**UP COUNTER**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

use IEEE.STD\_LOGIC\_unsigned.ALL;

use IEEE.numeric\_STD.ALL;

entity upc is

Port ( clk : in STD\_LOGIC;

rst : in STD\_LOGIC;

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

end upc;

architecture Behavioral of upc is

signal counter\_up : STD\_LOGIC\_VECTOR (3 downto 0);

begin

process(clk,rst)

begin

if rising\_edge(clk) then

if (rst='1') then

counter\_up<=x"0";

else

counter\_up<= counter\_up-x"1";

end if;

end if;

end process;

counter<= counter\_up;

end Behavioral;

**TB**

LIBRARY ieee;

USE ieee.std\_logic\_1164.ALL;

ENTITY upc\_tb IS

END upc\_tb;

ARCHITECTURE behavior OF upc\_tb IS

-- Component Declaration for the Unit Under Test (UUT)

COMPONENT upc

PORT(

clk : IN std\_logic;

rst : IN std\_logic;

counter : OUT std\_logic\_vector(3 downto 0)

);

END COMPONENT;

--Inputs

signal clk : std\_logic := '0';

signal rst : std\_logic := '0';

--Outputs

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

-- Clock period definitions

constant clk\_period : time := 10 ns;

BEGIN

-- Instantiate the Unit Under Test (UUT)

uut: upc PORT MAP (

clk => clk,

rst => rst,

counter => counter

);

-- Clock process definitions

clk\_process :process

begin

clk <= '0';

wait for 10 ns;

clk <= '1';

wait for 10 ns;

end process;

-- Stimulus process

stim\_proc: process

begin

rst<='1';

wait for 20 ns;

rst<='0';

wait ;

end process;

END;

**UPDOWN COUNTER**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

use IEEE.NUMERIC\_STD.ALL;

use IEEE.STD\_LOGIC\_unsigned.ALL;

entity up\_down is

Port ( clk : in STD\_LOGIC;

rst : in STD\_LOGIC;

up\_dow : in STD\_LOGIC;

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

end up\_down;

architecture Behavioral of up\_down is

signal counter\_up:STD\_LOGIC\_VECTOR (3 downto 0);

begin

process(clk,rst)

begin

if rising\_edge(clk) then

if (rst='1') then

counter\_up<=x"0";

elsif up\_dow='1' then

counter\_up<= counter\_up-x"1";

else

counter\_up<= counter\_up+x"1";

end if;

end if;

end process;

counter<= counter\_up;

end Behavioral;

**TB**

LIBRARY ieee;

USE ieee.std\_logic\_1164.ALL;

ENTITY up\_down\_tb IS

END up\_down\_tb;

ARCHITECTURE behavior OF up\_down\_tb IS

-- Component Declaration for the Unit Under Test (UUT)

COMPONENT up\_down

PORT(

clk : IN std\_logic;

rst : IN std\_logic;

up\_dow : IN std\_logic;

counter : OUT std\_logic\_vector(3 downto 0)

);

END COMPONENT;

--Inputs

signal clk : std\_logic := '0';

signal rst : std\_logic := '0';

signal up\_dow : std\_logic := '0';

--Outputs

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

-- Clock period definitions

constant clk\_period : time := 10 ns;

BEGIN

-- Instantiate the Unit Under Test (UUT)

uut: up\_down PORT MAP (

clk => clk,

rst => rst,

up\_dow => up\_dow,

counter => counter

);

-- Clock process definitions

clk\_process :process

begin

clk <= '0';

wait for 10 ns;

clk <= '1';

wait for 10 ns;

end process;

-- Stimulus process

stim\_proc: process

begin

rst <= '1';

up\_dow<='0';

wait for 20 ns;

rst <= '0';

wait for 300 ns;

up\_dow<='1';

wait;

end process;

END;

**DECADE COUNTER**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

use ieee.std\_logic\_unsigned.all;

entity decade is

Port ( clk : in STD\_LOGIC;

rst : in STD\_LOGIC;

