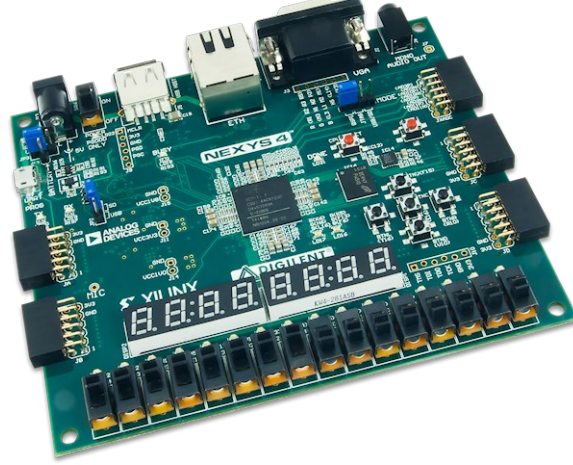


2. Kullanılan Yazılım ve Kart

Bu kitapta Bölüm 10'da verilen uygulamalar Digilent firması tarafından piyasaya sürülmüş olan NEXYS 4 deney kartı üzerinde çalıştırılacaktır. NEXYS 4 kartı üzerinde Xilinx firmasına ait Artix-7 100t FPGA yongası bulunmaktadır. Yeni başlayan kullanıcılar için pek çok uygulamayı doğrudan gerçekleştirmeye imkân sunacak bileşenler de kart üzerinde mevcuttur. Şekil 2-1'de kullanacağımız kartın görüntüsü verilmiştir.



Şekil 2-1 Nexy4 kartı

Kart üzerinde bulunan Artix-7 100t FPGA'sının temel özellikleri şu şekildedir:

- 15.850 Mantık Ünitesi (Logic Slices)
- 4860 Kbits Block Ram
- 240 DSP Ünitesi
- 450MHz'e kadar dahili saat hızı (Clock Speed)
- Dahili Analog-Dijital dönüştürücü (XADC)

Kart üzerinde kullanıcılara sağlanan donanım bileşenleri ise şu şekildedir:

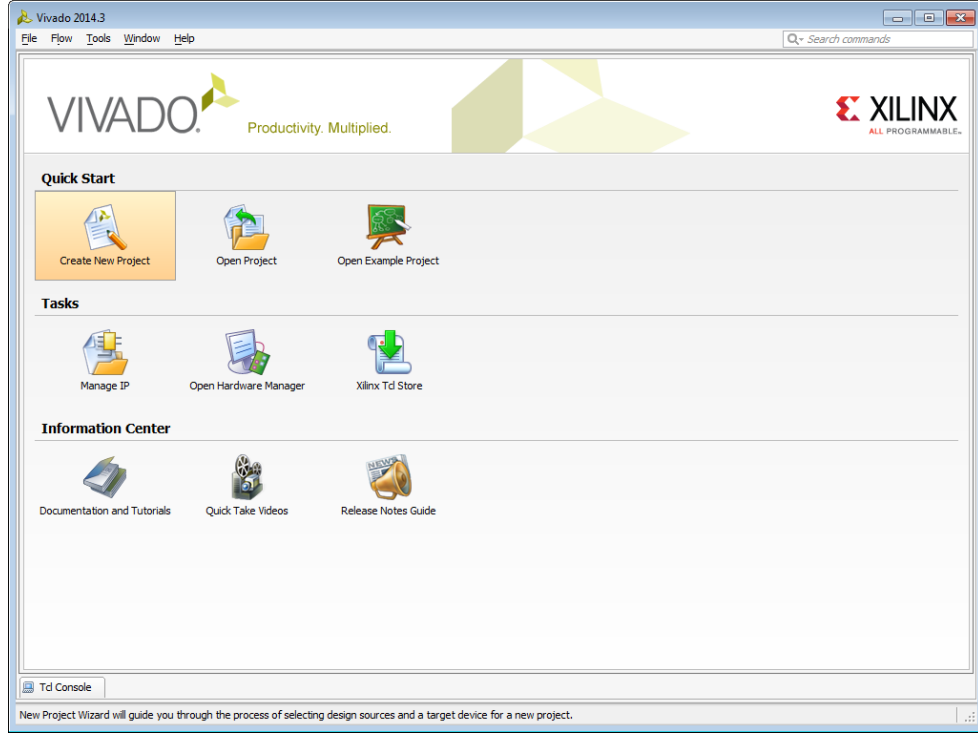
- 16 Adet Anahtar (switch)
- 16 Adet Led
- 2 Adet üç renkli led
- 2 Adet dört basamaklı 7 Segment Display
- 12 Bit VGA çıkışı
- 10/100 Mbs Ethernet Portu
- USB HID Desteği (Host)
- USB-UART Bağlantı Birimi
- Micro SD Kart Yuvası
- 4 Adet PMOD* Bağlantı Yuvası
- 1 Adet XADC PMOD Bağlantı Yuvası
- Sıcaklık Algılayıcısı
- 3 Eksen İvme Ölçer
- PDM Mikrofon
- PWM Ses Çıkışı
- 16Mbyte CellularRam
- Seri Flash
- Digilent USB-JTAG Bağlantısı

*PMOD : Digilent firması tarafından FPGA'larda ve Mikrodenetleyicilerde çevresel birimleri haberleştirmek için geliştirilmiş bir arabirim standardıdır.

2.1. Xilinx Vivado İle Yeni Proje Oluřturma

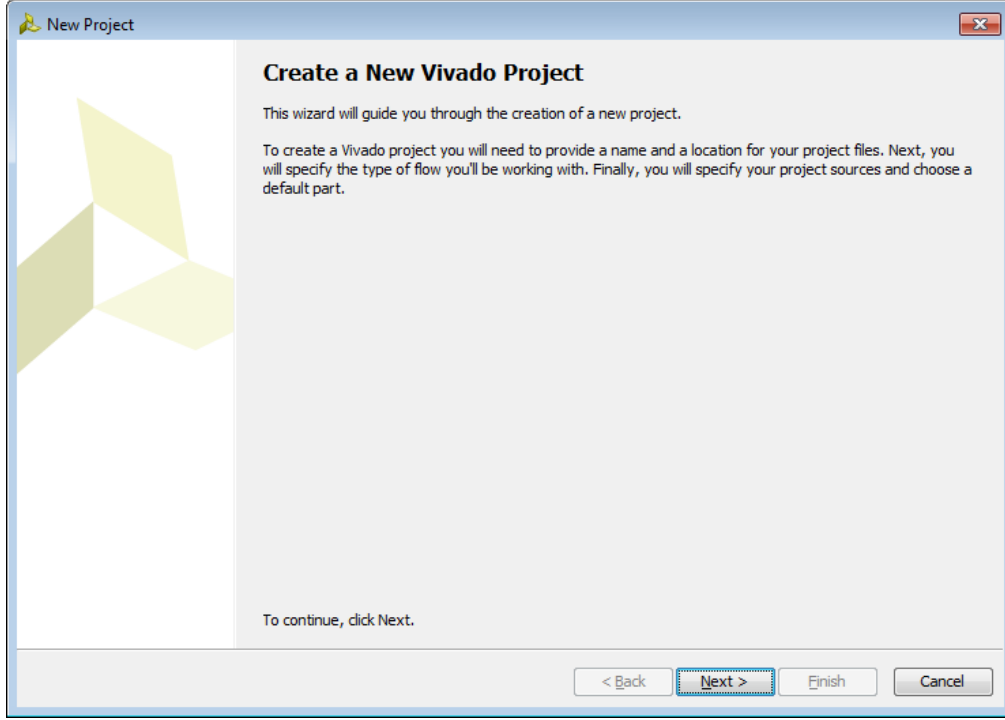
Bu kitapta anlatılan tüm çalışmalar Xilinx firması tarafından sağlana Xilinx Vivado Design Suite (WebPACK) kullanılarak gerekleřtirilmiřtir. Xilinx Vivado Design Suite (WebPACK) geliřtirme ortamı www.xilinx.com adresinden indirilebilir ve ücretsiz bir řekilde kullanılabilir. Bu kitap yazılırken var olan en güncel sürüm Xilinx Vivado 2014.3 sürümüdür ve anlatımlar bu sürüm temel alınarak yapılmıřtır.

Vivado programını kurduktan sonra programı bařlattığınızda karřınıza řekil 2-2’de gösterilen ekran gelecektir.



