



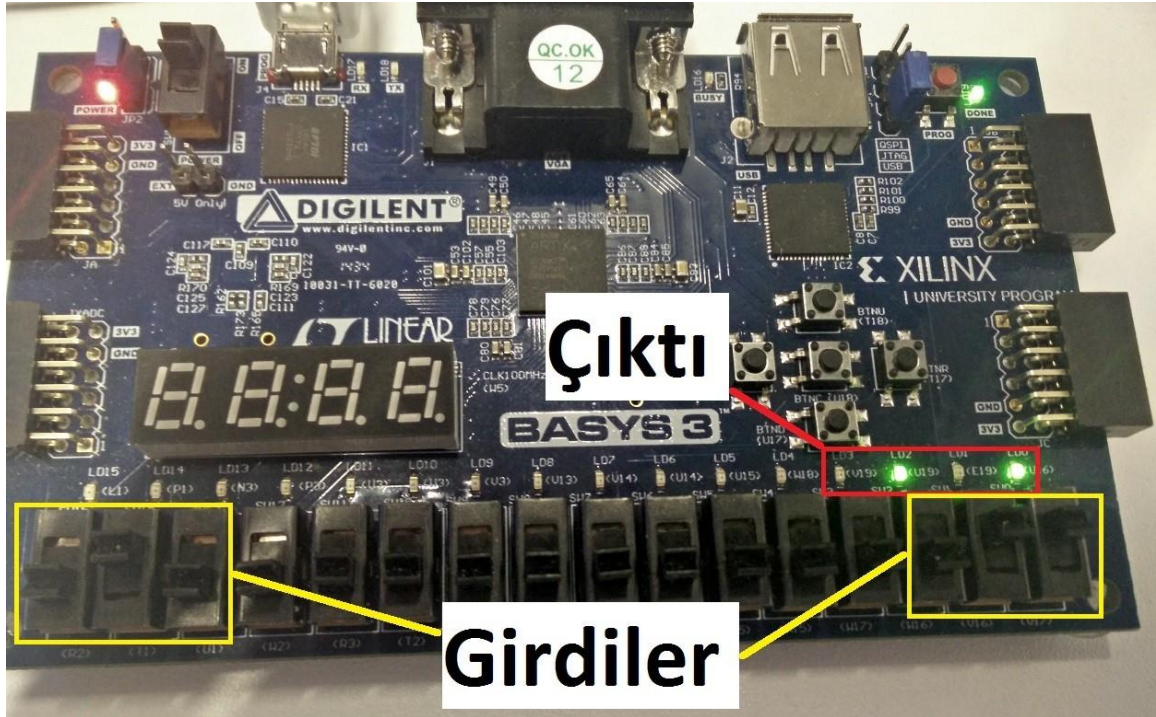
## YARIM VE TAM TOPLAYICILAR

### 1. GİRİŞ

Yarım Toplayıcı (Half-Adder) ve Tam Toplayıcı (Full-Adder) sayısal değerlerin toplanması için yoğun olarak kullanılan dijital devre elemanlarıdır. Bu ders kapsamında, Yarım ve Tam Toplayıcı devreleri Verilog dili kullanılarak kapı seviyesinde (Gate-Level) gerçekleştirilecektir.

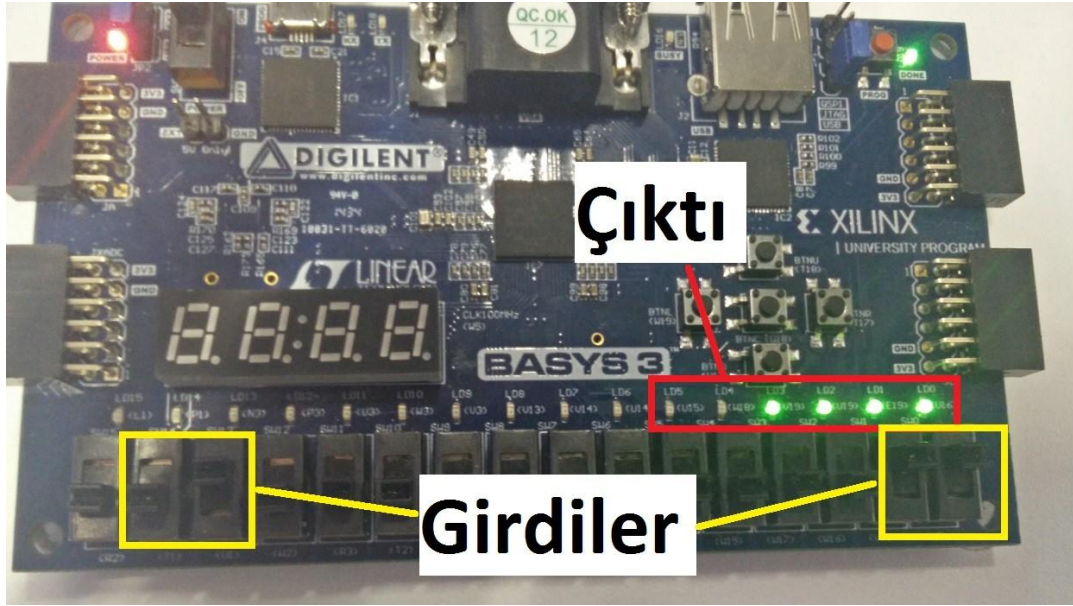
Daha sonra, bu devre elemanları kullanılarak Basys3 FPGA kartı üzerinde iki farklı uygulama geliştirilecektir. Bu uygulamalardan ilkinde, giriş (input) olarak verilen 3 bitlik iki sayının toplamını ikilik tabanda bir sayı olarak, FPGA kartı üzerindeki LED'leri yakarak göstermeniz istenmektedir.

