**İSTANBUL ÜNİVERSİTESİ**

**BİLGİSAYAR MÜHENDİSLİĞİ**

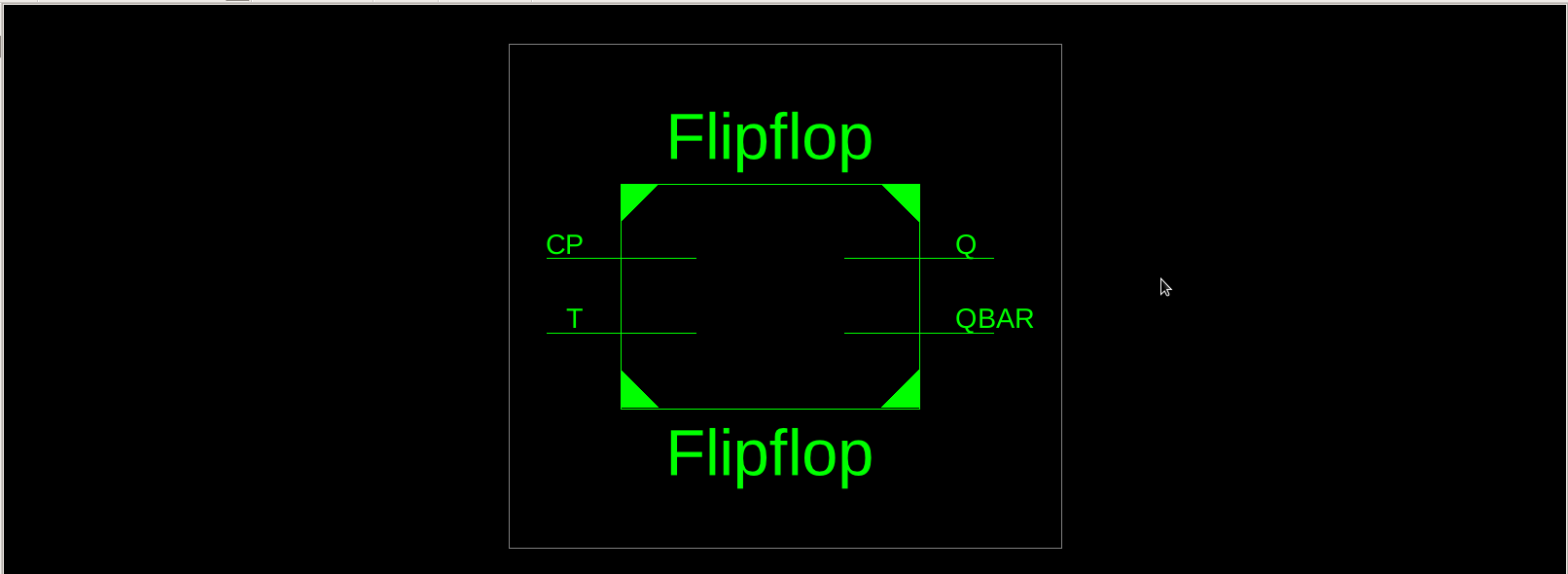
**Bilgisayar Organizasyonu ve**

**Tasarımı Laboratuvarı**

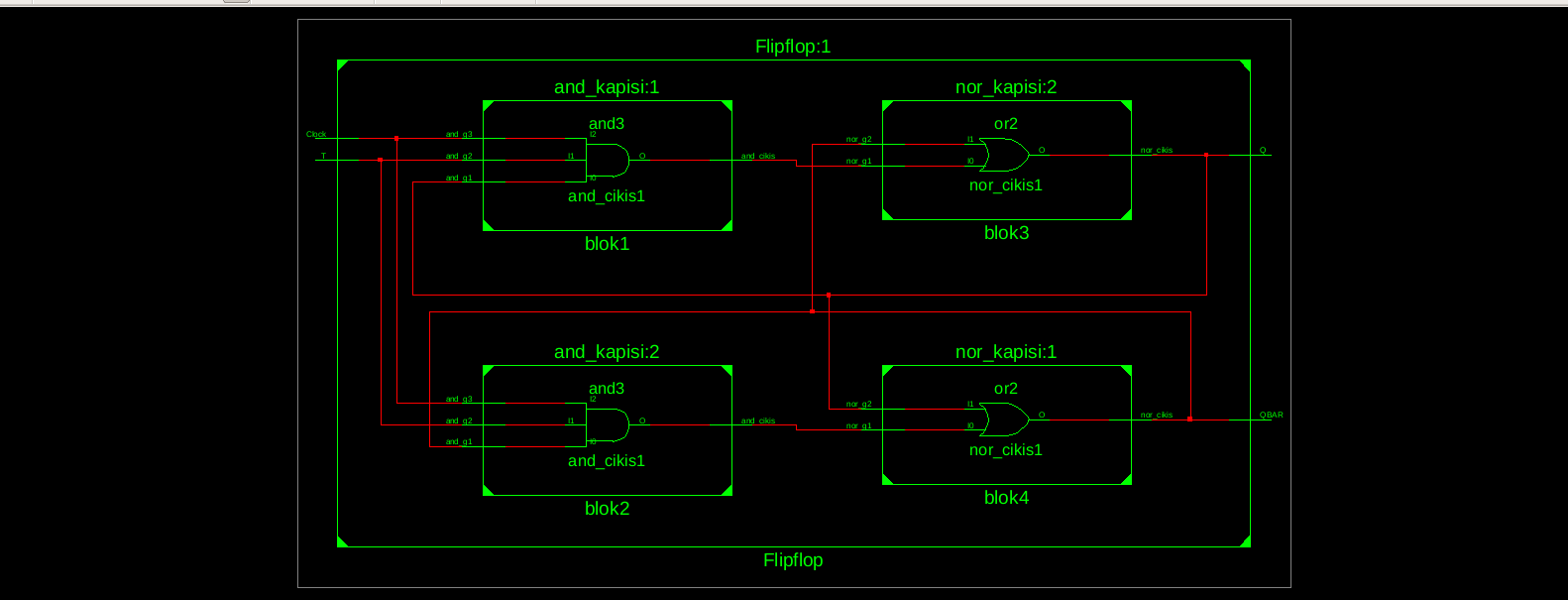
**Uygulama 3 :**

T Flipflop and Asynchronous Counter Using T flipflop

1.T Flipflop

****

1.1-Detaylı şema

****

Kod

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity nor\_kapisi is

port( nor\_g1: in STD\_LOGIC;

nor\_g2: in STD\_LOGIC;

nor\_cikis: out STD\_LOGIC);

end nor\_kapisi;

architecture Behavioral of nor\_kapisi is

begin

process(nor\_g1,nor\_g2)

begin

nor\_cikis <= nor\_g1 or nor\_g2;

end process;

end Behavioral;

----------------------------------------------------------------------------------

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity and\_kapisi is

port( and\_g1: in STD\_LOGIC;

and\_g2: in STD\_LOGIC;

and\_g3: in STD\_LOGIC;

and\_cikis: out STD\_LOGIC);

end and\_kapisi;

architecture Behavioral of and\_kapisi is

begin

process(and\_g1,and\_g2,and\_g3)

begin

and\_cikis <= and\_g1 and and\_g2 and and\_g3;

end process;

end Behavioral;

----------------------------------------------------------------------------------

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity Flipflop is

port( T: in STD\_LOGIC;

CP : in STD\_LOGIC;

Q : inout STD\_LOGIC;

