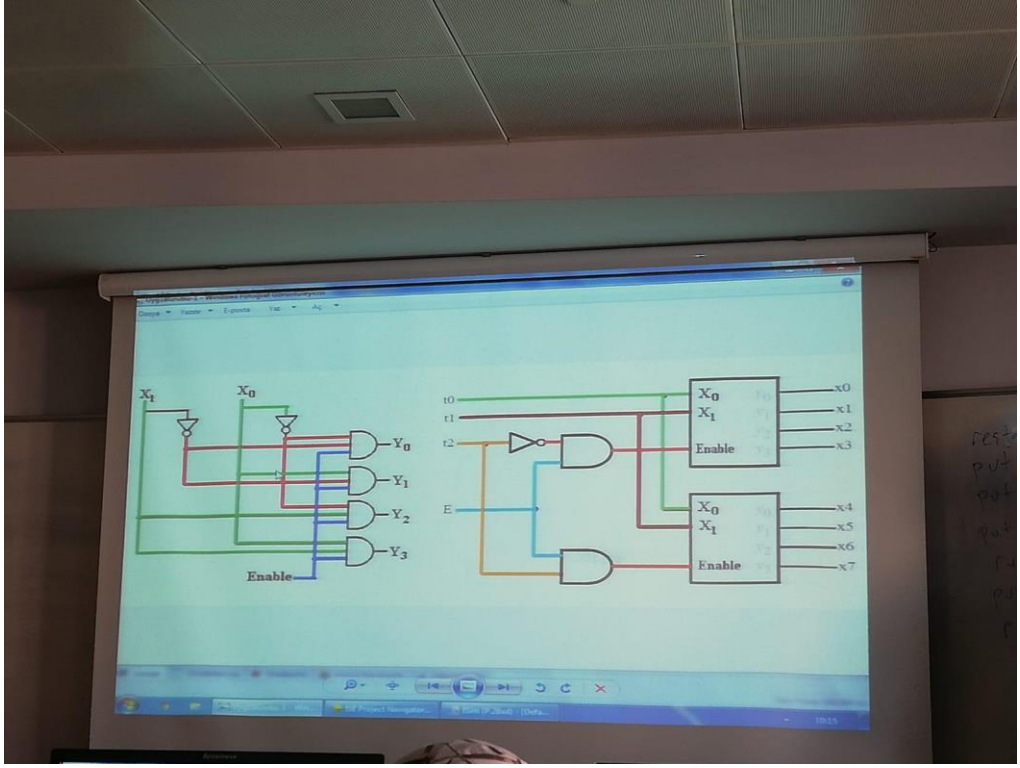


VHDL KODU

```
----------------------------------------------------------------------------------
-- Create Date:
-- Design Name:
-- Module Name:
----------------------------------------------------------------------------------
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity VE_KAPISI is
    port(ve_g1:in STD_LOGIC;
         ve_g2:in STD_LOGIC;
         ve_cikis:out STD_LOGIC);
end VE_KAPISI;

architecture veri_akisi of VE_KAPISI
is begin
    process(ve_g1,ve_g2)
    begin
```

```
        ve_cikis <= ve_g1 and ve_g2;
    end process;
end veri_akisi;
----------------------------------------------------------------------------------
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity VE_KAPISI3 is
```

```
    port(ve_g1:in STD_LOGIC;
         ve_g2:in STD_LOGIC;
         ve_g3:in STD_LOGIC;
         ve_cikis:out STD_LOGIC);
end VE_KAPISI3;

architecture veri_akisi of VE_KAPISI3
is begin
    process(ve_g1,ve_g2,ve_g3)
    begin
        ve_cikis <= ve_g1 and ve_g2 and
    ve_g3; end process;
end veri_akisi;
--------------------------------------------------------------------------------
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity NOT_KAPISI is
    port(not_giris:in STD_LOGIC;
         not_cikis:out STD_LOGIC);
end NOT_KAPISI;

architecture veri_akisi of NOT_KAPISI
is begin
    process(not_giris)
    begin
        not_cikis <= NOT not_giris;
```

