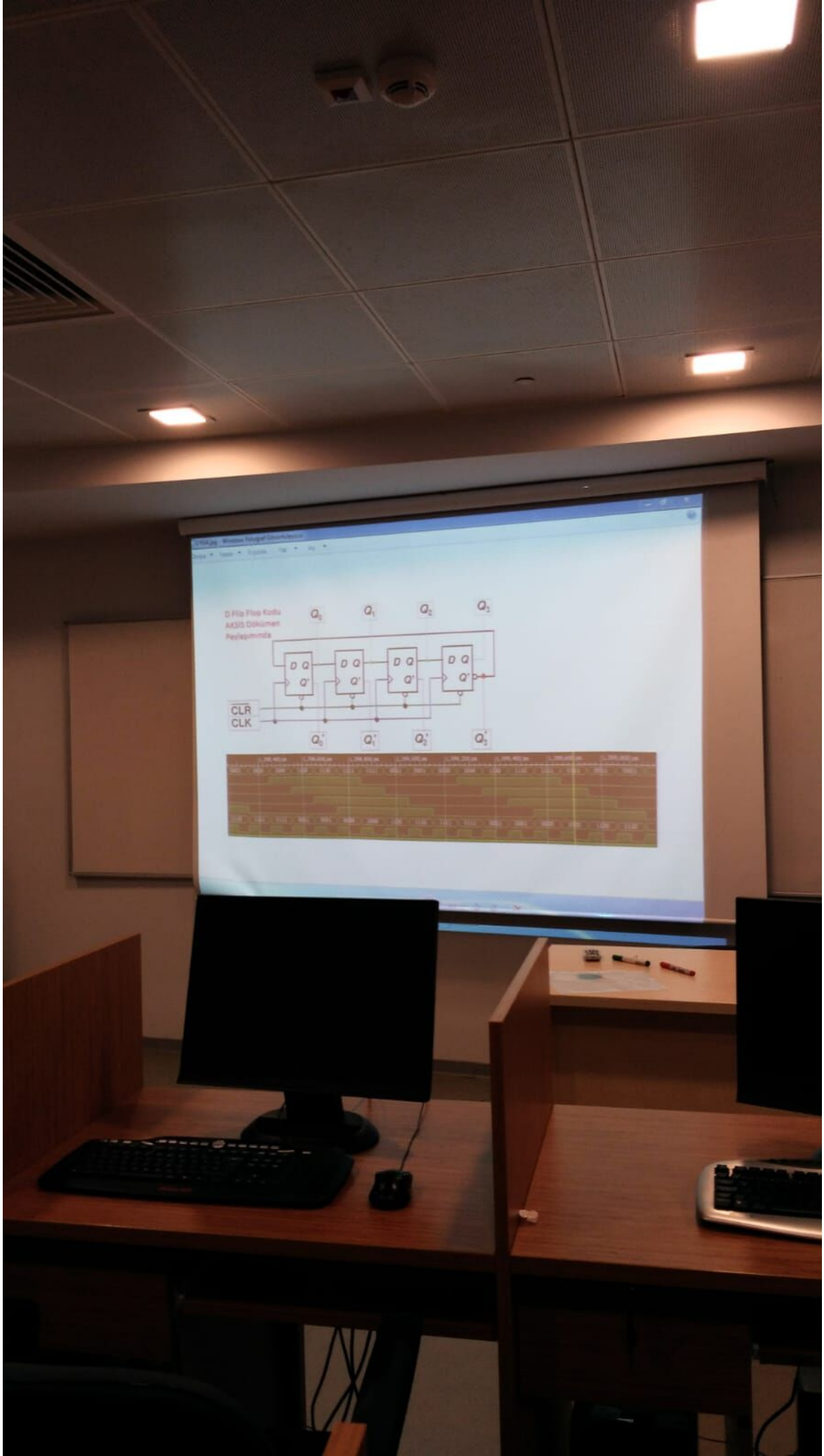
D Flip Flop Kodu
AKSİS Döküman
Paylaşımında
CLR
CLK

```
----------------------------------------------------------------------------
---------
-- Company:
-- Engineer: Yunus Emre Ak 1306150001 , Okan Uslu {no}
--
-- Create Date:    12:46:16 12/06/2018
-- Design Name:
-- Module Name:    Devre - Behavioral
-- Project Name:   5. Uygulama - D
-- Target Devices:
-- Tool versions:
-- Description: 5. Uygulamada çizilmesi gereken devrenin vhdl kodları
--
-- Dependencies:
--
-- Revision:
-- Revision 0.01 - File Created
-- Additional Comments:
--
----------------------------------------------------------------------------
---------
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity Circuit is
    port( -- Setting input and output ports
        CLR, CLK, D: in STD_LOGIC;
        Q0, Q1, Q2, Q3: inout STD_LOGIC;
    );
end Circuit;

-- Defining the behavioral of the architecture which is Circuit Flip-Flop
architecture Behavioral of Circuit is
    -- Adding D-FlipFlop component
    component DFF is
    port(
        Q, nQ: inout std_logic;
        CLK, CLR, D: in  std_logic
     );
    end component;

    -- Adding necessary signals
    signal o0: STD_LOGIC; -- DFF0 q output (the left)
    signal o1: STD_LOGIC; -- DFF1 q output
    signal o2: STD_LOGIC; -- DFF2 q output
    signal o3: STD_LOGIC; -- DFF3 q output

    begin -- Behavioral
        p0: DFF port map(o0, Q0, CLK, CLR, Q3);
        p1: DFF port map(o1, Q1, CLK, CLR, o0);
        p2: DFF port map(o2, Q2, CLK, CLR, o1);
        p3: DFF port map(o3, Q3, CLK, CLR, o2);
end Behavioral;
----------------------------------------------------------------------------
---------
Library IEEE;
USE IEEE.Std_logic_1164.all;
entity DFF is
   port(
      Q, nQ: inout std_logic;
```

```
      CLK, CLR, D: in  std_logic
   );
end DFF;
architecture Behavioral of DFF is
begin
 process(CLK)
 begin
    if(rising_edge(CLK)) then
            if(CLR='0') then
                  Q <= '0';
                  nQ <= '1';
            elsif (CLR='1') then
                  Q <= D;
                  nQ <= not(D);
            end if;
    end if;
 end process;
end Behavioral;
```