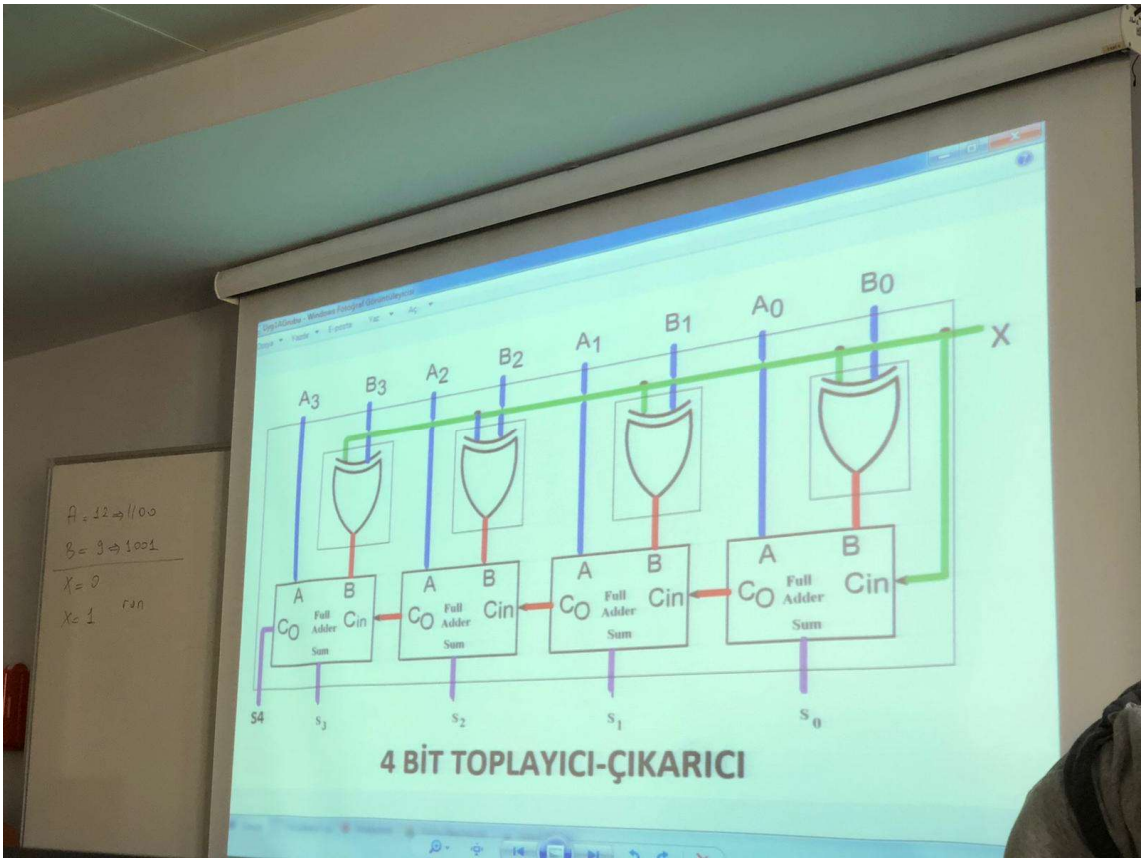



# VHDL KODU

Company:
-- Engineer:
-- Create Date:  09:29:28 10/18/2019
-- Design Name:  4 bit toplayıcı-çıkarıcı devresi
-- Module Name:  ToplayiciCikarici - Behavioral
-- Project Name:
-- Target Devices:
-- Tool versions:
-- Description:
-- Dependencies:
-- Revision:
-- Revision 0.01 - File Created
-- Additional Comments:
----------

```
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 VEYA_KAPISI is
   port(    veya_g1: in STD_LOGIC;
        veya_g2: in STD_LOGIC;
        veya_cikis: out STD_LOGIC);
end VEYA_KAPISI;

architecture veri_akisi of VEYA_KAPISI is
begin
   process(veya_g1, veya_g2)
   begin
      veya_cikis <= veya_g1 or veya_g2;
   end process;
end veri_akisi;
----------------------------------------------------
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity XOR_KAPISI is
   port(    xor_g1: in STD_LOGIC;
        xor_g2: in STD_LOGIC;
        xor_cikis: out STD_LOGIC);
end XOR_KAPISI;

architecture veri_akisi of XOR_KAPISI is
begin
   process(xor_g1, xor_g2)
```