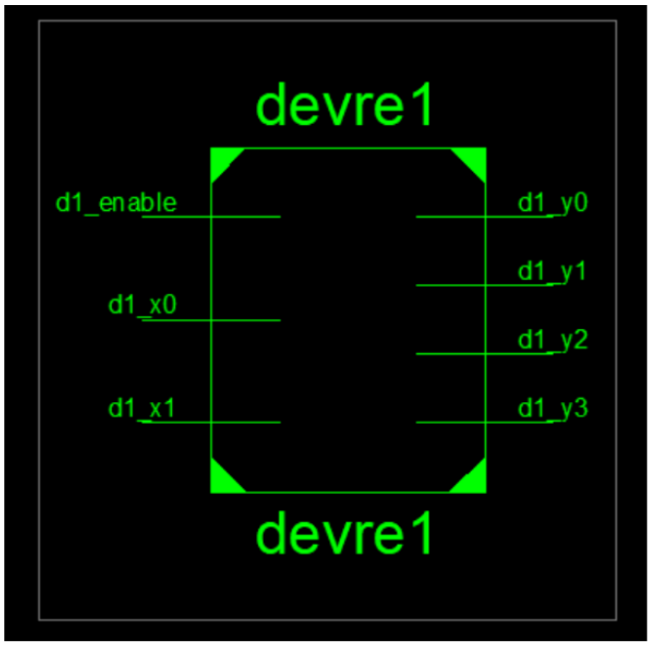
```
    end process;
end veri_akisi;
--------------------------------------------------------------------------------
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity devre1 is
    port(d1_x1: in STD_LOGIC;
         d1_x0: in STD_LOGIC;
         d1_enable: in STD_LOGIC;
         d1_y0: out STD_LOGIC;
         d1_y1: out STD_LOGIC;
         d1_y2: out STD_LOGIC;
         d1_y3: out STD_LOGIC);
end devre1;

architecture yapisal of devre1 is
    component NOT_KAPISI is
        port(not_giris:in STD_LOGIC;
             not_cikis:out STD_LOGIC);
    end component;
    component VE_KAPISI is
         port(ve_g1:in STD_LOGIC;
              ve_g2:in STD_LOGIC;
              ve_cikis:out STD_LOGIC);
    end component;
```

