

DENEY NO: 01

DENEY ADI: Ve Kapısı'nın VHDL ile Yapısal Tasarımı

Deneyin Amacı: A ve b giriş, x çıkış bitlerinin kullanılmasıyla AND Kapısının davranışının VHDL ile yapısal olarak tasarımının yapılması.

Deneyin Yapılışı: Devre aşağıdaki gibi çalışma göstermektedir.

GİRİŞ		ÇIKIŞ
A	B	X
0	0	0
0	1	0
1	0	0
1	1	1

Tablo 1. Devrenin doğruluk tablosu

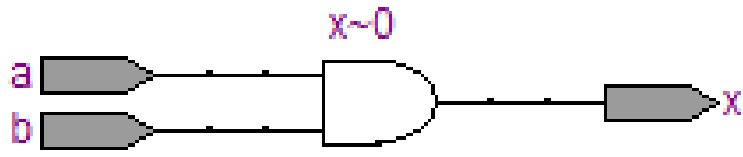
Tablo 1.'de verilen doğruluk tablosu aşağıdaki gibi kodlanmıştır.

```
LIBRARY ieee;
USE ieee.std_logic_1164.all;




ENTITY ve_kapisi IS
PORT (a:IN BIT;
      b:IN BIT;
      x:OUT BIT);
END ve_kapisi;

ARCHITECTURE mimari1 OF ve_kapisi IS
BEGIN
x<=a AND b;
END mimari1;
```

Deney Sonuçları: Tasarlanan devrenin RTL diyagramı ve benzetim sonuçları aşağıdaki gibi elde edilmiştir.



Şekil 1. Devrenin RTL diyagramı

Messages									
 /ve_kapisi/a	1								
 /ve_kapisi/b	1								
 /ve_kapisi/x	1								

Şekil 2. Devrenin benzetim sonuçları

DENEY NO: 02

DENEY ADI: Veya Kapısı'nın VHDL ile Yapısal Tasarımı

Deneyin Amacı: A ve b giriş, x çıkış bitlerinin kullanılmasıyla OR Kapısının davranışının VHDL ile yapısal olarak tasarımının yapılması.

Deneyin Yapılışı: Devre aşağıdaki gibi çalışma göstermektedir.

GİRİŞ		ÇIKIŞ
A	B	X
0	0	0
0	1	1
1	0	1
1	1	1

Tablo 1. Devrenin doğruluk tablosu

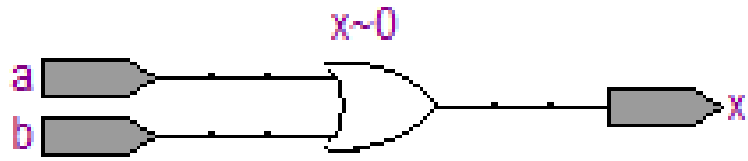
Tablo 1.'de verilen doğruluk tablosu aşağıdaki gibi kodlanmıştır.

```
LIBRARY ieee;
USE ieee.std_logic_1164.all;

ENTITY veya_kapisi IS
PORT (a:IN BIT;
      b:IN BIT;
      x:OUT BIT);
END veya_kapisi;

ARCHITECTURE mimari1 OF veya_kapisi IS
BEGIN
x<=a OR b;
END mimari1;
```

Deney Sonuçları: Tasarlanan devrenin RTL diyagramı ve benzetim sonuçları aşağıdaki gibi elde edilmiştir.



Şekil 1. Devrenin RTL diyagramı

Messages									
♦ /veya_kapisi/a	1								
♦ /veya_kapisi/b	1								
♦ /veya_kapisi/x	1								

Şekil 2. Devrenin benzetim sonuçları

DENEY NO: 03

DENEY ADI: Ya da Kapısı'nın VHDL ile Yapısal Tasarımı

Deneyin Amacı: A ve b giriş, x çıkış bitlerinin kullanılmasıyla XOR Kapısının davranışının VHDL ile yapısal olarak tasarımının yapılması.

Deneyin Yapılışı: Devre aşağıdaki gibi çalışma göstermektedir.

GİRİŞ		ÇIKIŞ
A	B	X
0	0	0
0	1	1
1	0	1
1	1	0

Tablo 1. Devrenin doğruluk tablosu

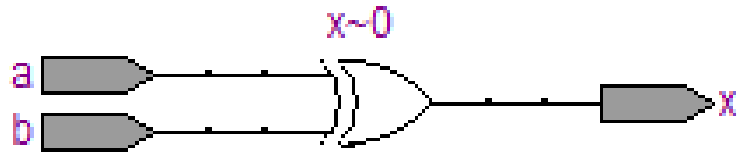
Tablo 1.'de verilen doğruluk tablosu aşağıdaki gibi kodlanmıştır.

```
LIBRARY ieee;
USE ieee.std_logic_1164.all;

ENTITY yada_kapisi IS
PORT (a:IN BIT;
      b:IN BIT;
      x:OUT BIT);
END yada_kapisi;

ARCHITECTURE mimari1 OF yada_kapisi IS
BEGIN
x<=a XOR b;
END mimari1;
```

Deney Sonuçları: Tasarlanan devrenin RTL diyagramı ve benzetim sonuçları aşağıdaki gibi elde edilmiştir.



Şekil 1. Devrenin RTL diyagramı

Messages		
/yada_kapisi/a	1	
/yada_kapisi/b	1	
/yada_kapisi/x	0	

Şekil 2. Devrenin benzetim sonuçları

DENEY NO: 04

DENEY ADI: Ve-Değil Kapısı'nın VHDL ile Yapısal Tasarımı

Deneyin Amacı: A ve b giriş, x çıkış bitlerinin kullanılmasıyla NAND Kapısının davranışının VHDL ile yapısal olarak tasarımının yapılması.

Deneyin Yapılışı: Devre aşağıdaki gibi çalışma göstermektedir.

GİRİŞ		ÇIKIŞ
A	B	X
0	0	1
0	1	1
1	0	1
1	1	0

Tablo 1. Devrenin doğruluk tablosu

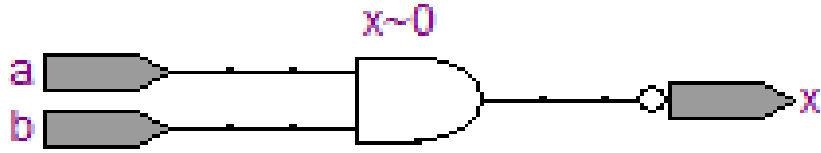
Tablo 1.'de verilen doğruluk tablosu aşağıdaki gibi kodlanmıştır.

```
LIBRARY ieee;
USE ieee.std_logic_1164.all;

ENTITY vedegil_kapisi IS
PORT (a:IN BIT;
      b:IN BIT;
      x:OUT BIT);
END vedegil_kapisi;

ARCHITECTURE mimari1 OF vedegil_kapisi IS
BEGIN
x<=a NAND b;
END mimari1;
```

Deney Sonuçları: Tasarlanan devrenin RTL diyagramı ve benzetim sonuçları aşağıdaki gibi elde edilmiştir.



Şekil 1. Devrenin RTL diyagramı

Messages									
◆ /vedegil_kapisi/a	1								
◆ /vedegil_kapisi/b	1								
◆ /vedegil_kapisi/x	0								

Şekil 2. Devrenin benzetim sonuçları

DENEY NO: 05

DENEY ADI: Veya-Değil Kapısı'nın VHDL ile Yapısal Tasarımı

Deneyin Amacı: A ve b giriş, x çıkış bitlerinin kullanılmasıyla NOR Kapısının davranışının VHDL ile yapısal olarak tasarımının yapılması.

Deneyin Yapılışı: Devre aşağıdaki gibi çalışma göstermektedir.

GİRİŞ		ÇIKIŞ
A	B	X
0	0	1
0	1	0
1	0	0
1	1	0

Tablo 1. Devrenin doğruluk tablosu

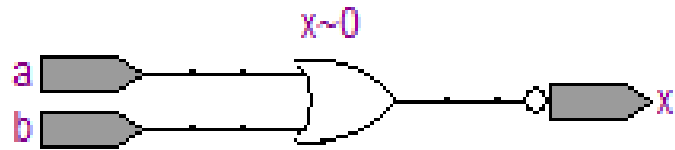
Tablo 1.'de verilen doğruluk tablosu aşağıdaki gibi kodlanmıştır.

```
LIBRARY ieee;
USE ieee.std_logic_1164.all;




ENTITY veyadegil_kapisi IS
PORT (a:IN BIT;
      b:IN BIT;
      x:OUT BIT);
END veyadegil_kapisi;

ARCHITECTURE mimari1 OF veyadegil_kapisi IS
BEGIN
x<=a NOR b;
END mimari1;
```

Deney Sonuçları: Tasarlanan devrenin RTL diyagramı ve benzetim sonuçları aşağıdaki gibi elde edilmiştir.



Şekil 1. Devrenin RTL diyagramı

Messages									
 /veyadegil_kapisi/a	1								
 /veyadegil_kapisi/b	1								
 /veyadegil_kapisi/x	0								

Şekil 2. Devrenin benzetim sonuçları

DENEY NO: 06

DENEY ADI: Ya da-Değil Kapısı'nın VHDL ile Yapısal Tasarımı

Deneyin Amacı: A ve b giriş, x çıkış bitlerinin kullanılmasıyla XNOR Kapısının davranışının VHDL ile yapısal olarak tasarımının yapılması.

Deneyin Yapılışı: Devre aşağıdaki gibi çalışma göstermektedir.

