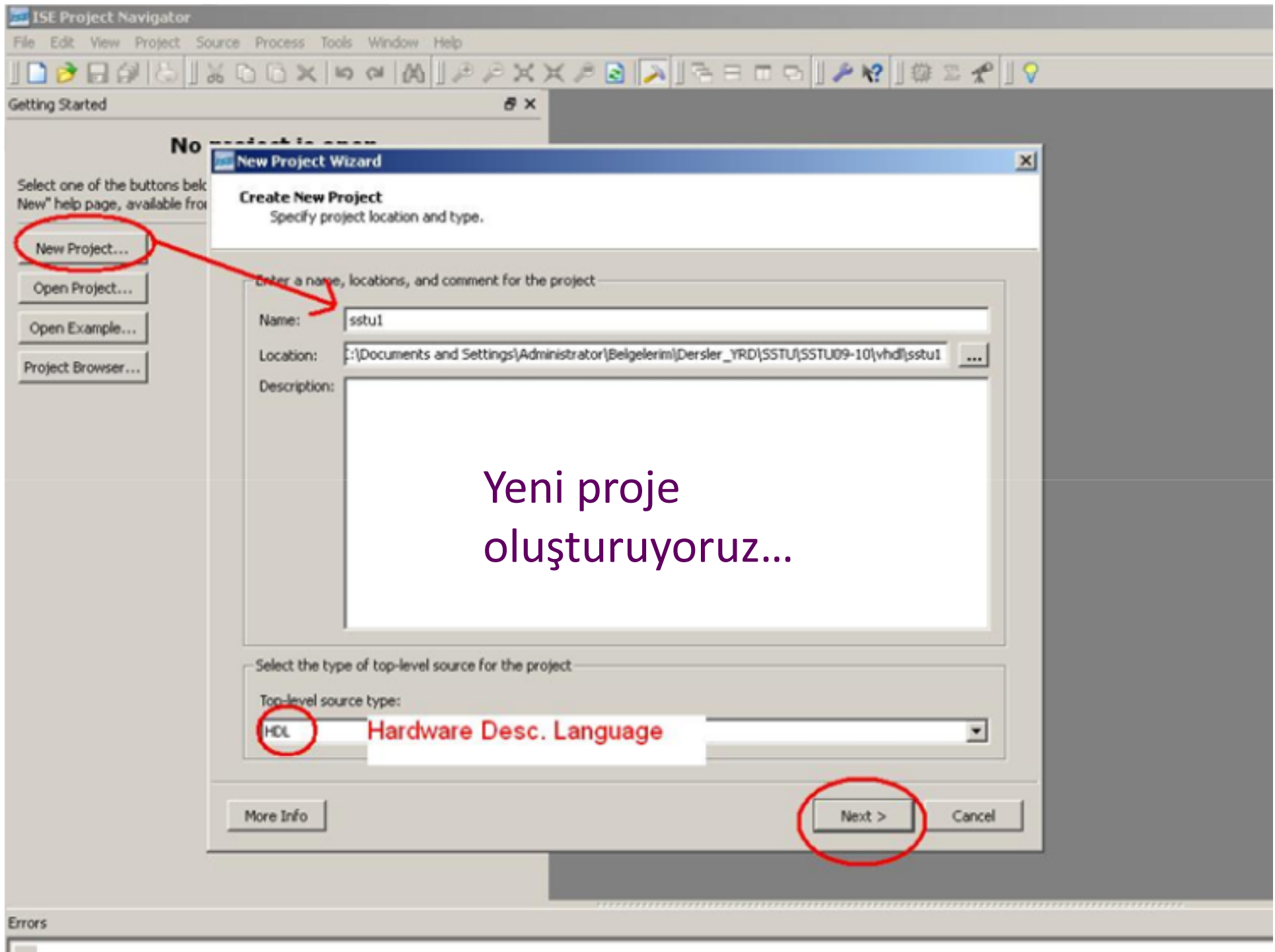


Sayısal Sistem Tasarım Uygulamaları

Xilinx – Yeni proje oluşturmak



New Project Wizard

Device Properties
Specify device and project properties.