Devrenizin girdileri, FPGA kartı üzerindeki switchler (Anahtar) kullanılarak ayarlanacaktır. Uygulamanın çalışması sırasında alınmış bir ekran görüntüsü aşağıda verilmiştir.



Girdi değerleri gösteren FPGA kartı üzerindeki ilk (sol) ve son (sağ) 3'er Switch, şekildeki örnekte sırasıyla 3'b010 ve 3'b011 durumundadır. Onluk tabanda 2 ve 3'e karşılık gelen bu değerlerin toplamının 5 olması beklenir. Toplama işleminin sonucunu göstermek üzere kullanılan en sağdaki 4 LED ışık, şekilde görüldüğü gibi 4'b0101, yani onluk tabanda 5 değerini göstermektedir.

İkinci kısımda geliştireceğiniz uygulamada ise 2 bitlik iki sayının toplamı kadar LED'in yanmasını sağlayacaksınız. LED'leri yakmaya önce en sağdaki LED'den başlanmalıdır. Bu uygulamanın çalışması sırasında alınmış örnek bir görüntü aşağıda verilmiştir.

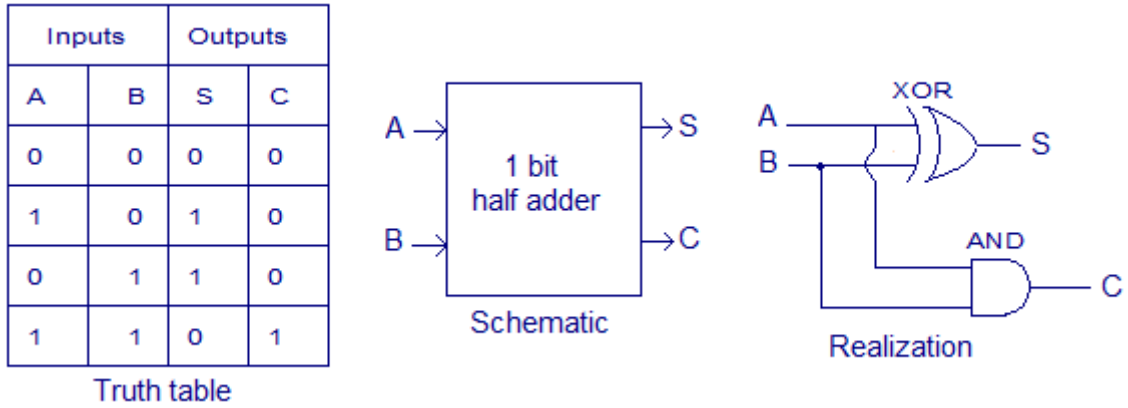


Yukarıdaki şekilde görüldüğü gibi, 2'b01 (1) ve 2'b11 (3) durumunda olan girdi değerlerin toplamı 4 olduğundan, FPGA kartı üzerindeki LED'lerden en sağdaki 4 tanesi yanmaktadır.

Bu dökümanın devamında, bu uygulamaları adım adım nasıl gerçekleştireceğiniz anlatılacaktır.

## 2. YARIM TOPLAYICI (HALF ADDER)

Yarım toplayıcı, iki adet 1-bitlik sayıyı toplar. **Toplam (Sum)** ve **Elde (Carry)** olarak iki adet birer bitlik çıkışı vardır. Yarım toplayıcının sayısal devresi aşağıdaki şekilde gösterilmiştir.



Şekil 1: Yarım Toplayıcı Devresi.

