

**DOĞU AKDENİZ ÜNİVERSİTESİ**  
**BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**  
**BLGM223 SAYISAL MANTIK TASARIMI**

**DENEY IV: QUARTUS II TASARIM ORTAMI: TEMEL VeriLog HDL KULLANIMI**

**AMAÇLAR:**

ALTERA tarafından geliştirilen son teknoloji elektronik tasarım otomasyonu yazılımı olan QUARTUS II ortamında VeriLog HDL dilini kullanmaya başlayacağız. QUARTUS II ile ilgili tasarım arayüzünü ve temel VHDL yapılarına ait bilgilerimizi tazeleyeceğiz. Uygulamalar yaparak becerilerimizi geliştireceğiz.

**Deney Öncesi Çalışma:**

*Aşağıdaki Açıklamalar Bölümünü okuyunuz ve Deney Adımlarını yapmaya çalışınız.*

**Açıklamalar: VeriLog HDL ile Tasarım İçin Temel Adımlar**

VeriLog HDL dilinde her tasarım işlevsel modüllerden oluşur ve bu modüllerin birbirlerine bir amaç doğrultusunda bağlanmasıyla sistemler elde edilir. Her modül, giriş terminalleri, çıkış terminalleri ve bu terminallere bağlanacak sinyallerin tipleri ile tanımlanır. Sistemin nasıl çalıştığına dair mantıksal ifadeler modülün bir blok olarak tanımlanmasının ardından sıralanır. Böylece, bir VeriLog HDL programının jenerik yapısı aşağıdaki gibi olmalıdır:

**module** *module\_adi*(Devre Giriş/Çıkış sinyalleri);

**input** Giriş sinyalleri listesi;

**output** Çıkış sinyalleri listesi;

**wire** Ara sinyal terimleri listesi;

**assign** Mantıksal ifadeler;

**assign** Mantıksal ifadeler;

    .....

**endmodule**

Aşağıdaki açıklamalar detaylı olarak QUARTUS II ortamında bir projeyi tanımlamanızı ve VeriLog HDL dilini kullanarak proje ile ilgili tasarımınızı yazılım ortamına girmenizin adımlarını göstermektedir.

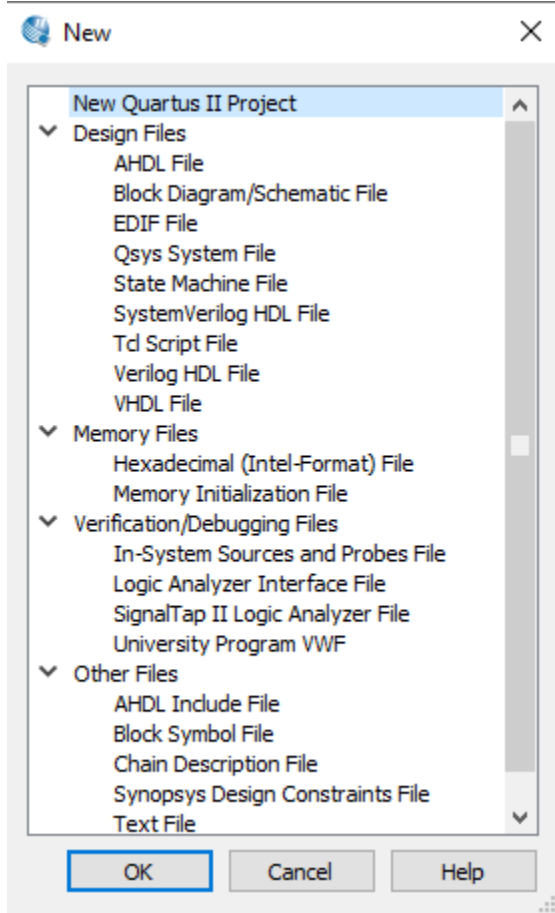
## VHDL ile Tasarım Girişİ

Devre şeması girişine bir alternatif olarak, VeriLog HDL tasarım tanımlama ve girişi için oldukça etkili bir yol olarak kullanılabilir. Büyük boyutlu tasarımlarda, VeriLog HDL ile tasarım üretkenliğin artırılmasında ve tasarım zamanının önemli ölçüde kısaltılmasında önemli katkı sağlar. VeriLog HDL ile sadeleştirme, bağlantı şeması, devre elemanlarının çoklanması gibi işlemler otomatik derleyici ve sentez araçları tarafından yapılır.

