

VHDL

VHDL:

- * Donanımsal modellenme ve uygulama dizisinden düşür.
- * Very High Speed Integrated Circuit Hardware Description Language

Architecture Behaviour

- Entity'nin davranış şeklini bildirir.

Kullanım

```
architecture behaviour of <entity> is
  <component_tanımlaması>
  <signal_tanımlaması>
begin
  <etiket> : <modul> port map (<portlar>);
  <etiket> : ... ;
end behaviour;
```

giris ve çıkış portları

eğer içinde logic işlem yapıyorsan

process (<portlar>)

begin

<logic_islem>;

end process;

logic isleme yapılacak portlar

Case Condition

- * Koşul yapısı için kullanılır.

Kullanım

```
if (<boolean>) then
```

```
  <logic_islemler>
```

```
elsif (<boolean>) then
```

```
  <logic_islem>
```

```
else
```

```
  <logic_islem>
```

```
end if;
```

then olmaz!

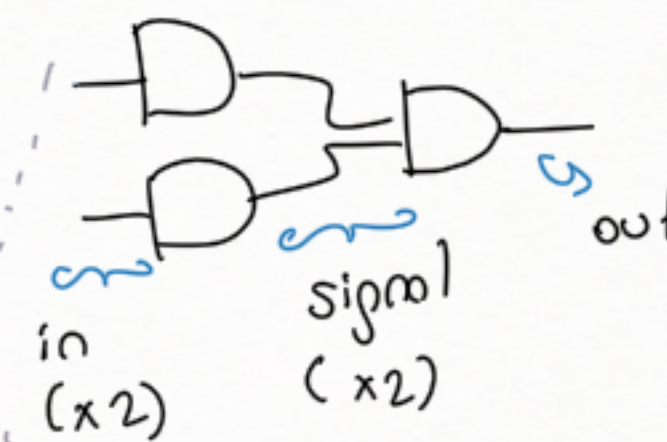
Kutüphane

- * Bahsedilen tanımlamaların yapılabilmesi için gereken kodların bulunduğu yer. (?)

Kullanım

```
library IEEE;
```

```
use IEEE.STDLOGIC_1164.ALL;
```



Kullanım

```
signal <signal_ismi> : STD_LOGIC;
```

Ayırac

Yeni dosya açarak yerine kodları atarak işi tutabilir.

Kullanım

Kullanım

```
entity <modul_ismi> is
  port (
```

```
    <isim1, isim2 ...> : <mod> STD_LOGIC;
```

```
    <isim3, isim4 ...> : <mod> STD_LOGIC;
```

```
  );
```

```
end <modul_ismi>;
```

<mod>

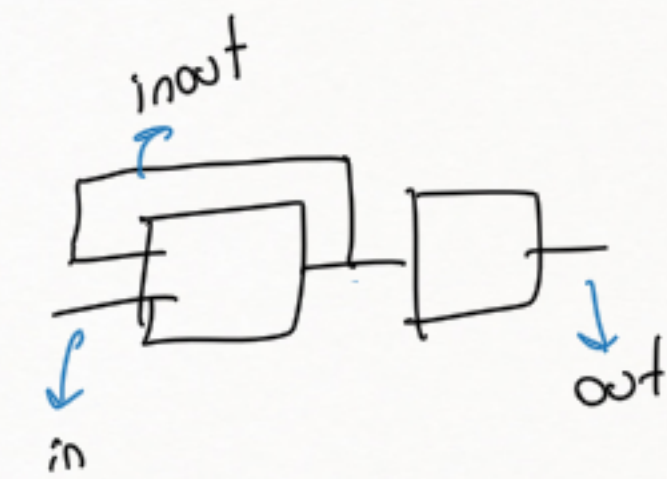
- in: Okunabilir.

- out: Yazılabilir.

- inout: Her ikisi de

- En sonunda ";" olmaz.