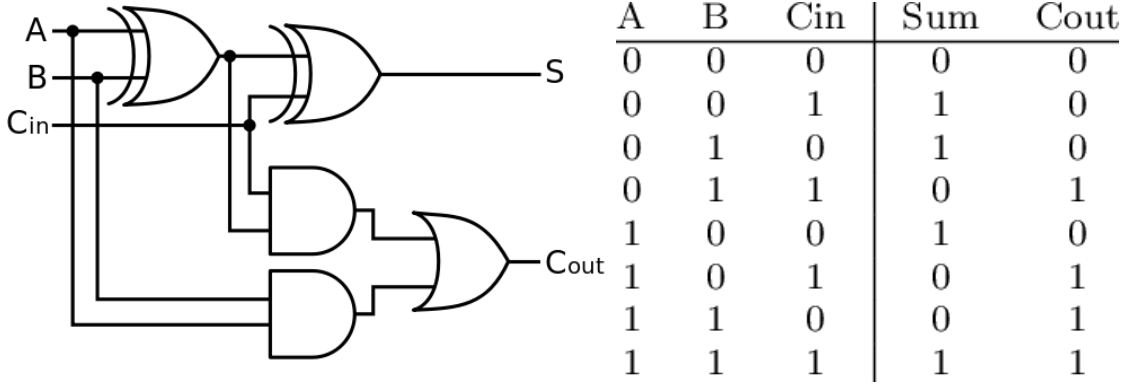
S ve C çıkışları sırasıyla toplamı ve eldeyi göstermektedir. A ve B girişleri ise toplamı hesaplanacak olan sayılardır.

- Yarım Toplayıcı devresini Verilog ile kapı seviyesinde Vivado yazılımını kullanarak gerçekleştirin. Modülünüzün girişlerine "a" ve "b", çıkışlarına ise "sum" ve "carry\_out" isimlerini verin.
- Yarım Toplayıcı devresini doğrulamak için bir testbench kodu yazın ve simülasyon sonucunda üretilen waveformu inceleyerek devrenin istenildiği gibi çalıştığından emin olun. Yazacağınız testbench, yarım toplayıcı devresinin alabileceği tüm farklı giriş değerlerini test edebilmelidir.



### 3. TAM TOPLAYICI (FULL ADDER)

Tam Toplayıcı, iki adet tek bitlik sayıyı ve bir bitlik elde girişini toplar, çıkış olarak yarım toplayıcıda olduğu gibi Toplam (Sum) ve Elde (Carry) sayısal sinyallerini üretir. Bu elemanın devresini, önceki bölümde gerçekleştirdiğiniz gibi mantık kapılarını kullanarak tasarlamalısınız. İki adet Yarım Toplayıcı ve bir adet VEYA (OR) kapısı kullanarak tasarlanmış Tam Toplayıcı devresi aşağıdaki şekilde gösterilmiştir



- Tam Toplayıcı devresini yukarıdaki şekilde gösterildiği gibi Verilog ile kapı seviyesinde tasarım yaparak gerçekleştirin. Tam Toplayıcı modülünde girişler “a”, “b” ve “carry\_in”, çıkışlar ise “sum” ve “carry\_out” şeklinde olmalıdır.
- Tam Toplayıcı devresini doğrulamak için testbench kodu yazın ve simülasyon sonucunda üretilen waveformu inceleyerek devrenin istenildiği gibi çalıştığından emin olun. Yazacağınız testbench, tam toplayıcı devresinin alabileceği tüm farklı giriş değerlerini test edebilmelidir.

### 4. 3 BİTLİK TOPLAYICI BİRİMİ

Bu bölümde üç bit genişliğindeki iki sayıyı toplayan ve dört bit genişliğinde bir sonuç veren bir modül gerçekleştirmelisiniz.

- 3 bitlik iki sayıyı toplayacak olan modüle “Adder3Bit” ismini verin. 3 bitlik toplanacak sayılara “sayi1” ve “sayi2”, 4 bitlik çıkışa ise “toplam” ismini verin. Modülü yine Verilog ile kapı seviyesinde gerçekleştirmelisiniz.
- “Adder3Bit” modülünün doğrulamasını yapan bir testbench geliştirin. Devrenin simülasyonunu yaparak seçtiğiniz iki sayının toplamın doğru sonucu verdiğini gözlemleyin.