QBAR : inout STD\_LOGIC);

end Flipflop;

architecture Behavioral of Flipflop is

component nor\_kapisi is

port( nor\_g1: in STD\_LOGIC;

nor\_g2: in STD\_LOGIC;

nor\_cikis: out STD\_LOGIC);

end component;

component and\_kapisi is

port( and\_g1: in STD\_LOGIC;

and\_g2: in STD\_LOGIC;

and\_g3: in STD\_LOGIC;

and\_cikis: out STD\_LOGIC);

end component;

signal arakablo1:STD\_LOGIC;

signal arakablo2:STD\_LOGIC;

begin

blok1: and\_kapisi port map(and\_cikis=>arakablo1,and\_g1=>Q, and\_g2=>T, and\_g3=>CP);

blok2: and\_kapisi port map(and\_cikis=>arakablo2,and\_g1=>QBAR,and\_g2=>T,and\_g3=>CP);

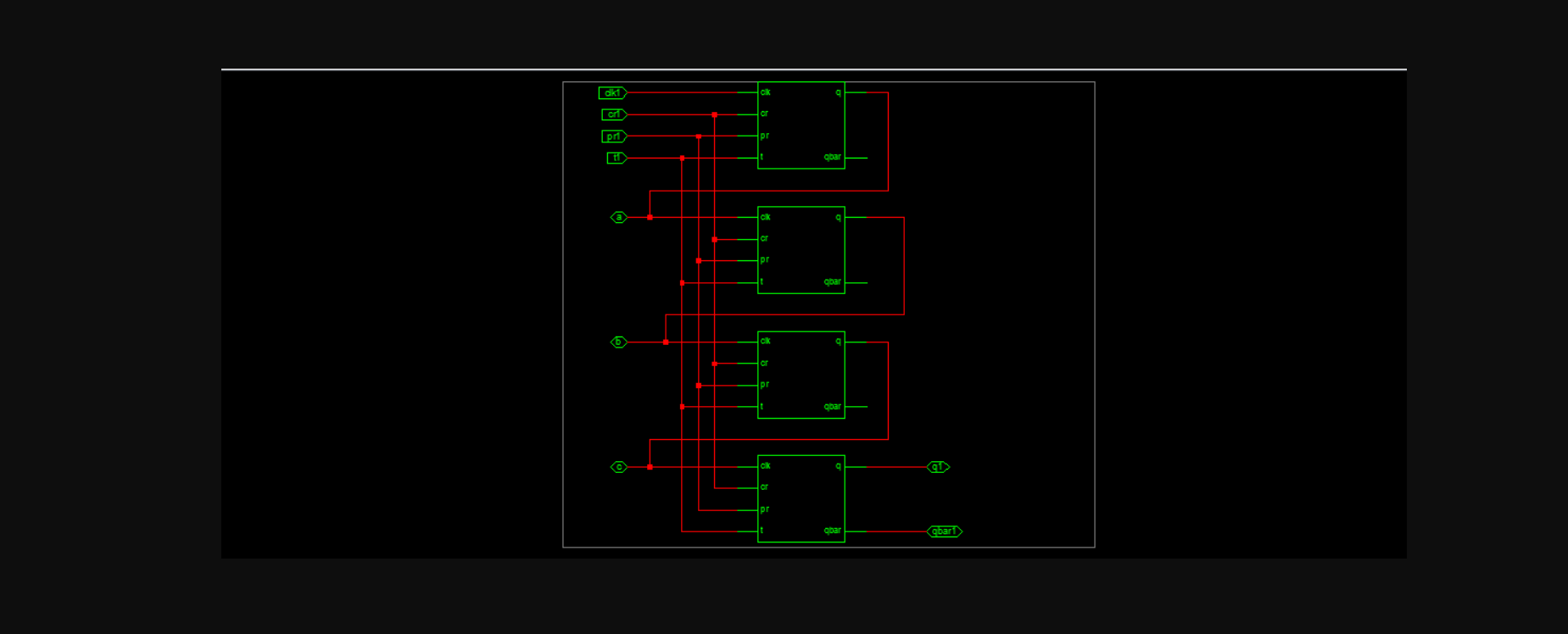
blok3: nor\_kapisi port map(nor\_cikis=>Q,nor\_g1=>arakablo1,nor\_g2=>QBAR);

blok4: nor\_kapisi port map(nor\_cikis=>QBAR,nor\_g1=>arakablo2,nor\_g2=>Q);