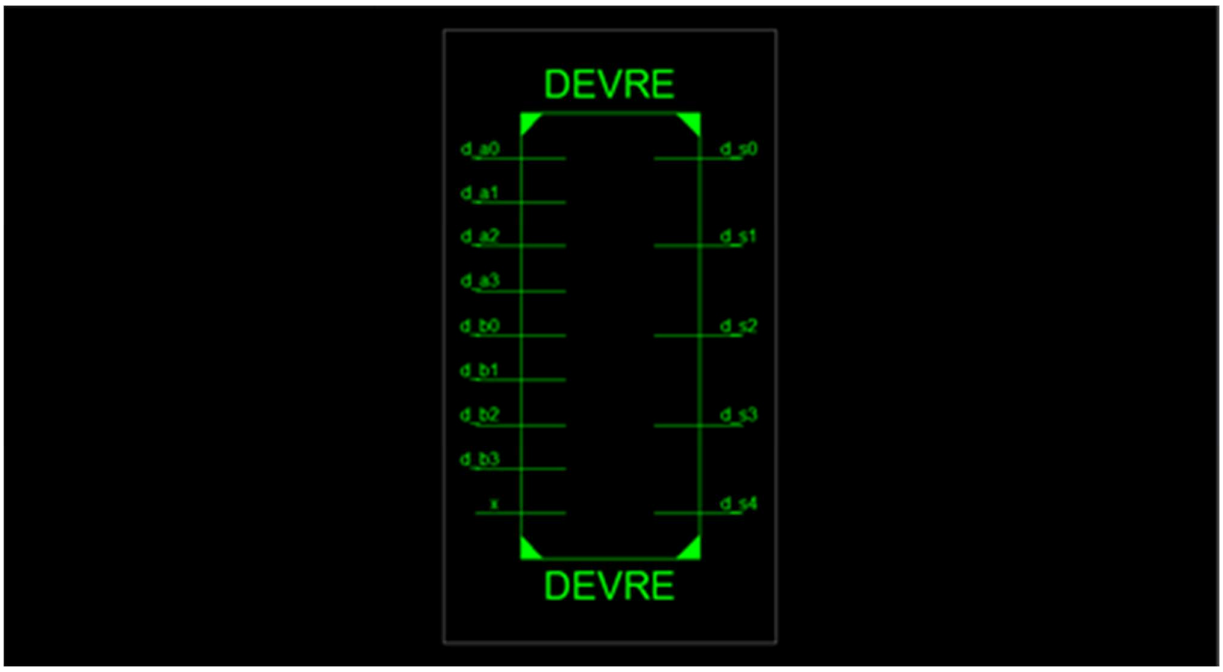
```
    begin
        xor_cikis <= xor_g1 xor xor_g2;
    end process;
end veri_akisi;
-------------------------------------------------
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity YT is
    port(    yt_g1: in STD_LOGIC;
             yt_g2: in STD_LOGIC;
             yt_toplam: out STD_LOGIC;
             yt_eldecikisi: out STD_LOGIC);
end YT;

architecture yapisal of YT is
    component VE_KAPISI is
        port(    ve_g1:in STD_LOGIC;
                 ve_g2:in STD_LOGIC;
                 ve_cikis:out STD_LOGIC);
    end component;
    component XOR_KAPISI is
        port(    xor_g1: in STD_LOGIC;
                 xor_g2: in STD_LOGIC;
                 xor_cikis: out STD_LOGIC);
    end component;
begin
    blok1: XOR_KAPISI port map(xor_g1 => yt_g1, xor_g2 => yt_g2, xor_cikis
=> yt_toplam);
    blok2: VE_KAPISI port map(ve_g1 => yt_g1, ve_g2 => yt_g2, ve_cikis =>
yt_eldecikisi);
end yapisal;
-------------------------------------------------------------------------
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity TT is
    port(    tt_g1: in STD_LOGIC;
             tt_g2: in STD_LOGIC;
             tt_eldegirisi: in STD_LOGIC;
             tt_toplam: out STD_LOGIC;
```

