library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

use IEEE.NUMERIC\_STD.ALL;

entity lab6 is

generic(WIDTH : integer := 32);

port( A : in unsigned(WIDTH-1 downto 0);

X : in unsigned(WIDTH-1 downto 0);

prod : out unsigned(2\*WIDTH-1 downto 0)

);

end lab6;

architecture Behavioral of lab6 is

--MUX signals

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

signal shift\_cnt : unsigned(2 downto 0);

--MUX I/O will be 35 bits when applying Radix 16

signal zeroA, oneA : unsigned((WIDTH+3)-1 downto 0);

signal twoA, threeA, fourA : unsigned((WIDTH+3)-1 downto 0);

signal fiveA, sixA, sevenA : unsigned((WIDTH+3)-1 downto 0);

signal mux\_out : unsigned((WIDTH+3)-1 downto 0);

--Shift and add

signal sum : unsigned((WIDTH+3)-1 downto 0);

signal shift35 : unsigned((WIDTH+3)-1 downto 0);

signal shift32 : unsigned(WIDTH-1 downto 0);

signal bit\_bkt : unsigned(WIDTH-1 downto 0);

signal prod\_concat : unsigned(2\*WIDTH-1 downto 0);

--sel <= (X(2) & X(1) & X(0)) when 2 \* shift\_cnt ~= WIDTH else sel;

begin

mux: process(A,X)

begin

--Initialize MUX inputs

zeroA <= (others => '0');

oneA <= resize(A, (WIDTH+3));

twoA <= resize((A sll 1), (WIDTH+3));

threeA <= twoA + A;

fourA <= resize((A sll 2), (WIDTH+3));

fiveA <= fourA + A;

sixA <= fiveA + A;

sevenA <= sixA + A;

--Aassign select line

if((2\*shift\_cnt) /= WIDTH) then

sel <= X(2) & X(1) & X(0);

shift\_cnt <= shift\_cnt + 1;

else

prod <= prod\_concat;

shift\_cnt <= (others => '0');

end if;

--Observe LSBs of X input using Radix 8

case sel is

--Pass P0

when "000" =>

mux\_out <= zeroA;

--Pass A

when "001" =>

mux\_out <= oneA;

--Pass 2A (shift by 1)

when "010" =>

mux\_out <= twoA;

---Pass 3A (2A + A)

when "011" =>

mux\_out <= threeA;

---Pass 4A (shift by 2)

when "100" =>

mux\_out <= fourA;

---Pass 5A (4A + A)

when "101" =>

mux\_out <= fiveA;

--Pass 6A (5A + A)

when "110" =>

mux\_out <= sixA;

---Pass 7A (6A + A)

when "111" =>

mux\_out <= sevenA;

when others =>

mux\_out <= mux\_out;

end case;

end process;

sum <= mux\_out + shift35;

shift32 <= resize((sum srl 3), 32);

prod\_concat <= shift32 & bit\_bkt;

end Behavioral;

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

use IEEE.NUMERIC\_STD.ALL;

entity tb\_lab6 is

-- Port ( );

end tb\_lab6;

architecture Behavioral of tb\_lab6 is

component lab6 is

generic(WIDTH : integer := 32);

port( A : in unsigned(WIDTH-1 downto 0);

X : in unsigned(WIDTH-1 downto 0);

prod : out unsigned(2\*WIDTH-1 downto 0)

);

end component;

constant WIDTH : integer := 32;

signal A,X : unsigned(WIDTH-1 downto 0);

signal prod : unsigned(2\*WIDTH-1 downto 0);

begin

UUT: lab6

port map(A => A, X => X, prod => prod);

A <= to\_unsigned(16#02F170A6#, A'length);

X <= to\_unsigned(16#000101AD#, A'length);

end Behavioral;