ECE 3561 MT Register Set

12/03/19

Matt Gambill

***regLine Code Listing***

LIBRARY IEEE;

USE IEEE.STD\_LOGIC\_1164.ALL;

ENTITY regLine IS

PORT (

ABUS, BBUS : INOUT std\_logic\_vector(15 DOWNTO 0);

Aload, Bload, Adrive, Bdrive, Arsel, Brsel : IN std\_logic

);

END regLine;

ARCHITECTURE one OF regLine IS

-- define subcomponents

COMPONENT busdr IS

PORT (

dr : IN std\_logic;

datain : IN std\_logic\_vector(15 DOWNTO 0);

dataout : OUT std\_logic\_vector(15 DOWNTO 0)

);

END COMPONENT;

FOR ALL : busdr USE ENTITY work.busdr(one);

COMPONENT reg16 IS

PORT (

din : IN std\_logic\_vector(15 DOWNTO 0);

dout : OUT std\_logic\_vector(15 DOWNTO 0);

load : IN std\_logic

);

END COMPONENT;

FOR ALL : reg16 USE ENTITY WORK.reg16(one);

COMPONENT mux2\_to\_1x16 IS

PORT (

a, b : IN std\_logic\_vector(15 DOWNTO 0);

sa, sb : IN std\_logic;

muxout : OUT std\_logic\_vector(15 DOWNTO 0)

);

END COMPONENT;

SIGNAL mux0\_out, regOut : std\_logic\_vector(15 DOWNTO 0);

SIGNAL regLoad, Abusdr\_load, Bbusdr\_load : std\_logic;

BEGIN

mux0 : mux2\_to\_1x16

PORT MAP(ABUS, BBUS, Arsel, Brsel, mux0\_out);

reg0 : reg16 PORT MAP(mux0\_out, regOut, regLoad);

Abusdr : busdr PORT MAP(Abusdr\_load, regOut, ABUS);

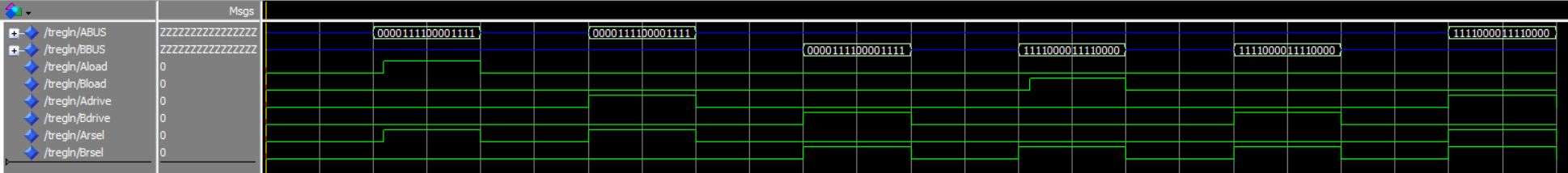
Bbusdr : busdr PORT MAP(Bbusdr\_load, regOut, BBUS);

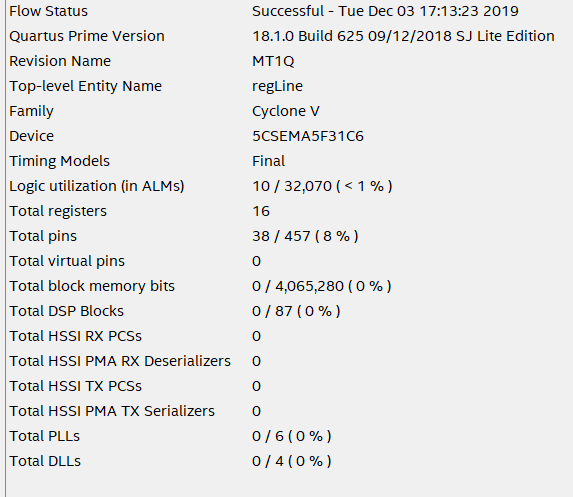
Abusdr\_load <= (Adrive) AND (Arsel);