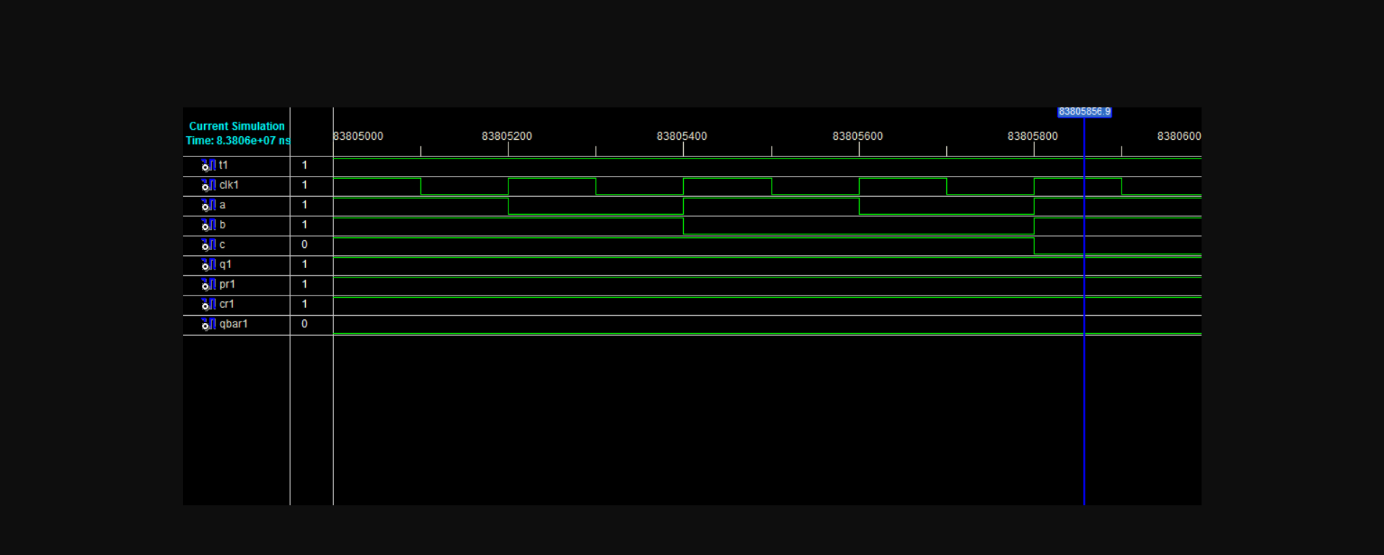
end Behavioral;

