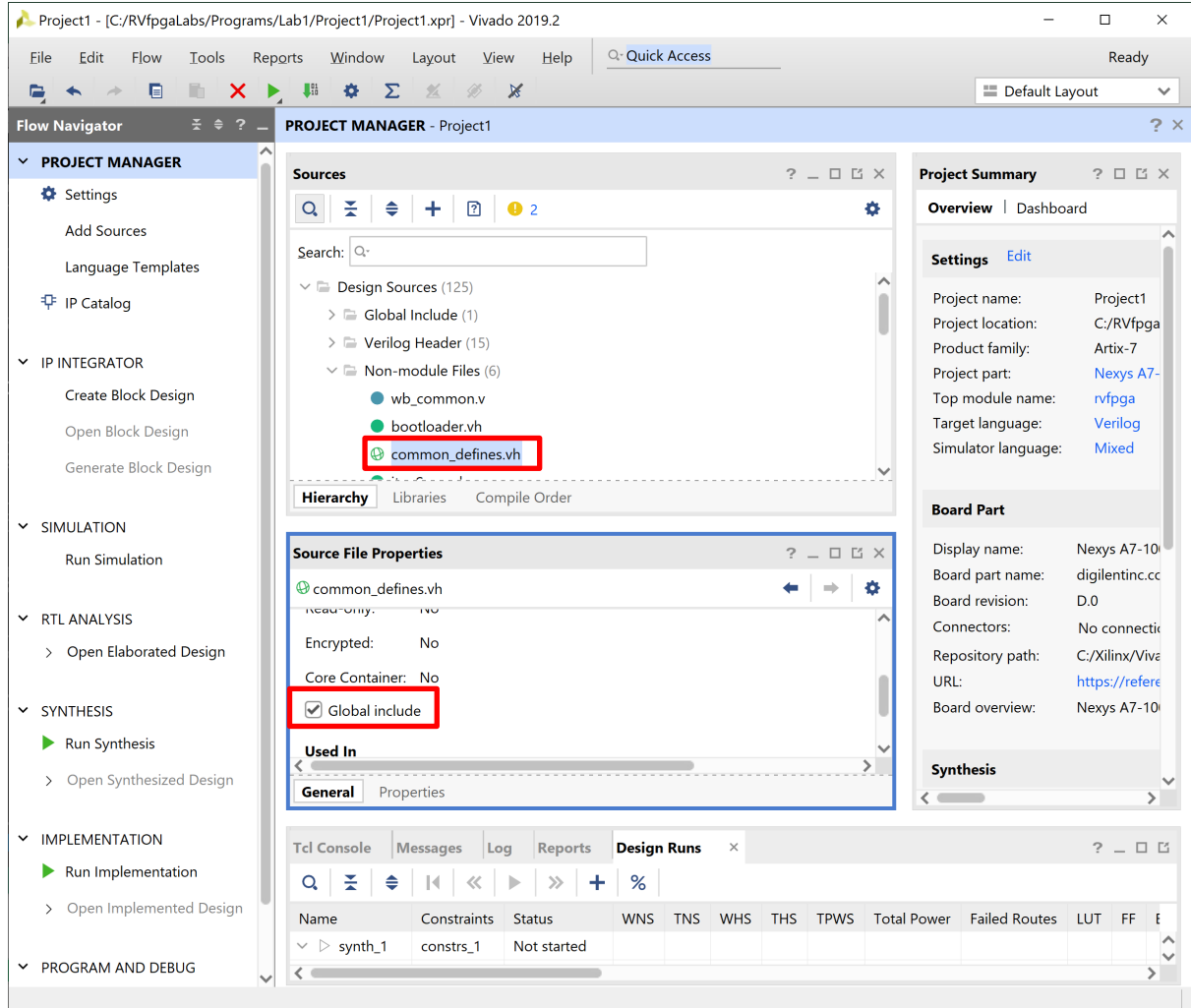
Adım 5. rvfpga'i Üst Modül olarak, common_defines.vh'i genel olarak ayarla

Proje ilk değerlerini alacak. Şimdi rvfpga modülünü üst modül olarak ayarlayacaksınız. Sources (Kaynaklar) panelinde Design Sources'ın (Tasarım Kaynakları) altına kaydır, rvfpga modülüne sağ tıkla, Set as Top'ı (Üst olarak Ayarla) (Figür 9'a göz at) seç. Gösterildiği gibi rvfpga modülünü bu adı arama kutusuna yazarak da bulabilirsin. Bu rvfpga'i hiyerarşide en yüksek düzeyli modül olarak ayarlayıp, Alanda-Programlanabilir Geçit Dizisine sentezlemek, gerçekleştirmek üzere hedef olarak ayarlar. rvfpga'i üst modül olarak ayarladıktan sonra hiyerarşi güncellenecek.