Select the device and design flow for the project

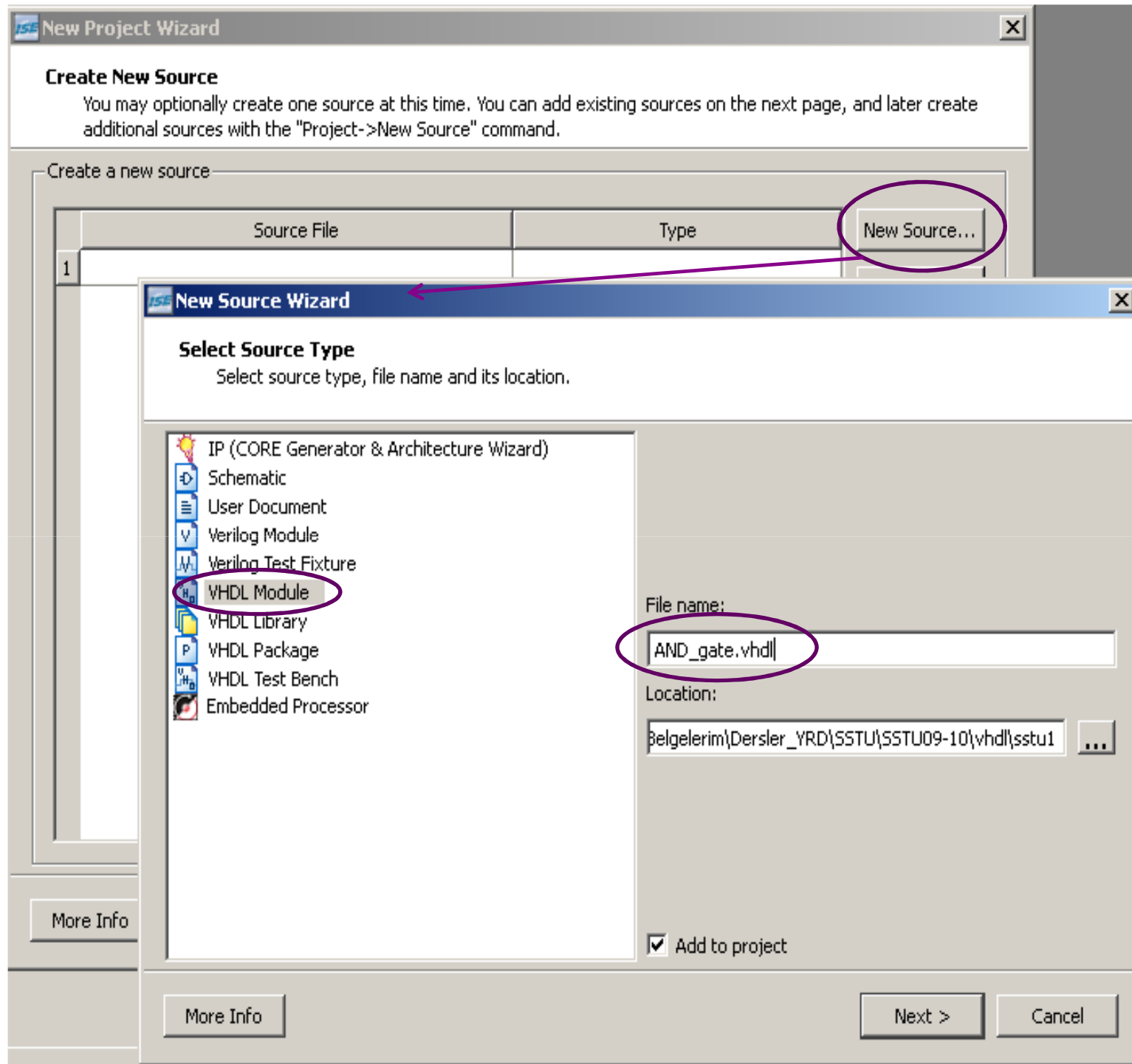
Property Name	Value
Product Category	General Purpose
Family	Spartan3E
Device	XC3S500E
Package	CP132
Speed	-5
Top-Level Source Type	HDL
Synthesis Tool	XST (VHDL/Verilog)
Simulator	ISim (VHDL/Verilog)
Preferred Language	VHDL
Manual Compile Order	☐
Enable Enhanced Design Summary	☒
Enable Message Filtering	☐
Display Incremental Messages	☐

More Info

< Back Next > Cancel

→ Gerçeklemeyeceğiniz için burada seçtikleriniz çok önemli değil.

→ Modelsim XE – VHDL de seçebilirsiniz.



VHDL Sunum
I.pdf içindeki ilk
örneği
deneyeceğiz.
Bunun için önce
AND_gate.vhd
adlı modülü
oluşturalım.

Define Module

Specify ports for module.

Entity name AND_gate

Architecture name Behavioral

Entity oluşturmak için tek yol bu değil,
ancak modülün giriş çıkışlarını burada
belirtirseniz .vhd dosyanızda otomatik
olarak görünecektir. (tüm giriş çıkışlarım
birer bitlik- hiçbir bus değil)

Port Name	Direction	Bus	MSB	LSB
A	in	☐		
B	in	☐		
X	out	☐		
	in	☐		
	in	☐		
	in	☐		
	in	☐		
	in	☐		
	in	☐		
	in	☐		
	in	☐		

More Info

< Back

Next >

Cancel

- Next, next Finish diyerek projeyi oluşturunuz. Karşımıza “Design Summary” dosyası geliyor. Bu dosya şu anda tertemiz..
- Biz AND_gate.vhd’yi açıp architecture yazıyoruz. (Architecture için bkz: VHDL_sunuml.pdf) Herşey hazır geldiğinden yazacağımız tek satır:

`X <= A and B;`

Bir sonraki aşamaya geçmeden önce:

- AND_gate modülünü nasıl oluşturduysak, bir de OR_gate modülü oluşturun.
- (ipucu: veya işlemi or ile tanımlanmıştır.)