### 5. FPGA KARTI İLE UYGULAMA

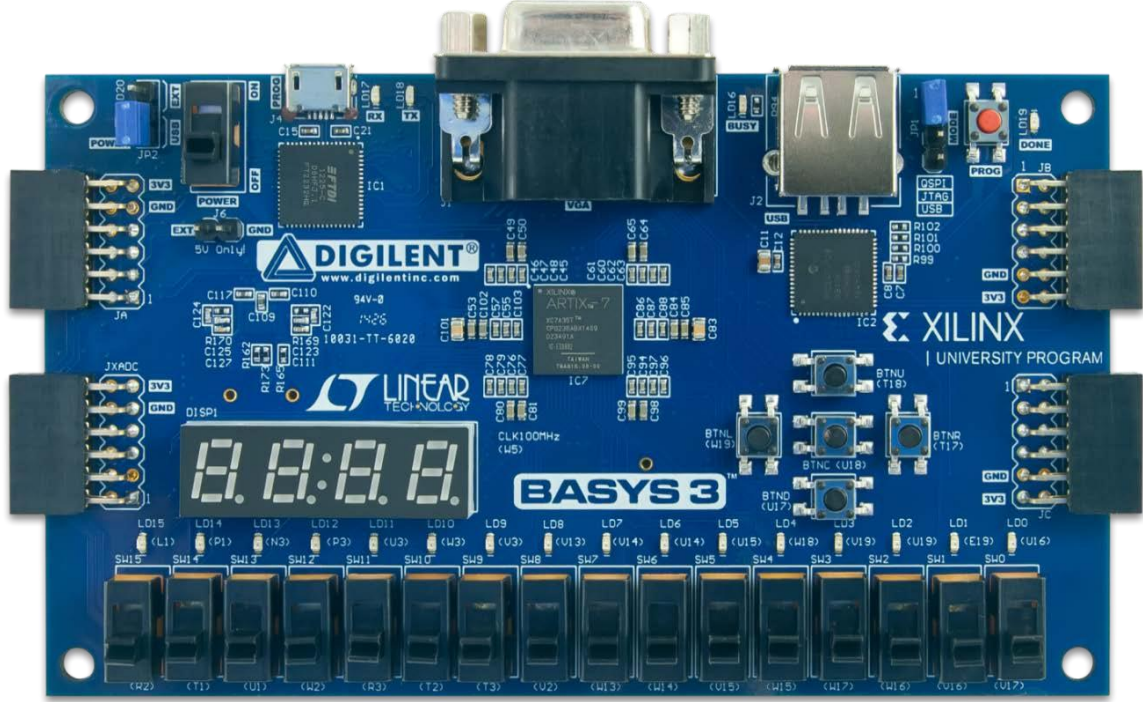
Bu bölümde, bir önceki bölümde yazmış olduğunuz modülü kullanarak bir uygulamayı nasıl geliştirebileceğiniz ve geliştirdiğiniz uygulamayı Basys3 kartına yüklemek için hangi adımları takip etmeniz gerektiği açıklanmıştır.

Derslerde kullanacağımız Basys3 kartını aşağıdaki şekilde görebilirsiniz. Kartın üzerinde USB, VGA, Seven Segment Display, 16 adet LED ışık, 16 adet Switch, 5 adet Push-Button, 100 MHz'lık saat gibi bileşenler bulunmaktadır. Uygulamalar sırasında kullanmanız gerekecek, kart ve her bir bileşeni ile ilgili ayrıntılı kullanım detayları içeren referans dökümanından (<https://reference.digilentinc.com/basys3/refmanual>) faydalanabilirsiniz.





**BİL264L - Mantıksal Devre Tasarımı Laboratuvarı**  
**ELE263L - Sayısal Sistem Tasarımı Laboratuvarı**



İlk olarak, asistanlardan bir Basys3 FPGA kartı ve USB kablosu teslim alın. Teslim aldığınız kartın seri numarası kutunun üzerinde ve FPGA kartının kendisinde bulunmaktadır. Kartı ve kabloyu teslim ederken, asistanlar size bu seri numaraya sahip kartı teslim ettiklerine dair imza alacaklar. Ders bitiminde kartı teslim ederken, FPGA kartı ile kutunun aynı seri numaraya sahip olduğundan emin olun (yanlışlıkla arkadaşınızın kartını kendi kutunuza koymayın). Kartların seri numaralarının nerede yazdığını asistanlara sorabilirsiniz.