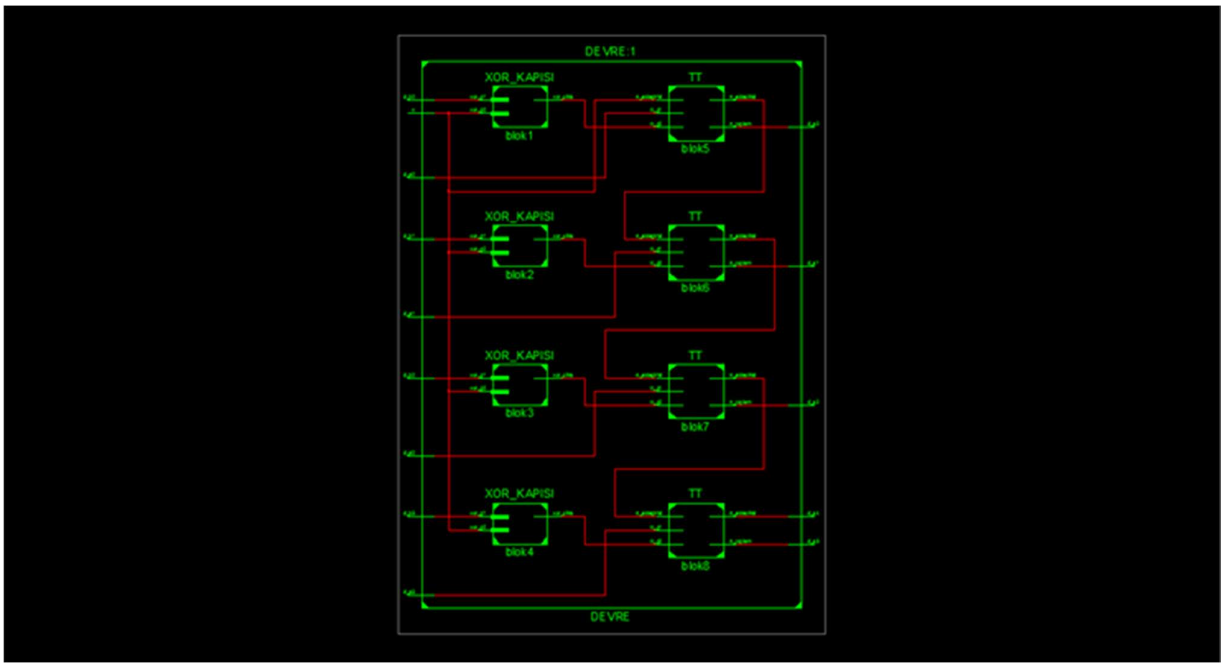
```
        tt_eldecikisi: out STD_LOGIC
    );
end TT;

architecture yapisal of TT is
    component YT is
        port(    yt_g1: in STD_LOGIC;
                 yt_g2: in STD_LOGIC;
                 yt_toplam: out STD_LOGIC;
                 yt_eldecikisi: out STD_LOGIC);
    end component;
    component  VEYA_KAPISI is
        port(    veya_g1: in STD_LOGIC;
                 veya_g2: in STD_LOGIC;
                 veya_cikis: out STD_LOGIC);
    end component;
    signal arakablo1: STD_LOGIC;
    signal arakablo2: STD_LOGIC;
    signal arakablo3: STD_LOGIC;
begin
    blok1: YT port map(yt_g1 => tt_g1, yt_g2 => tt_g2, yt_eldecikisi =>
arakablo2,
    yt_toplam => arakablo1);
    blok2: YT port map(yt_g1 => arakablo1, yt_g2 => tt_eldegirisi,
    yt_eldecikisi => arakablo3, yt_toplam => tt_toplam);
    blok3: VEYA_KAPISI port map(veya_g1 => arakablo2, veya_g2 =>
arakablo3,
    veya_cikis => tt_eldecikisi);
end yapisal;

--------------------------------------------------------------------------------
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity DEVRE is
    port(    d_a0:in STD_LOGIC;
    d_a1:in STD_LOGIC;
    d_a2:in STD_LOGIC;
    d_a3:in STD_LOGIC;
    d_b0:in STD_LOGIC;
    d_b1:in STD_LOGIC;
```

```
    d_b2:in STD_LOGIC;
    d_b3:in STD_LOGIC;
    d_s0:out STD_LOGIC;
    d_s1:out STD_LOGIC;
    d_s2:out STD_LOGIC;
    d_s3:out STD_LOGIC;
    d_s4:out STD_LOGIC;
    x:in STD_LOGIC);
end DEVRE;


architecture yapisal of DEVRE is
    component TT is
