

**DOĞU AKDENİZ ÜNİVERSİTESİ**  
**BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ (TÜRKÇE)**  
**BLGM223 SAYISAL MANTIK TASARIMI**

**DENEY II: QUARTUS II TASARIM ORTAMINA**  
**VE**  
**VeriLog HDL PROGRAMLAMA İLE TASARIMA GİRİŞ**

**Amaçlar:**

Bu deneyde INTEL-ALTERA tarafından geliştirilen en son Elektronik Tasarım Otomasyonu yazılımlarından biri olan QUARTUS II'yi kullanmaya başlayacağız. QUARTUS II'nin tasarım arayüzü ve VeriLogHDL diline yönelik en temel kavramları öğreneceğiz.

**Deney Öncesi Çalışma:**

*Aşağıda verilen deney çalışmasını dikkatlice okuyunuz ve eğer mümkünse deney aşamalarını laboratuara gitmeden önce denemeye çalışınız..*

**Deney Çalışması:**

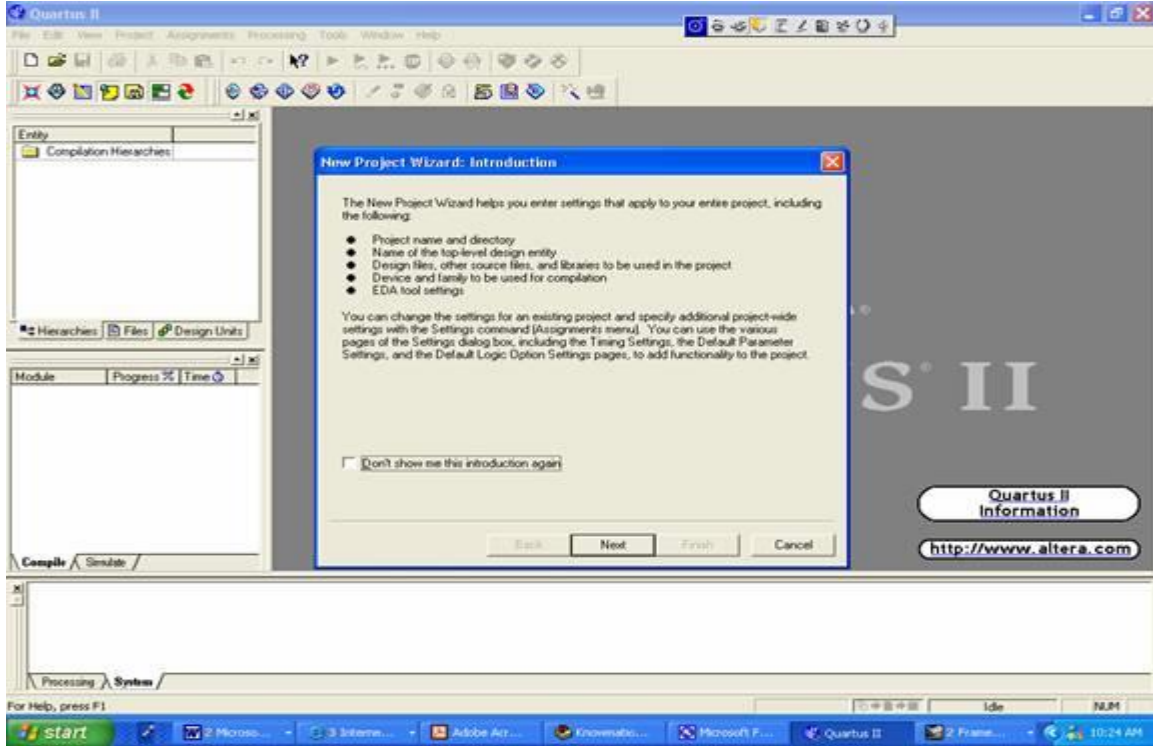
Aşağıda verilen açıklamalar sizin için QUARTUS II tasarım ortamına yönelik detaylı bir rehber niteliğindedir. Bu açıklamaları takip ederek QUARTUS II tasarım ortamında bir proje başlatabilir ve bu proje ile ilgili temel tasarım aşamalarını gerçekleştirebilirsiniz.

**Adım 1. Proje Tanımlanması:** QUARTUS II ortamındaki her tasarım bir proje çerçevesinde olmalıdır. Bir proje, projenin adı, projenin bulunduğu kütük (directory) tasarımı amaçlanan fiziksel sitemin adı, tasarım dosyaları, kütüphaneler, zamanlama bilgisi, devre elemanları, v.s. den oluşur. QUARTUS II üçüncü taraflarca sağlanan tasarım otomasyonu araçlarının kullanımına izin verir.