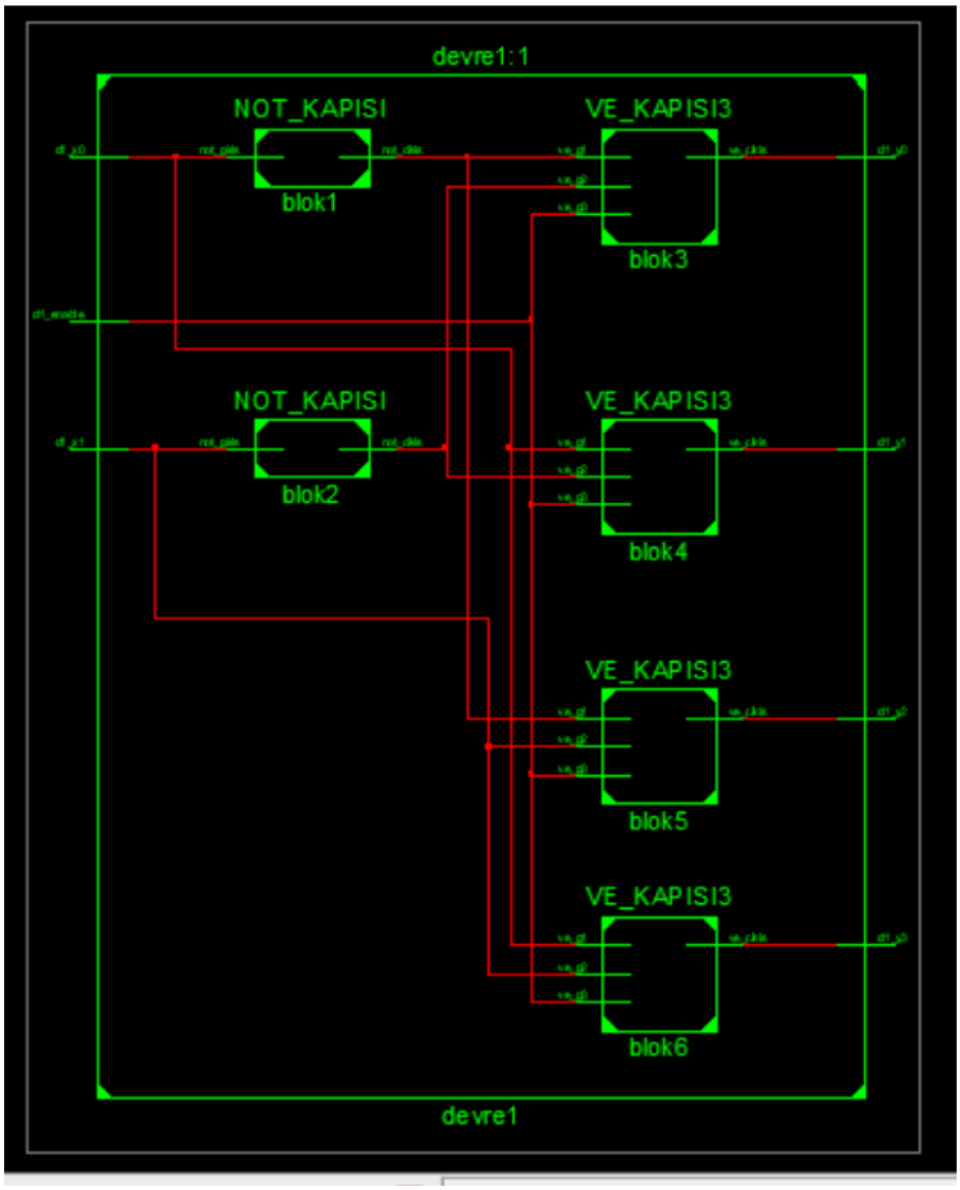
```
component VE_KAPISI3 is
      port(ve_g1:in STD_LOGIC;
            ve_g2:in STD_LOGIC;
            ve_g3:in STD_LOGIC;
            ve_cikis:out STD_LOGIC);
    end component;

    signal arakablo0: STD_LOGIC;
    signal arakablo1: STD_LOGIC;

begin
    blok1: NOT_KAPISI port map(not_giris =>d1_x0 , not_cikis=>arakablo0);
    blok2: NOT_KAPISI port map(not_giris=>d1_x1 , not_cikis=>arakablo1);
    blok3: VE_KAPISI3 port map(ve_g1=>arakablo0 , ve_g2=>arakablo1
    , ve_g3=>d1_enable , ve_cikis=>d1_y0);
    blok4: VE_KAPISI3 port map(ve_g1 =>d1_x0 , ve_g2=>arakablo1 ,
    ve_g3=>d1_enable , ve_cikis=>d1_y1);
    blok5: VE_KAPISI3 port map(ve_g1=>arakablo0 , ve_g2=>d1_x1
    , ve_g3=>d1_enable , ve_cikis=>d1_y2);
    blok6: VE_KAPISI3 port map(ve_g1=>d1_x0 , ve_g2 =>d1_x1
    , ve_g3=>d1_enable , ve_cikis=>d1_y3);
end yapisal;
--------------------------------------------------------------------------------
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity DEVRE is
```