AND_gate ve OR_gate modülleriniz hazırsa, bunları bir üst modülde nasıl kullanacağımızı görelim:

- Üst modül olan ao_logic.vhd için de ister wizard kullanarak ister dosyayı oluşturduktan sonra yazarak modul entity sini giriyoruz.
- Bu modulun mimarisi altında 2 AND_gate alt modülü var. Bunları eklemek için hierarcy'de AND_gate.vhd seçili iken, Process → design utilites altında “View HDL Instantenation Template” çalıştırılır. Karşımıza şu gelir:

```
COMPONENT AND_gate
PORT(
  A : IN std_logic;
  B : IN std_logic;
  X : OUT std_logic
);
END COMPONENT;
```

```
Inst_AND_gate: AND_gate PORT MAP(
  A => ,
  B => ,
  X =>
);
```

Component bir defa üst modüle taşınırsa, onu farklı farklı isimlerle istediğimiz sayıda kullanabiliriz. Modüle ve giriş çıkışlarına isim verme işi Port map ile yapılır. PORT MAP, alt modüldeki A,B ve X sinyallerini üst modülde nasıl isimlendirdiğimizi göstermek içindir. Template AND-OR-logic altına kopyalanır ve temel sunumun ilgili sayfasındaki şekle göre isimler yenilenir:

```
G1: AND_gate PORT MAP(
  A => IN1,
  B => IN2,
  X => OUT1
);
```

```
G2: AND_gate PORT MAP(
  A => IN3,
  B => IN4,
  X => OUT2
);
```

OR_gate modüllünü de G3 ismi ile siz ekleyin!

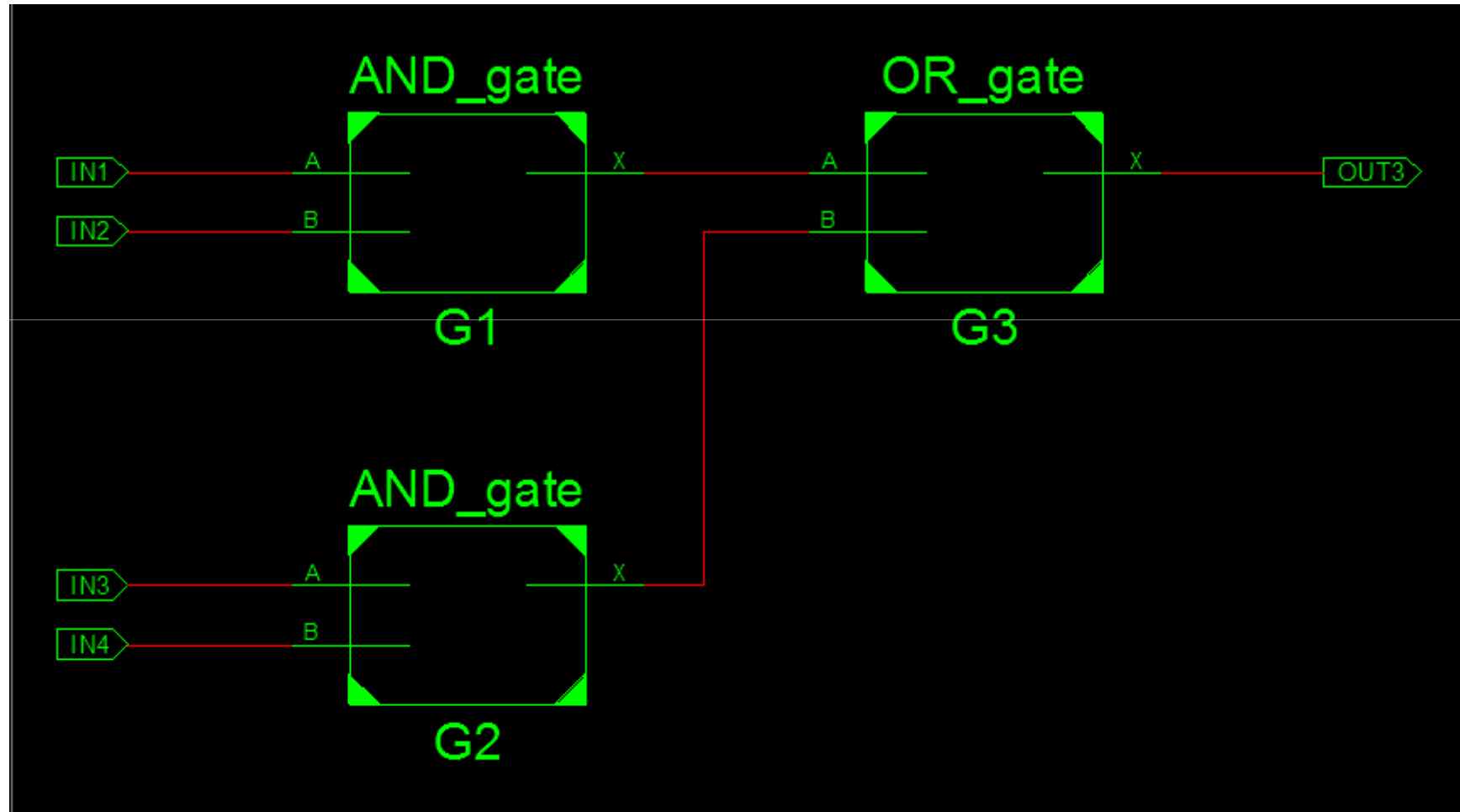
- Herşeyi eklediniz, ama devrenizin içindeki bağlantıları da tanımlamanız gerekir:

```
signal OUT1, OUT2:std_logic;
```