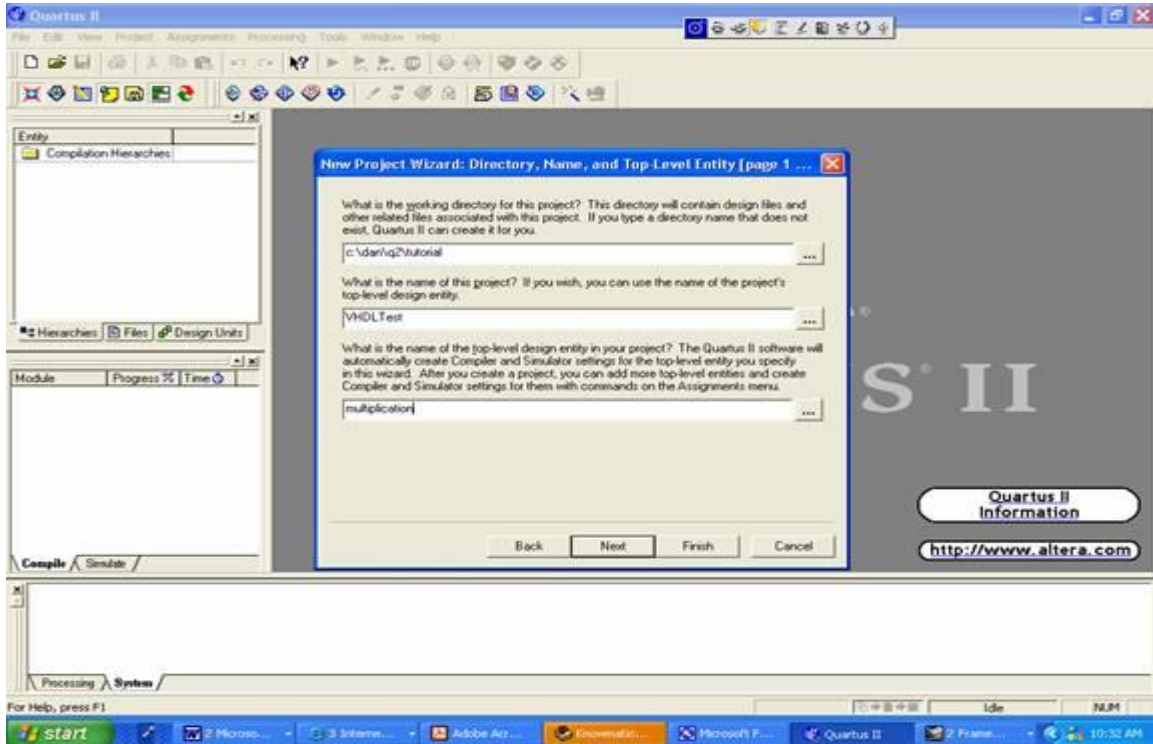
Şimdi, önce sisteminize yüklenmiş bulunan QUARTUS II yazılımını başlatalım ve yazılımın açılış ekranından

*[File]->[New Project Wizard]*

seçeneğini çalıştıralım. Aşağıda görünümü verilen ekran karşınıza çıkacak ve size yeni bir proje tanımı için gereken içerik sunulacaktır.