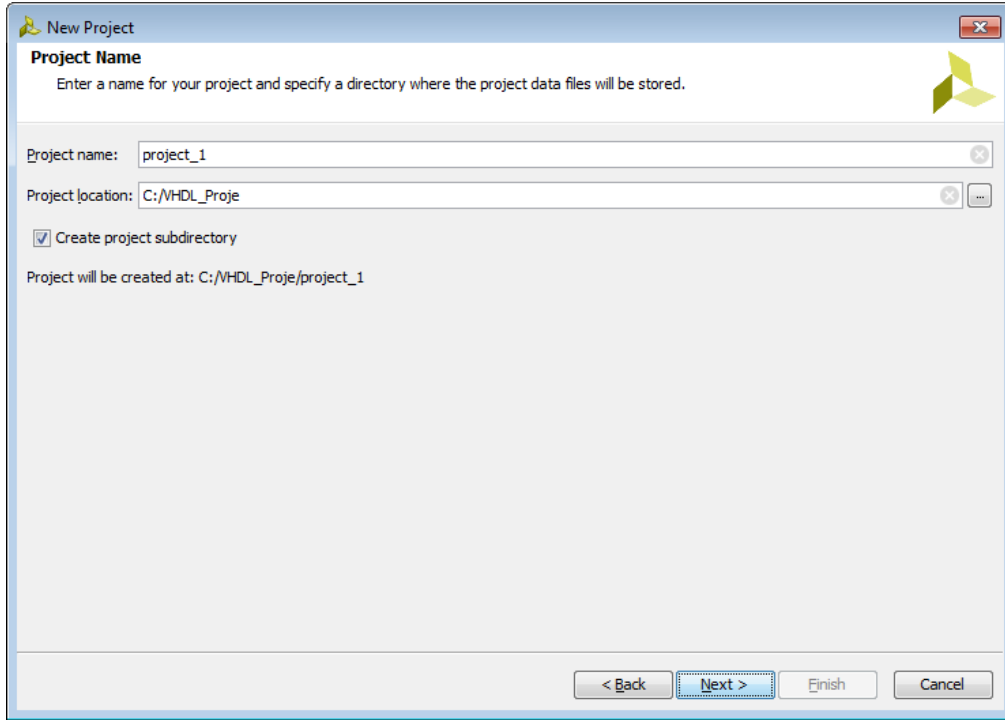
řekil 2-2 Aılıř Ekranı

Aılıř ekranı karřımıza geldikten sonra **Create New Project** sembolünü seçerek devam ediyoruz. Bir sonraki ekranda **Next** diyerek ilerliyoruz (řekil 2-3).



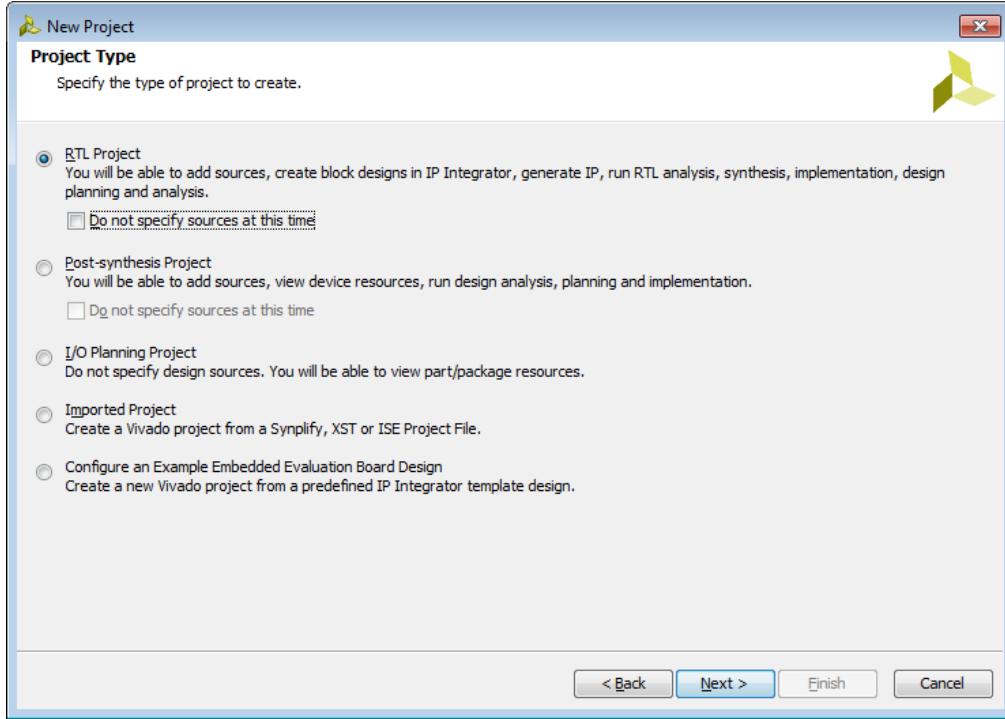
Şekil 2-3 Proje Oluşturma - 1

İlk olarak projemizin ismini ve çalışma dizinimizi belirliyoruz. Şekil 2-4'te bu ekrana ait görüntü verilmiştir.



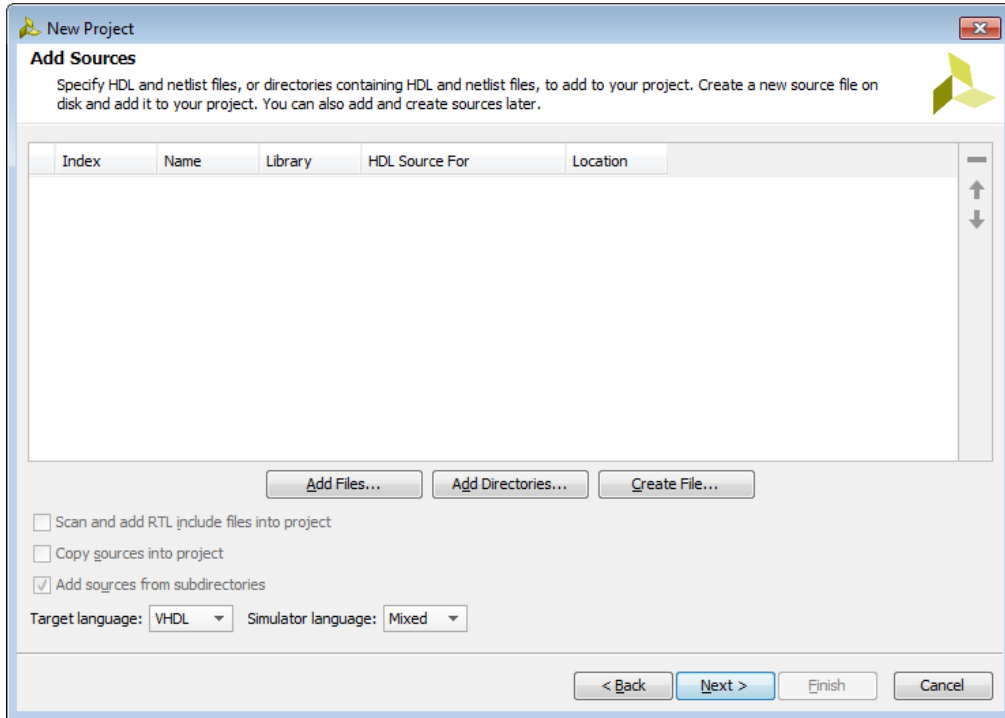
Şekil 2-4 Proje Oluşturma - 2

Bir sonraki ekranda ise oluşturacağımız projenin türünü seçiyoruz. Bu ekranda **RTL Project** seçeneğini seçerek **Next** tuşu ile devam ediyoruz. Şekil 2-5'te bu ekrana ait görünüm verilmiştir.