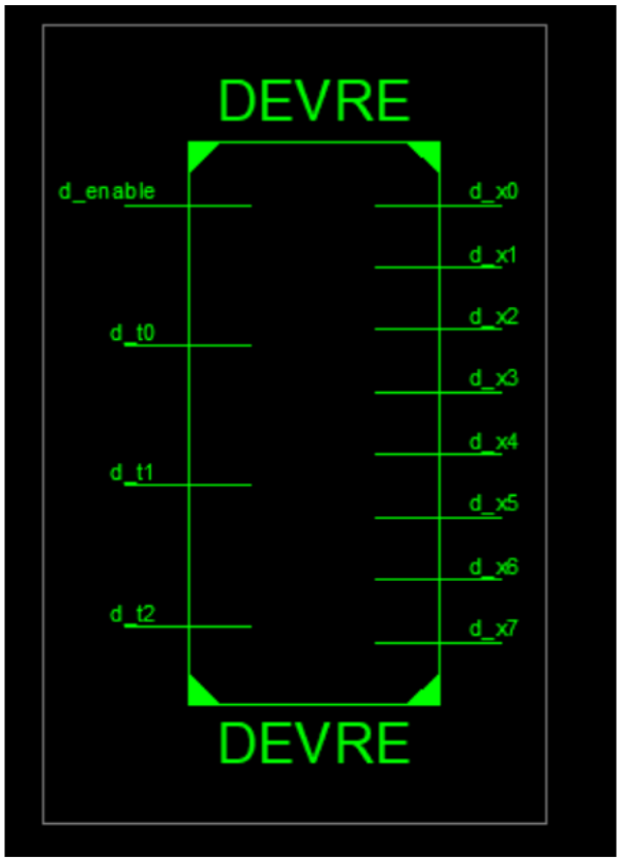
```
    port(d_t0:in STD_LOGIC;
         d_t1:in STD_LOGIC;
         d_t2:in STD_LOGIC;
         d_enable:in STD_LOGIC;
         d_x0:out STD_LOGIC;
         d_x1:out STD_LOGIC;
         d_x2:out STD_LOGIC;
         d_x3:out STD_LOGIC;
         d_x4:out STD_LOGIC;
         d_x5:out STD_LOGIC;
         d_x6:out STD_LOGIC;
         d_x7:out STD_LOGIC);
end DEVRE;

 architecture yapisal of DEVRE is
    component NOT_KAPISI is
        port(not_giris:in STD_LOGIC;
             not_cikis:out STD_LOGIC);
    end component;
    component VE_KAPISI is
        port(ve_g1:in STD_LOGIC;
             ve_g2:in STD_LOGIC;
             ve_cikis:out STD_LOGIC);
    end component;
    component devre1 is
        port(d1_x1: in STD_LOGIC;
             d1_x0: in STD_LOGIC;
```