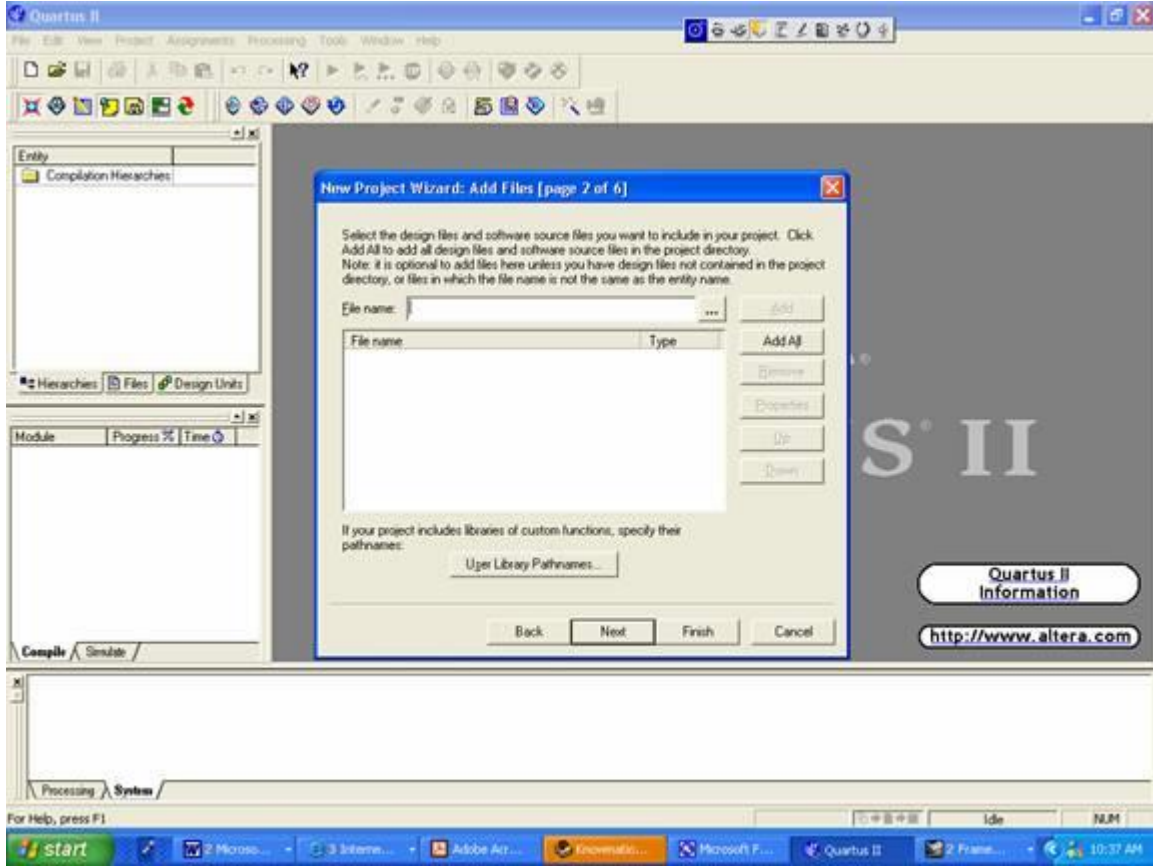


[Next] seçiniz : Çalıştığımız kütüğün adını, projenizin adını, tasarlayacağınız sistemin adını soran bir pencere ile karşılaşacaksınız. İlgili yerleri uygun şekilde doldurunuz.

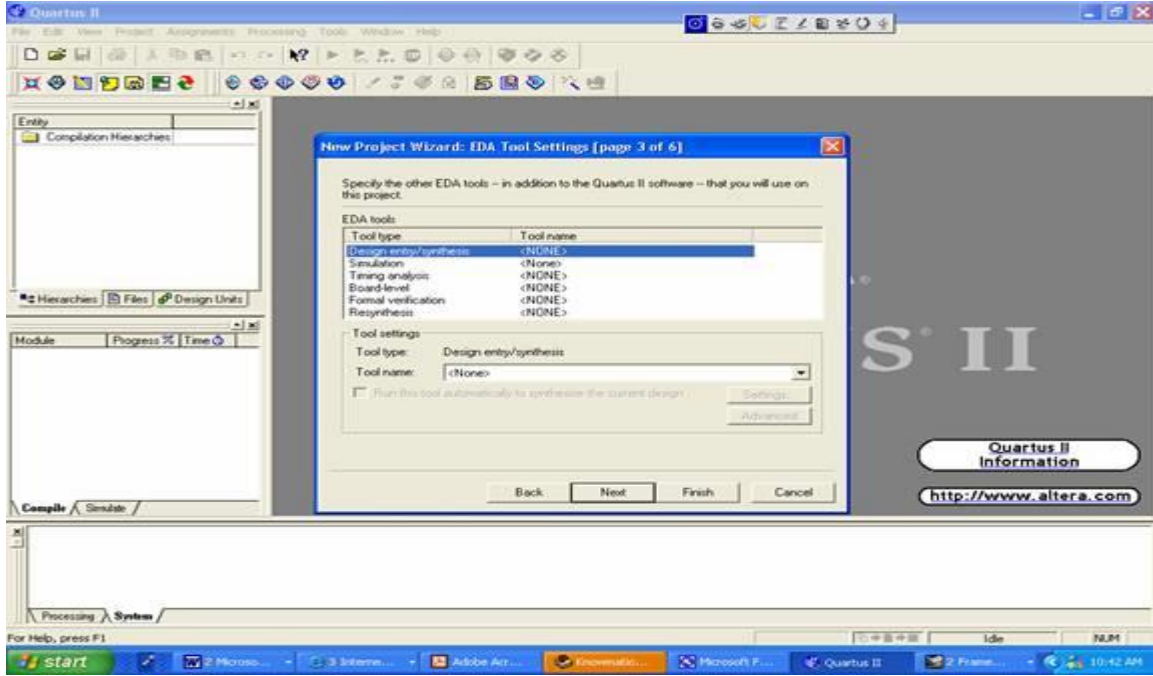


**Örneğin;** çalıştığınız kütük ismi olarak “c:\BenimDosyam\BLGM223\Projeler” yazabilirsiniz. Bu sizin projeniz ile dosyaların saklandığı kütük olacaktır. Sonraki kutucuğun içine “Quartus\_Proje1” yazabilirsiniz. *Quartus\_Proje1* projenizin ismi olacaktır. Üçüncü kutucuğun içine “İlk\_Devrem” yazarak tasarımı yapacağınız sistemin ismini de belirleyebilirsiniz. **Burada dikkat edeceğimiz konu, seçtiğiniz dosya ismi ile VeriLog ortamında kodunu yazacağınız modülün isminin aynı olmasıdır.** Bir projenin içinde birden fazla sistem bulunabilir ve bunlar daha sonra birbirlerine bağlanabilirler. Bunu ilerleyen derslerde göreceksiniz.