        port(    tt_g1: in STD_LOGIC;
            tt_g2: in STD_LOGIC;
            tt_eldegirisi: in STD_LOGIC;
            tt_toplam: out STD_LOGIC;
            tt_eldecikisi: out STD_LOGIC    );
    end component;
    component  XOR_KAPISI is
        port(xor_g1: in STD_LOGIC;
            xor_g2: in STD_LOGIC;
            xor_cikis: out STD_LOGIC);
    end component;

   signal arakablo1: STD_LOGIC;
    signal arakablo2: STD_LOGIC;
    signal arakablo3: STD_LOGIC;
    signal arakablo4: STD_LOGIC;
    signal arakablo5: STD_LOGIC;
    signal arakablo6: STD_LOGIC;
    signal arakablo7: STD_LOGIC;

begin
    blok1: XOR_KAPISI port map(xor_g1 =>d_b0 ,xor_g2  => x,
    xor_cikis => arakablo1 );
    blok2: XOR_KAPISI port map( xor_g1=>d_b1 ,xor_g2  => x,
    xor_cikis => arakablo2);
    blok3: XOR_KAPISI port map(xor_g1 =>d_b2 ,xor_g2 => x ,
    xor_cikis => arakablo3);
    blok4: XOR_KAPISI port map(  xor_g1=>d_b3,xor_g2=>  x ,
```

```
    xor_cikis =>arakablo4 );
    blok5: TT port map( tt_g1=>d_a0 ,tt_g2  => arakablo1, tt_eldecikisi =>
arakablo5, tt_eldegirisi =>x ,
    tt_toplam => d_s0);
    blok6: TT port map(tt_g1=>d_a1 ,tt_g2  => arakablo2, tt_eldecikisi =>
arakablo6, tt_eldegirisi =>arakablo5 ,
    tt_toplam => d_s1);
    blok7: TT port map(tt_g1=>d_a2 ,tt_g2  => arakablo3, tt_eldecikisi =>
arakablo7, tt_eldegirisi =>arakablo6,
    tt_toplam => d_s2);
    blok8: TT port map(tt_g1=>d_a3 ,tt_g2  => arakablo4, tt_eldecikisi => d_s4,
tt_eldegirisi =>arakablo7 ,
    tt_toplam => d_s3);

end yapisal;
```