**Uyarı:** Basys3 kartlarını dikkatli bir şekilde zarar vermeden kullanın.

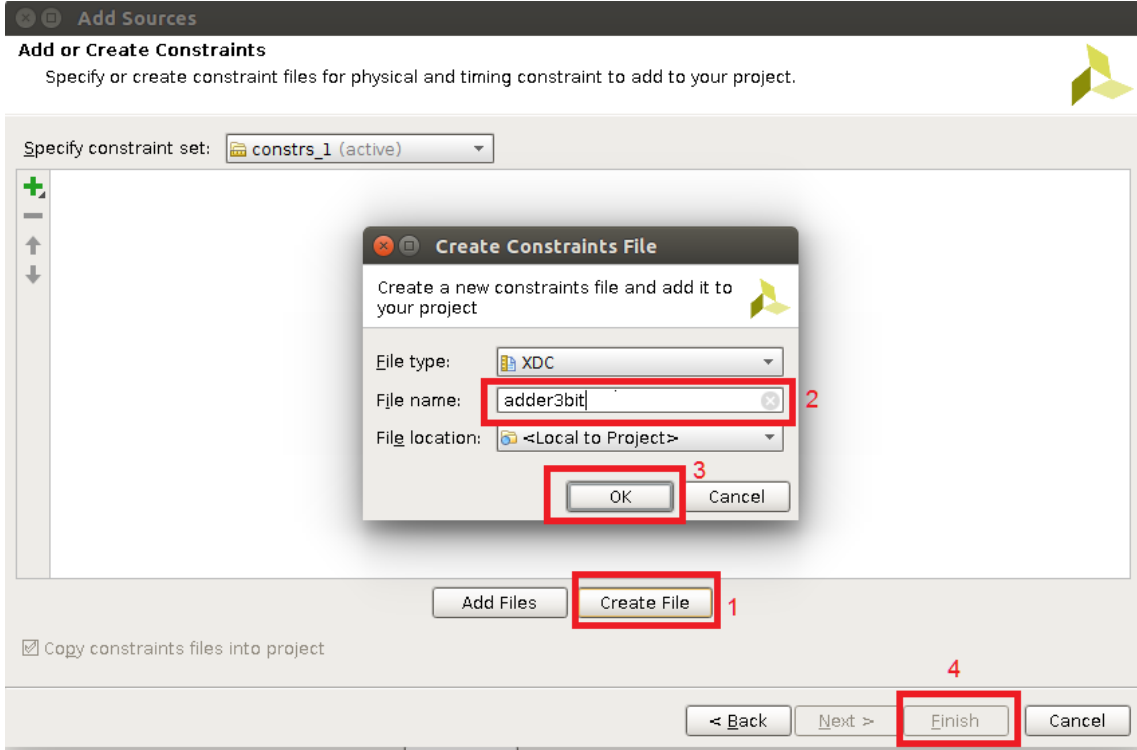
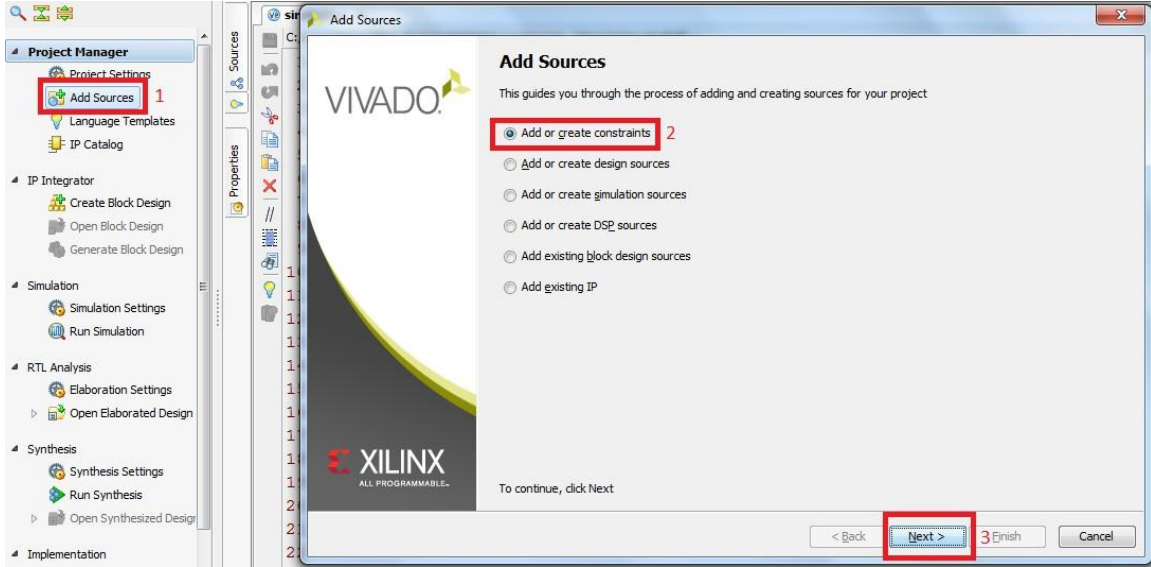
Bu uygulamada, toplama işleminin sonucunu doğru verdiğinden emin olduğunuz “Adder3Bit” modülünü kullanan bir uygulamayı, anlatıldığı şekilde adım adım yazmalısınız.

- Toplanacak 3 bitlik sayılardan “sayı1”i, en solda bulunan 3 switch ile, “sayı2”yi ise en sağda bulunan 3 switch ile ifade edin.
- Toplamı en sağdaki 4 LED’i kullanarak gösterin.