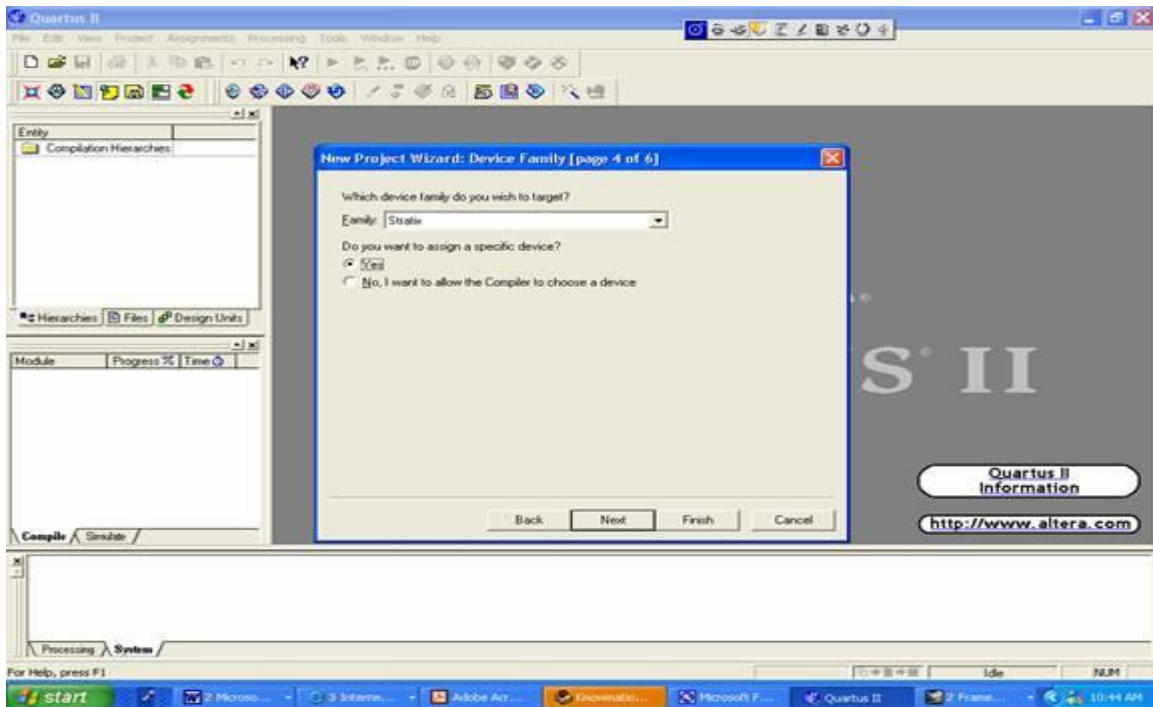
*Yukarıdaki işlemleri tamamlayınca [Next] seçiniz: Eğer proje kütüğü olarak belirlediğiniz isim henüz açılmış değil ise, aşağıdaki pencere açılıp bu kütüğün oluşturulmasını onaylamanız istenecektir. [Yes] seçeneğini çalıştırınca karşınıza aşağıdaki pencere çıkacaktır.*



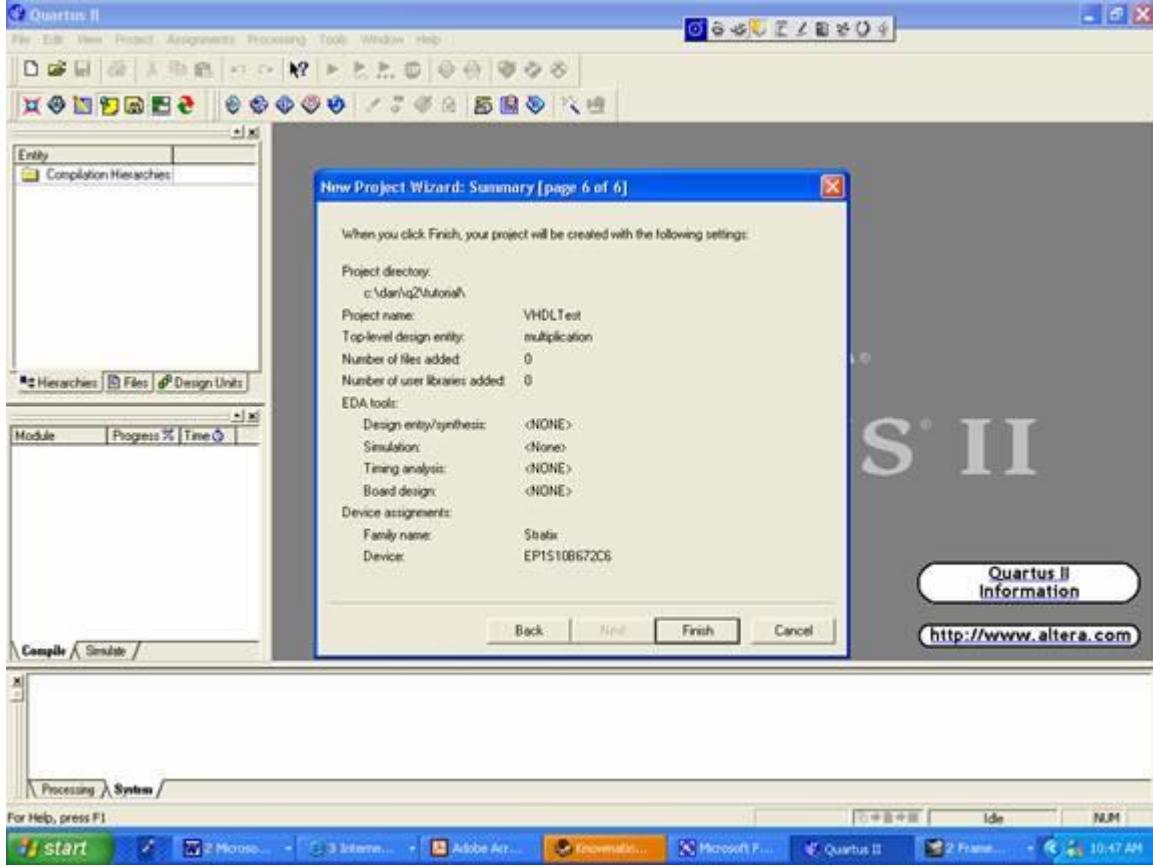
Burada istenilen: projenize dahil etmek istediğiniz hazır kaynak dosyaların ve/veya kütüphanelerin olup olmadığıdır. Eğer var ise, bunlarla ilgili yolları tanımlamanız istenmektedir. Bu aşamada bu seçeneği atlamak *[Next] seçeneğini çalıştırılm.* Bir sonraki pencere bu projede kullanılacak tasarım araçlarını göstermektedir:



Şimdilik sadece QUARTUSII yazılımına ihtiyacımız olduğundan [Next] seçeneğini çalıştıralım. QUARTUSII ortamında yapılan tasarımlar nihai olarak bir FPGA yongası aracılığıyla gerçekleştirildiğinden, bu aşamada donanım olarak hangi FPGA yongasını kullanacağımızı belirlememiz gerekiyor. Aşağıdaki pencerede size varolan tüm seçenekler sunuluyor.



“Cycloen II” ailesi seçeneğini işaretleyelim ve bir sonraki pencerede [Yes] seçimini yapalım. [Next] seçeneğini takiben Cyclone II ailesinden hangi elemanı kullanacağımızı seçmemiz istenecektir.



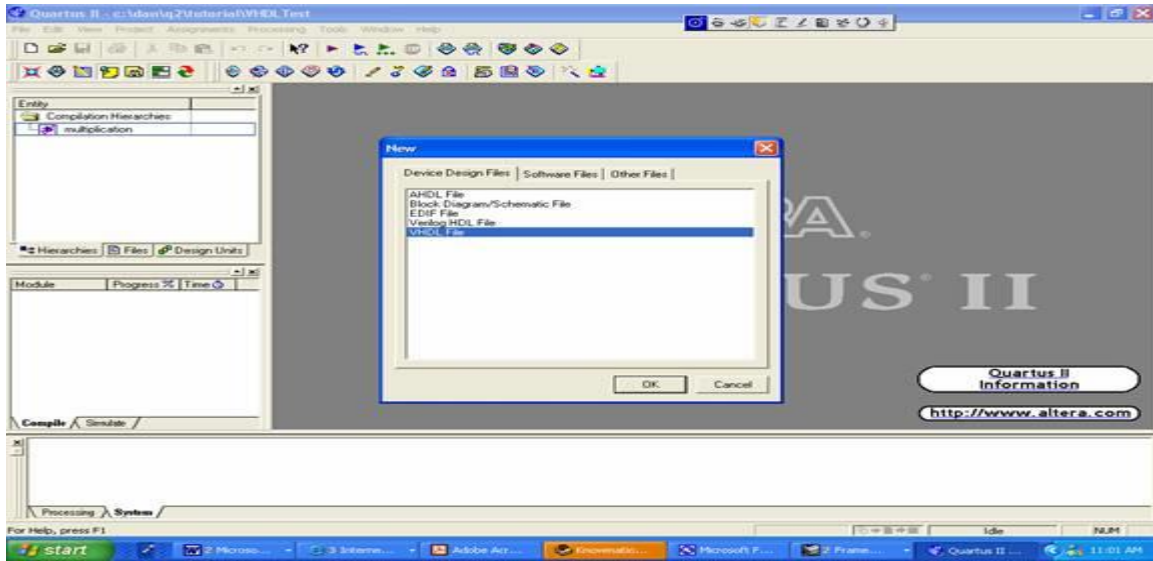
“EP2C5AF256A7” elemanını işaretleyip [Next]seçimini yapalım.

Tebrikler. Şu ana kadar, üzerinde çalışacağınız bir proje yaratıp hedef sayısal sisteminizin donanım ortamını belirlemiş oldunuz. [Finish] seçeneğini çalıştırarak proje yaratma işlemini tamamlamış olunuz.