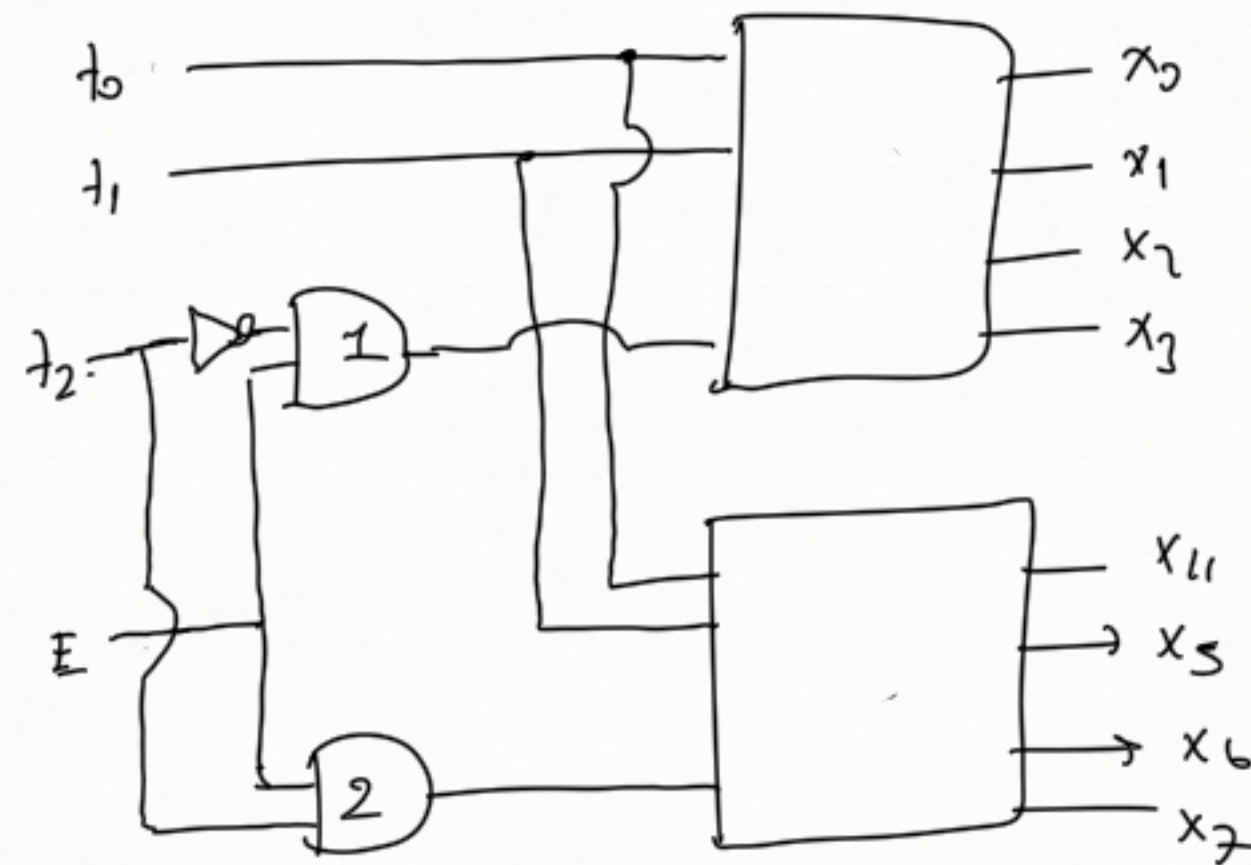
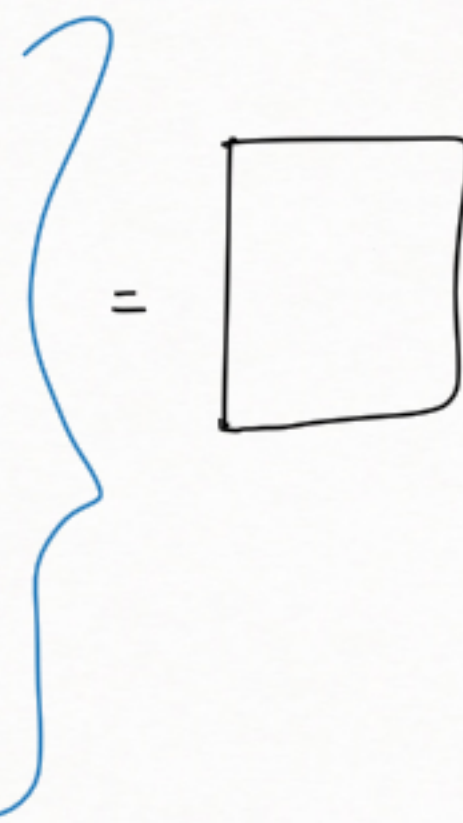
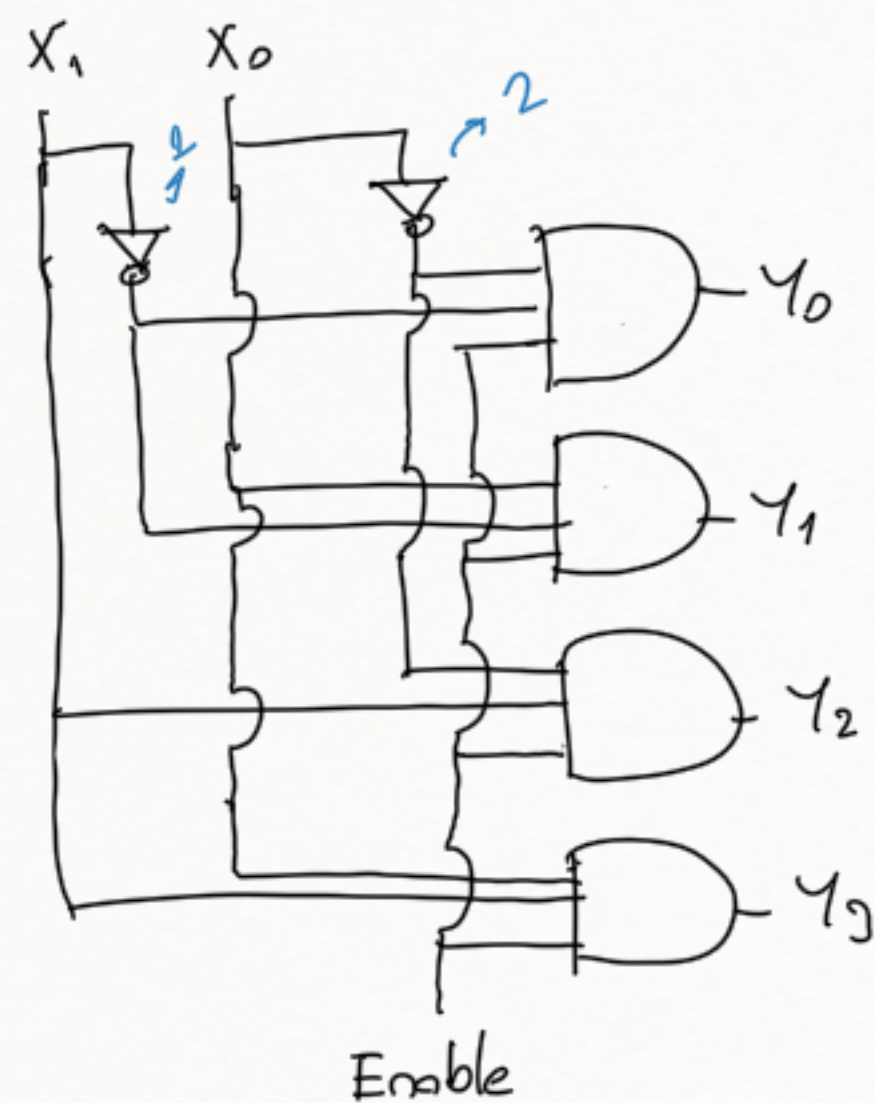
! ; olmaz.



