library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL; use IEEE.STD\_LOGIC\_ARITH.ALL;

use IEEE.STD\_LOGIC\_UNSIGNED.ALL;

* Uncomment the following library declaration if instantiating
* any Xilinx primitives in this code.

--library UNISIM;

--use UNISIM.VComponents.all;

entity test11 is

port(

B, const1, F, Wb, Wc : in std\_logic\_vector (3 downto 0); clock, reset: in std\_logic;

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

z,r: out std\_logic\_vector (3 downto 0); --z: out std\_logic;

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

end test11;

architecture Behavioral of test11 is

component D\_latch5 is

port( Data\_in5 : in std\_logic\_vector(3 downto 0); latch\_strobe\_R3: in std\_logic;

Data\_out5: out std\_logic\_vector(3 downto 0)

); end component;

component mux4\_2\_to\_1 is port (

const1: in std\_logic\_vector(3 downto 0); Wb: in std\_logic\_vector(3 downto 0); selector\_R3: in std\_logic;

O4: out std\_logic\_vector (3 downto 0) );

end component;

component mux3\_2\_to\_1 is port (

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

R2out: in std\_logic\_vector(7 downto 0); selector\_R3: in std\_logic;

O3: out std\_logic\_vector (7 downto 0)

);

end component;

component D\_latch4 is

port( Data\_in4 : in std\_logic\_vector(7 downto 0); latch\_strobe\_R3: in std\_logic;

Data\_out4: out std\_logic\_vector(7 downto 0) );

end component;

component divcas4 is -- 8 bit dividend, 4 bit divisor port (

enable\_R3 : in std\_logic;

dividend : in std\_logic\_vector(7 downto 0); divisor : in std\_logic\_vector(3 downto 0); quotient : out std\_logic\_vector(3 downto 0); remainder : out std\_logic\_vector(3 downto 0)

); end component;

component D\_latch6 is

port( Data\_in6 : in std\_logic\_vector(3 downto 0); output\_strobe\_R3: in std\_logic;

Data\_out6: out std\_logic\_vector(3 downto 0) );

end component;

component demux2\_2\_to\_1 is port (

Data\_out6: in std\_logic\_vector(3 downto 0);

R1in, RegM: out std\_logic\_vector(3 downto 0); Deselector\_R3: in std\_logic

); end component;

component D\_latch8 is

port( Data\_in8 : in std\_logic\_vector(3 downto 0); latch\_strobe\_R1: in std\_logic;

Data\_out8: out std\_logic\_vector(3 downto 0) );

end component;

component D\_latch7 is

port( Data\_in7 : in std\_logic\_vector(3 downto 0); latch\_strobe\_R1: in std\_logic;

Data\_out7: out std\_logic\_vector(3 downto 0) );

end component;

component Adder is port

(Data\_out7: in std\_logic\_vector(3 downto 0); enable\_R1: in std\_logic;

Data\_out8: in std\_logic\_vector(3 downto 0); Data\_in9 : out std\_logic\_vector(3 downto 0) );

end component;

component D\_latch9 is port(

Data\_in9 : in std\_logic\_vector( 3 downto 0); output\_strobe\_R1: in std\_logic;

R2input: out std\_logic\_vector(3 downto 0) );

end component;

component mux\_2\_to\_1 is port (

Wc: in std\_logic\_vector(3 downto 0);

R1out: in std\_logic\_vector(3 downto 0); selector\_R2: in std\_logic;

O: out std\_logic\_vector (3 downto 0)

);

end component;

component D\_latch is

port( Data\_in : in std\_logic\_vector(3 downto 0); latch\_strobe\_R2: in std\_logic;

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

End component;

component multiplier is

port( enable\_R2: in std\_logic;

Data\_out: in std\_logic\_vector(3 downto 0); Data\_out2: in std\_logic\_vector(3 downto 0); Data\_in3 : out std\_logic\_vector(7 downto 0) );

end component;

component D\_latch2 is

port( Data\_in2 : in std\_logic\_vector(3 downto 0); latch\_strobe\_R2: in std\_logic;

Data\_out2: out std\_logic\_vector(3 downto 0) );

end component;

component D\_latch3 is

port( Data\_in3 : in std\_logic\_vector(7 downto 0); output\_strobe\_R2: in std\_logic;

Data\_out3: out std\_logic\_vector(7 downto 0) );

end component;

component Demux1\_2\_to\_1 is port (

Data\_out3: in std\_logic\_vector(7 downto 0);

R3in, RegY: out std\_logic\_vector(7 downto 0); Deselector\_R2: in std\_logic

); end component;

component registerY is port

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

Y : out std\_logic\_vector(7 downto 0)

);

end component;

component registerF is port

( F: in std\_logic\_vector(3 downto 0); strobe: in std\_logic;

regF : out std\_logic\_vector(3 downto 0)

);

end component;

component registerB is port

( B: in std\_logic\_vector(3 downto 0); strobe: in std\_logic;

regB : out std\_logic\_vector(3 downto 0)

);

end component;

component registerM is port

( regm: in std\_logic\_vector(3 downto 0); Dstrobe: in std\_logic;

Data\_in7 : out std\_logic\_vector(3 downto 0)

