**BİLGİSAYAR ORGANİZASYONU VE TASARIMI LABORATUVARI UYGULAMA RAPORU**

**UYGULAMA NO:** 4

**UYGULAMA TARİHİ:** 14 ARALIK 2016

**GRUP NO:**  G8

**GRUP ÜYELERİ**

1358130030 Anıl Ertürk

1358130060 Gülçin Çelebi

**Bölüm 1**

-- Description: toplama cikarma islemi yapar. put a ... put b ... put m 1- cikar 0 topla

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL; --standart kütüphane bildirimi

entity tam is

--ötelemeli toplayıcıyı yapısal mimari ile tasarlamak için önce tam toplayıcı tasarlamamız gerekir

port(

x,y,z : in std\_logic;

c,s: out std\_logic

); --tam toplayıcının üç giriş (x, y, z) ve iki çıkış (c ve s) portu bulunur

end tam;

architecture dataflow of tam is -–veri akışı mimarisi kullanıyoruz

begin

C <= (x and y) or (z and y) or (z and x); --C ve S’nin karakteristik denklemleri

S <= (not z and not x and y) or (not z and x and not y) or (z and not x and not y) or (z and x and y);

end dataflow;

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL; --standart kütüphane bildirimi

entity nbitadder is

--n bit toplayicinin üç giriş (A, B, M) ve iki çıkış (S ve C) portu bulunur

port(

A,B: in STD\_LOGIC\_VECTOR (3 downto 0);

S: out STD\_LOGIC\_VECTOR (3 downto 0);

M: in std\_logic;

C: out std\_logic

);

end nbitadder;

architecture Structural of nbitadder is

component tam is --component bildirimi

port(

x,y,z : in std\_logic;

c,s: out std\_logic

);

end component;

signal x0,x1,x2,x3,c1,c2,c3 : std\_logic;

--x sinyalleri xor dan gelen xor’lanmi’ bitlerini tutar. c sinyalleri ise tam toplayıcılardan tam

--toplayıcılara giden üç adet eldeyi ifade tutar

begin

x0 <= M xor A(3); --xor mapping işlemleri

x1 <= M xor A(2);

x2 <= M xor A(1);

x3 <= M xor A(0);

tam1: tam port map(x0,B(3),M,c1,S(0)); --tam toplayici mapping işlemleri

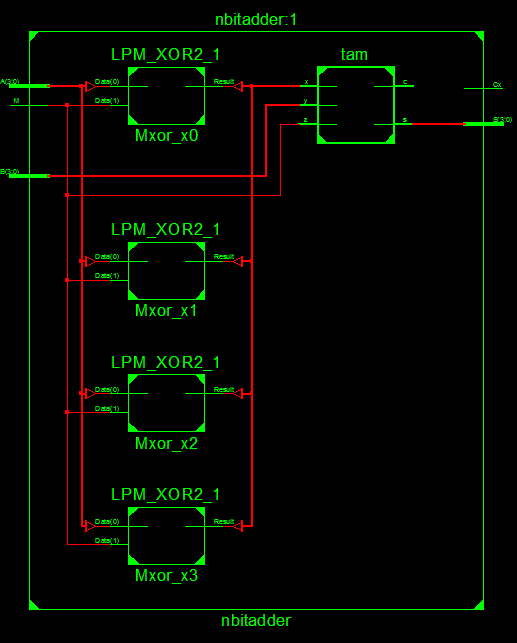
tam2: tam port map(x1,B(2),c1,c2,S(1));

tam3: tam port map(x2,B(1),c2,c3,S(2));

tam4: tam port map(x3,B(0),c3,C,S(3));

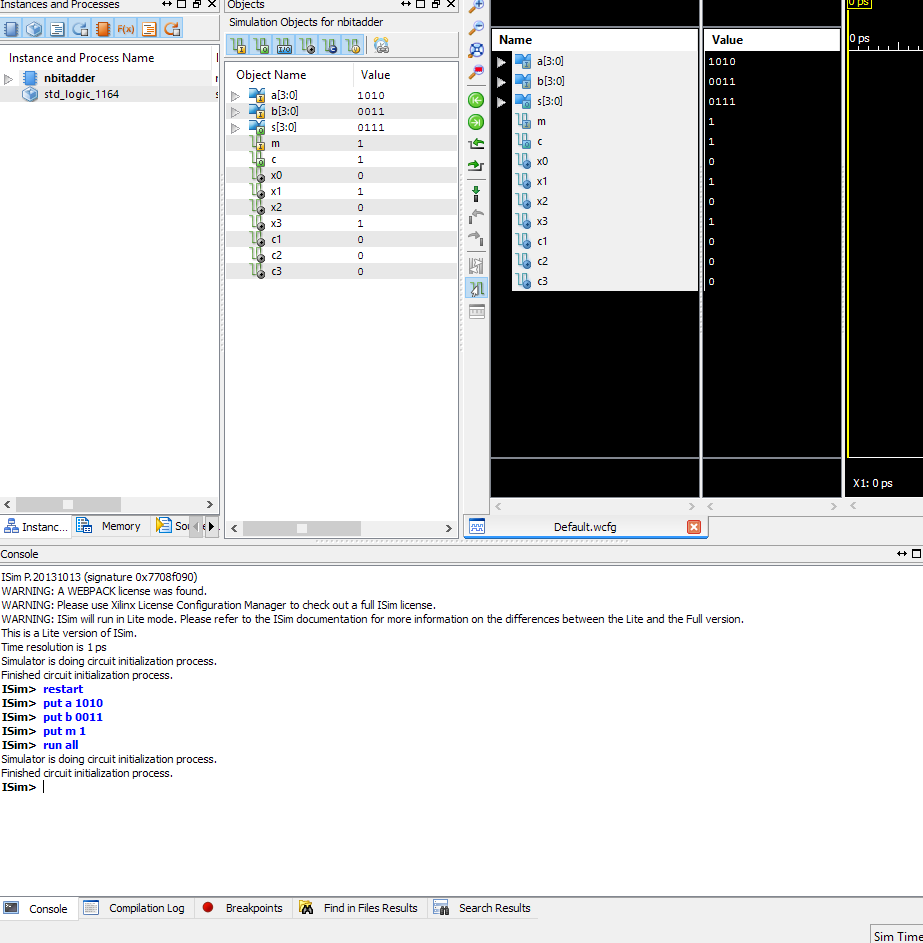
end Structural;

**Bölüm 2**



RTL şeması

**Bölüm 3**



Waveform Diagram