GİRİŞ		ÇIKIŞ
A	B	X
0	0	1
0	1	0
1	0	0
1	1	1

Tablo 1. Devrenin doğruluk tablosu

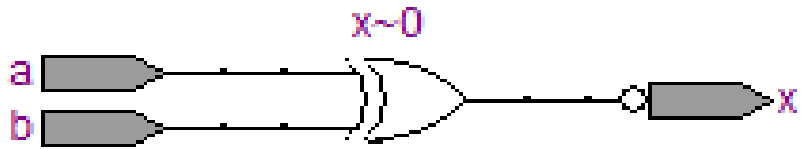
Tablo 1.'de verilen doğruluk tablosu aşağıdaki gibi kodlanmıştır.

```
LIBRARY ieee;
USE ieee.std_logic_1164.all;




ENTITY yadadegil_kapisi IS
PORT (a:IN BIT;
      b:IN BIT;
      x:OUT BIT);
END yadadegil_kapisi;

ARCHITECTURE mimari1 OF yadadegil_kapisi IS
BEGIN
x<=a XNOR b;
END mimari1;
```

Deney Sonuçları: Tasarlanan devrenin RTL diyagramı ve benzetim sonuçları aşağıdaki gibi elde edilmiştir.



Şekil 1. Devrenin RTL diyagramı

Messages									
 /yadadegil_kapisi/a	1								
 /yadadegil_kapisi/b	1								
 /yadadegil_kapisi/x	1								

Şekil 2. Devrenin benzetim sonuçları

DENEY NO: 07

DENEY ADI: Değil Kapısı'nın VHDL ile Yapısal Tasarımı

Deneyin Amacı: A giriş ve x çıkış bitinin kullanılmasıyla NOT Kapısının davranışının VHDL ile yapısal olarak tasarımının yapılması.

Deneyin Yapılışı: Devre aşağıdaki gibi çalışma göstermektedir.

GİRİŞ	ÇIKIŞ
A	X
0	1
1	0

Tablo 1. Devrenin doğruluk tablosu

Tablo 1.'de verilen doğruluk tablosu aşağıdaki gibi kodlanmıştır.

```
LIBRARY ieee;
USE ieee.std_logic_1164.all;

ENTITY degil_kapisi IS
    Port ( a: IN  STD_LOGIC;
          x: OUT  STD_LOGIC);
END degil_kapisi;

ARCHITECTURE mimari1 OF degil_kapisi IS
BEGIN
    x <= NOT a;
END mimari1;
```

Deney Sonuçları: Tasarlanan devrenin RTL diyagramı ve benzetim sonuçları aşağıdaki gibi elde edilmiştir.



Şekil 1. Devrenin RTL diyagramı

Messages				
/degil_kapisi/a	1			
/degil_kapisi/x	0			

Şekil 2. Devrenin benzetim sonuçları

DENEY NO: 08

DENEY ADI: Buffer Kapısı'nın VHDL ile Yapısal Tasarımı

Deneyin Amacı: A giriş ve x çıkış bitinin kullanılmasıyla Buffer(Tampon) Kapısının davranışının VHDL ile yapısal olarak tasarımının yapılması.

Deneyin Yapılışı: Devre aşağıdaki gibi çalışma göstermektedir.

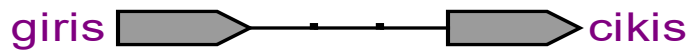
GİRİŞ	ÇIKIŞ
A	X
0	0
1	1

Tablo 1. Devrenin doğruluk tablosu

Tablo 1.'de verilen doğruluk tablosu aşağıdaki gibi kodlanmıştır.

```
LIBRARY ieee;  
USE ieee.std_logic_1164.all;  
  
ENTITY bufer_kapisi IS  
PORT (giris:IN BIT;  
      cikis:OUT BIT);  
END bufer_kapisi;  
  
ARCHITECTURE mimari1 OF bufer_kapisi IS  
BEGIN  
  cikis<=giris;  
END mimari1;
```

Deney Sonuçları: Tasarlanan devrenin RTL diyagramı ve benzetim sonuçları aşağıdaki gibi elde edilmiştir.



Şekil 1. Devrenin RTL diyagramı

Messages						
◆ /bufer_kapisi/giris	1					
◆ /bufer_kapisi/cikis	1					

Şekil 2. Devrenin benzetim sonuçları

DENEY NO: 09

DENEY ADI: Yarım Toplayıcının VHDL ile Yapısal Tasarımı

Deneyin Amacı: A ve b giriş, s ve c çıkış bitlerinin kullanılmasıyla yarım toplayıcının nasıl çalışma gösterdiğinin VHDL ile yapısal olarak tasarımının yapılması.

Deneyin Yapılışı: Devre aşağıdaki gibi çalışma göstermektedir.

GİRİŞ		ÇIKIŞ	
A	B	S	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Tablo 1. Devrenin doğruluk tablosu

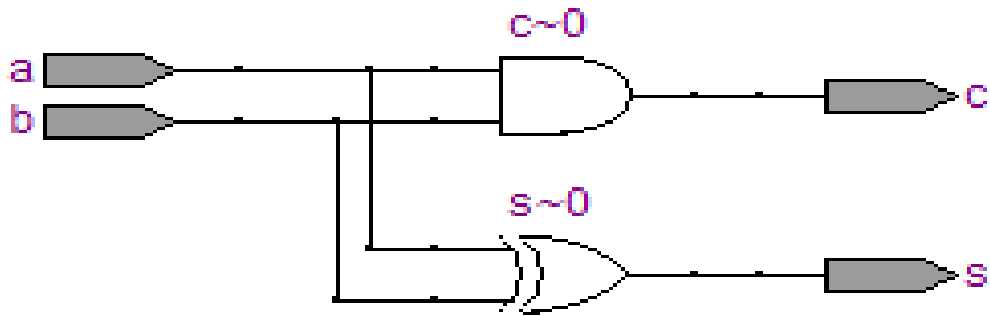
Tablo 1.'de verilen doğruluk tablosu aşağıdaki gibi kodlanmıştır.

```
LIBRARY ieee;
USE ieee.std_logic_1164.all;





ENTITY yarim_toplayici IS
PORT (a:IN BIT;
      b:IN BIT;
      c:OUT BIT;
      s:OUT BIT);
END yarim_toplayici;

ARCHITECTURE mimari1 OF yarim_toplayici IS
BEGIN
s<=a XOR b;
c<=a AND b;
END mimari1;
```

Deney Sonuçları: Tasarlanan devrenin RTL diyagramı ve benzetim sonuçları aşağıdaki gibi elde edilmiştir.



Şekil 1. Devrenin RTL diyagramı

Messages									
 /yarim_toplayici/a	1								
 /yarim_toplayici/b	1								
 /yarim_toplayici/s	0								
 /yarim_toplayici/c	1								

Şekil 2. Devrenin benzetim sonuçları

DENEY NO: 10**DENEY ADI: Yarım Toplayıcının VHDL ile Davranışsal Tasarımı**

Deneyin Amacı: A ve b giriş, s ve c çıkış bitlerinin kullanılmasıyla yarım toplayıcının nasıl çalışma gösterdiğinin VHDL ile davranışsal olarak tasarımının yapılması.

Deneyin Yapılışı: Devre aşağıdaki gibi çalışma göstermektedir.

GİRİŞ		ÇIKIŞ	
A	B	S	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Tablo 1. Devrenin doğruluk tablosu

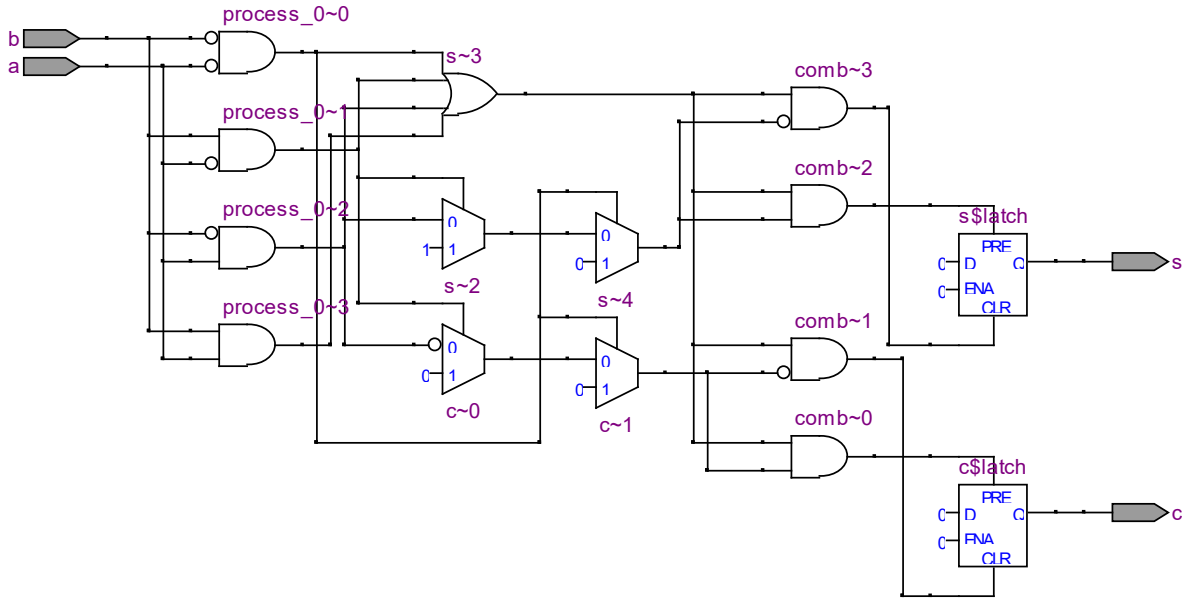
Tablo 1.'de verilen doğruluk tablosu aşağıdaki gibi kodlanmıştır.

```
LIBRARY ieee;
USE ieee.std_logic_1164.all;





ENTITY yarımToplayıcı_dav IS
PORT (a,b:IN STD_LOGIC;
      s,c:OUT STD_LOGIC);
END yarımToplayıcı_dav;

ARCHITECTURE mimari1 OF yarımToplayıcı_dav IS
BEGIN
  PROCESS (a,b)
  BEGIN
    if(a='0' and b='0') then
      s<='0';
      c<='0';
    elsif(a='0' and b='1') then
      s<='1';
      c<='0';
    elsif(a='1' and b='0') then
      s<='1';
      c<='0';
    elsif(a='1' and b='1') then
      s<='0';
      c<='1';
    end if;
  END PROCESS;
END mimari1;
```

Deney Sonuçları: Tasarlanan devrenin RTL diyagramı ve benzetim sonuçları aşağıdaki gibi elde edilmiştir.