**BİL264L - Mantıksal Devre Tasarımı Laboratuvarı**  
**ELE263L - Sayısal Sistem Tasarımı Laboratuvarı**

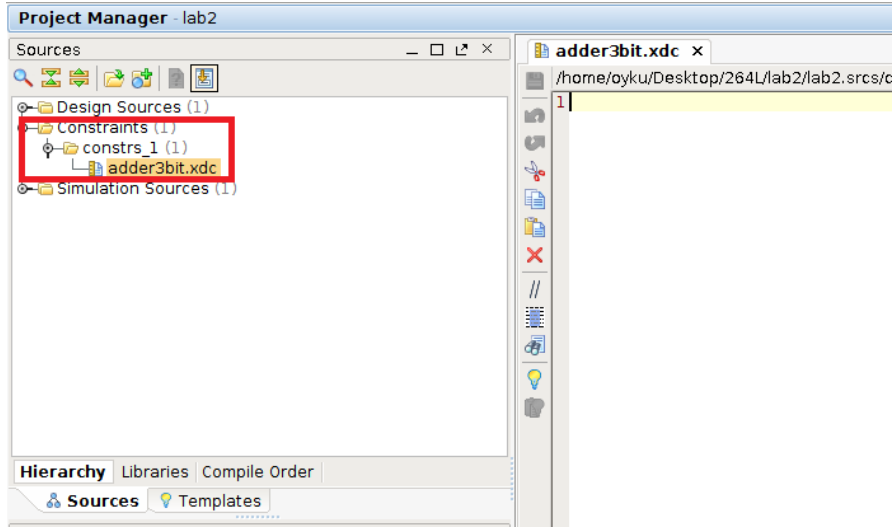
Yazacağınız uygulamanın giriş/çıkış sinyallerini FPGA üzerindeki hangi pinlere bağlamak istediğimizi göstermek için bir “constraint” (kısıt) dosyası oluşturmalsınız. Bu dosyayı nasıl oluşturacağınız aşağıdaki iki şekilde anlatılmıştır.



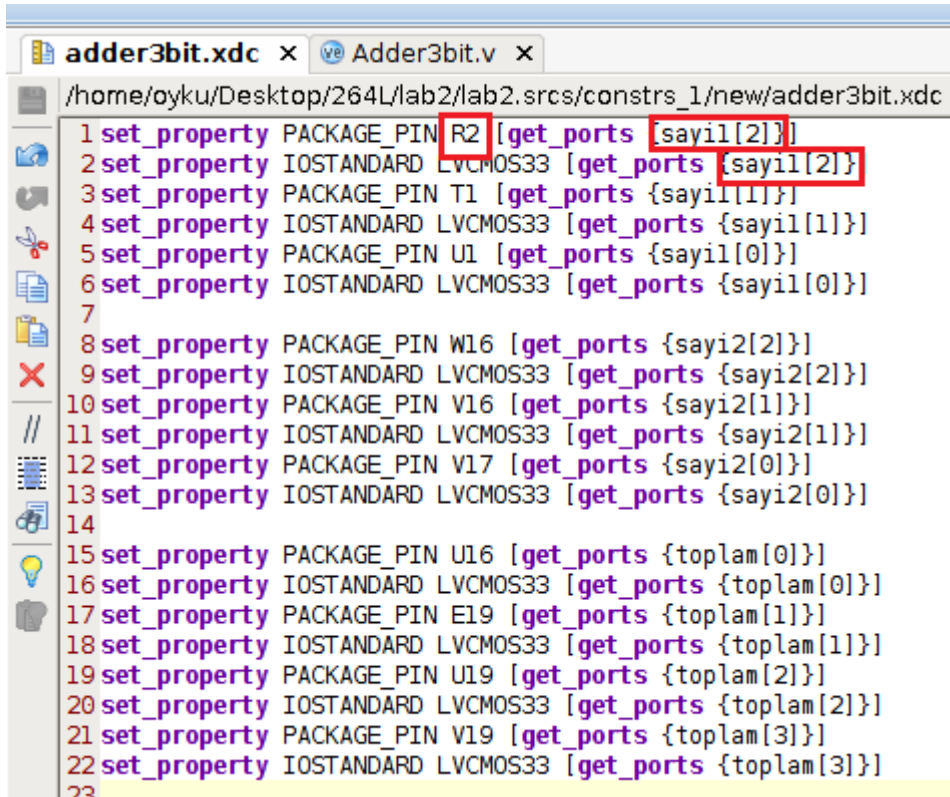
Böylece “adder3bit.xdc” isimli bir constraint dosyası oluşturmuş oldunuz. Daha sonra, constraint dosyasını açarak, devrenizdeki giriş ve çıkışların, FPGA’deki hangi pinlere bağlanacağını belirlemelisiniz.



BİL264L - Mantıksal Devre Tasarımı Laboratuvarı  
ELE263L - Sayısal Sistem Tasarımı Laboratuvarı



Oluşturduğunuz dosyaya aşağıdaki şekilde gösterilen satırları yazın. Burada her iki satır sizin Verilog ile tanımladığınız bir giriş/çıkış bitini, FPGA üzerindeki bir pine bağlamanızı sağlar. Farklı pinler için her iki satırda değiştirmeniz gereken yerler, ilk iki satırda kırmızı ile işaretlidir. “**get\_ports**” ifadesinin sağına, yazdığımız koddaki giriş/çıkış pinini yazın. “**PACKAGE\_PIN**” ifadesinin sağına [basys3 basic io](https://basys3.basys3.io) linkinde gösterilen uygun pinlerden birinin kodunu yazın.