Figür 9. rvfpga'ı üst modül olarak ayarla

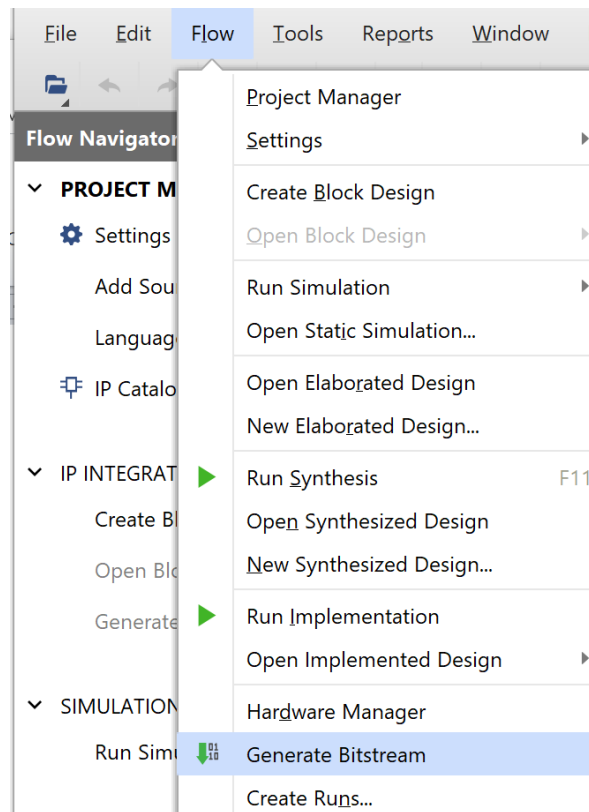
Şimdi, yine Design Sources (Tasarım Kaynakları) altındaki Sources (Kaynaklar) panelinde, Non-modules (Modül-olmayanlar) dosya grubunu genişletip common_defines.vh'e tıkla. Ardından dosyanın özellikleri, Sources (Kaynaklar) panelinin hemen altında, Source File Properties (Kaynak Dosya Özellikleri) panelinde açılacak. Kutuyu işaretlemek için Global Include'a (Genel İçerilecek) (Figür 10'a göz at) tıkla. Hiyerarşi şimdi güncellenip dosyayı Design Sources/Global Include'da (Tasarım Kaynakları/Genel İçerilecek) içerecektir. Önemli olarak Syntax Error Files (Sözdizim Hata Dosyaları) gelecek bir adımda yok olacak.



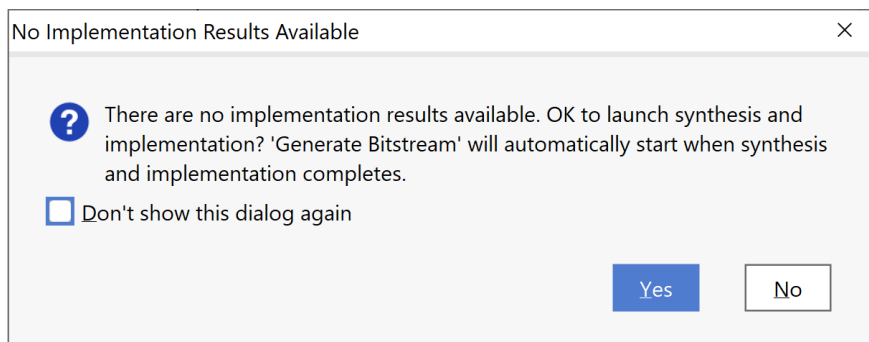
Figür 10. common_defines.vh'i Genel içerecek dosya olarak ayarla