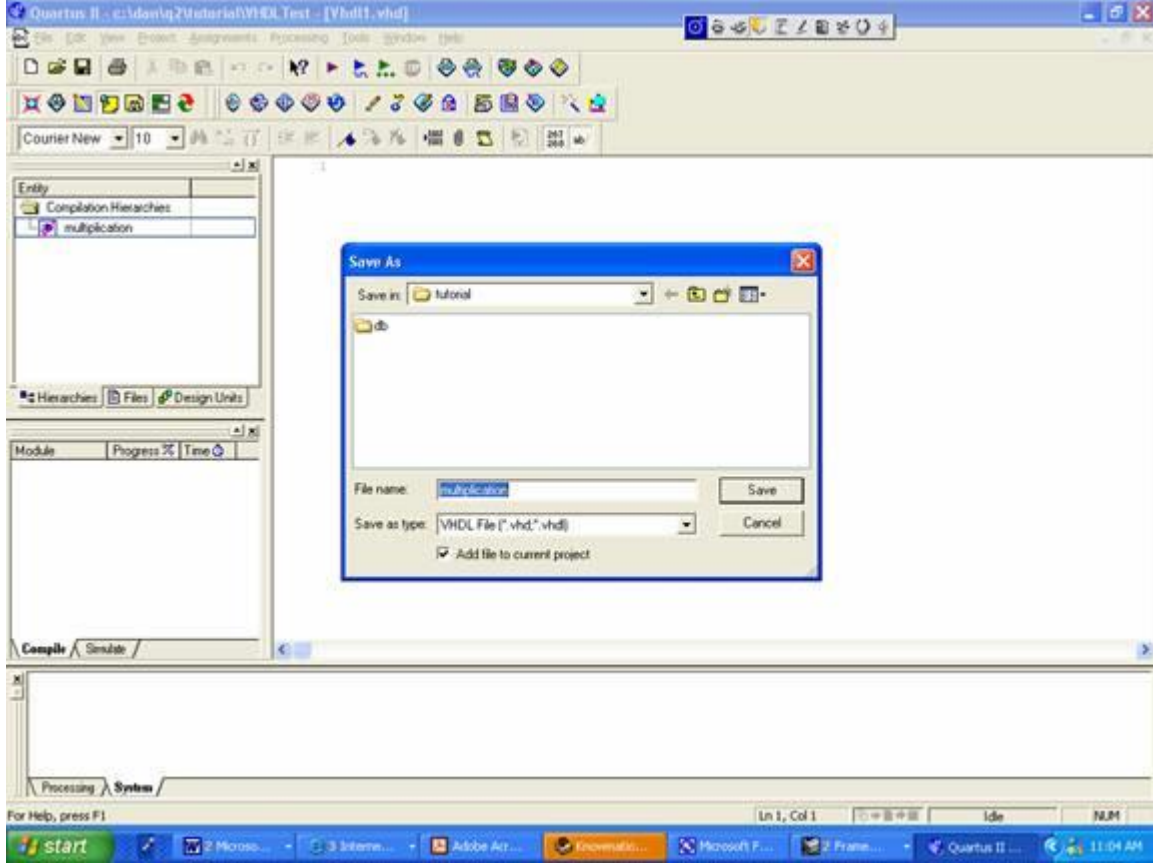
## Adım 2: VeriLog HDL Programlama Diliyle Tasarım Girişi

Devre şeması girişine alternatif olarak VeriLog HDL veya benzeri bir donanım tanımlama dilini kullanarak da yapacağımız tasarımları gerçekleştirebiliriz. Büyük boyutlu tasarımlarda donanım dillerinin kullanımı üretkenliğin artmasına yardımcı olacağı gibi tasarım zamanının kısaltılmasına ve hataların en aza indirilmesine de büyük oranda katkı sağlar. Mantık sadeleştirmelerinin ve ağ bağlantılarının otomatik olarak derleyici tarafından yapılıyor olması tasarımcıyı pek çok detay yükten kurtararak tasarımda daha yaratıcı olmasına neden olur. Şimdi

[File]→[New] menüsüne tıklayıp, “VeriLog HDL File” seçeneğini [OK] ile olurlayalım.



[File]->[Save As] seçeneğini tıklayalım ve dosya ismi olarak otomatik olarak gelen proje ismini onaylayalım ve [Save] seçeneğiyle de açtığımız program giriş dosyasını diske kaydedelim. .