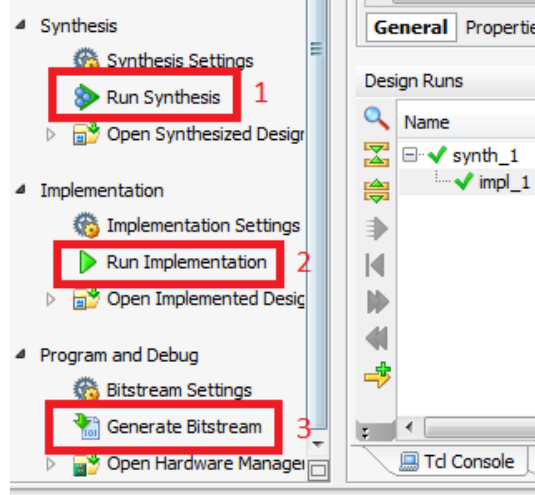


## BİL264L - Mantıksal Devre Tasarımı Laboratuvarı ELE263L - Sayısal Sistem Tasarımı Laboratuvarı

Yukarıdaki şeklin ilk 6 satırında, çarpılacak ilk sayının her bir bitinin hangi pin'e bağlanacağı ve IO standardı belirtilmiştir. R2 switchi ilk sayının en anlamlı bitini (most significant bit) göstermek için, U1 switchi ise ilk sayının en anlamsız bitini (least significant bit) göstermek için kullanılmaktadır. 8. satırdan başlayan 6 satırda ise ikinci sayı için pin atamaları yapılmıştır.

Daha sonra, sırasıyla Sentez, İmplementasyon ve Bitfile oluşturulması adımlarını çalıştırın. Sentez işlemi bittiğinde karşınıza çıkan küçük pencerede **"Run Implementation"** seçeneğini seçip, çıkan pencerede OK tuşuna tıklarsanız implemantasyon, aşağıdaki şekilde 2 numara ile gösterilen yere tıklamadan başlamış olur.

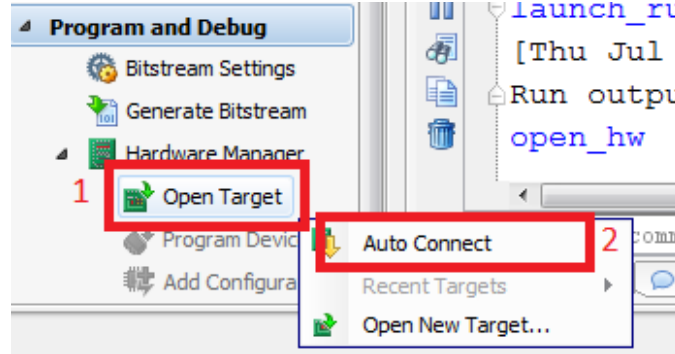
Benzer şekilde İmplementasyondan sonra Bitfile oluşturulması aşamasını da başlatabilirsiniz.



Bu uygulamada **"clock"** kullanmadığımız için Vivado bu konuda bir uyarı verebilir, bu uyarıyı dikkate almayın.

Eğer yukarıdaki 3 adımı hatasız şekilde tamamladıysanız, oluşturulan Bitfile'ı Basys3 kartına yükleyebilirsiniz. Bunun için öncelikle kartı USB kablosunu kullanarak bilgisayarınıza bağlayın. Kartın USB girişinin yanında bulunan switchi **"ON"** durumuna getirerek kartı çalıştırın. Kartı çalıştırdığınızda, kartta yüklü olarak gelen üretici firmanın örnek uygulaması çalışacaktır. Switchleri açıp kapatarak ve butonlara basarak uygulamanın ne yaptığını gözlemleyebilirsiniz. Bu örnek uygulama kartın flash belleğinde yüklü olduğu için kartı her kapatıp açtığınızda bu uygulama geri yüklenecektir (kendi uygulamanızı yüklemiş olsanız bile).