Adım 6. Veri Akışı oluştur

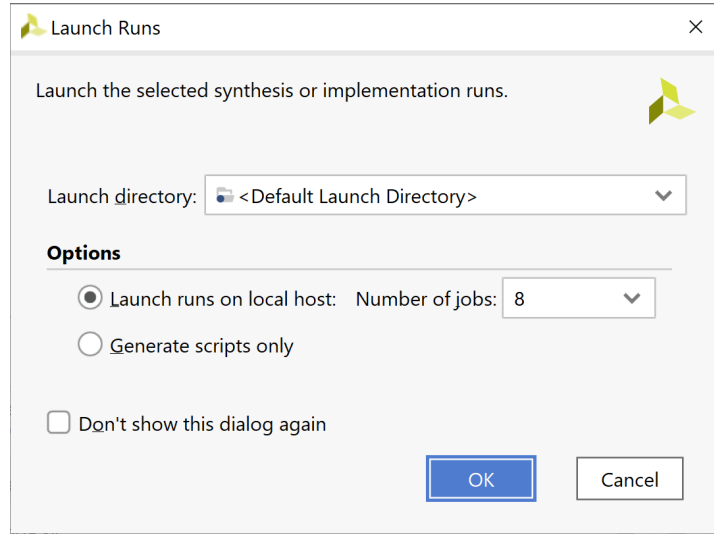
Şimdi Figür 11'de gösterildiği gibi Flow (Akış) → Generate Bitstream'e (Veri Akışı Oluştur) tıkla. Erişilebilir gerçekleştirme sonucu olmadığını söyleyip sentezle gerçekleştirme (Figür 12'yi gör) başlatmanı soran bir pencere belirebilir. Yes'e (Evet) tıkla. Ardından Launch Runs (Çalıştırmaları Başlat) penceresinde (Figür 13'e göz at) OK'a tıkla. Bu adım RVfpga'i (projedeki Verilog ile SystemVerilog dosyalarınca tanımlandığı gibi) sentezler, Alanda-Programlanabilir Geçit Dizisine eşler, veri akışını oluşturur. Bu işlem normalde, bilgisayarının hızına bağlı olarak, 20-50 dakika arası sürer.



Figür 11. Veri Akışı Oluştur

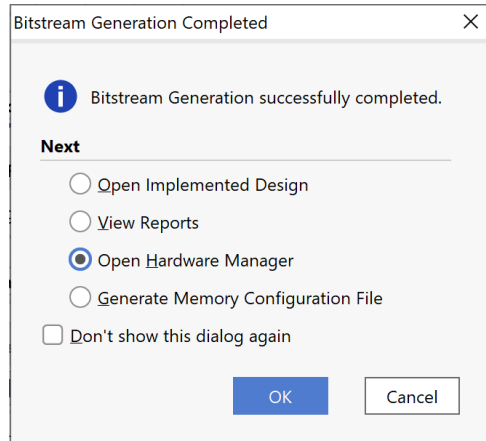


Figür 12. Sentezleme ile gerçekleştirme başlatma penceresi



Figür 13. Çalıştırmaları Başlat

Veri akışı oluşmasının ardından Figür 14'te gösterildiği gibi bir pencere belirecek. Pencereyi kapatmak için sağ-üst köşedeki **X** butonuna tıkla.



Figür 14. Veri Akışı Oluşumu Bitirildi

Şimdi RVfpga sistemini kendin kurguladığına göre, 6-10 arası Deneylerde RVfpga'e değişiklikler yaptıktan sonra yeniden kurgulayabileceksin. Şimdilik PlatformIO'yu kullanarak programlar indirip çalıştırmak için az önce kurguladığın RVfpga sistemini kullanmaya başlayacaksın.

RVfpga'i Nexys A7 kartına indirmek için PlatformIO'yu kullanman önerilir. Bu yöntem RVfpga İlk Kullanım Kılavuzunun (GSG) 6.A bölümünde detaylıca tanımlanmıştı. Yine GSG'de değişik örnekler için tanımlandığı gibi (6.B'den 6.H'ye), RVfpga sistemini Nexys A7 kartındaki Alanda-Programlanabilir Geçit Kartına indirmenin ardından, RVfpga'e programları indirip, çalıştırmak/ayıklamak için PlatformIO'yu kullanacaksın.

RVfpga İlk Kullanım Kılavuzunun Bölüm 7'sinde tanımlandığı gibi RVfpga'de çalışan programların simülasyonunu yapmak için Verilator'ü, bir Donanım Tanım Dili simütatörü, de kullanabilirsin. Bu Yazmaç-Aktarım Düzeyli simülasyonlar yazılım programı çalışırken alçak-düzeyli donanım sinyallerini görmeni sağlar. Sen 6-10 arası Deneylerde RVfpga'i genişletip, yaptığın değişiklikleri deneyip ayıklarken Verilator'ü sıkça kullanacağız.