**Calculator without simulation**

by Lini Mestar

ECE 424 Lab

January 26th, 2016

**Top**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

-- Uncomment the following library declaration if using

-- arithmetic functions with Signed or Unsigned values

--use IEEE.NUMERIC\_STD.ALL;

-- Uncomment the following library declaration if instantiating

-- any Xilinx primitives in this code.

--library UNISIM;

--use UNISIM.VComponents.all;

entity calculator is

port( sub,clk,reset:in STD\_LOGIC;

a,b:in std\_logic\_vector(3 downto 0);

PD0,QD0,RD0,PD1,QD1,RD1,PD2: out std\_logic\_vector(3 downto 0));

end calculator;

architecture Behavioral of calculator is

component B2BCD

port(p:in std\_logic\_vector(7 downto 0);

Q,R: in std\_logic\_vector(3 downto 0);

PD0,QD0,RD0,PD1,QD1,RD1,PD2: out std\_logic\_vector(3 downto 0);

cout,clk,reset:in std\_logic

);

end component;

component AddSub

Port ( a,b : in STD\_LOGIC\_VECTOR (3 downto 0);

op : in STD\_LOGIC;

ov, cout : out STD\_LOGIC;

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

end component;

component fourBit\_multi

Port ( a,b : in std\_logic\_vector (3 downto 0);

p: out std\_logic\_vector (7 downto 0));

end component;

component full\_divider

Port (a,b: in std\_logic\_vector (3 downto 0);

q1 : out std\_logic\_vector (3 downto 0));

end component;

--signal : std\_logic\_vector( 2 downto 0 ):="000";

--signal : std\_logic\_vector( 3 downto 0 ):="0000";

signal WADD, WADD\_RAM,RADD\_RAM : std\_logic\_vector( 4 downto 0 ):="00000";

signal B2BCD\_R,B2BCD\_Q : std\_logic\_vector( 3 downto 0 ):="0000";

signal B2BCD\_P : std\_logic\_vector( 7 downto 0 ):="00000000";

signal cout,ov : std\_logic:='0';

--signal Dataout : std\_logic\_vector(3 downto 0);

--signal Control : std\_logic\_vector(2 downto 0);

begin

divider: full\_divider port map(a,b,B2BCD\_Q);

multiplier: fourBit\_multi port map(a,b,B2BCD\_P);

AdderSubr: AddSub port map(a,b,sub,ov,cout,B2BCD\_R);

B2BCD2: B2BCD port map(B2BCD\_P,B2BCD\_Q,B2BCD\_R,PD0,QD0,RD0,PD1,QD1,RD1,PD2,cout,clk,reset);

end Behavioral;

**Divider**

library IEEE;

use ieee.std\_logic\_1164.ALL;

use ieee.std\_logic\_arith.ALL;

use ieee.std\_logic\_unsigned.ALL;

use ieee.numeric\_std.all;

entity full\_divider is

Port (a,b: in std\_logic\_vector (3 downto 0);

q1 : out std\_logic\_vector (3 downto 0));

end full\_divider;

architecture Behavioral of full\_divider is

Signal X : std\_logic\_vector (9 downto 1):= "000000000"; -- for souts

Signal Y: std\_logic\_vector (12 downto 1):= "000000000000"; -- for aouts

--Signal Z : std\_logic\_vector (3 downto 1):= "000"; -- for 0's

Signal A1: std\_logic\_vector (11 downto 1):= "00000000000"; -- for dont cares

--Signal B1 : std\_logic\_vector (2 downto 1):= "11"; -- for 1's

Signal B2 : std\_logic\_vector (16 downto 1):= "0000000000000000"; -- for 1's

Signal L: std\_logic\_vector (12 downto 1):= "000000000000"; -- for couts

Signal q : std\_logic\_vector (4 downto 1):= "0000";

component divider is

Port ( ain , zin, cin, sin : in std\_logic;

aout, zout, cout, sout: out std\_logic);

end component;

begin

Div0 : divider Port map (a(0), '1', '1', b(3),

Y(1), B2(1), L(1), X(1));