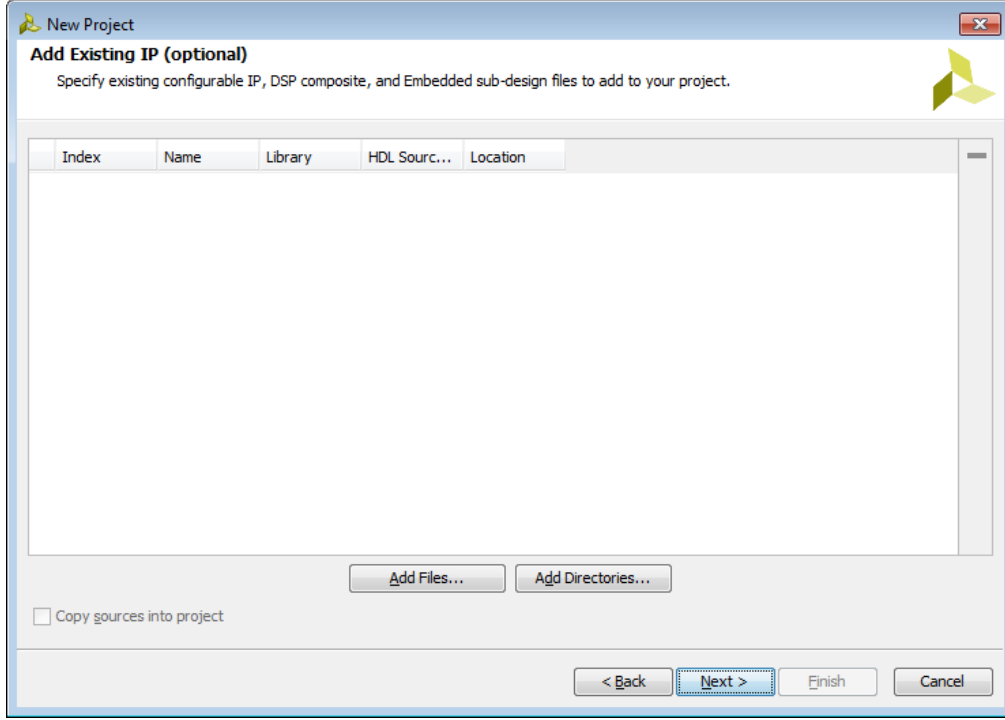
Şekil 2-5 Proje Oluşturma - 3

Proje türünü seçtikten sonra projede kullanılacak kaynak dosyaların seçiminin yapıldığı ekranla devam ediyoruz. Burada **Target Language** kısmında VHDL dilini seçiyoruz. Bu ekranda daha önceden kullanılan dosyaları seçmek mümkün olduğu gibi, yeni kaynak dosyaları da oluşturmak mümkündür. Fakat şimdilik bu ekrana **Next** diyerek devam ediyoruz.



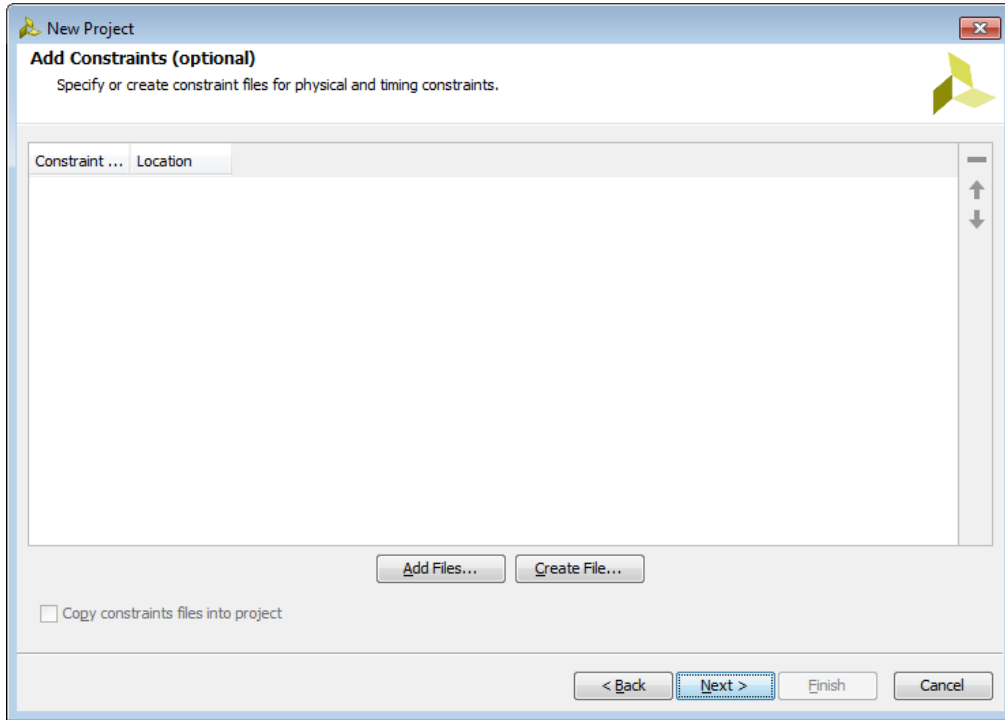
Şekil 2-6 Proje Oluşturma - 4

Kaynak dosya seçim ekranından sonra var olan IP'lerin (Intellectual Property Core) eklendiği ekranlar karşılaşmaktayız. Bu ekranda hiçbir şey değiştirmeden **Next** ile devam ediyoruz (Şekil 2-7).



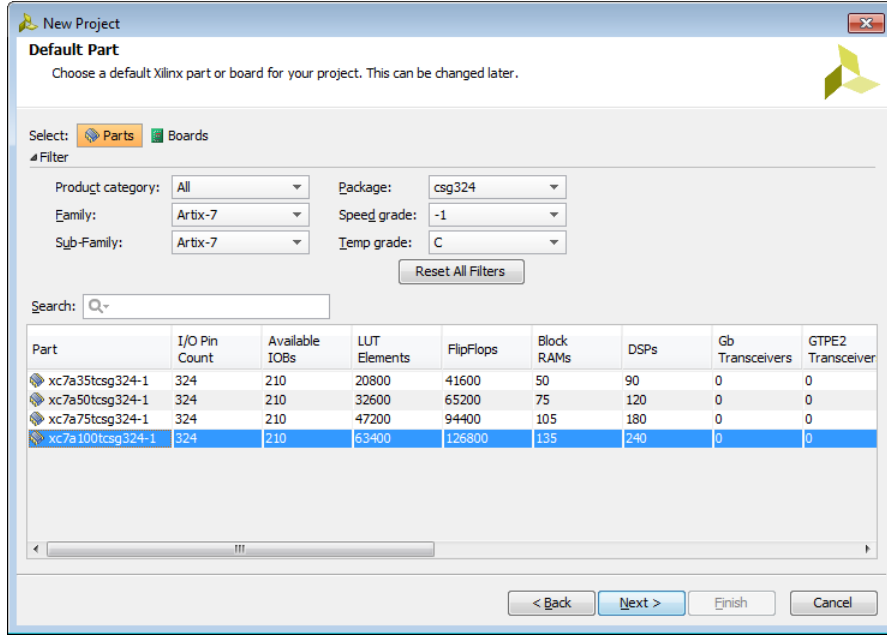
Şekil 2-7 Proje Oluşturma - 5

Sıradaki ekranda ise tasarımla alakalı kısıtlamaların belirlendiği dosyaların eklemek ya da yenisini oluşturmak mümkün olmaktadır. Bu aşamada herhangi bir ayar yapmayacağımız için **Next** diyerek devam ediyoruz (Şekil 2-8).



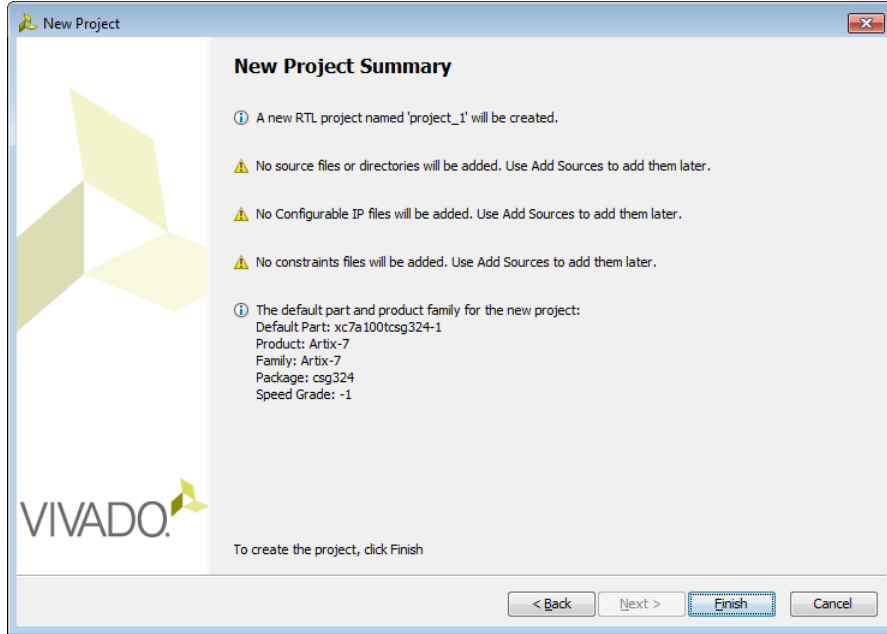
Şekil 2-8 Proje Oluşturma - 6

Açılan ekranda karşımıza parça seçim ekranı gelmektedir. Bu aşamada elimizdeki karta uygun ayarlamaların yapılması gerekmektedir. NEXYS4 kartı için gerekli seçimlere ait görüntü Şekil 2-9'da verilmiştir. Burada mevcut olan listeden **xc7a100tcsg324-1** olarak adlandırılmış cihazı seçiyoruz.