### Adım 1: Proje Yaratma ve Kod Girişİ

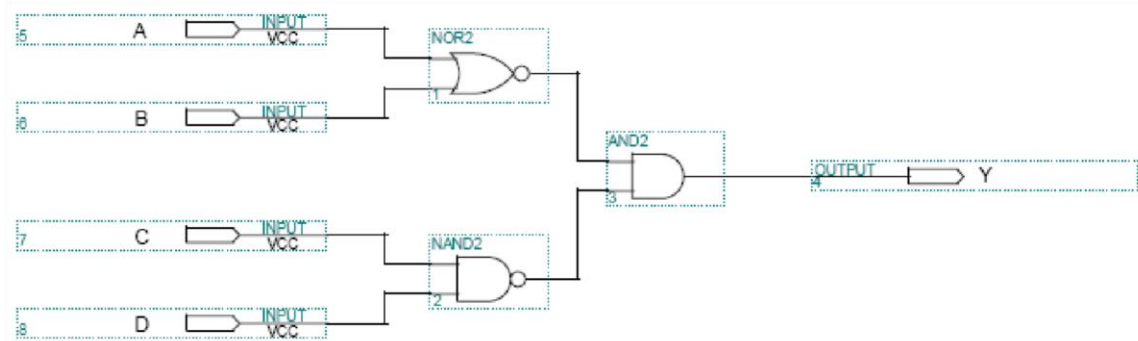
Yeni bir proje tanımlamak için: [File]⇒[New Project Wizard] seçeneğinden sonra proje yaratmak için önceki deneylerde belirtilen adımları uygulayınız. Projenizin adını “BLGM223\_Deney4” olarak belirleyiniz.

[File]⇒[New] menüsünden “SystemVeriLog HDL File” seçiniz ve [Ok] butonuna tıklayınız.



[File]->[Save As] seçeneğiyle dosyanıza “BLGM223\_Deney4” adını veriniz ve [Save] ile kaydediniz.

VHDL kodu yazacağımız devre şeması aşağıdaki gibidir:

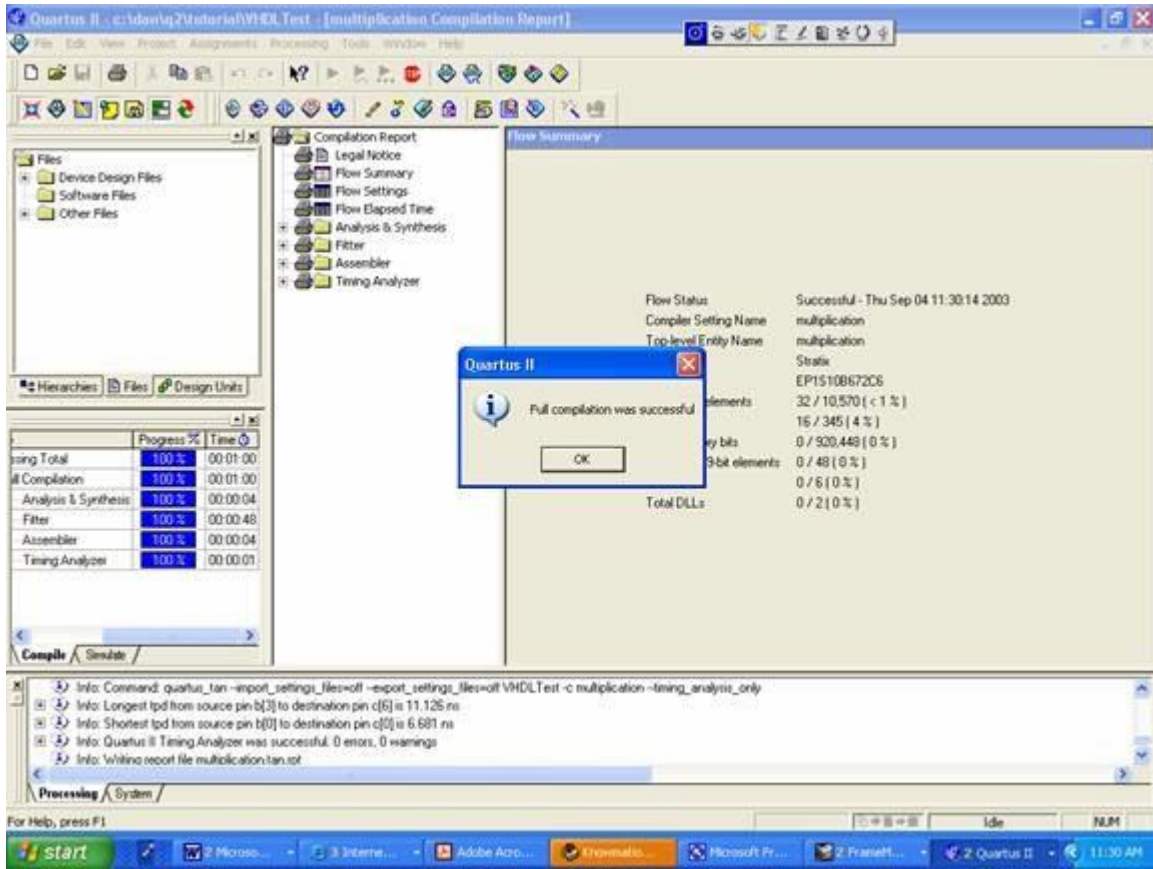


Bu devre ile ilgili olarak aşağıdaki VeriLog HDL kodunu editör pencersine. giriniz.

```
module BLGM223_Deney4(A,B,C,D,Y);  
  input A,B,C,D;  
  output Y;  
  wire T1,T2;  
  
  assign T1= ~(A | B);  
  assign T2= ~(C & D);  
  
  assign Y=T1 & T2;  
  
endmodule
```

### Adım 2. Projenizin Derlenmesi:

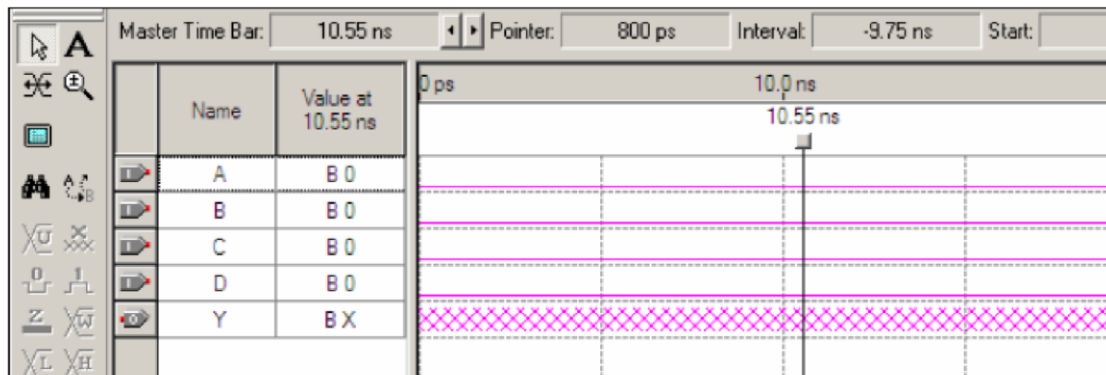
1. **Processing**  $\Rightarrow$  **Start Compilation** seçeneğiyle projenizi derleyiniz.
2. Derleme sonunda **Full compilation was successful** mesajını görmelisiniz. Bu derlemenin başarılı olduğunu ifade eder.
3. Projeniz 0 hata ile sonlanmalıdır. Eğer hatalar var ise, yazdığınız kodu kontrol ederek bu hataları düzeltiniz.
4. Başarılı bir derlemenin sonunda aşağıdaki ekranı gözlemlemelisiniz. Bu durumda derleme penceresini kapatınız.



“Timing Analyzer” menüsünden “Timing Analyzer Summary” seçeneğine tıklayınız. Bu seçenekten “the worst-case tpd” (giriş-çıkış sinyal geçikmesi) işlevini inceleyiniz.

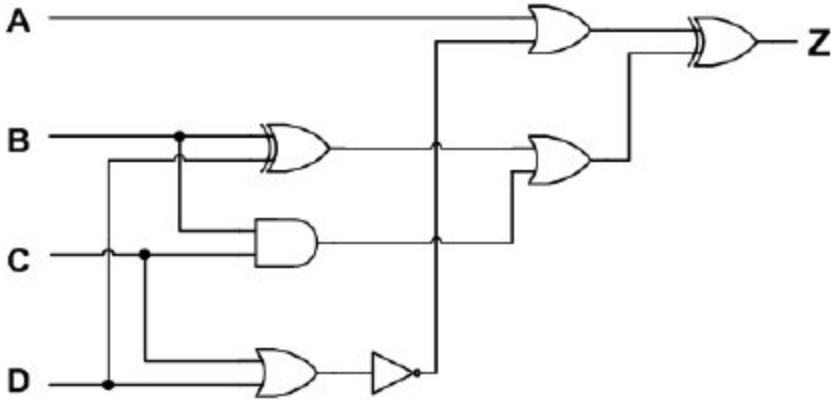
### Adım 3: Projenizin Benzetimi (Simülasyonu):

Deney 2 ve Deney 3’deki adımları uygulayınız.



Dalga formü editör penceresi

Adım 4: Aşağıdaki devreyi dikkate alarak deney adımlarını tekrarlayınız.



BAŞARILAR DİLERİM.

Doç. Dr. Adnan ACAN