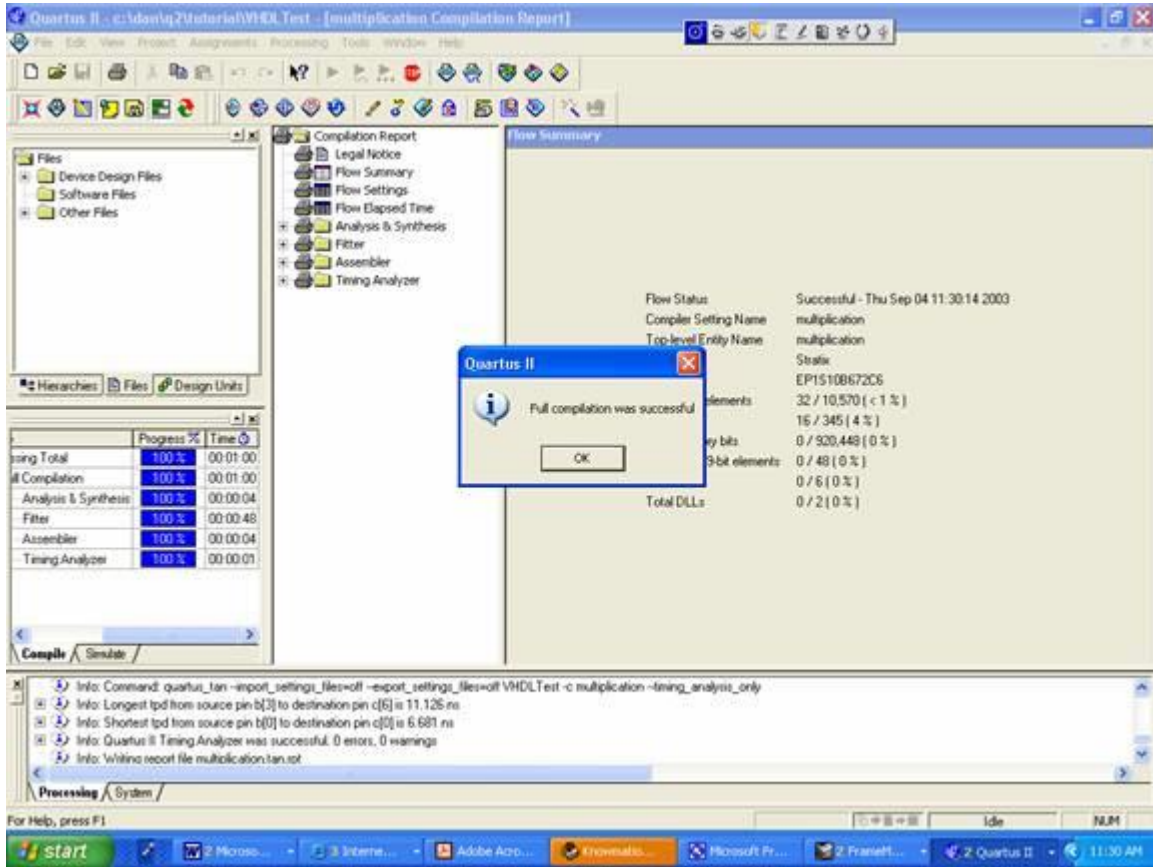
Program giriş pencersine aşağıdaki VHDL programını yazalım.

```
module VeriLogDeney1 (a,b,c,d,y);
    input a,b,c,d;
    output y;
    wire s1,s2;

    assign s1= ~(a | b);
    assign s2= ~(c & d);
    assign y = s1 & s2;

endmodule
```

Bu kaynak dosyasını diske kaydettikten sonra , Click [Processing]→[Start Compilation] seçeneğiyle bu programı derleyelim. Başarılı bir derleme sonrasında aşağıdaki ekranı görüyor olmalısınız.



## Tasarımınızın Benzetimi

**İşlevsel Benzetimin yapılması:** VeriLog HDL programları için işlevsel benzetim süreci, daha önce devre tasarım şeması girişi deneyinde anlatılan adımların aynısıdır. **Yeni bir** dalga biçimi yaratıp File | Save As seçeneğinden dosya ismi olarak *FileName.vwf* yazınız.

Devre tasarım şeması girişi deneyinde yaptığımız gibi giriş ve çıkış terminallerini (iğnelerini) dalga biçimi editor penceresine taşıyınız. Giriş terminallerine ait dalga biçimlerini oluşturup digger adımları da aynen takip ediniz. Bu adımlar hatırlamanız amacıyla aşağıda yeniden listelenmiştir.

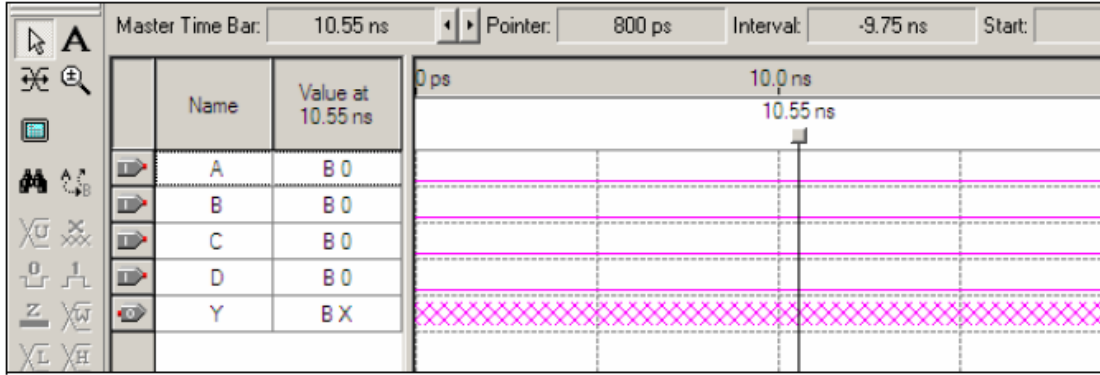
### **Adım 3: Projenizi Bezetimleme (simulation):**