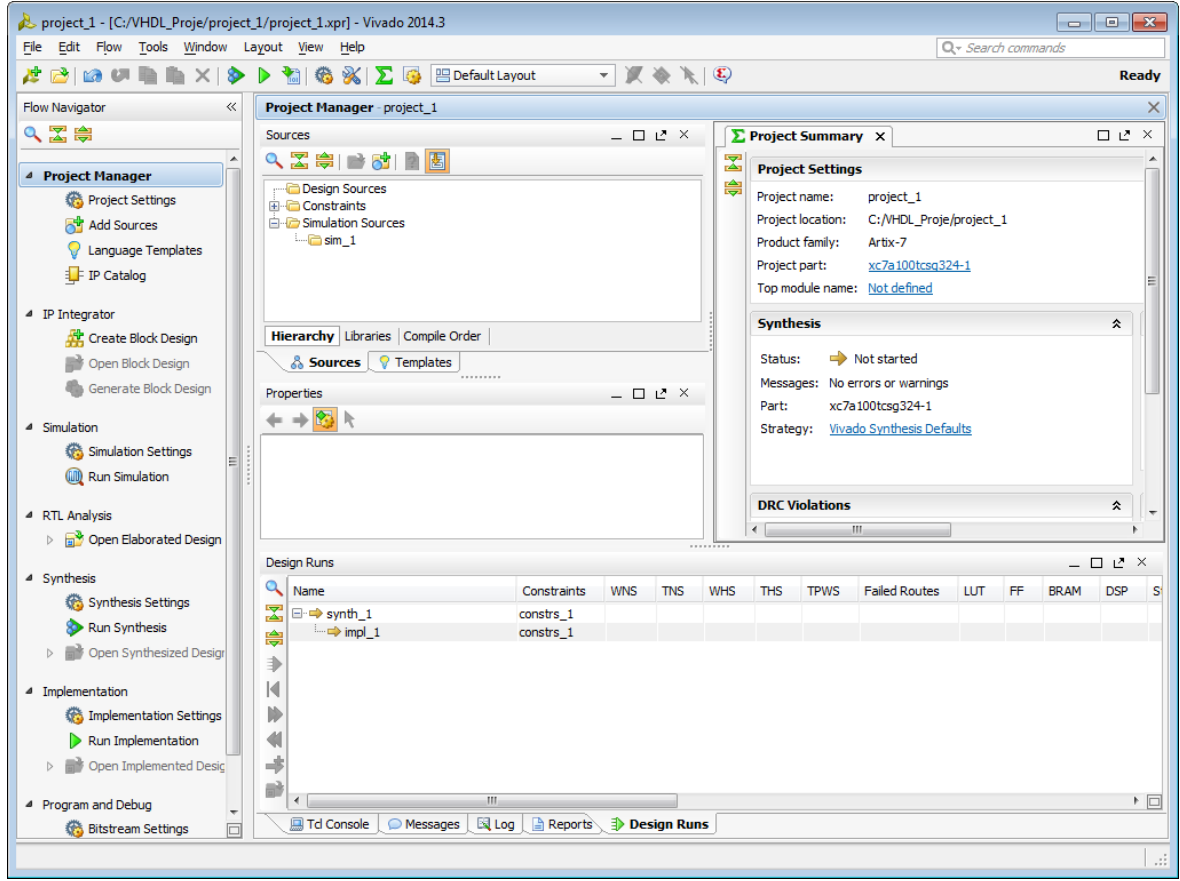
Şekil 2-9 Proje Oluşturma - 7

Son olarak karşımıza projemiz ile ilgili özet bilgilerin yer aldığı ekran gelmektedir. Bu ekranı da **Finish** diyerek kapatıyoruz (Şekil 2-10).



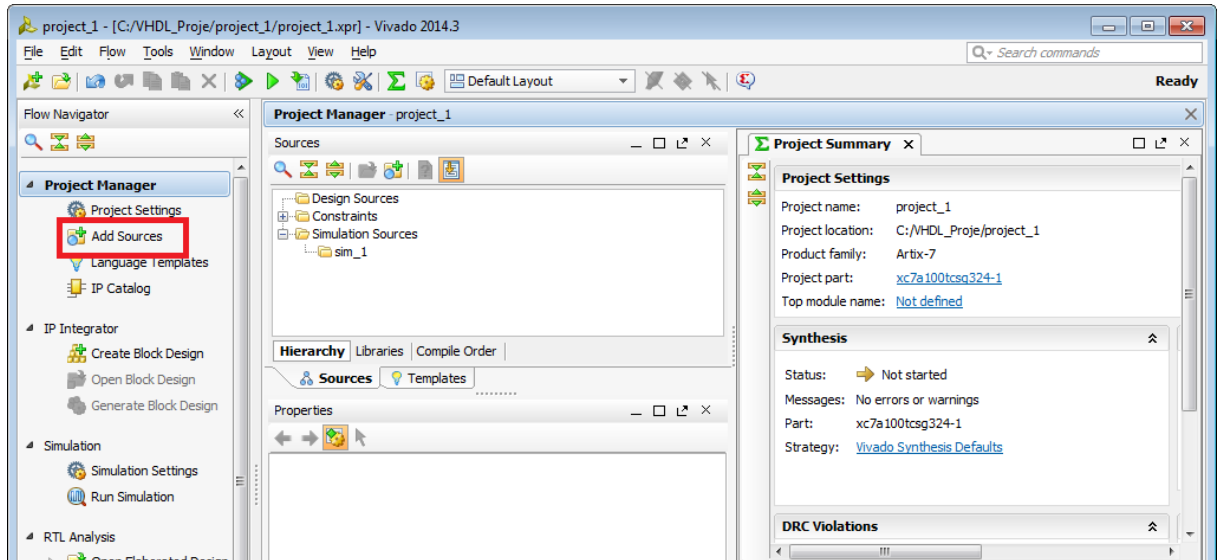
Şekil 2-10 Proje Oluşturma - 8

Tüm bu basamakları sorunsuz bir şekilde tamamladığımızda karşımıza gelecek ekran Şekil 2-11'de verilmiştir.



Şekil 2-11 Proje Ekranının Genel Görüntüsü

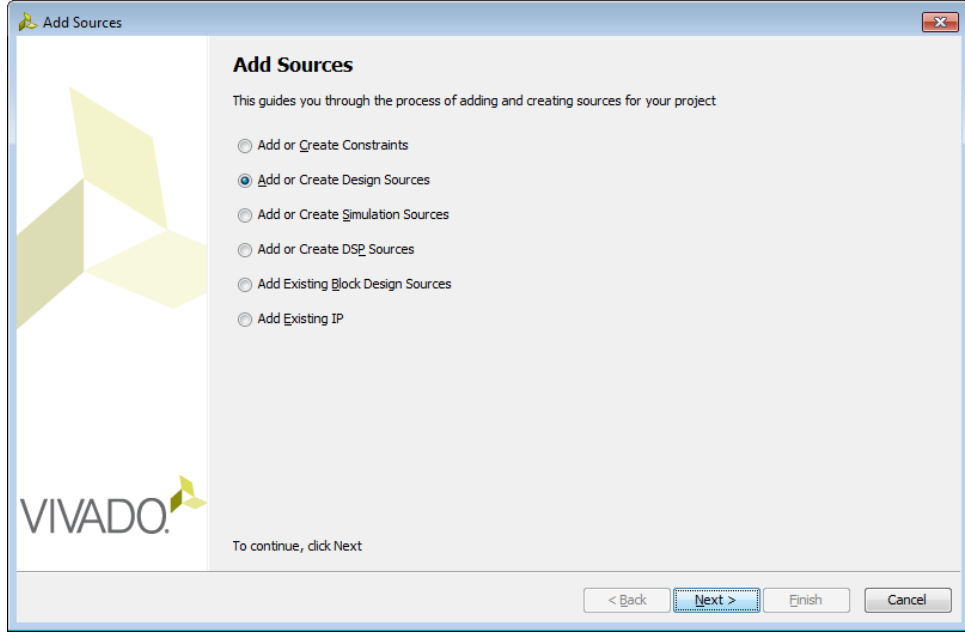
Şu ana kadar olan adımlarda sadece boş bir proje oluşturduk. Bundan sonraki adımlarda ise projemize sıfırdan kaynak dosyaları oluşturacağız. Bunun için ilk olarak ekranın sol tarafında bulunan **Project Manager** bölmesinden **Add Sources** başlığını seçiyoruz (Şekil 2-12).