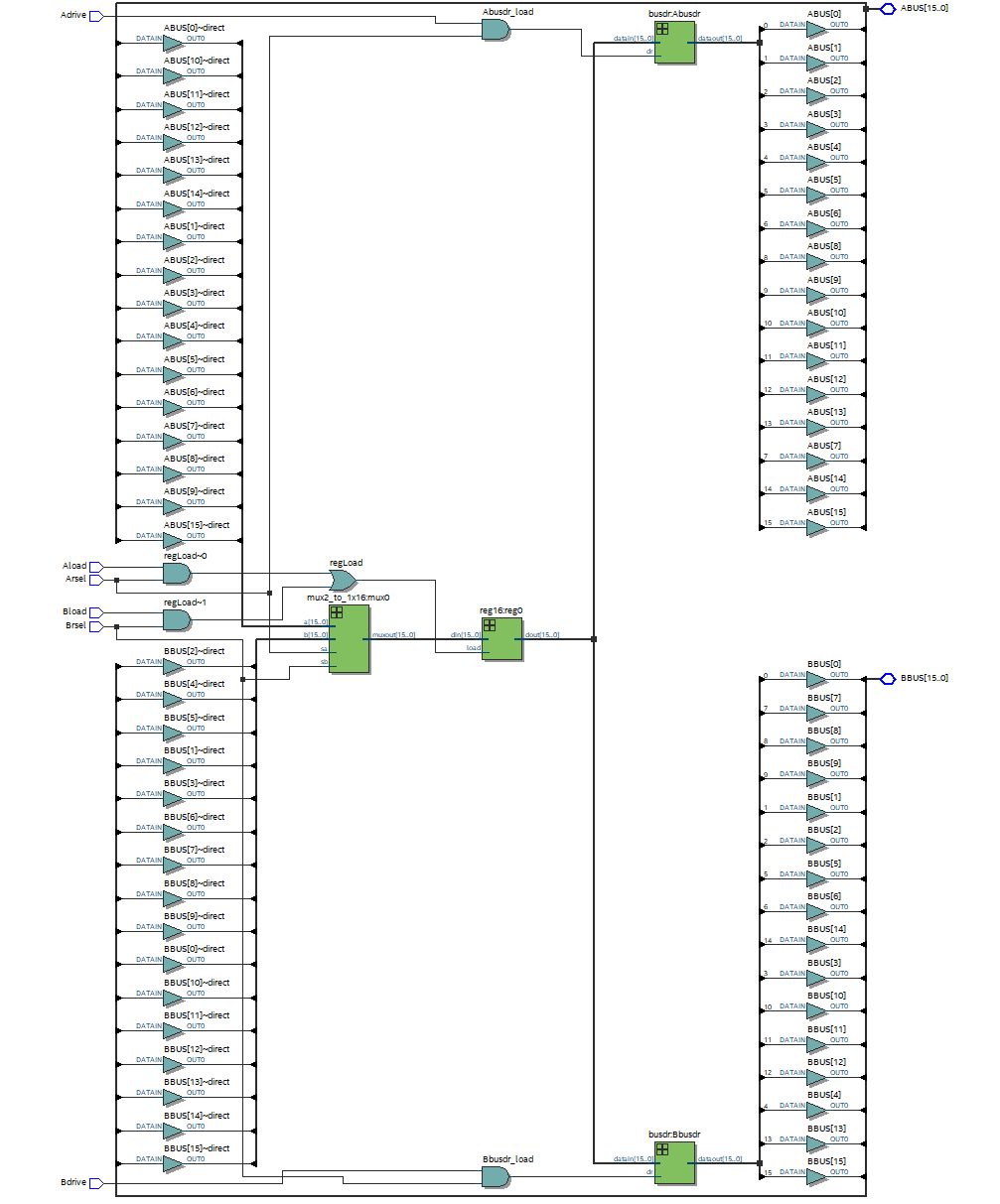
Bbusdr\_load <= (Bdrive) AND (Brsel);

regLoad <= (Aload AND Arsel) OR (Bload AND Brsel);

END one;







***regSet Code Listing***

LIBRARY IEEE;

USE IEEE.STD\_LOGIC\_1164.ALL;

ENTITY regSet IS

PORT (

ABUS, BBUS : INOUT std\_logic\_vector(15 DOWNTO 0);

Aload, Bload, Adrive, Bdrive : IN std\_logic;

Aregno,Bregno : IN std\_logic\_vector(2 downto 0)

);

END regSet;

ARCHITECTURE one OF regSet IS

-- define subcomponents

COMPONENT regLine IS

PORT (

ABUS, BBUS : INOUT std\_logic\_vector(15 DOWNTO 0);

Aload, Bload, Adrive, Bdrive, Arsel, Brsel : IN std\_logic

);

END COMPONENT;

COMPONENT pri3to8 IS

PORT (pri : IN std\_logic\_vector(2 downto 0);

p0,p1,p2,p3,p4,p5,p6,p7 : OUT std\_logic);

END COMPONENT;

SIGNAL Arsel0,Arsel1,Arsel2,Arsel3,Arsel4,Arsel5,Arsel6,Arsel7 : std\_logic;

SIGNAL Brsel0,Brsel1,Brsel2,Brsel3,Brsel4,Brsel5,Brsel6,Brsel7 : std\_logic;

BEGIN

regLine0 : regLine PORT MAP(ABUS,BBUS,Aload,Bload,Adrive,Bdrive,Arsel0, Brsel0);

regLine1 : regLine PORT MAP(ABUS,BBUS,Aload,Bload,Adrive,Bdrive,Arsel1, Brsel1);

regLine2 : regLine PORT MAP(ABUS,BBUS,Aload,Bload,Adrive,Bdrive,Arsel2, Brsel2);

regLine3 : regLine PORT MAP(ABUS,BBUS,Aload,Bload,Adrive,Bdrive,Arsel3, Brsel3);

regLine4 : regLine PORT MAP(ABUS,BBUS,Aload,Bload,Adrive,Bdrive,Arsel4, Brsel4);

regLine5 : regLine PORT MAP(ABUS,BBUS,Aload,Bload,Adrive,Bdrive,Arsel5, Brsel5);

regLine6 : regLine PORT MAP(ABUS,BBUS,Aload,Bload,Adrive,Bdrive,Arsel6, Brsel6);

regLine7 : regLine PORT MAP(ABUS,BBUS,Aload,Bload,Adrive,Bdrive,Arsel7, Brsel7);

Aencoder : pri3to8 PORT MAP(Aregno,Arsel0,Arsel1,Arsel2,Arsel3,Arsel4,Arsel5,Arsel6,Arsel7);

Bencoder : pri3to8 PORT MAP(Bregno,Brsel0,Brsel1,Brsel2,Brsel3,Brsel4,Brsel5,Brsel6,Brsel7);

END one;