Şekil 1. Devrenin RTL diyagramı

Messages									
 /yarimtoplayici_dav/a	1								
 /yarimtoplayici_dav/b	1								
 /yarimtoplayici_dav/s	0								
 /yarimtoplayici_dav/c	1								

Şekil 2. Devrenin benzetim sonuçları

DENEY NO: 11**DENEY ADI: Tam Toplayıcının VHDL ile Yapısal Tasarımı**

Deneyin Amacı: A, b ve EG(elde) giriş, toplam ve EC çıkış bitlerinin kullanılmasıyla tam toplayıcının nasıl çalışma gösterdiğinin VHDL ile yapısal olarak tasarımının yapılması.

Deneyin Yapılışı: Devre aşağıdaki gibi çalışma göstermektedir.

GİRİŞ			ÇIKIŞ	
A	B	EG	Toplam	EC
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Tablo 1. Devrenin doğruluk tablosu

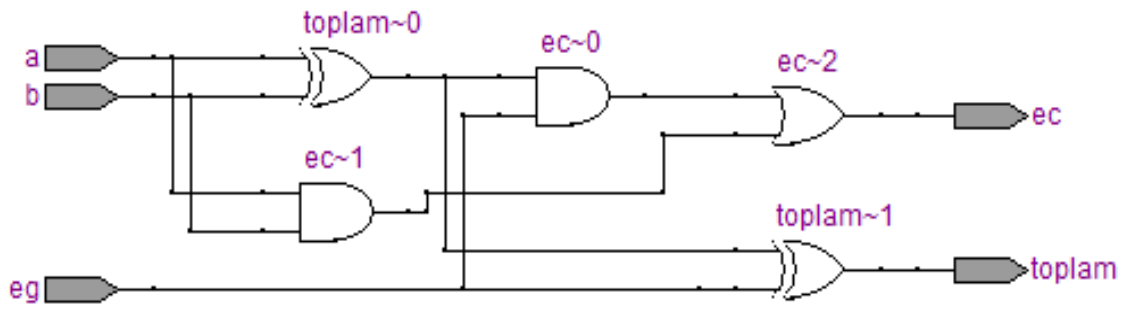
Tablo 1.'de verilen doğruluk tablosu aşağıdaki gibi kodlanmıştır.

```
LIBRARY ieee;
USE ieee.std_logic_1164.all;

ENTITY tam_toplayici IS
PORT (a,b,eg:IN STD_LOGIC;
      toplam,ec:OUT STD_LOGIC);
END tam_toplayici;

ARCHITECTURE mimari1 OF tam_toplayici IS
BEGIN
toplam<=(a xor b) xor eg;
ec<=(eg and (a xor b)) or (a and b);
END mimari1;
```

Deney Sonuçları: Tasarlanan devrenin RTL diyagramı ve benzetim sonuçları aşağıdaki gibi elde edilmiştir.



Şekil 1. Devrenin RTL diyagramı

Messages		
♦ /tam_toplayici/a	1	
♦ /tam_toplayici/b	1	
♦ /tam_toplayici/eg	1	
♦ /tam_toplayici/toplam	1	
♦ /tam_toplayici/ec	1	

Şekil 2. Devrenin benzetim sonuçları

DENEY NO: 12**DENEY ADI: Tam Toplayıcının VHDL ile Davranışsal Tasarımı**

Deneyin Amacı: A, b ve EG(elde) giriş, toplam ve EC çıkış bitlerinin kullanılmasıyla tam toplayıcının nasıl çalışma gösterdiğinin VHDL ile davranışsal olarak tasarımının yapılması.

Deneyin Yapılışı: Devre aşağıdaki gibi çalışma göstermektedir.

GİRİŞ			ÇIKIŞ	
A	B	EG	Toplam	EC
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Tablo 1. Devrenin doğruluk tablosu

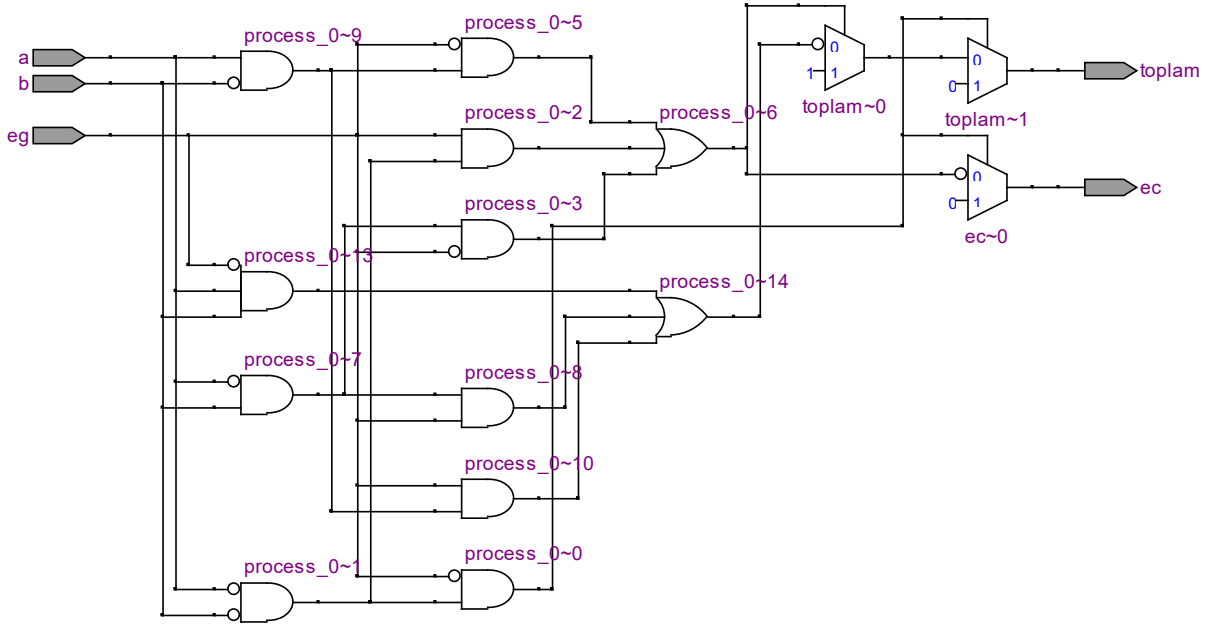
Tablo 1.'de verilen doğruluk tablosu aşağıdaki gibi kodlanmıştır.

```
LIBRARY ieee;
USE ieee.std_logic_1164.all;

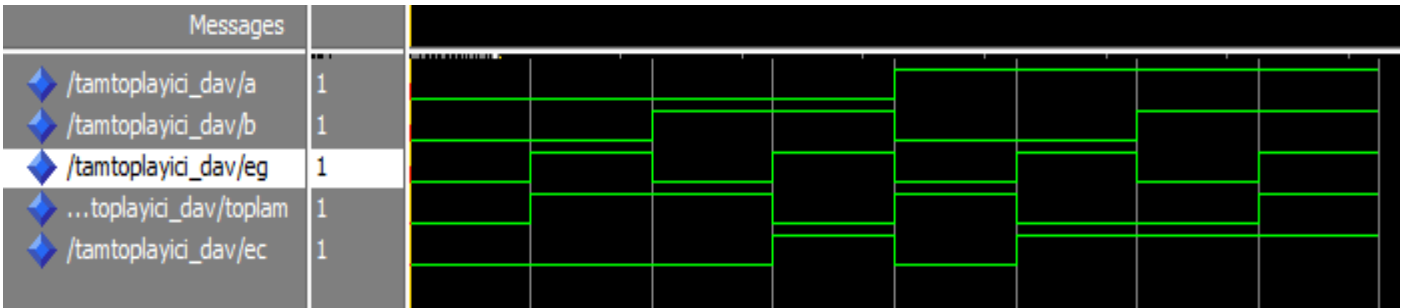
ENTITY tamToplayici_dav IS
PORT(a,b,eg:IN STD_LOGIC;
      toplam,ec:OUT STD_LOGIC);
END tamToplayici_dav;

ARCHITECTURE mimari1 OF tamToplayici_dav IS
BEGIN
  PROCESS(a,b,eg)
  BEGIN
    if (A='0' and B='0' and EG='0') then
      toplam <= '0';
      ec <= '0';
    elsif (A='0' and B='0' and EG='1') or
      (A='0' and B='1' and EG='0') or
      (A='1' and B='0' and EG='0') then
      toplam <= '1';
      ec <= '0';
    elsif (A='0' and B='1' and EG='1') or
      (A='1' and B='0' and EG='1') or
      (A='1' and B='1' and EG='0') then
      toplam <= '0';
      ec <= '1';
    else
      toplam <= '1';
      ec <= '1';
    END if;
  end PROCESS;
end mimari1;
```

Deney Sonuçları: Tasarlanan devrenin RTL diyagramı ve benzetim sonuçları aşağıdaki gibi elde edilmiştir.