Şekil 2-12 Dosya Ekleme-1

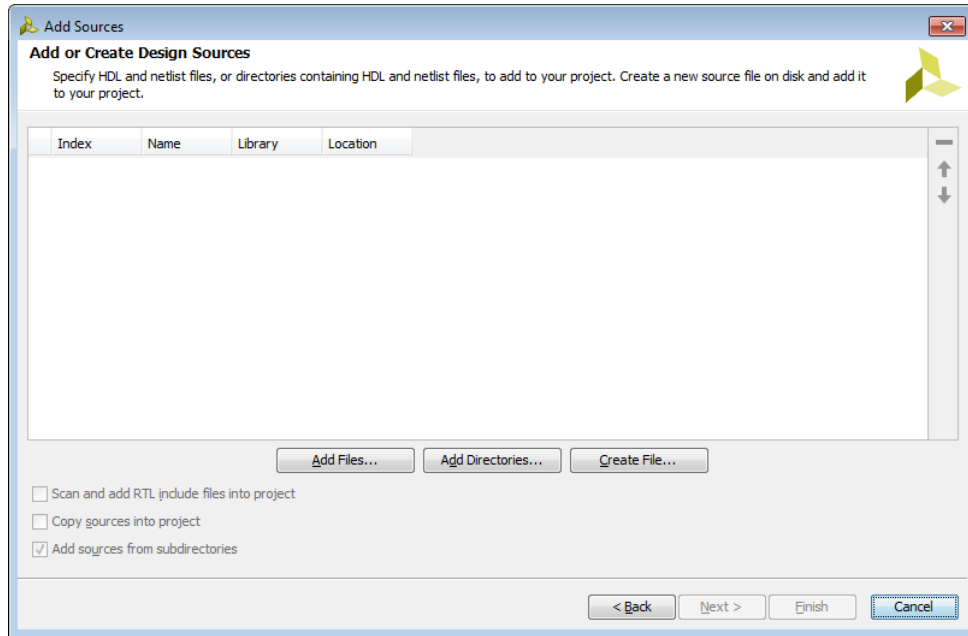
Karşımıza gelecek ekranda farklı amaçlar için seçenekler mevcut olmakla beraber bizi ilgilendiren seçenek **Add or Create Design Source** seçeneğidir. Diğer seçenekler yeni başlayanlar için bu aşamada önemli

değildir. Seçim ekranına ait görüntü Şekil 2-13'de verilmiştir. Bu ekrandan **Next** seçeneği ile bir sonraki aşamaya geçiyoruz



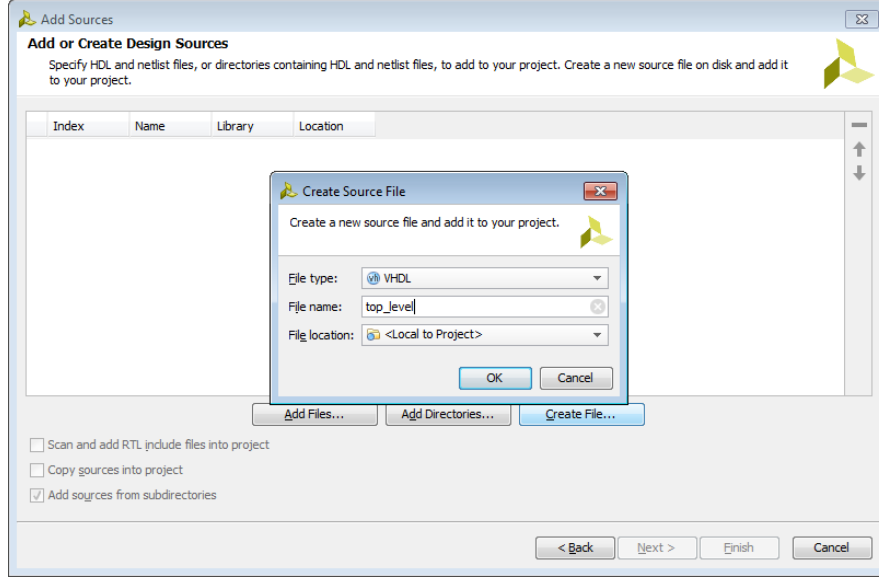
Şekil 2-13 Dosya Ekleme-2

Karşımıza gelen ekranı daha önceki proje oluşturma aşamasında görmüştük. Aslında projeye ait kaynak dosyaları daha o ekranda iken de oluşturulabilirdi fakat tüm işleyişi göstermek adına o ekran atlanıp ayrıca nasıl dosya oluşturulacağından bahsedilmiştir. Bu ekranda **Create File** yazısını seçip devam ediyoruz (Şekil 2-14).



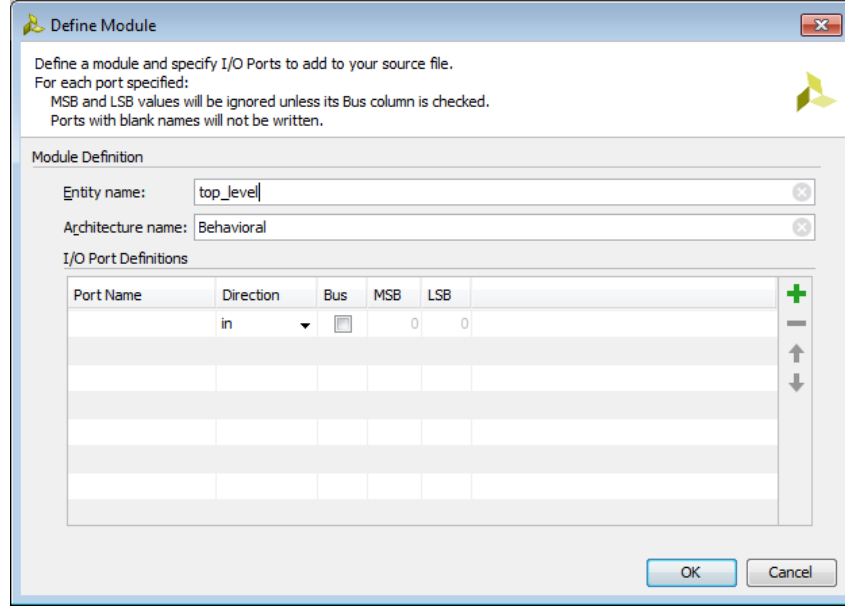
Şekil 2-14 Dosya Ekleme-3

Karşımıza çıkacak olan ufak pencerede oluşturacağımız dosyaya ait temel ayarlamaları yapıyoruz. **File Type** kısmından **VHDL**'i seçiyoruz. **File Name** kısmında ise oluşturacağımız dosyaya ait isimlendirmeyi yapıyoruz. Bu noktada isim verirken Türkçe'ye has ı,ş,ö,ç,ğ,ü gibi karakterleri kullanmamaya dikkat ediyoruz. Bir diğer önemli ayrıntı ise dosya isminde boşluk kullanılmaması gerektiğidir. Eğer birden fazla kelime içeren bir isim kullanılacaksa alt çizgi (_) karakteri kullanılarak isimlendirme yapılabilir. Gerekli bilgileri girdikten sonra **OK** tuşuna basarak pencereyi kapatıyor ve **Next** tuşuna basarak sonraki aşamaya geçiyoruz (Şekil 2-15).



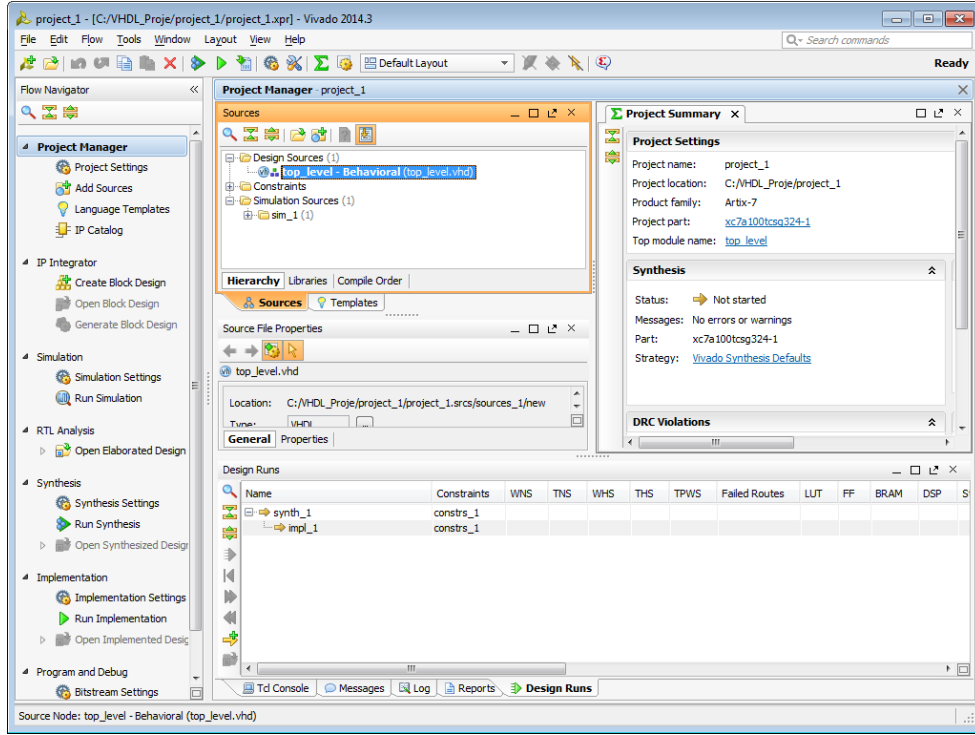
Şekil 2-15 Dosya Ekleme-4

Bu ekranda oluşturduğumuz dosyaya ait temel VHDL tanımlamalarının yapıldığı görülmektedir. Yeni başlayanlar için bu kısım şu anda bir şey ifade etmese de kitapta ilerledikçe burada var olan tanımlamaların ne işe yaradığı anlaşılacaktır. O yüzden bu ekrana da **OK** diyerek sonraki aşamaya geçiyoruz. Bu örnekte dosya ismi olarak **top_level** ismi kullanılmıştır (Şekil 2-16).