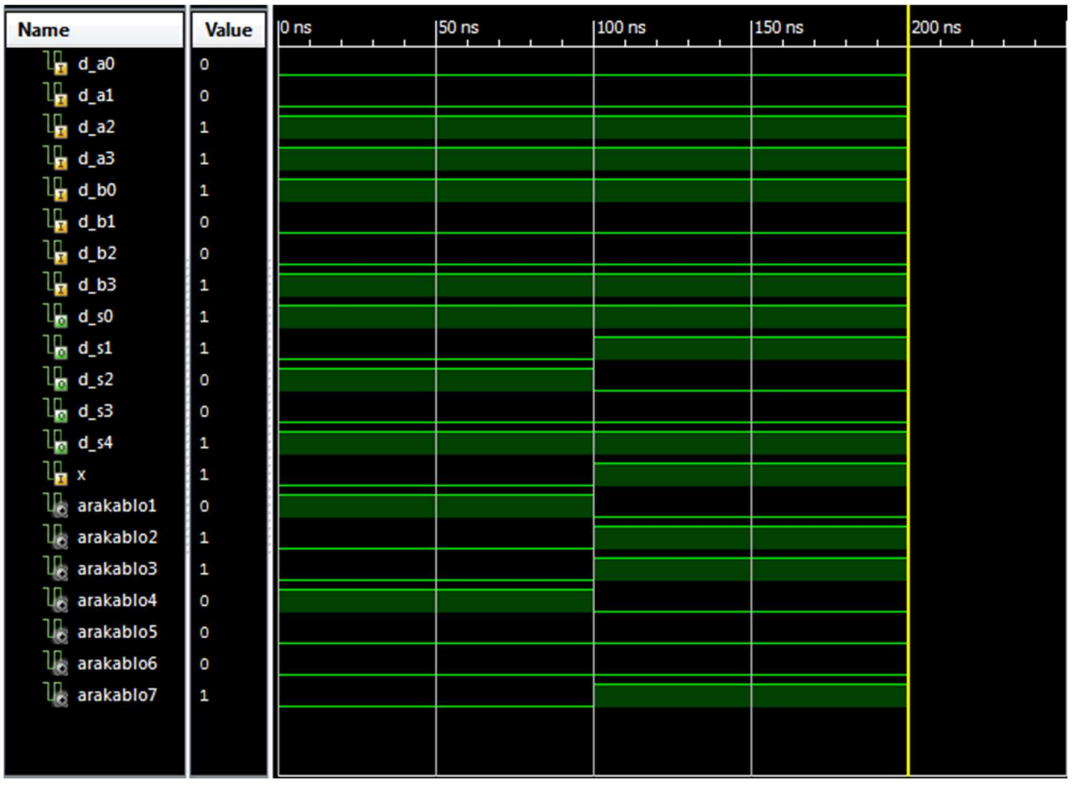
# RTL ŞEMASI

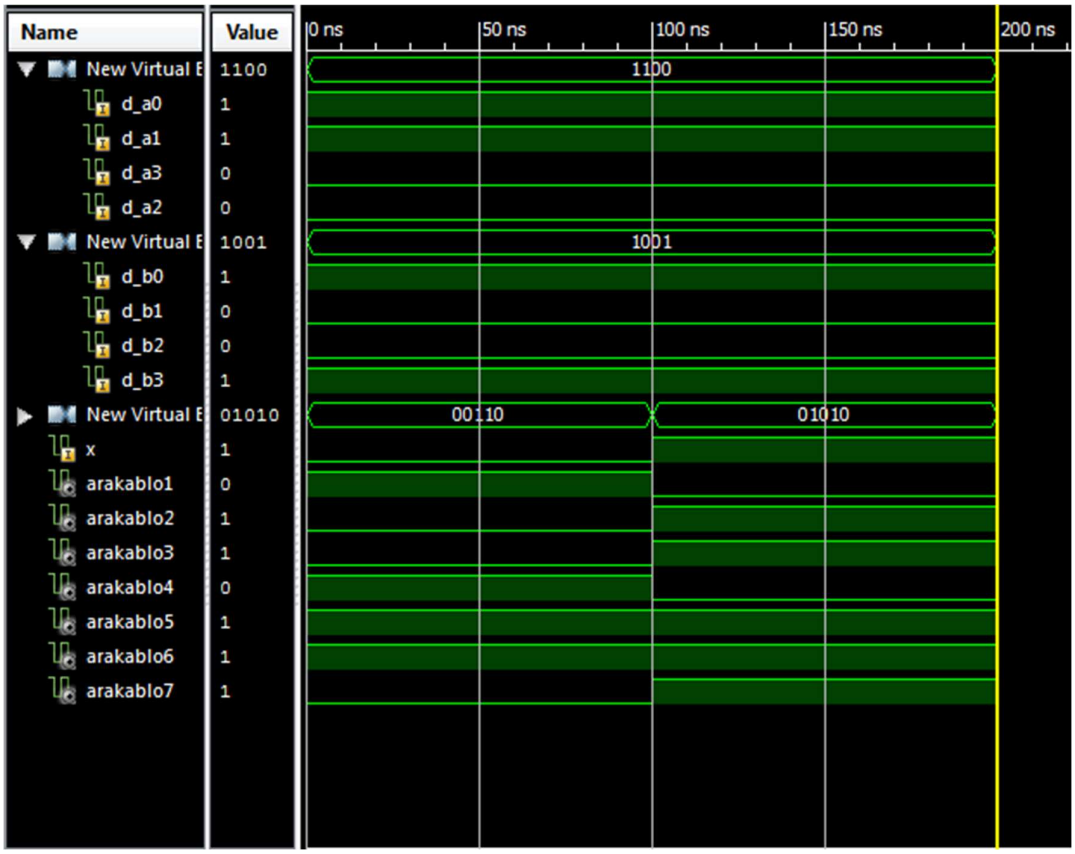





# SİMÜLASYON DALGA FORMU

✓ 12 ve 9 değerleri için toplama ve çıkarma işlemi yapılmıştır.




Name
Value
0 ns
50 ns
100 ns
150 ns
200 ns
New Virtual E
1100
d_a0
1
d_a1
1
d_a3
0
d_a2
0
New Virtual E
1001
d_b0
1
d_b1
0
d_b2
0
d_b3
1
New Virtual E
01010
x
1
arakablo1
0
arakablo2
1
arakablo3
1
arakablo4
0
arakablo5
1
arakablo6
1
arakablo7
1
1100
1001
00110
01010