Div1 : divider Port map (a(1), '1', L(1), '0',

Y(2), B2(2), L(2), X(2));

Div2 : divider Port map (a(2), '1', L(2), '0',

Y(3),B2(3), L(3), X(3));

Div3 : divider Port map (a(3), '1', L(3), '0',

Y(4), B2(4), q(4), A1(1));

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

Div4 : divider Port map (Y(1), q(4), q(4), b(2),

Y(5), B2(5), L(4), X(4));

Div5 : divider Port map (Y(2), B2(5), L(4), X(1),

Y(6), B2(6), L(5), X(5));

Div6 : divider Port map (Y(3), B2(6), L(5), X(2),

Y(7), B2(7), L(6), X(6));

Div7 : divider Port map (Y(4), B2(7), L(6), X(3),

Y(8), B2(8), q(3), A1(2));

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

Div8 : divider Port map (Y(5), q(3), q(3), b(1),

Y(9), B2(9), L(7), X(7));

Div9 : divider Port map (Y(6), B2(9), L(7), X(4),

Y(10), B2(10), L(8), X(8));

Div10 : divider Port map (Y(7), B2(10), L(8), X(5),

Y(11), B2(11), L(9), X(9));

Div11 : divider Port map (Y(8), B2(11), L(9), X(6),

Y(12), B2(12), q(2), A1(3));

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

Div12 : divider Port map (Y(9), q(2), q(2), b(0),

A1(4), B2(13), L(10), A1(5));

Div13 : divider Port map (Y(10), B2(13), L(10), X(7),

A1(6), B2(14), L(11), A1(7));

Div14 : divider Port map (Y(11), B2(14), L(11), X(8),

A1(8), B2(15), L(12), A1(9));

Div15 : divider Port map (Y(12), B2(15), L(12), X(9),

A1(10),B2(16), q(1), A1(11));

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

q1(3) <= q(4);

q1(2) <= q(3);

q1(1) <= q(2);

q1(0) <= q(1);

end Behavioral;

**Multiplier**

library IEEE;

use ieee.std\_logic\_1164.ALL;

use ieee.std\_logic\_arith.ALL;

use ieee.std\_logic\_unsigned.ALL;

use ieee.numeric\_std.all;

entity fourBit\_multi is

Port ( a,b : in std\_logic\_vector (3 downto 0);

p: out std\_logic\_vector (7 downto 0));

end fourBit\_multi;

architecture Behavioral of fourBit\_multi is

signal f : std\_logic\_vector(11 downto 1) := "00000000000"; -- to use 0 as a std\_log\_vector

signal c,y,z: std\_logic\_vector (12 downto 1) := "000000000000"; -- inner carryout, aout , bout

signal s : std\_logic\_vector (9 downto 1 ) := "000000000"; -- inner sums

signal x: std\_logic\_vector (12 downto 1) := "000000000000"; -- for dont care signals

component multi is

Port (ain, bin, cin, sin : in std\_logic;

aout, bout, cout, sout : out std\_logic);

end component;

begin

multi0 : multi Port map ( a(0), b(0), f(1) ,f(2),

y(1), z(1), c(1), p(0));

multi1 : multi Port map ( a(1), z(1), c(1) , f(3),

y(2), z(2), c(2), s(1));

multi2 : multi Port map ( a(2), z(2), c(2) , f(4),

y(3), z(3), c(3), s(2));

multi3 : multi Port map ( a(3), z(3), c(3) , f(5),

y(4), x(1), x(2), s(3));

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

multi4 : multi Port map ( y(1), b(1), f(6) , s(1),

y(5), z(4), c(4), p(1));

multi5 : multi Port map ( y(2), z(4), c(4) , s(2),

y(6), z(5), c(5), s(4));

multi6 : multi Port map ( y(3), z(5), c(5), s(3),

y(7), z(6), c(6), s(5));

multi7 : multi Port map ( y(4), z(6), c(6) , f(7),

y(8), x(3), x(4), s(6));