- Dikkat etmeniz gerekenler:
 - Birbiri ile eşleşen sinyaller uyumlu mu? (aynı türde mi tanımlanmış? Örn: X:std_logic iken, OUT1 de std_logic mi?)
 - Syntax doğru mu?

Hepsi tamamsa, RTL schematic oluşturulabilir.
(bknz: Xilinx ISE nasıl kurulur?)

Şunu elde edeceksiniz:



Test Bench

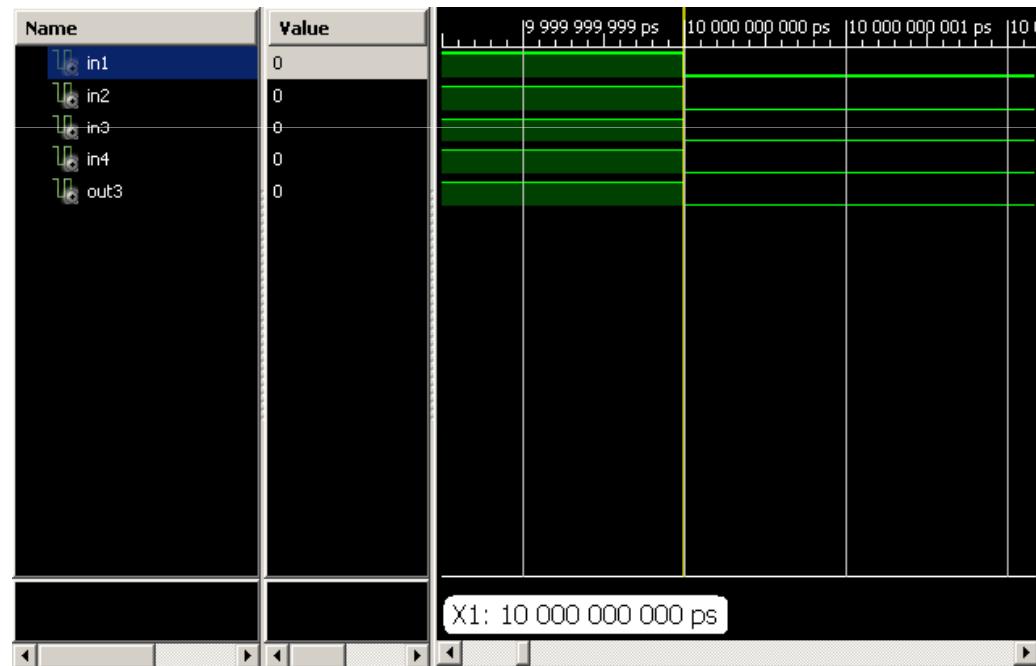
- Simulasyon için bir test bench oluşturalım. Projeye yeni bir “VHDL Test Bench” ekleyerek işe başlayın. (Test dosyasını ao_logic modülü için hazırlayacaksınız.)
- Uzun uzun bir şeyler gelecek önünüze, ama sizin şimdilik clock ve reset ile işiniz olmadığından <clock>_process :process ve stim_proc: process işlemlerini görmezden gelip, yeni bir process yazmanız gerekiyor.
- Ben şunu yazdım:

ve elde ettiğim:

```
test: process
begin
  IN1<='1';
  IN2<='1';
  IN3<='1';
  IN4<='1';

  wait for 10ms;
  IN1<='0';
  IN2<='0';
  IN3<='0';
  IN4<='0';
  wait;

end process;
```



Farklı test processleri yazarak devrenizi deneyin