Şekil 1. Devrenin RTL diyagramı



Şekil 2. Devrenin benzetim sonuçları

DENEY NO: 13

DENEY ADI: Yarım Çıkarıcının VHDL ile Yapısal Tasarımı

Deneyin Amacı: A ve b giriş, fark ve borç çıkış bitlerinin kullanılmasıyla yarım çıkarıcının nasıl çalışma gösterdiğinin VHDL ile yapısal olarak tasarımının yapılması.

Deneyin Yapılışı: Devre aşağıdaki gibi çalışma göstermektedir.

GİRİŞ		ÇIKIŞ	
A	B	Fark	Borç
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

Tablo 1. Devrenin doğruluk tablosu

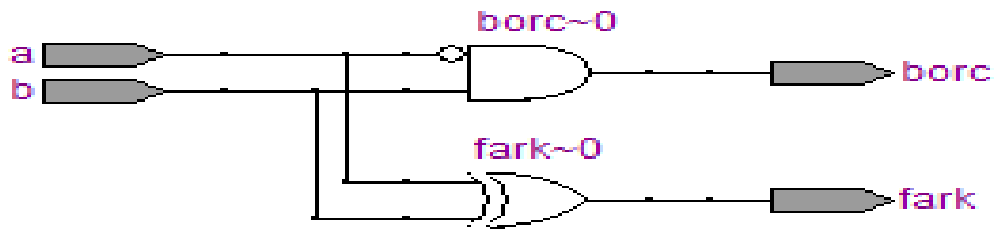
Tablo 1.'de verilen doğruluk tablosu aşağıdaki gibi kodlanmıştır.

```
LIBRARY ieee;
USE ieee.std_logic_1164.all;

ENTITY yari_cikarici IS
PORT (a,b:IN BIT;
      fark,borc:OUT BIT);
END yari_cikarici;

ARCHITECTURE mimari1 OF yari_cikarici IS
BEGIN
fark<=a XOR b;
borc<=(not a) and b;
END mimari1;
```

Deney Sonuçları: Tasarlanan devrenin RTL diyagramı ve benzetim sonuçları aşağıdaki gibi elde edilmiştir.



Şekil 1. Devrenin RTL diyagramı

Messages									
◆ /yari_cikarici/a	1								
◆ /yari_cikarici/b	1								
◆ /yari_cikarici/fark	0								
◆ /yari_cikarici/borc	0								

Şekil 2. Devrenin benzetim sonuçları

DENEY NO: 14**DENEY ADI: Yarım Çıkarıcının VHDL ile Davranışsal Tasarımı**

Deneyin Amacı: A ve b giriş, fark ve borç çıkış bitlerinin kullanılmasıyla yarım çıkarıcının nasıl çalışma gösterdiğinin VHDL ile davranışsal olarak tasarımının yapılması.

Deneyin Yapılışı: Devre aşağıdaki gibi çalışma göstermektedir.

GİRİŞ		ÇIKIŞ	
A	B	Fark	Borç
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

Tablo 1. Devrenin doğruluk tablosu

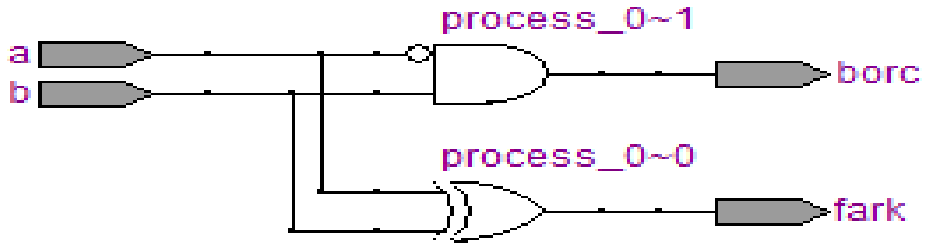
Tablo 1.' de verilen doğruluk tablosu aşağıdaki gibi kodlanmıştır.

```
LIBRARY ieee;
USE ieee.std_logic_1164.all;

ENTITY yarim_cikarici IS
PORT (a,b:IN BIT;
      fark,borc:OUT BIT);
END yarim_cikarici;

ARCHITECTURE mimari1 OF yarim_cikarici IS
BEGIN
  process(a,b)
  begin
    if(a/=b) then fark<='1';
    else fark<='0';
    end if;
    if(a='0') and (b='1') then borc<='1';
    else borc<='0';
    end if;
  end process;
end mimari1;
```

Deney Sonuçları: Tasarlanan devrenin RTL diyagramı ve benzetim sonuçları aşağıdaki gibi elde edilmiştir.



Şekil 1. Devrenin RTL diyagramı






Messages									
♦ /yarim_cikarici/a	1								
♦ /yarim_cikarici/b	1								
♦ /yarim_cikarici/fark	0								
♦ /yarim_cikarici/borc	0								

Şekil 2. Devrenin benzetim sonuçları

DENEY NO: 15**DENEY ADI: SR Tipi Yaz-Boz'un Davranışsal Olarak VHDL ile Tasarımı**

Deneyin Amacı: S, R girişlerinin ve saat vuruşlarının kullanılmasıyla, SR tipi yaz-boz' un çıkışının, bir sonraki saat vuruşunda nasıl çalışma gösterdiğinin VHDL ile davranışsal olarak tasarımının yapılması.

Deneyin Yapılışı: Devre aşağıdaki gibi çalışma göstermektedir.

GİRİŞ			SAAT	Bir Sonraki Saat vuruşu
S	R	Reset	Clk	Q
0	0	0		Q(t)
0	1	0		0 (Sıfır Yapma)
1	0	0		1 (Bir Yapma)
1	1	0		Tanımsız
X	X	1		0

Tablo 1. Devrenin karakteristik tablosu

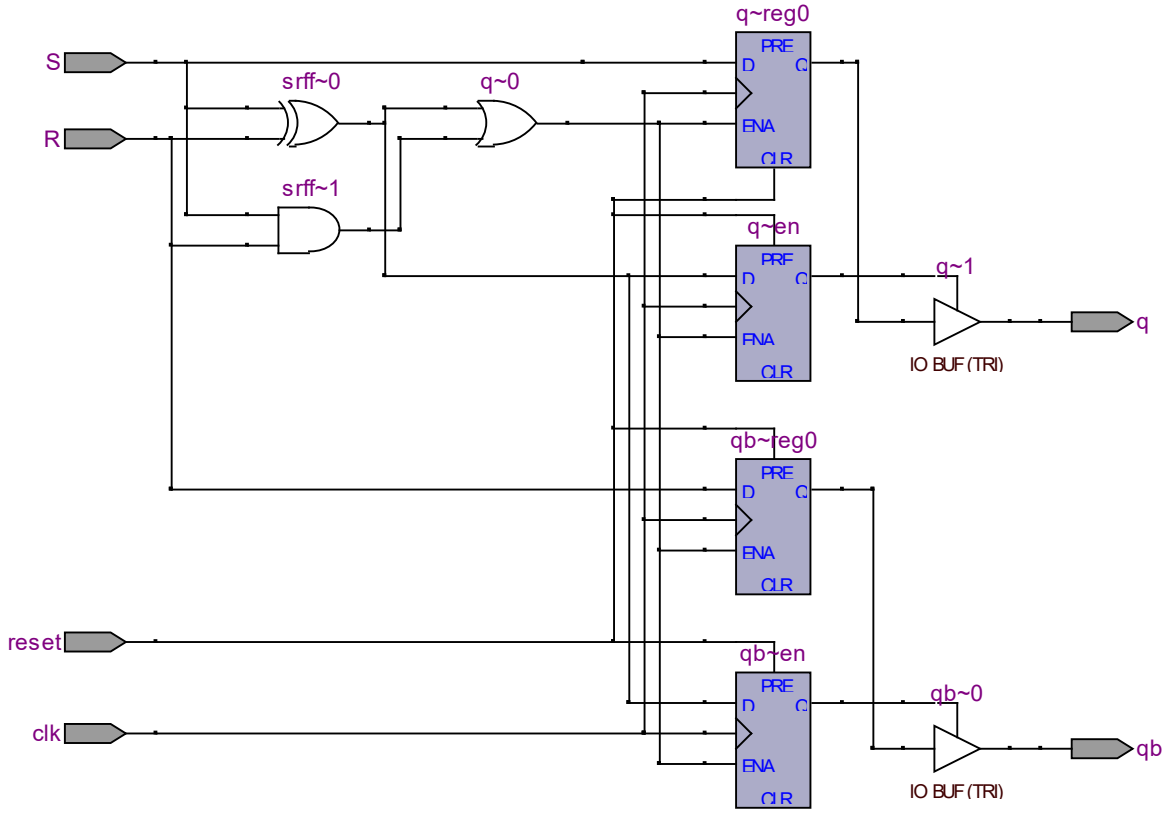
Tablo 1.'de verilen doğruluk tablosu aşağıdaki gibi kodlanmıştır.

```

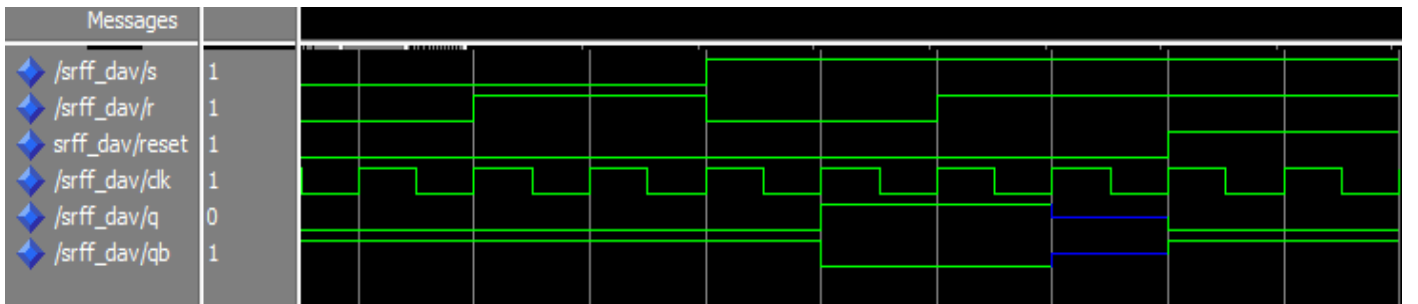
library IEEE;
use IEEE.STD_LOGIC_1164.all;
entity SRff_dav is
    port (S,R,clk: in STD_LOGIC;
          reset : in STD_LOGIC;
          q,qb: out STD_LOGIC);
end SRff_dav;
architecture mimari1 of SRff_dav is
begin
    SRff : process (s,r,clk,reset) is
    begin
        if (reset='1') then
            q <= '0';
            qb <= '1';
        elsif (rising_edge (clk)) then
            if (s/=r) then
                q <= s;
                qb <= r;
            elsif (s='1' and r='1') then
                q <= 'Z';
                qb <= 'Z';
            end if;
        end if;
    end process SRff;
end mimari1;