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

multi8 : multi Port map ( y(5), b(2), f(8) , s(4),

y(9), z(7), c(7), p(2));

multi9 : multi Port map ( y(6), z(7), c(7) , s(5),

y(10), z(8), c(8), s(7));

multi10 : multi Port map ( y(7), z(8), c(8) , s(6),

y(11), z(9), c(9), s(8));

multi11 : multi Port map ( y(8), z(9), c(9) , f(9),

y(12), x(5), x(6), s(9));

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

multi12 : multi Port map ( y(9), b(3), f(10) , s(7),

x(7), z(10), c(10), p(3));

multi13 : multi Port map ( y(10), z(10), c(10) , s(8),

x(8), z(11), c(11), p(4));

multi14 : multi Port map ( y(11), z(11), c(11) , s(9),

x(9), z(12), c(12), p(5));

multi15 : multi Port map ( y(12), z(12), c(12) , f(11),

x(10), x(11), x(12), p(6));

p(7) <= x(12);

end Behavioral;

**AdderSubr**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

--use IEEE.STD\_LOGIC\_ARITH.ALL;

--use IEEE.STD\_LOGIC\_UNSIGNED.ALL;

entity AddSub is

Port ( a,b : in STD\_LOGIC\_VECTOR (3 downto 0);

op : in STD\_LOGIC;

ov, cout : out STD\_LOGIC;

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

end AddSub;

architecture Behavioral of AddSub is

signal z:std\_logic\_vector(3 downto 0);

component Adder\_4 is

Port ( x,y : in STD\_LOGIC\_VECTOR (3 downto 0);

cin : in STD\_LOGIC; cout,ov : out STD\_LOGIC;

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

end component;

begin

z<= b xor op&op&op&op;

add:Adder\_4 Port map( a,z ,op ,cout,ov ,s);

end Behavioral;

**B2bcd2**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

use IEEE.STD\_LOGIC\_ARITH.ALL;

use IEEE.STD\_LOGIC\_UNSIGNED.ALL;

use IEEE.math\_real.all;

-- Uncomment the following library declaration if using

-- arithmetic functions with Signed or Unsigned values

--use IEEE.NUMERIC\_STD.ALL;

-- Uncomment the following library declaration if instantiating

-- any Xilinx primitives in this code.

--library UNISIM;

--use UNISIM.VComponents.all;

entity B2BCD is

port (

p:in std\_logic\_vector(7 downto 0);

Q,R: in std\_logic\_vector(3 downto 0);

PD0,QD0,RD0,PD1,QD1,RD1,PD2: out std\_logic\_vector(3 downto 0);

cout,clk,reset:in std\_logic

);

end B2BCD;

architecture Behavioral of B2BCD is

signal state : std\_logic\_vector( 2 downto 0 ):="000";

signal D1,D2 : std\_logic\_vector( 3 downto 0 );

signal A : std\_logic\_vector( 7 downto 0 );

begin

process

begin

if (reset='1') then

state<="000";

elsif (clk'event and clk='1') then

case state is

when "000"=> D2<="0000"; D1<="0000"; A<=P; state<="001";

when "001"=> if(A>"01100011") then

A<=A-"01100100"; D2<=D2+"0001"; state<="001";

else

if(A>"00001001") then

A<=A-"00001010"; D1<=D1+"0001"; state<="001";

else

PD2<=D2; PD1<=D1; PD0<=A(3 downto 0); A<="0000"&Q; D1<="0000"; state<="010";

end if;

end if;

when "010"=> if(A>9) then

A<=A-"00001010"; D1<=D1+"0001"; state<="010";

else

QD1<=D1; QD0<=A(3 downto 0); A<="000"&cout&R; D1<="0000"; state<="011";

end if;

when "011"=> if(A>9) then

A<=A-10; D1<=D1+1; state<="011";

else

RD1<=D1; RD0<=A(3 downto 0); state<="000";

end if;

when others=> state<="000";

end case;

end if;

end process;

end Behavioral;