Signal

- * 2 model (entity) arası kabloların verilerini tutar. (ara kablo)

Ev



Karb

```

library IEEE;
use IEEE.STD-LOGIC-1164.ALL;

entity not_gate is
    port(
        x0 : in STD-LOGIC;
        y0 : out STD-LOGIC
    );
end not_gate;

architecture behavior of not_gate is

```

1) Kapıları belirleme

\neg = not-gate

\equiv = triand-gate

= decoder

D = and-gate

ALL = circuit

kapi / devre
entity

koddaki ismi
entity name

2) Sinyal belirleme (signal)

1, 2 çıkışlı sinyal'dir.

1, 2 çıkışlı sinyal'dir.

```

entity Decoder is
    port(
        x0, x1, E : in STD-LOGIC;
        y0, y1, y2, y3 : out STD-LOGIC
    );
end Decoder;

```

architecture behavior of Decoder is

component triand-gate is

```

    port(
        x0, x1 : in STD-LOGIC;
        y0 : out STD-LOGIC
    );
end component;

```

component not_gate is

```

    port(
        x0 : in STD-LOGIC;
        y0 : out STD-LOGIC
    );
end component;

```

```

library IEEE;
use IEEE.STD-LOGIC-1164.ALL;

```

entity triand-gate is

```

    port(
        x0, x1, x2 : in STD-LOGIC;
        y0 : out STD-LOGIC
    );
end triand-gate;

```

architecture behavior of triand-gate is

```

begin
    process (x0, x1, x2)
    begin
        y0 <= x0 and x1 and x2;
    end process;
end behavior;

```

```

begin
    process (x0)

```

```

begin
    y0 <= NOT x0;
end process;
end behavioral;

```

signal n0 : STD-LOGIC;
signal n1 : STD-LOGIC;

~~begin~~
~~process (x0, x1, E)~~ } *logik islem uygulanıyor.*

~~begin~~
p0: not_gate port map (x0, n0);
p1: not_gate port map (x1, n1);
p2: triand-gate port map (n0, n1, E, y0);
...

~~end process;~~ } *process olmadigindan olmaz.*
end Behavioral;

...