```

Deney Sonuçları: Tasarlanan devrenin RTL diyagramı ve benzetim sonuçları aşağıdaki gibi elde edilmiştir.



Şekil 1. Devrenin RTL diyagramı








Şekil 2. Devrenin benzetim sonuçları

DENEY NO: 16**DENEY ADI: JK Tipi Yaz-Boz' un Davranışsal Olarak VHDL ile Tasarımı**

Deneyin Amacı: J, K girişlerinin ve saat vuruşlarının kullanılmasıyla, JK tipi yaz-boz' un çıkışının, bir sonraki saat vuruşunda nasıl çalışma gösterdiğinin VHDL ile davranışsal olarak tasarımının yapılması.

Deneyin Yapılışı: Devre aşağıdaki gibi çalışma göstermektedir.

GİRİŞ			SAAT	Bir Sonraki Saat vuruşu
J	K	Reset	Clk	Q(t+1)
0	0	0		Q(t)
0	1	0		0 (Sıfır Yapma)
1	0	0		1 (Bir Yapma)
1	1	0		Q'(t)
X	X	1		0

Tablo 1. Devrenin karakteristik tablosu

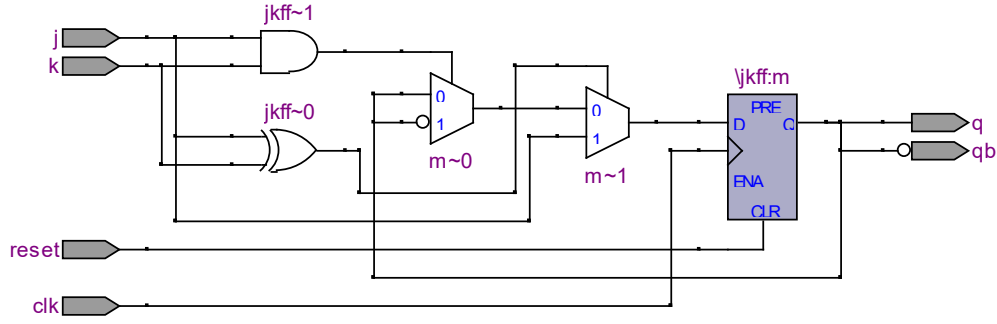
Tablo 1.'de verilen doğruluk tablosu aşağıdaki gibi kodlanmıştır.

```

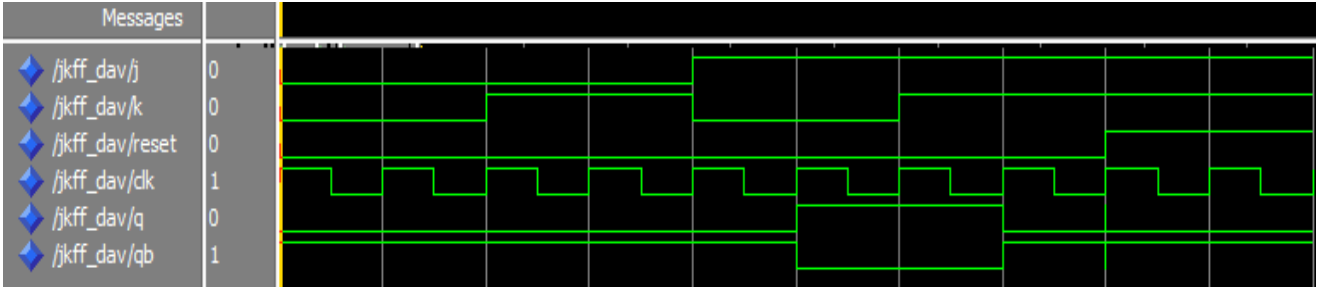
library IEEE;
use IEEE.STD_LOGIC_1164.all;
entity JKff_dav is
    port(j,k,clk : in STD_LOGIC;
          reset : in STD_LOGIC;
          q ,qb: out STD_LOGIC);
end JKff_dav;
architecture mimari1 of JKff_dav is
begin
    jkff : process (j,k,clk,reset) is
        variable m : std_logic := '0';
    begin
        if (reset='1') then
            m := '0';
        elsif (rising_edge (clk)) then
            if (j/=k) then
                m := j;
            elsif (j='1' and k='1') then
                m := not m;
            end if;
        end if;
        q <= m;
        qb <= not m;
    end process jkff;
end mimari1;

```

Deney Sonuçları: Tasarlanan devrenin RTL diyagramı ve benzetim sonuçları aşağıdaki gibi elde edilmiştir.



Şekil 1. Devrenin RTL diyagramı




Şekil 2. Devrenin benzetim sonuçları

DENEY NO: 17**DENEY ADI: D Tipi Yaz-Boz' un Davranışsal Olarak VHDL ile Tasarımı**

Deneyin Amacı: D girişinin ve saat vuruşlarının kullanılmasıyla, D tipi yaz-boz' un çıkışının, bir sonraki saat vuruşunda nasıl çalışma gösterdiğinin VHDL ile davranışsal olarak tasarımının yapılması.

Deneyin Yapılışı: Devre aşağıdaki gibi çalışma göstermektedir.

GİRİŞ	SAAT	Bir Sonraki Saat vuruşu
D	Saat	Q(t+1)
0		0
1		1

Tablo 1. Devrenin karakteristik tablosu

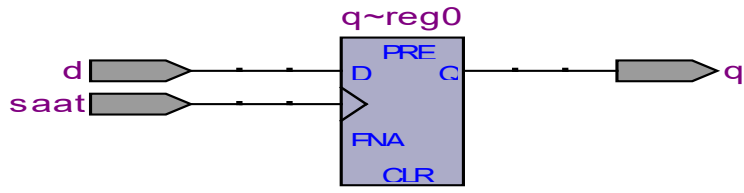
Tablo 1.'de verilen doğruluk tablosu aşağıdaki gibi kodlanmıştır.

```
library ieee;
use ieee.std_logic_1164.all;

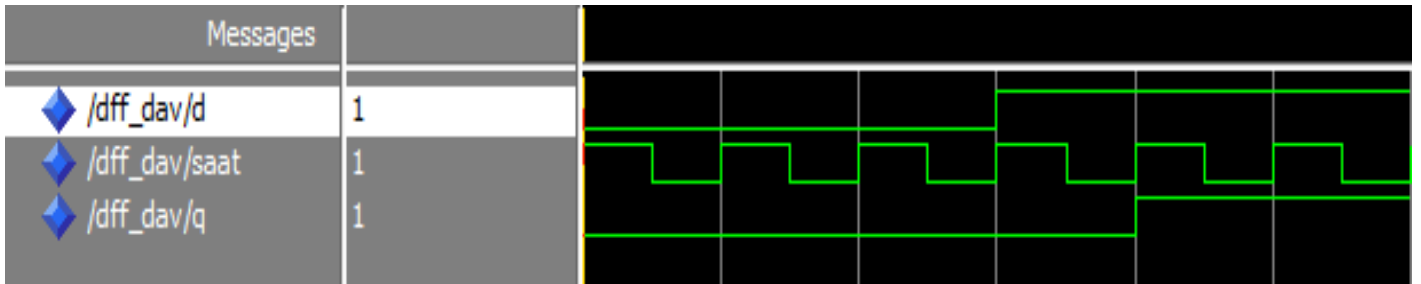
entity Dff_dav is
port (d,saat :in bit;
      q:out bit);
end Dff_dav;

architecture mimari1 of Dff_dav is
begin
  process (saat)
  begin
    if (saat='1' and saat'event) then q<=d;
    end if;
  end process;
end mimari1;
```

Deney Sonuçları: Tasarlanan devrenin RTL diyagramı ve benzetim sonuçları aşağıdaki gibi elde edilmiştir.