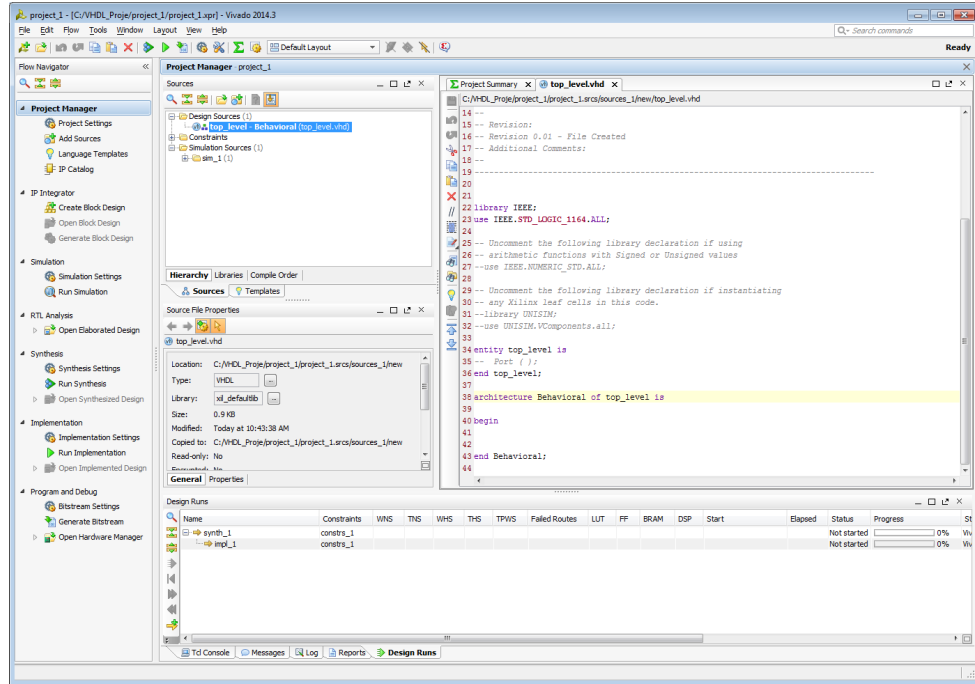
Şekil 2-16 Dosya Ekleme-5

Son olarak tekrar proje ekranına döndüğümüzde oluşturduğumuz dosyanın **Sources** bölümünde **Design Sources** başlığı altında gösterildiği görülmekte olacaktır. Şekil 2-17’de bu duruma ait görüntü verilmiş olup örnek dosya adı olarak **top_level** seçildiği unutulmamalıdır.



Şekil 2-17 Dosya Ekleme-6

Son olarak oluşturduğunuz dosyaya çift tıklarsanız, dosya içeriği ekranın sağ tarafında görüntülenecek ve Şekil 2-18’de verilen ekrana benzer bir ekran ile karşılaşılacaktır.



Şekil 2-18 Genel Görünüm

2.2. Xilinx Vivado ile Benzetim

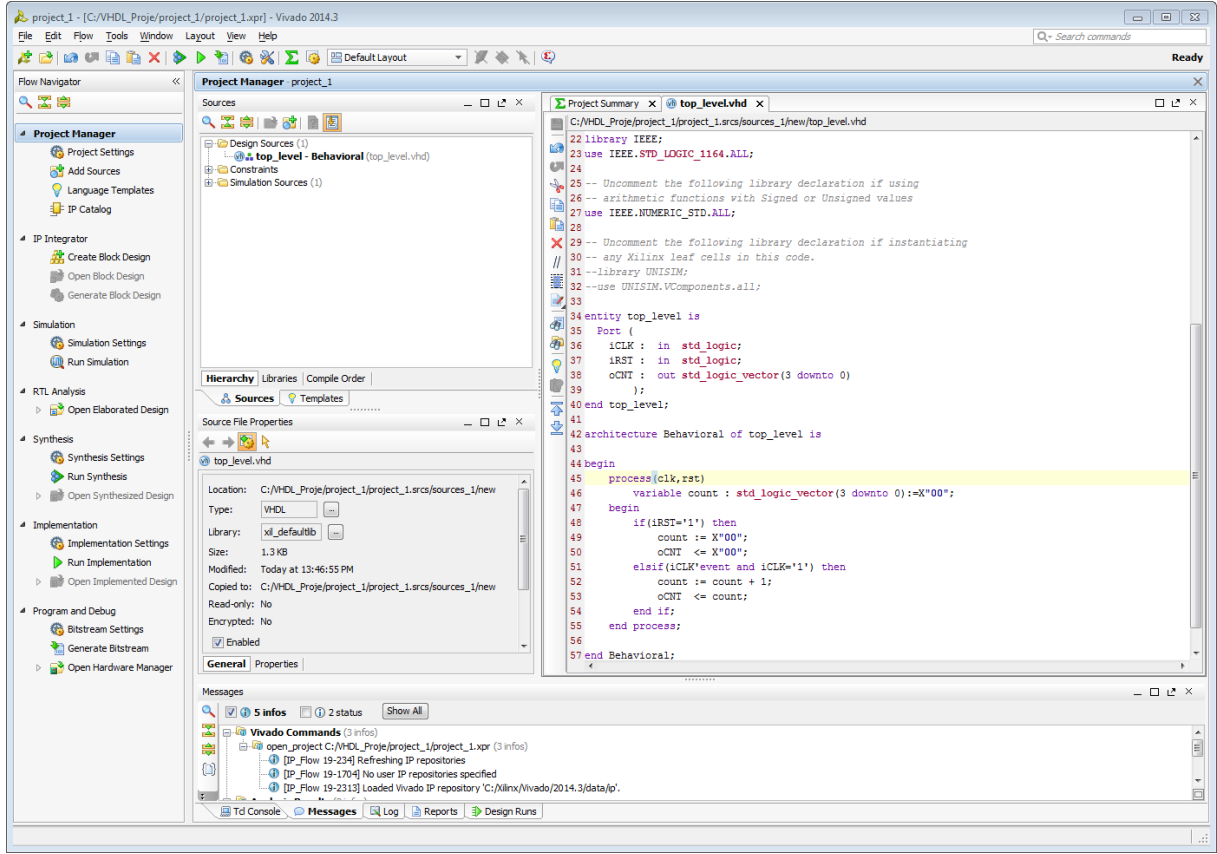
VHDL ile tasarım yapmanın önemli adımlarından biri de, yaptığımız tasarımın doğru çalışıp çalışmadığını öğrenmek için yapmamız gereken benzetim (simülasyon) adımıdır. Bu adım bize tasarımımızda bulunan olası

sorunları ortaya çıkarma fırsatı sunmaktadır. Bu sayede tasarımımızı fiziksel ortamda sınamak yerine sanal bir ortamda sınayıp doğru çalıştığına emin olduktan sonra sentezleme adımına geçebiliriz. Yapılan tasarıma bağlı olmakla birlikte sentezleme işlemi saatler sürebilmektedir. Kimse saatler süren bir sentezleme işleminden sonra yaptığı tasarımın hatalı olduğunu görmek istemez çünkü harcanan tüm emek ve saatler süren sentezleme işlemi boşa gitmiştir. Bu nedenlerden dolayı yaptığımız tasarımları mutlaka benzetim ortamında denemeliyiz.

Vivado çalışma ortamı bize gayet kullanışlı bir benzetim ortamı sunmaktadır. Yazdığımız kodu kolayca bu ortama geçirip, çalıştığına emin olduktan sonra sentezleme işlemine geçeriz. Bu anlatımda aşağıda kodu verilen basit bir 4 bitlik sayıcı tasarlanmış ve benzetimi yapılmıştır. Verilen örnek VHDL kodunda bulunan kavramlar bu aşamada bir şey ifade etmeyebilir fakat kitabın ilerleyen bölümlerinde bu kavramlar hakkında gerekli açıklamalar yapılmıştır.