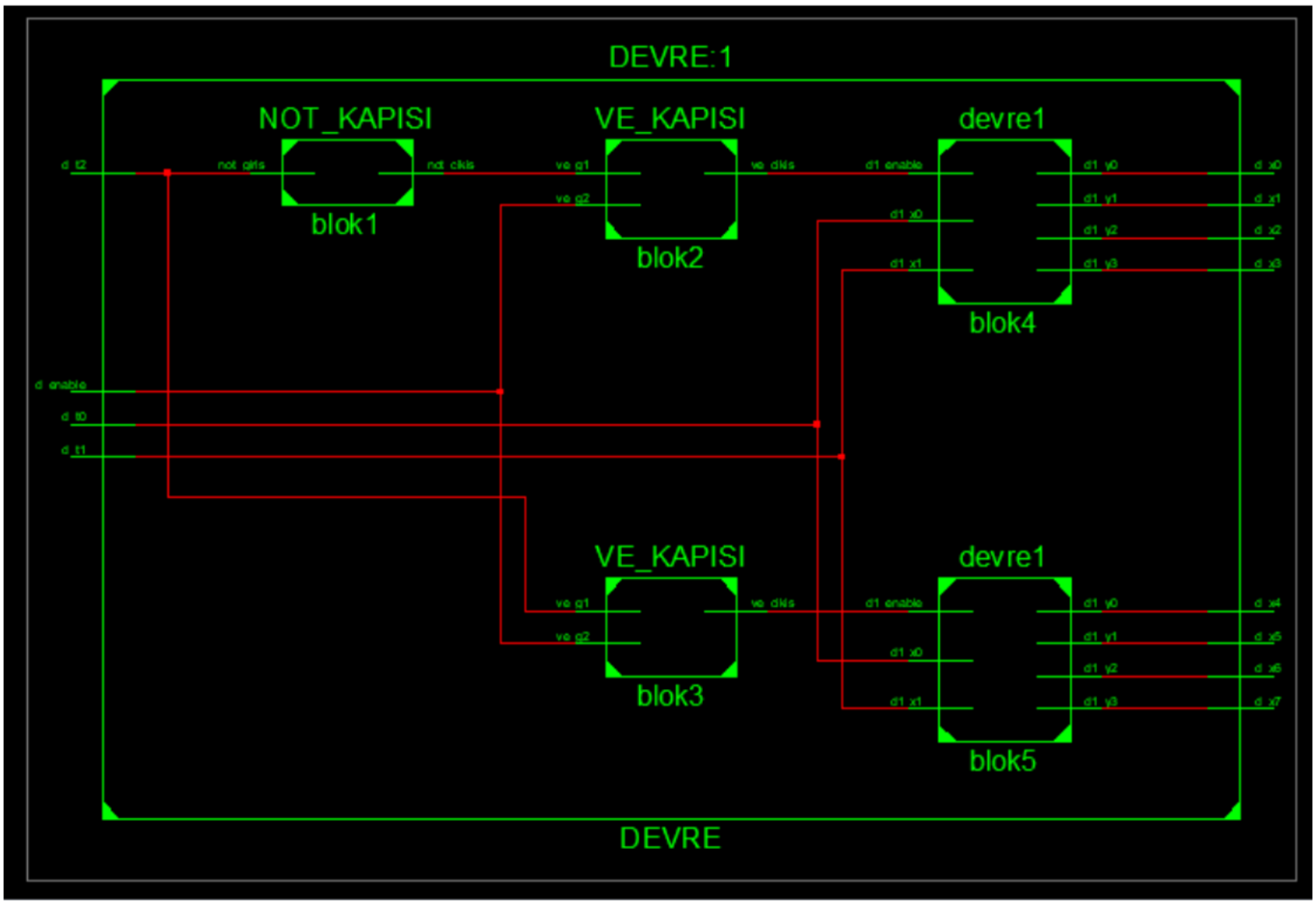
```
            d1_enable: in STD_LOGIC;
            d1_y0: out STD_LOGIC;
            d1_y1: out STD_LOGIC;
            d1_y2: out STD_LOGIC;
            d1_y3: out STD_LOGIC);
    end component;

    signal arakablo1: STD_LOGIC;
    signal arakablo2: STD_LOGIC;
    signal arakablo3: STD_LOGIC;

begin
    blok1: NOT_KAPISI port map(not_giris=>d_t2 , not_cikis=>arakablo1);
    blok2: VE_KAPISI port map(ve_g1=>arakablo1 , ve_g2=>d_enable ,
    ve_cikis=>arakablo2);
    blok3: VE_KAPISI port map(ve_g1=>d_t2 , ve_g2=>d_enable ,
    ve_cikis=>arakablo3);
    blok4: devre1 port map(d1_x0=>d_t0 , d1_x1=>d_t1 , d1_enable=>arakablo2 ,
d1_y0=>d_x0 , d1_y1=>d_x1 , d1_y2=>d_x2 , d1_y3=>d_x3 );
    blok5: devre1 port map(d1_x0=>d_t0 , d1_x1=>d_t1 , d1_enable=>arakablo3
, d1_y0=>d_x4 , d1_y1=>d_x5 , d1_y2=>d_x6 , d1_y3=>d_x7 );
end yapisal;
```

RTL ŞEMASI


devre1
d1_enable
d1_x0
d1_x1
d1_y0
d1_y1
d1_y2
d1_y3
devre1


devre1:1
NOT_KAPISI
blok1
VE_KAPISI3
blok3
NOT_KAPISI
blok2
VE_KAPISI3
blok4
VE_KAPISI3
blok5
VE_KAPISI3
blok6
devre1


DEVRE
d_enable
d_t0
d_t1
d_t2
d_x0
d_x1
d_x2
d_x3
d_x4
d_x5
d_x6
d_x7
DEVRE


DEVRE:1
NOT_KAPISI
VE_KAPISI
devre1
d_t2
not giris
not cikis
blok1
ve g1
ve g2
ve cikis
blok2
d1 enable
d1 x0
d1 x1
d1 y0
d1 y1
d1 y2
d1 y3
d_x0
d_x1
d_x2
d_x3
blok4
d_enable
d_t0
d_t1
VE_KAPISI
devre1
ve g1
ve g2
ve cikis
blok3
d1 enable
d1 x0
d1 x1
d1 y0
d1 y1
d1 y2
d1 y3
d_x4
d_x5
d_x6
d_x7
blok5
DEVRE

SİMÜLASYON DALGA FORMU