Şekil 1. Devrenin RTL diyagramı






Şekil 2. Devrenin benzetim sonuçları

DENEY NO: 18**DENEY ADI: T Tipi yaz-boz' un Davranışsal Olarak VHDL ile Tasarımı**

Deneyin Amacı: T girişinin ve saat vuruşlarının kullanılmasıyla T tipi yaz-boz' un çıkışının, bir sonraki saat vuruşunda nasıl çalışma gösterdiğinin VHDL ile davranışsal olarak tasarımının yapılması.

Deneyin Yapılışı: Devre aşağıdaki gibi çalışma göstermektedir.

GİRİŞ		SAAT	Bir Sonraki saat vuruşu
T	Reset	Clk	Q(t+1)
0	0		Q (t)
1	0		Q'(t)
X	1		0

Tablo 1. Devrenin karakteristik çizelgesi

Tablo 1.'de verilen doğruluk tablosu aşağıdaki gibi kodlanmıştır.

```
library ieee;
use ieee.std_logic_1164.all;

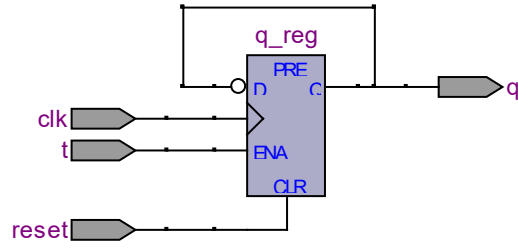
entity Tff_dav is
port (clk, reset, t: in std_logic;
      q: out std_logic);
end Tff_dav;

architecture mimari of Tff_dav is
    signal q_reg: std_logic;
    signal q_next: std_logic;
begin
    process (clk)
    begin
        if (reset = '1') then
            q_reg <= '0';
        elsif (clk'event and clk = '1') then
            q_reg <= q_next;
        end if;
    end process;

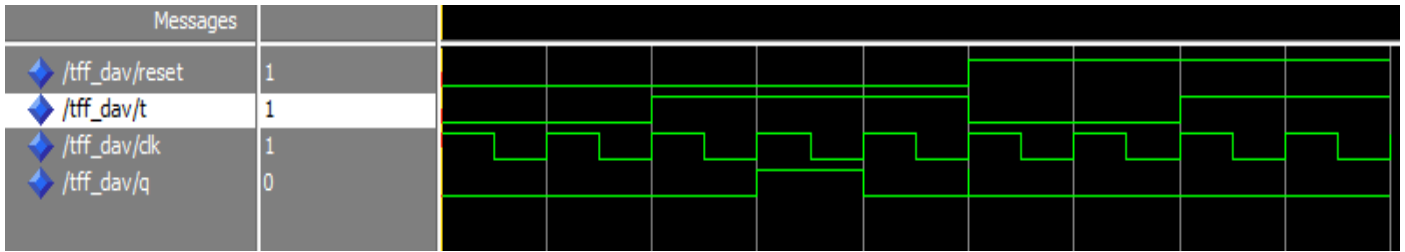
    q_next <= q_reg when t = '0' else
        not (q_reg);

    q <= q_reg;
end mimari;
```

Deney Sonuçları: Tasarlanan devrenin RTL diyagramı ve benzetim sonuçları aşağıdaki gibi elde edilmiştir.



Şekil 1. Devrenin RTL diyagramı



Şekil 2. Devrenin benzetim sonuçları

DENEY NO: 19**DENEY ADI: 8*1'lik Seçicinin Davranışsal Olarak VHDL ile Tasarımı**

Deneyin Amacı: S_2, S_1, S_0 seçim girişlerinin değerlerine göre I çıkışının 8*1'lik seçici ile belirlenmesinde nasıl çalışma gösterdiğinin davranışsal olarak tasarımın yapılması.

Deneyin Yapılışı: Devre aşağıdaki gibi çalışma göstermektedir.

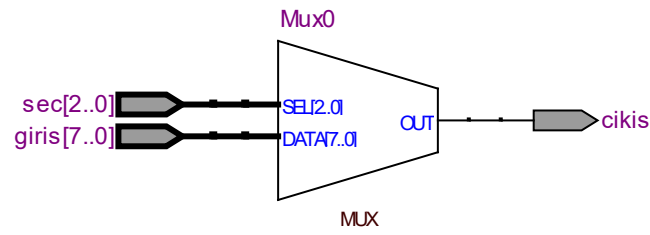
Secim Bitleri			GİRİŞ	ÇIKIŞ
S_2	S_1	S_0	$S_2S_1S_0$	I
0	0	0	Giriş(0)	Çıkış(0)
0	0	1	Giriş(1)	Çıkış(1)
0	1	0	Giriş(2)	Çıkış(2)
0	1	1	Giriş(3)	Çıkış(3)
1	0	0	Giriş(4)	Çıkış(4)
1	0	1	Giriş(5)	Çıkış(5)
1	1	0	Giriş(6)	Çıkış(6)
1	1	1	Giriş(7)	Çıkış(7)

Tablo 1. Devrenin doğruluk tablosu

Tablo 1.'de verilen doğruluk tablosu aşağıdaki gibi kodlanmıştır.

```
library ieee;
use ieee.std_logic_1164.all;
ENTITY mux_8x1 IS
PORT(sec : in std_logic_vector(2 downto 0);
      giris : in std_logic_vector(7 downto 0);
      cikis : out std_logic);
end mux_8x1;
architecture mimari1 of mux_8x1 IS
begin
  process(sec,giris)
  begin
    case sec is
      when "000" => cikis <= giris(0);
      when "001" => cikis <= giris(1);
      when "010" => cikis <= giris(2);
      when "011" => cikis <= giris(3);
      when "100" => cikis <= giris(4);
      when "101" => cikis <= giris(5);
      when "110" => cikis <= giris(6);
      when "111" => cikis <= giris(7);
    end case;
  end process;
end mimari1;
```

Deney Sonuçları: Tasarlanan devrenin RTL diyagramı ve benzetim sonuçları aşağıdaki gibi elde edilmiştir.



Şekil 1. Devrenin RTL diyagramı

Messages									
+ /mux_8x1/sec + /mux_8x1/giris + /mux_8x1/cikis	000	000	001	010	011	100	101	110	111
	01010101	01010101							
	1								

Şekil 2. Devrenin benzetim sonuçları

DENEY NO: 20

DENEY ADI: 3*8 Decoder' in Davranışsal Olarak VHDL ile Tasarımı

Deneyin Amacı: n bitlik ikili (binary) veriyi, 2^n bitlik onlu(decimal) veriye dönüştüren, sayısal eleman olan decoder' in(kod çözücü), giriş bitlerine göre çıkışın belirlenmesinin, VHDL ile davranışsal olarak tasarlanması.

Deneyin Yapılışı: Devre aşağıdaki gibi çalışma göstermektedir.

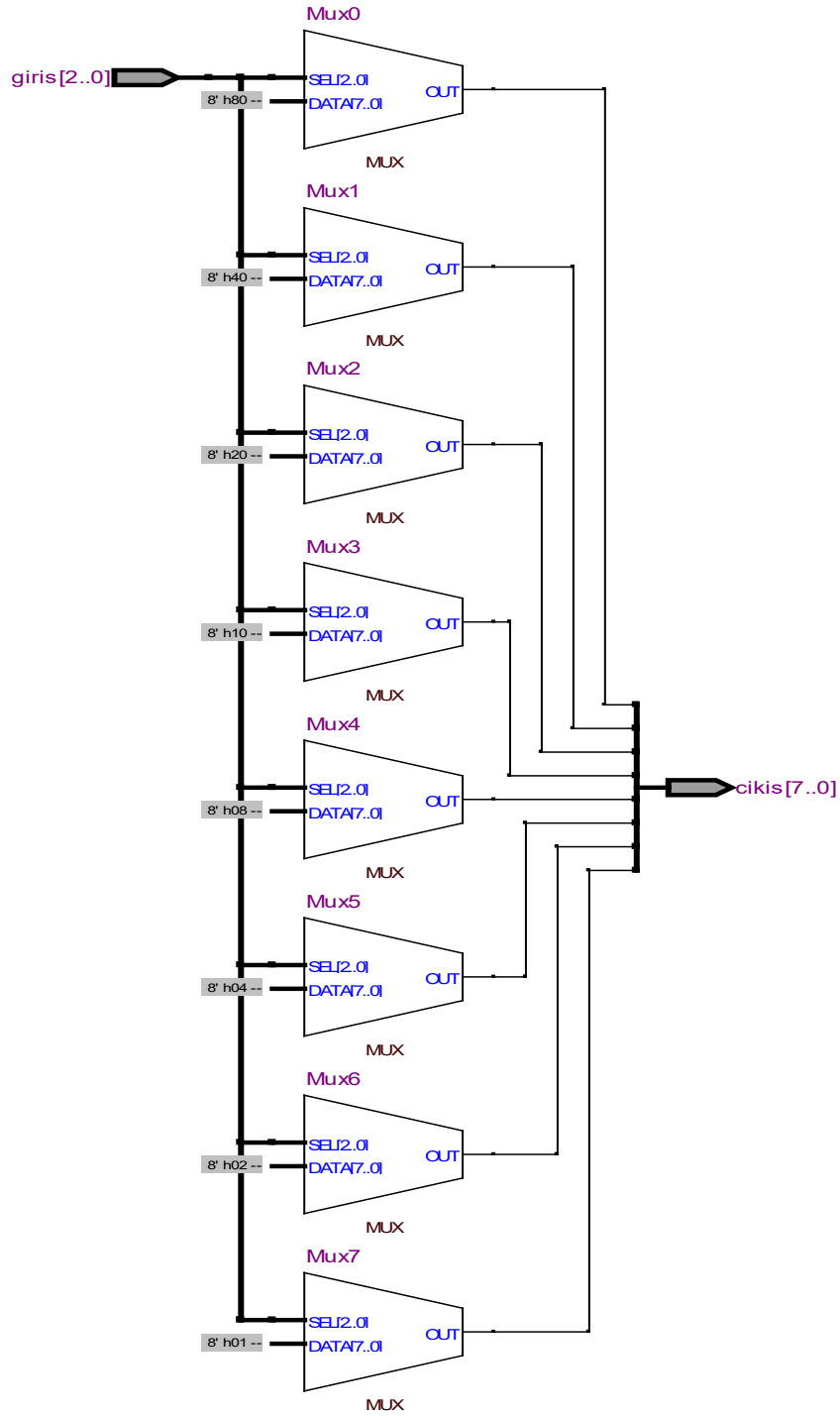
GİRİŞ BİTLERİ			ÇIKIŞ BİTLERİ							
Giriş(2)	Giriş(1)	Giriş(0)	Çıkış(0)	Çıkış(1)	Çıkış(2)	Çıkış(3)	Çıkış(4)	Çıkış(5)	Çıkış(6)	Çıkış(7)
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0	1	0	0
1	1	0	0	0	0	0	0	0	1	0
1	1	1	0	0	0	0	0	0	0	1