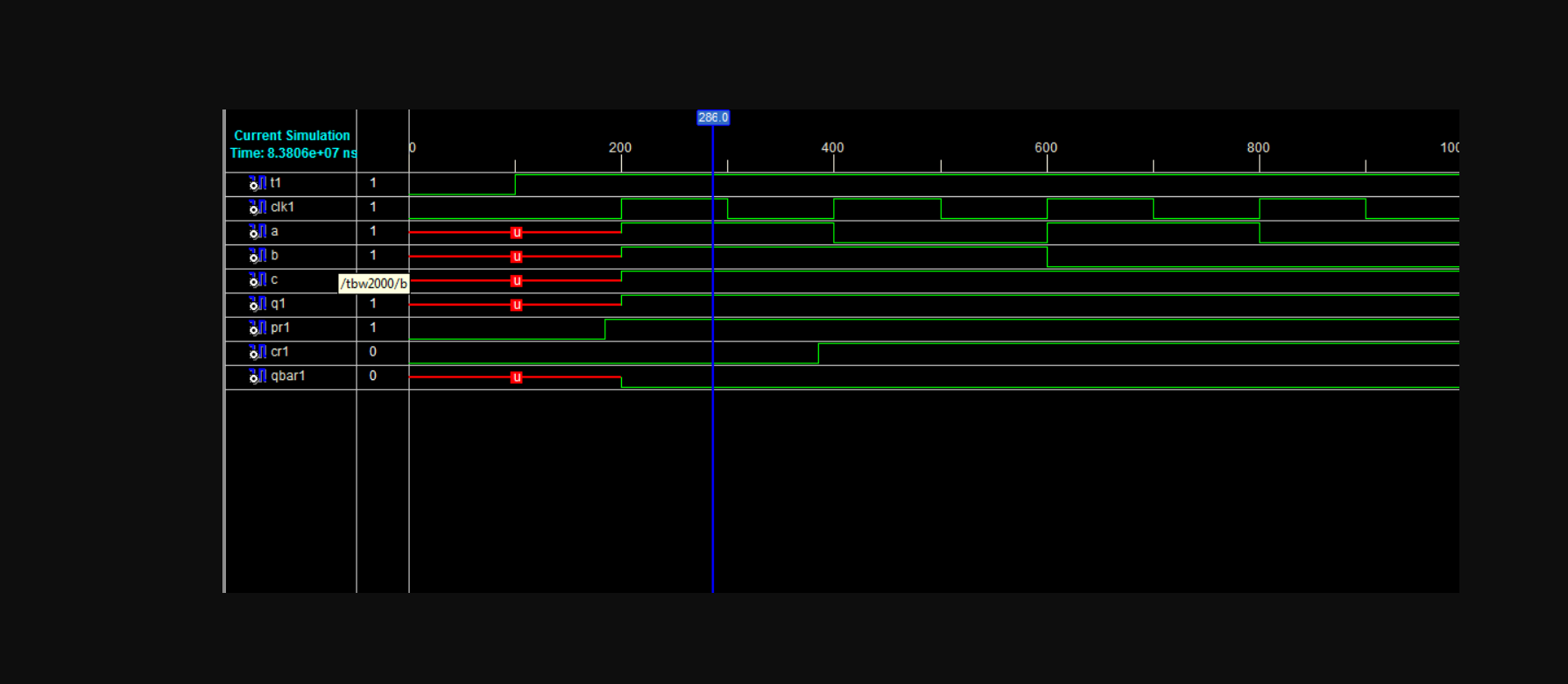
2. Asynchronous Counter

RTL Şematik



Simülasyon





KOD

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

use IEEE.STD\_LOGIC\_ARITH.ALL;

use IEEE.STD\_LOGIC\_UNSIGNED.ALL;

entity async is

Port ( t1 : in STD\_LOGIC;

clk1 : in STD\_LOGIC;

a : inout STD\_LOGIC;

b : inout STD\_LOGIC;

c : inout STD\_LOGIC ;

q1 : inout STD\_LOGIC;

pr1: in STD\_LOGIC;

cr1: in STD\_LOGIC;

qbar1 : inout STD\_LOGIC);

end async;

architecture Behavioral of async is

component t\_ff4 is

Port ( t : in STD\_LOGIC;

clk : in STD\_LOGIC;

pr: in STD\_LOGIC;

cr: in STD\_LOGIC;

q : inout STD\_LOGIC;

qbar : inout STD\_LOGIC);

end component;

signal abar,bbar,cbar:STD\_LOGIC;

begin

tff1: t\_ff4 port map (t1,clk1,pr1,cr1,a,abar);

tff2: t\_ff4 port map (t1,a,pr1,cr1,b,bbar);

tff3: t\_ff4 port map (t1,b,pr1,cr1,c,cbar);

tff4: t\_ff4 port map (t1,c,pr1,cr1,q1,qbar1);

end Behavioral;