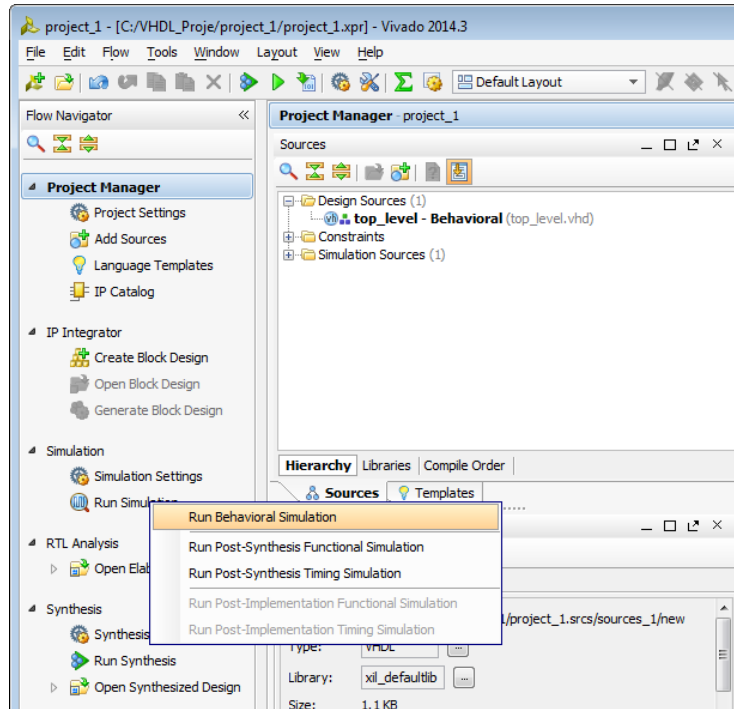
```
1. library ieee;  
2. use ieee.std_logic_1164.all;  
3. use ieee.numeric_std.all;  
4.  
5. entity top_level is  
6.   port(  
7.     iCLK : in std_logic;  
8.     iRST : in std_logic;  
9.     oCNT : out std_logic_vector(3 downto 0)  
10.  );  
11.end top_level;  
12.  
13.architecture Behavioral of top_level is  
14.  
15.begin  
16.  
17.   process(clk, rst)  
18.     variable counter : std_logic_vector(3 downto 0) := "0000";  
19.   begin  
20.     if(iRST = '1') then  
21.       null;  
22.     elsif(iCLK'event and iCLK = '1') then  
23.       count := count + 1;  
24.       oCNT  <= count;  
25.     end if;  
26.end process;  
27.end Behavioral;
```

Verilen kodu Vivado ortamında oluşturduğumuz projeye ekleyerek ilk aşamayı geçebiliriz. Kodu yazdıktan sonra ekranda oluşacak görüntü Şekil 2-19'de verilmiştir.



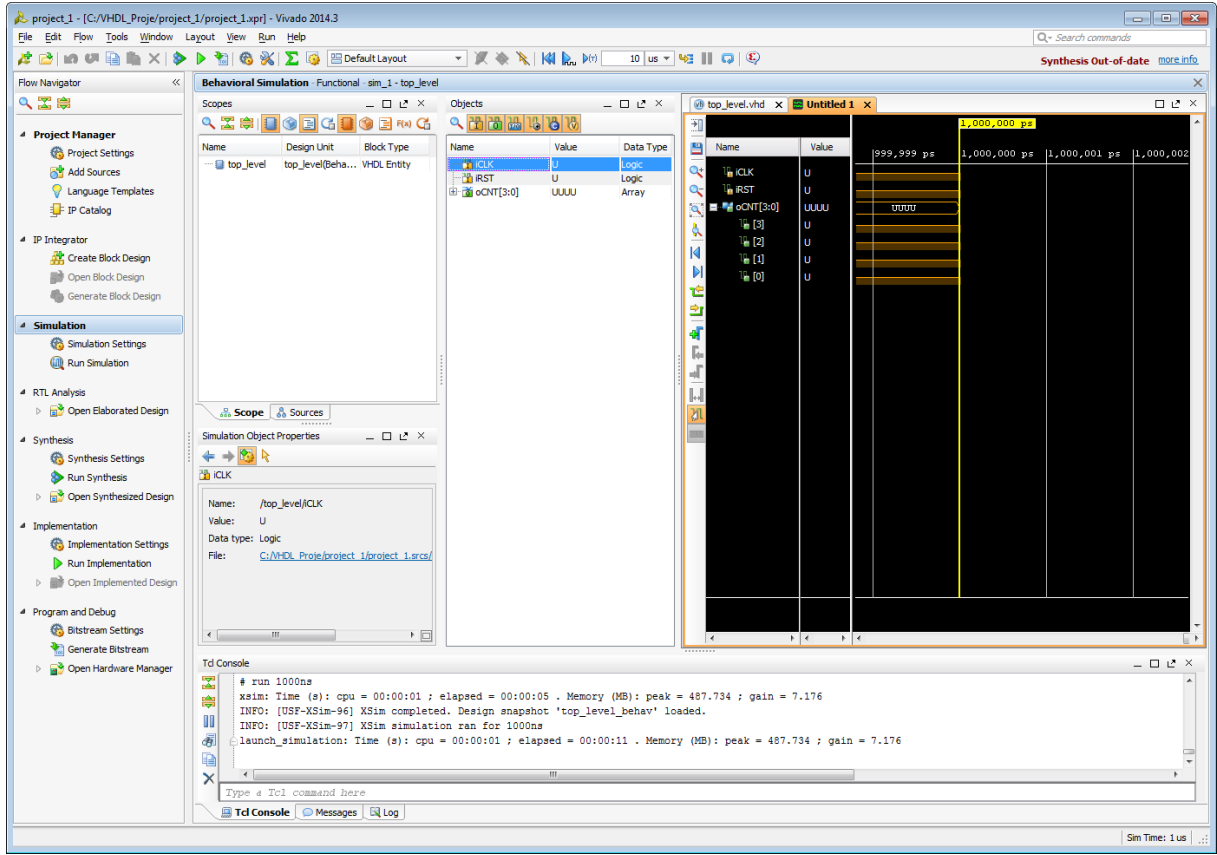
Şekil 2-19 Kod'a ait ekran görüntüsü

Benzetim işlemine başlamak için ekranın sol tarafında var olan başlıklardan **Simulation** başlığı altındaki **Run Simulation** yazısına tıkladığınızda açılan menüden **Run Behavioral Simulation** seçeneğini seçeriz. İlgili seçime ait ekran görüntüsü Şekil 2-20'de verilmiştir.



Şekil 2-20 Benzetim ortamının başlatılması.

Bu aşamadan sonra yazdığımız kod hızlıca benzetim ortamında aktarılmış olur. Eğer yazdığınız VHDL kodunda bir hata olursa bu aşama hata ile sonlanır ve benzetim ortamı açılmayarak, hata mesajı gösterilir. Eğer yazılan VHDL kodunda bir sorun yoksa Şekil 2-21’de verilen çalışma ortamı açılacaktır.

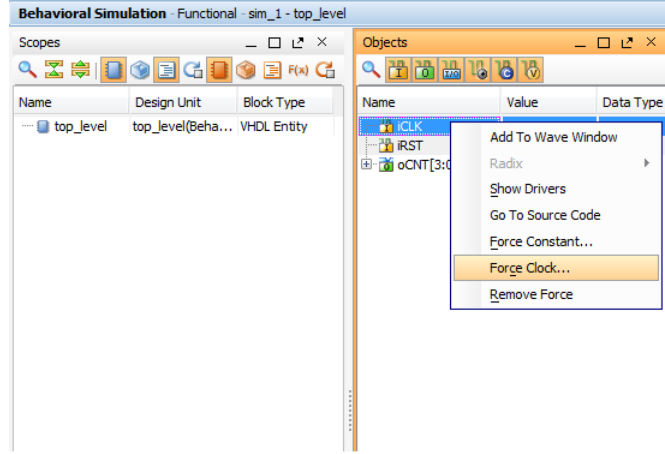


Şekil 2-21 Benzetim ortamının genel görüntüsü.

Açılan pencere temelde iki kısımdan oluşmaktadır. Ekranın sağ tarafındaki siyah bölgede tasarımımda var olan sinyallerin, giriş ve çıkışların o anki değerleri görülmektedir. Bizim yazdığımız kodda başlangıç değerleri ayarlanmadığı için giriş ve çıkış portlarına ait değerler U değerini göstermektedir. Burada U ifadesi, belirtilen sinyal değerinin bilinmediğini (unknown) ifade etmektedir.