Tablo 1. Devrenin doğruluk çizelgesi

Tablo 1.'de verilen doğruluk tablosu aşağıdaki gibi kodlanmıştır.

```
LIBRARY ieee;
USE ieee.std_logic_1164.all;
ENTITY decoder_dav IS
PORT (giris: IN std_logic_vector(2 downto 0);
      cikis : OUT std_logic_vector(7 downto 0));
END decoder_dav;
ARCHITECTURE mimari1 OF decoder_dav IS
BEGIN
  PROCESS(giris)
  BEGIN
    case giris is
      when "000" => cikis <= "00000001";
      when "001" => cikis <= "00000010";
      when "010" => cikis <= "00000100";
      when "011" => cikis <= "00001000";
      when "100" => cikis <= "00010000";
      when "101" => cikis <= "00100000";
      when "110" => cikis <= "01000000";
      when "111" => cikis <= "10000000";
    end case;
  end process;
END mimari1;
```

Deney Sonuçları: Tasarlanan devrenin RTL diyagramı ve benzetim sonuçları aşağıdaki gibi elde edilmiştir.



Şekil 1. Devrenin RTL diyagramı

Messages									
<div> <div>/decoder_dav/giris</div> <div>(2)</div> <div>(1)</div> <div>(0)</div> </div>	000	000	001	010	011	100	101	110	111
	0								
	0								
	0								
<div> <div>/decoder_dav/cikis</div> <div>(7)</div> <div>(6)</div> <div>(5)</div> <div>(4)</div> <div>(3)</div> <div>(2)</div> <div>(1)</div> <div>(0)</div> </div>	00000001	00000001	00000010	00000100	00001000	00010000	00100000	01000000	10000000
	0								
	0								
	0								
	0								
	0								
	0								
	0								
	0								
	0								
	1								

Şekil 2. Devrenin benzetim sonuçları

DENEY NO: 21**DENEY ADI: 8*3 Encoder' in Davranışsal Olarak VHDL ile Tasarımı**

Deneyin Amacı: 2^n bitlik onlu(decimal) veriyi, n bitlik ikili(binary) veriye dönüştüren, sayısal eleman olan, encoder' in(kodlayıcı), giriş bitlerine göre çıkışın belirlenmesinin VHDL ile davranışsal olarak tasarlanması.

Deneyin Yapılışı: Devre aşağıdaki gibi çalışma göstermektedir.

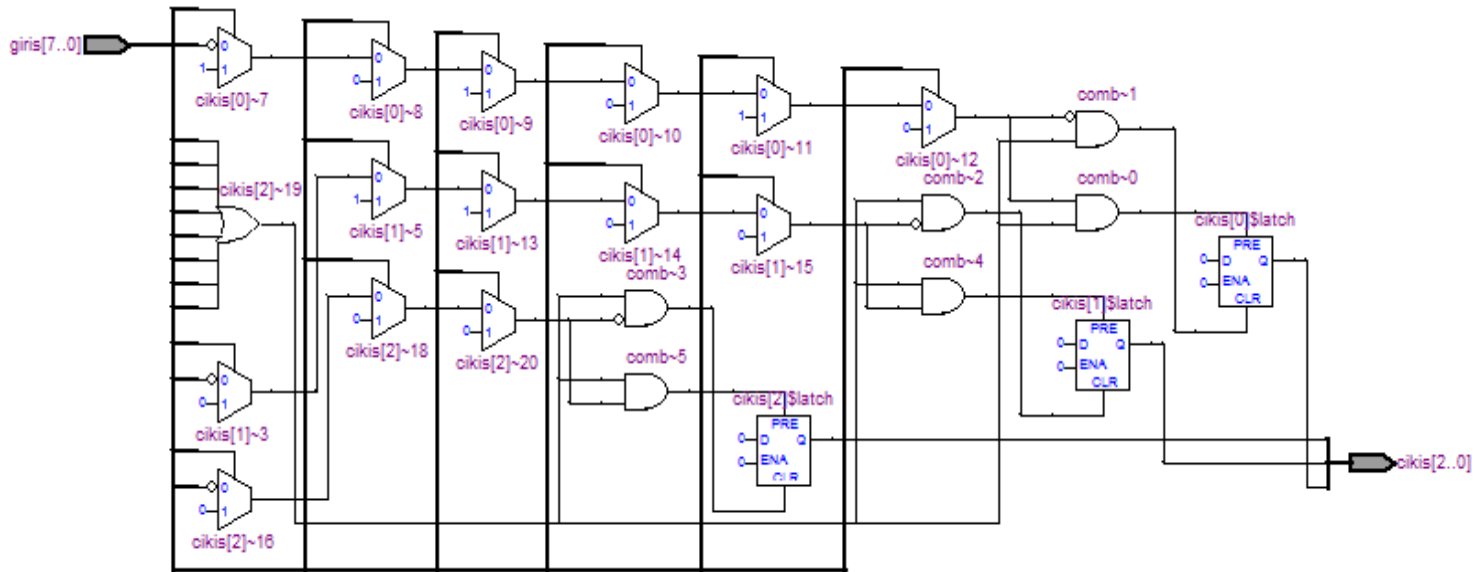
GİRİŞ BİTLERİ								ÇIKIŞ		
Giriş(7)	Giriş(6)	Giriş(5)	Giriş(4)	Giriş(3)	Giriş(2)	Giriş(1)	Giriş(0)	Çıkış(2)	Çıkış(1)	Çıkış(0)
0	0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	1	0	0	0	1
0	0	0	0	0	1	0	0	0	1	0
0	0	0	0	1	0	0	0	0	1	1
0	0	0	1	0	0	0	0	1	0	0
0	0	1	0	0	0	0	0	1	0	1
0	1	0	0	0	0	0	0	0	1	0
1	0	0	0	0	0	0	0	0	1	1

Tablo 1. Devrenin doğruluk çizelgesi

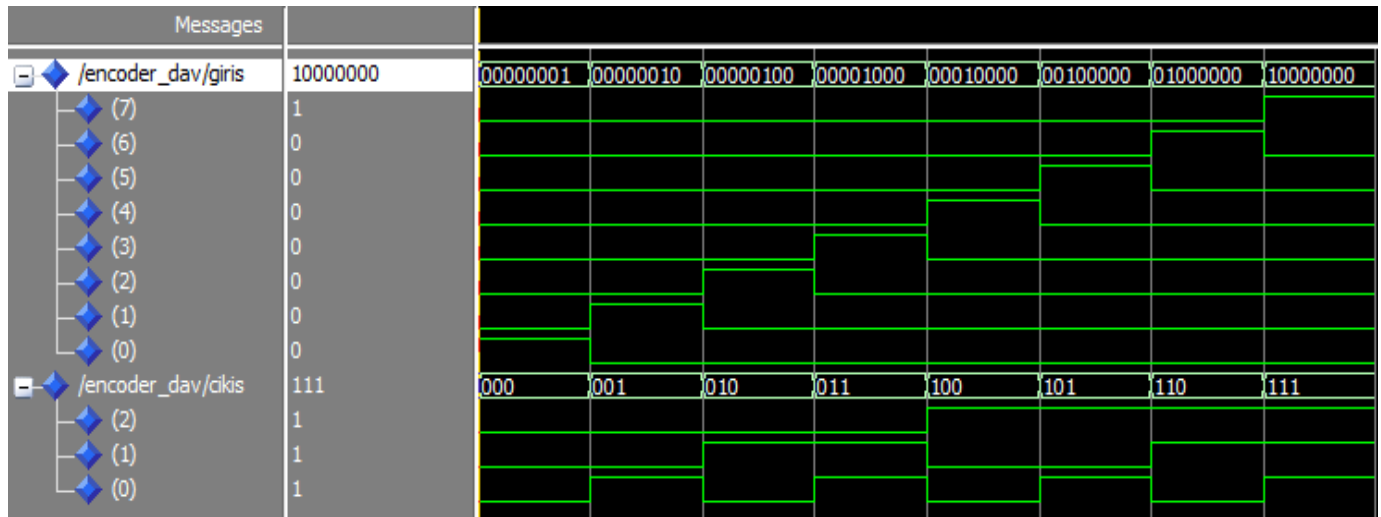
Tablo 1.'de verilen doğruluk tablosu aşağıdaki gibi kodlanmıştır.

```
LIBRARY ieee;
USE ieee.std_logic_1164.all;
ENTITY encoder_dav IS
PORT (giris: IN std_logic_vector(7 downto 0);
      cikis: OUT std_logic_vector(2 downto 0));
end encoder_dav;
ARCHITECTURE mimari1 OF encoder_dav IS
BEGIN
  PROCESS(giris)
  BEGIN
    if(giris(0)='1') then cikis<="000";
    elsif(giris(1)='1') then cikis<="001";
    elsif(giris(2)='1') then cikis<="010";
    elsif(giris(3)='1') then cikis<="011";
    elsif(giris(4)='1') then cikis<="100";
    elsif(giris(5)='1') then cikis<="101";
    elsif(giris(6)='1') then cikis<="110";
    elsif(giris(7)='1') then cikis<="111";
    end if;
  end PROCESS;
end mimari1;
```


Deney Sonuçları: Tasarlanan devrenin RTL diyagramı ve benzetim sonuçları aşağıdaki gibi elde edilmiştir.