);

end component;

component register1 is port

( const1: in std\_logic\_vector(3 downto 0); strobe: in std\_logic;

reg1: out std\_logic\_vector(3 downto 0)

);

end component;

component registerWb is port

( wb: in std\_logic\_vector(3 downto 0);

strobe: in std\_logic;

regwb : out std\_logic\_vector(3 downto 0)

);

end component;

component registerWc is port

( wc: in std\_logic\_vector(3 downto 0); strobe: in std\_logic;

Data\_in1 : out std\_logic\_vector(3 downto 0)

);

end component;

component mux2\_2\_to\_1 is port (

1. in std\_logic\_vector(3 downto 0);
   1. in std\_logic\_vector(3 downto 0); selector\_R2: in std\_logic;

O2: out std\_logic\_vector (3 downto 0)

);

end component;

component registerTp is port

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

regtp : out std\_logic\_vector(7 downto 0)

);

end component;

component control\_unit is

port(

clock, reset: in std\_logic; latch\_strobe\_R2,output\_strobe\_R2,enable\_R2,selector\_R2,Deselector\_R2: out

std\_logic;

selector\_R3,latch\_strobe\_R3,enable\_R3,output\_strobe\_R3,Deselector\_R3: out std\_logic;

--count1 :out INTEGER; latch\_strobe\_R1,enable\_R1,output\_strobe\_R1,clk,strobe\_regM, strobe\_regY:

out std\_logic

);

end component;

signal sig1, sig2, sig3, sig4, sig7, sig8, sig10 :std\_logic\_vector(3 downto 0); signal sig44, sig45, sig81 : std\_logic\_vector(3 downto 0);

signal sig39, sig40, sig43, sig46 : std\_logic\_vector (7 downto 0); signal sig47, sig51, sig54, sig13, sig86: std\_logic\_vector (7 downto 0); signal sig5, sig6, sig9, sig52, sig53 :std\_logic;

signal sig55, sig80, sig48 : std\_logic;

signal sig11, sig12,sig16, sig17, sig19, sig20: std\_logic\_vector(3 downto 0); signal sig21, sig22, sig25,sig61: std\_logic\_vector ( 3 downto 0);

signal sig14, sig15, sig18, sig23 : std\_logic; signal sig24, sig30, sig31, sig32: std\_logic; signal sig36, sig37, sig50, sig59: std\_logic; signal sig60,sig64, sig66: std\_logic;

signal sig62, sig68, sig70,sig71,sig74: std\_logic;

signal sig26, sig27, sig28, sig29: std\_logic\_vector (3 downto 0); signal sig33, sig34, sig35, sig56: std\_logic\_vector (3 downto 0); signal sig57, sig38, sig58, sig63: std\_logic\_vector (3 downto 0); signal sig65, sig67, sig69,sig72,sig73: std\_logic\_vector ( 3 downto 0); signal sigc, sigr: std\_logic;

begin

sigc<=clock;

sigr<=reset;

sig69<=B;

sig67<=F;

sig57<=Wc;

sig7<=Wb;

sig72<=const1;

sig86<=tp;

stage\_0 : mux3\_2\_to\_1 port map (sig39, sig54, sig9, sig43); stage\_1 : D\_latch5 port map (sig10, sig6, sig2);

stage\_2 : mux4\_2\_to\_1 port map (sig8, sig7, sig9,sig10); stage\_3 : D\_latch4 port map(sig43, sig6, sig13);

stage\_4 : divcas4 port map (sig80, sig13, sig2, sig16, sig17); stage\_5 : D\_latch6 port map(sig16, sig18, sig19);

stage\_6 : demux2\_2\_to\_1 port map (sig19, sig20, sig21, sig23); stage\_7 : D\_latch8 port map (sig20, sig24, sig25);

stage\_8 : D\_latch7 port map (sig26, sig24, sig27); stage\_9 : Adder port map (sig27, sig31, sig25, sig29); stage\_10: D\_latch9 port map (sig29, sig32, sig33);

stage\_11: mux\_2\_to\_1 port map (sig81, sig33,sig36,sig34); stage\_12: D\_latch port map ( sig34, sig37, sig38); stage\_13: multiplier port map (sig66, sig38, sig65, sig46); stage\_14: D\_latch3 port map (sig46, sig53, sig47);

stage\_15: Demux1\_2\_to\_1 port map (sig47, sig54, sig40, sig48); stage\_16: registerY port map (sig40, sig55, sig51);

stage\_17: registerWb port map (sig7, sig59, sig56); stage\_18: registerWc port map (sig57, sig59, sig81); stage\_19: mux2\_2\_to\_1 port map (sig58,sig61,sig36,sig63); stage\_20: D\_latch2 port map (sig63, sig37, sig65); stage\_21: registerF port map (sig67, sig59, sig58); stage\_22: registerB port map (sig69, sig59, sig61); stage\_23: registerM port map (sig21, sig74, sig26); stage\_24: register1 port map (sig72,sig59,sig8);

stage\_25: registerTp port map (sig86,sig59,sig39);

stage\_26:control\_unit port map (sigc, sigr, sig37, sig53, sig66, sig36, sig48, sig9, sig6, sig80, sig18, sig23, sig24, sig31, sig32, sig59, sig74, sig55);

Y<=sig51;

z<=sig16;

r<=sig29;

end Behavioral;