**Bir** sayısal devre iki şekilde benzetimlenebilir. En basit olanında tüm elemanlarının ve bağlantıların ideal olduğunu kabul ederiz ve bu durumda devre boyunca sinyallerin ulaşımında hiç bir geçikme olmaz. Bu benzetim yaklaşımına “işlevsel bezetim” (*functional simulation*) denilir. Daha karmaşık olan ikinci yaklaşımda ise bütün sinyal geçikmeleri dikkate alınarak benzetim yapılır ve yaklaşıma “zamanlama benzetimi” (*timing simulation*) denilir. Temel olarak




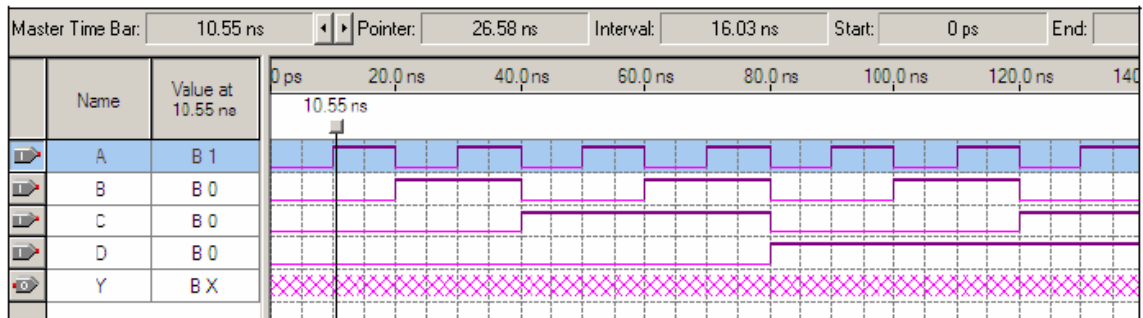
işlevsel benzetim bir devrenin/sistemin işlevsel doğruluğunu control etmek amacıyla kullanılır. Bu benzetim yaklaşımı az zaman alır ve kolayca yapılır. Bu deneyde işlevsel benzetimi uygulayacağız, zamanlama benzetimini de sonraki deneylerde öğreneceğiz. Önceki adımlarda QuartusII ortamında oluşturduğunuz sayısal devrenin işlevsel benzerimini şu şekilde yapabilirsiniz:

1. **File** → **New** → **University Program VWF File** seçiminden sonra **OK** ile onaylayınız.
2. Ana menüden **Edit** → **Insert Node or Bus** seçimini yapınız.
3. **Node Finder** üzerine tıklayınız.
4. **Pins: all** seçiminden sonra **Start** ile işlevsel benzetimi başlatınız.
- 5 **Nodes Found** listesinden A, B, C, D ve Y terminallerini işaretleyiniz.
6. Sağ ok ( → ) tuşu ile A, B, C, D, ve Y terminallerini **Selected Nodes** tarafına taşıyınız.
7. **OK** seçimlerini yaparak benzetimleme sürecinin ikinci aşamasına geçiniz.
8. Terminallerinin üzerine tıklayarak fare yardımıyla sıralarını değiştirebilirsiniz.



Sayısal Dalga Formu Penceresi

9. Şimdi, **Edit** → **End** seçiminin ardından **Time** seçeneğine **160.0 ns** giriniz ve benzetimleme zaman aralığını **0.0ns** ile **160.0ns** aralığına sabitleyiniz.
10. **View** → **Fit in Window** ile benzetim aralığının tamamını ekrana yansıtınız.
11. A terminaline tıklayınız ve yan cubuk üzerinden **Overwrite clock** tuşunu seçerek periyodu için 20.0 ns yazınız. 
12. Aynı işlemleri B: 40.0 ns, C: 80.0 ns, and D: 160.0 ns olarak tekrarlayınız.



## Waveform Editor Window with Inputs

13. **Assignments**→**Settings** seçimini yaparak ayarlar ekranını açınız. **Simulations Settings** seçiminin ardından **Mode** seçeneğine geçiniz ve benzetim modunu **Functional** olarak deęitiriniz.
14. **File**→**Save** seçeneğine giderek dosyanızı saklayınız. Dosya ismi projenizin ismiyle aynı olacaktır. **OK** seçeneğine tıklayınız.
15. Ana menüye giderek **Processing**→ **Start Simulation** seçimini yapınız.
16. Benzetimleme işlemi tamamlanınca **OK** ile onaylayınız.
17. **View**→**Fit in Window** seçimiyle tüm benzetimleme aralığını ekrana yansıtınız.
18. Elde ettiđiniz benzetimleme sonuçlarını deverenizin doğruluk tablosu ile karşılaştırınız.

BAŞARILAR DİLERİM.

Doç. Dr. Adnan ACAN