Ekranın sol yarısında ise tasarımımda var olan sinyallere ait ismi, türü değeri gibi genel bilgiler görüntülenmektedir. Benzetim işlemini başlatabilmek için bu aşamada bir takım ayarlamaların yapılması gerekmektedir.

İlk olarak tasarımımda bulunan girişlerden hangisinin, tasarımımda tetikleyen saat işareti olduğunun belirtilmesi gerekmektedir. Bizim örnek tasarımımda bir saat işaretine ihtiyaç duyduğu için bu adım zorunlu bir adımdır. Eğer tasarımımda bu tarz bir giriş yoksa bu adım es geçilebilir. Örnek çalışmamızdaki saat işareti girişi **iCLK** olarak isimlendirilmiştir. **iCLK** işaretine sağ tuşla tıkladığımızda açılan menüden **Force Clock** seçeneği seçilmelidir. Seçime ait ekran görüntüsü Şekil 2-22’de verilmiştir.

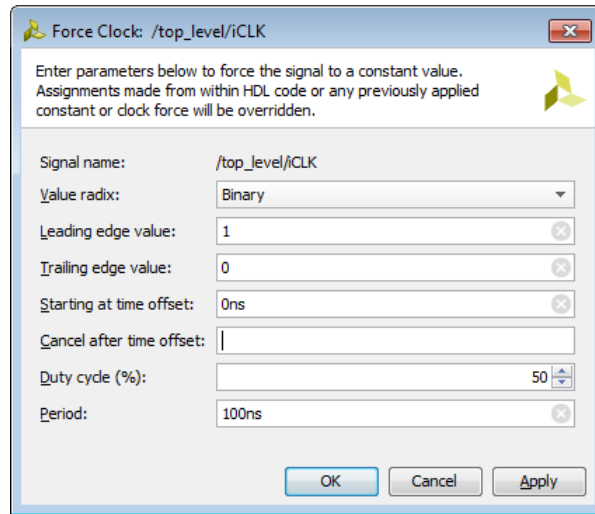


Şekil 2-22 Saat işareti belirleme ekranı.

Karşımıza gelen pencereden saat işaretine ait değerlerin belirlenmesi işlemi yapılacaktır. Bu ekranda saat işaretimizin yükselen kenardaki değerini, düşen kenardaki değerini, varsa saat işaretinin başlama gecikmesini, eğer belli bir süreden sonra saat işaretinin devre dışı kalması isteniyorsa bunun için gereken süre değerini, saat işaretinin yüzde cinsinden dalga doluluk oranını (ideal durum %50'dir) ve son olarak da saat işaretinin periyot değerini belirliyoruz.

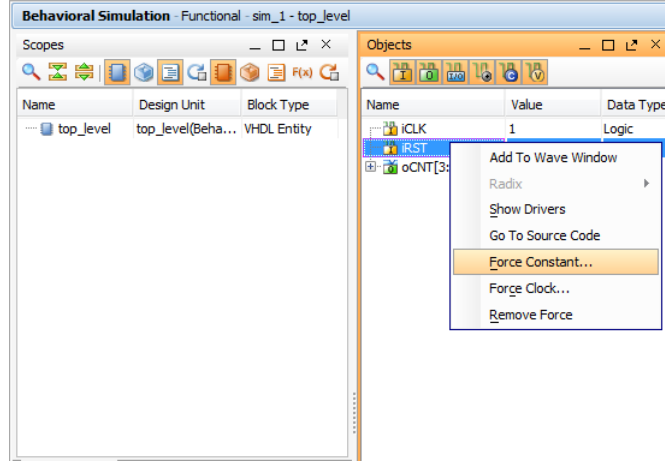
Şekil 2-23'de örnek bir ayarlama görüntüsü verilmiştir. Örnek ekranda periyot değeri 100ns olarak verilmiş olup, frekans olarak 10MHz'e tekabül etmektedir. Bu hesaba ilişkin bağıntı aşağıdaki formülde belirtilmiştir.

$$f = \frac{1}{t} = \frac{1}{100ns} = \frac{1}{100 \times 10^{-9}} = 10 \times 10^6 = 10 \text{ MHz}$$



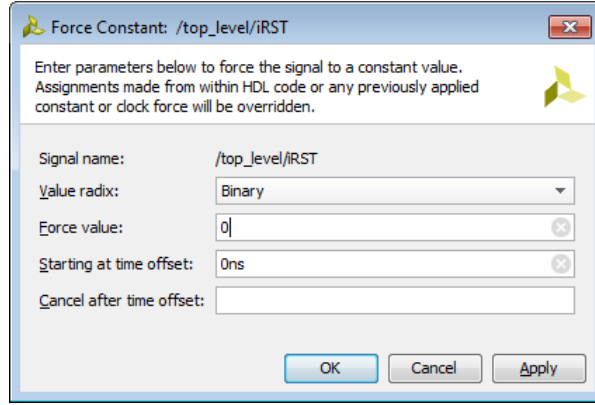
Şekil 2-23 Saat darbesi özelliklerinin belirlenmesi

Bu aşamadan sonra sıra diğer giriş işaretimiz olan **iRST** sinyalinin değerini ayarlamaya geldi. Verilen örnek kod parçası incelendiğinde **iRST** girişinin değeri '1' olduğunda devreyi sıfırlamakta (reset durumu) ve çıkışa "0000" değerini atamaktadır. Bu işaretin değerini belirlemek için üzerine sağ tuşla tıkladığımızda açılan menüden **Force Constant** seçeneğini seçmemiz gerekmektedir. Bu işleme ait ekran görüntüsü Şekil 2-24'de verilmiştir.



Şekil 2-24 Sinyale değer atanması

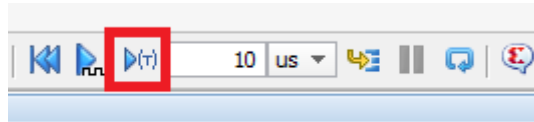
Bu seçenek seçildiğinde açılan pencereden belirtilen sinyale sabit bir değer ataması yapılabilir. Bizim örneğimizde **iRST** sinyali 1 bitlik bir lojik sinyal olduğu için yapılabilecek atama değerleri 1 ve 0 olmaktadır. Eğer belirtilen sinyal bundan farklı bir türde, örneğin **std_logic_vector** tipinde olsaydı alabileceği değer aralığı daha fazla olurdu. Belirtilen değer atama işlemine ait ekran görüntüsü Şekil 2-25'te gösterilmektedir.



Şekil 2-25 Sinyale sabit değer atanması.

Şekil 2-25'te verilen ekrandan belirtilen sinyale ait üç değeri ayarlayabilmekteyiz. Bu değerler sırası ile ; sinyalin alabileceği değer (örneğimizde 0), başlangıç gecikmesi varsa bunun değeri ve son olarak atanan değer belirlenir. Bu değerleri belirttikten sonraki adım ise benzetimin çalıştırılması olacaktır.

Benzetim işlemini başlatmak için program penceresinin üst kısmında bulunan işaretlerden Şekil 2-26'da verilmiş olan görüntüde kırmızı çerçeve içine alınmış olan işarete tıklanması yeterlidir. Bu sayede tasarladığımız yapı tıkladığımız işaretin yanındaki kutuda belirtilen süre kadar çalıştırılacaktır. Bizim örneğimizde bu süre 10 us (mikro saniye) olarak verilmiştir.



Şekil 2-26 Benzetimin 10us süreyle çalıştırılması

Şekil 2-26'da verilen görüntüde kırmızı çerçevenin solunda kalan işaretse benzetimin herhangi bir süre kısıtlaması olmadan çalışmasını sağlamaktadır. Eğer amacınız belli aralıklarla olan değişimleri gözlemlemek değilse, doğrudan tasarımın ürettiği çıkışı görmek istiyorsanız bu seçenek kullanışlı olacaktır.

Benzetim çalıştırıldıktan sonra ekranın sol kısmında olan son değişimler görüntülenmektedir. Çalıştırılan süre boyunca sinyallerde olan tüm değişimleri görüntülemek için siyah ekranın hemen sol yanında bulunan sembollerden büyüteç sembolleri arasında son sırada bulunan sembol seçilmelidir. Şekil 2-27'te verilen görüntüde en alttaki büyüteç sembolü bahsettiğimiz semboldür.



Şekil 2-27 Benzetim ekranının tamamının görüntülenmesi

Eğer tüm adımlar sorunsuz bir şekilde tamamlanırsa benzetim sonucunda elde edilen ekran Şekil 2-28’de verilen şekilde görüntülenecektir.



Şekil 2-28 Benzetim çıktısı ekran görüntüsü