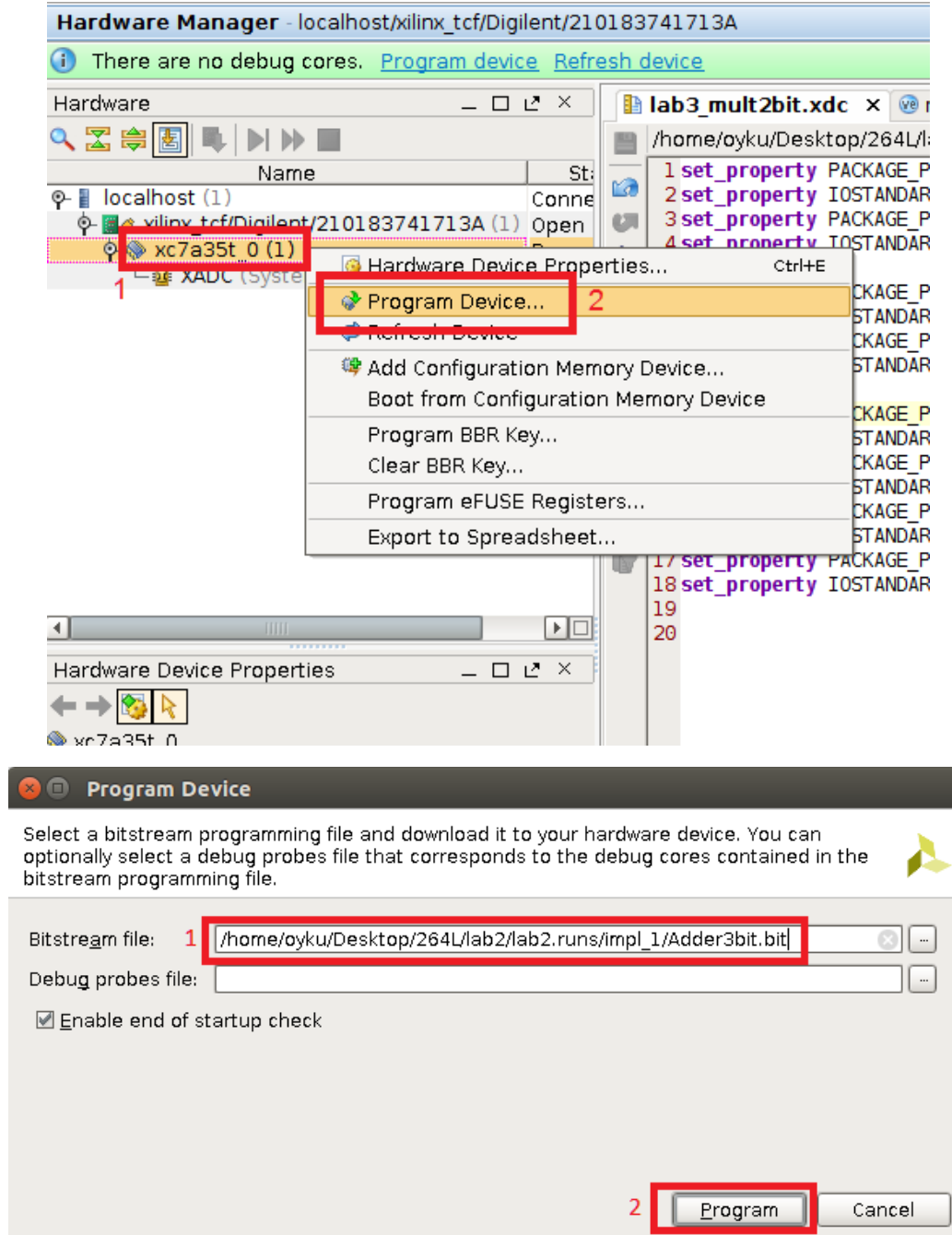
Basys3 kartını bilgisayarınıza bağladığınızda USB sürücüsü yüklenecektir, bu işlemin tamamlanmasını bekleyin. Daha sonra aşağıda gösterildiği gibi FPGA kartına bağlanın.





**BİL264L - Mantıksal Devre Tasarımı Laboratuvarı**  
**ELE263L - Sayısal Sistem Tasarımı Laboratuvarı**

Uygulamanızın Bitfile'ını kullanarak kartı programlayın.



Programlama işlemi tamamlandığında uygulamanızın Basys3 kartı üzerinde çalıştığını göreceksiniz.





## 6. 2 BİTLİK TOPLAYICI BİRİMİ

Bu uygulamada amaç, girdi olarak verilen iki sayının toplamı kadar LED'in FPGA kartı üzerinde yanıyor olmasını sağlamaktır. Bunun için, 2 bitlik iki sayıyı **4. Bölüm**'de gerçekleştirdiğiniz "Adder3Bit" modülünü kullanarak toplamalısınız. Daha sonra, bu modülün **ürettiği toplam kadar LED'i yakacak şekilde** devrenin geri kalanını tasarlamalısınız. 2 bitlik iki sayının toplamı en fazla 6 olabileceğinden, modülünüz en fazla 6 adet LED yakabilecektir.

- Yukarıda anlatıldığı şekilde, 2 bitlik iki sayıyı toplayıp, bu toplam kadar LED'i FPGA kartı üzerinde yakan modülü kapı seviyesinde gerçekleştirin. Modüle "AdderLEDs" ismini verip, 2 bitlik girişlere "sayi1" ve "sayi2", 6 bitlik çıkışa ise "leds" ismini verin.
- Gerçekleştirdiğiniz "AdderLEDs" modülünü doğrulayan testbench kodunu yazıp, devrenin simülasyonunu yapın.
- "AdderLEDs" modülünü gerçekleştirdikten sonra modülü, giriş ve çıkışlarını uygun Switch ve LED'lere bağlayarak bir önceki bölümde anlatıldığı şekilde FPGA kartına yükleyip deneyin.