Şekil 1. Devrenin RTL diyagramı



Şekil 2. Devrenin benzetim sonuçları

DENEY NO: 22**DENEY ADI: Yapısal ve Davranışsal Tasarımın Bir Devrede Kullanılarak VHDL ile Tasarımı**

Deneyin Amacı: A ve b girişlerinin ve dâhili sinyalin kullanılmasıyla, yapısal olarak, AND kapısı ve davranışsal olarak d yaz-boz' u kullanılarak, yapısal ve davranışsal tasarımın birlikte kullanılmasının VHDL ile tasarımının yapılması.

Deneyin Yapılışı: Devre aşağıdaki gibi çalışma göstermektedir.

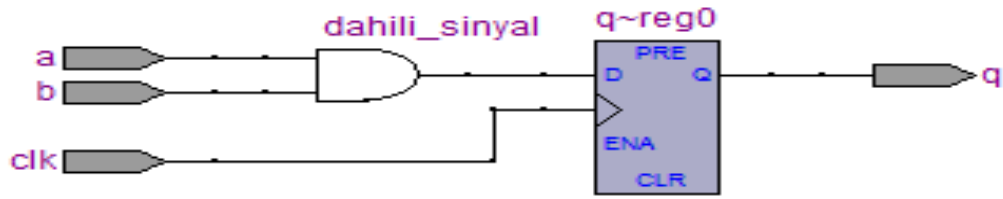
GİRİŞ			ÇIKIŞ
A	B	Dâhili sinyal	Q
0	0	0	0
0	1	0	0
1	0	0	0
1	1	1	1

Tablo 1. Devrenin doğruluk çizelgesi

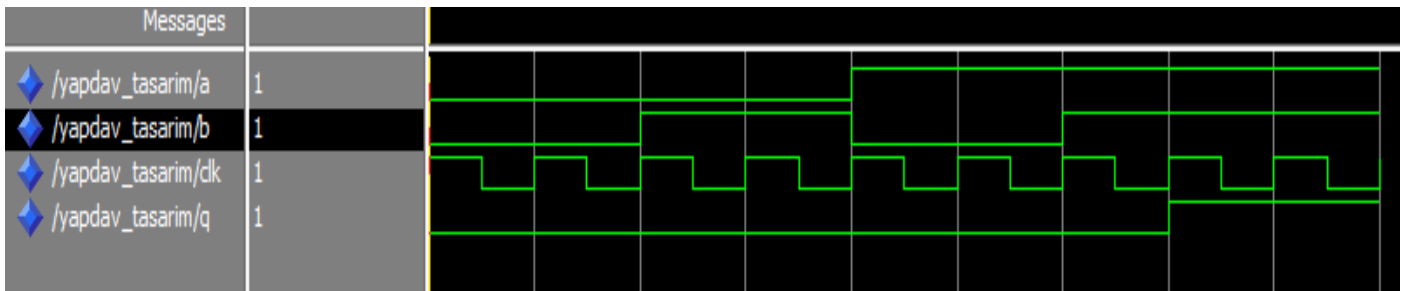
Tablo 1.'de verilen doğruluk tablosu aşağıdaki gibi kodlanmıştır.

```
LIBRARY ieee;
USE ieee.std_logic_1164.all;
ENTITY yapdav_tasarim IS
PORT (a,b,clk :IN BIT;
      q: OUT BIT);
END yapdav_tasarim;
ARCHITECTURE mimari1 OF yapdav_tasarim IS
-- sinyal tanımlamaları
signal dahili_sinyal :bit;
begin
--yapısal tasarım
dahili_sinyal <=a and b;
process (clk)
begin
--davranışsal tasarım
if (clk='1' and clk 'event) then q<=dahili_sinyal;
end if;
end process;
END mimari1;
```

Deney Sonuçları: Tasarlanan devrenin RTL diyagramı ve benzetim sonuçları aşağıdaki gibi elde edilmiştir.



Şekil 1. Devrenin RTL diyagramı



Şekil 2. Devrenin benzetim sonuçları

DENEY NO: 23**DENEY ADI: Basit bir RAM' in Davranışsal Olarak VHDL ile Tasarımı**

Deneyin Amacı: n ve k sabit değişkenlerini kullanarak, 2^k kelimelik n bitlik veri giriş ve çıkışına sahip, saat vuruşlarını kullanarak ve 'oku' veya 'yaz' girişlerinin aktifliklerine göre, k bitlik adres seçim yolunun gösterdiği bölgeden veya bölgeye, okuma veya yazma yapabilen basit bir RAM' in VHDL ile davranışsal tasarımı.

Deneyin Yapılışı: Devre aşağıdaki gibi çalışma göstermektedir.



Tablo 1. RAM Blok Şeması

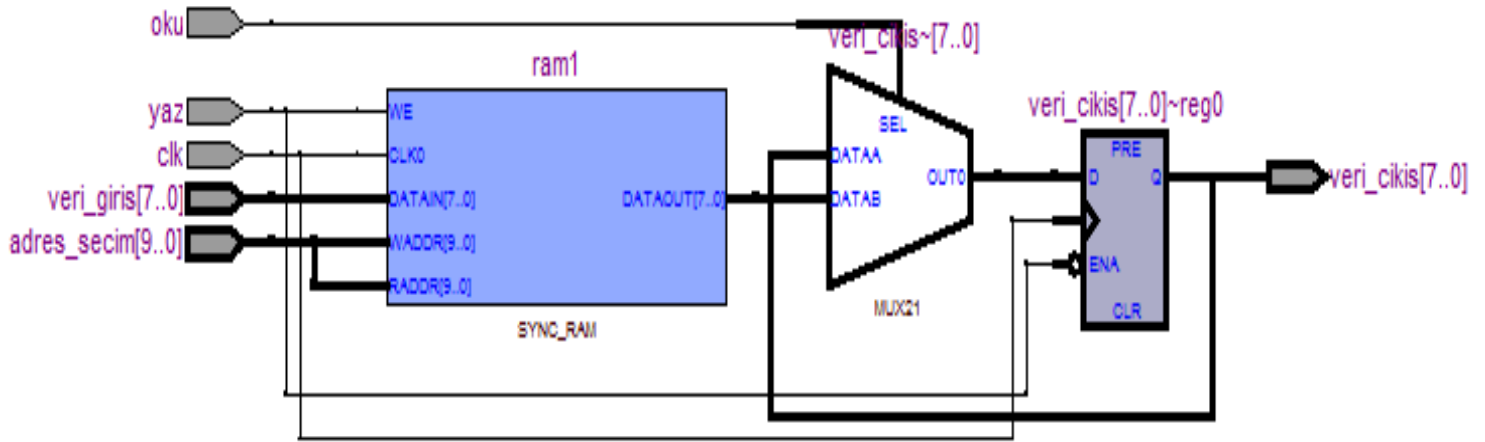
Tablo 1' de verilen 2^k kelimelik n bitlik RAM'ın VHDL ile kodlanması aşağıdaki gibi yapılmıştır.

```
LIBRARY ieee;
USE ieee.std_logic_1164.all;
USE ieee.std_logic_unsigned.all;

ENTITY ram IS
    GENERIC(n: integer:=8;
           k: integer:=10);
    PORT(clk,oku,yaz: IN STD_LOGIC;
         veri_giris: IN STD_LOGIC_VECTOR(n-1 downto 0);
         adres_secim: IN STD_LOGIC_VECTOR(k-1 downto 0);
         veri_cikis: OUT STD_LOGIC_VECTOR(n-1 downto 0));
END ram;

ARCHITECTURE mimari OF ram IS
    type ram_tipi is array (0 to 2**k) of std_logic_vector(n-1 downto 0);
    signal ram1:ram_tipi;
BEGIN
    PROCESS(clk)
    BEGIN
        if(clk='1' and clk'EVENT) then
            if(yaz='1') then
                ram1(conv_integer(adres_secim))<=veri_giris;
            elsif(oku='1') then
                veri_cikis<=ram1(conv_integer(adres_secim));
            end if;
        end if;
    END PROCESS;
END mimari;
```

Deney Sonuçları: Tasarlanan devrenin RTL diyagramı ve benzetim sonuçları aşağıdaki gibi elde edilmiştir.



Şekil 1. Devrenin RTL diyagramı

Messages		
♦ /ram/clock	1	
♦ /ram/oku	1	
♦ /ram/yaz	0	
+ ♦ /ram/veri_giris	01010101	00000000 10101010 01010101
+ ♦ /ram/adres_secim	0000000010	00000000 00000000 00000000 00000000 00000000 10
+ ♦ /ram/veri_cikis	01010101	00000000 10101010 01010101

Şekil 2. Devrenin benzetim sonuçları