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

end decade;

architecture Behavioral of decade is

begin

process(clk,rst)

variable temp:std\_logic\_vector(3 downto 0);

begin

if rst='1' then

temp := "0000";

y <= "0000";

elsif clk'event and clk='1' then

if temp < 9 then

temp := temp + 1;

else

temp := "0000";

end if;

y <= temp;

end if;

end process;

end Behavioral;

**TB**

LIBRARY ieee;

USE ieee.std\_logic\_1164.ALL;

ENTITY decade\_tb IS

END decade\_tb;

ARCHITECTURE behavior OF decade\_tb IS

-- Component Declaration for the Unit Under Test (UUT)

COMPONENT decade

PORT(

clk : IN std\_logic;

rst : IN std\_logic;

y : OUT std\_logic\_vector(3 downto 0)

);

END COMPONENT;

--Inputs

signal clk : std\_logic := '0';

signal rst : std\_logic := '0';

--Outputs

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

-- Clock period definitions

constant clk\_period : time := 10 ns;

BEGIN

-- Instantiate the Unit Under Test (UUT)

uut: decade PORT MAP (

clk => clk,

rst => rst,

y => y

);

-- Clock process definitions

clk\_process :process

begin

clk <= '0';

wait for 10 ns;

clk <= '1';

wait for 10 ns;

end process;

stim\_proc: process

begin

-- hold reset state for 100 ns.

rst <= '1';

wait for 20 ns;

rst <= '0';

wait;

end process;

END;

**HALF ADDER USING STRUCTURAL**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity fulladderstru is

Port ( a : in STD\_LOGIC;

b : in STD\_LOGIC;

s : out STD\_LOGIC;

c : out STD\_LOGIC);

end fulladderstru;

architecture Behavioral of fulladderstru is

begin

s<= a xor b;

c<=a and b;

end Behavioral;

///////////orgate/////

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity orgate is

Port ( x : in STD\_LOGIC;

y : in STD\_LOGIC;

z : out STD\_LOGIC);

end orgate;

architecture Behavioral of orgate is

begin

z<=x or y;

end Behavioral;

//////////main adder///

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity mainadder is

Port ( p : in STD\_LOGIC;

q : in STD\_LOGIC;

r : in STD\_LOGIC;

sum : out STD\_LOGIC;

carry : out STD\_LOGIC);

end mainadder;

architecture Behavioral of mainadder is

component fulladderstru is

Port ( a : in STD\_LOGIC;

b : in STD\_LOGIC;

s : out STD\_LOGIC;

c : out STD\_LOGIC);

end component;

component orgate is

Port ( x : in STD\_LOGIC;

y : in STD\_LOGIC;

z : out STD\_LOGIC);

end component;

signal sum1,carry1,carry2:std\_logic;

begin

u0:fulladderstru port map(p,q,sum1,carry1);

u1:fulladderstru port map(sum1,r,sum,carry2);

u2:orgate port map(carry1,carry2,carry);

end Behavioral;

**TB**

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;

ENTITY mainadder\_tb IS

END mainadder\_tb;

ARCHITECTURE behavior OF mainadder\_tb IS

-- Component Declaration for the Unit Under Test (UUT)

COMPONENT mainadder

PORT(

p : IN std\_logic;

q : IN std\_logic;

r : IN std\_logic;

sum : OUT std\_logic;

carry : OUT std\_logic

);

END COMPONENT;

--Inputs

signal p : std\_logic := '0';

signal q : std\_logic := '0';

signal r : std\_logic := '0';

--Outputs

signal sum : std\_logic;

signal carry : std\_logic;

-- No clocks detected in port list. Replace <clock> below with

-- appropriate port name

BEGIN

-- Instantiate the Unit Under Test (UUT)

uut: mainadder PORT MAP (

p => p,

q => q,

r => r,

sum => sum,

carry => carry

);

-- Stimulus process

stim\_proc: process

begin

p<='0';

q<='0';

r<='0';

wait for 10 ns;

p<='0';

q<='0';

r<='1';

wait for 10 ns;

p<='0';

q<='1';

r<='0';

wait for 10 ns;

p<='0';

q<='1';

r<='1';

wait for 10 ns;

p<='1';

q<='0';

r<='0';

wait for 10 ns;

p<='1';

q<='0';

r<='1';

wait for 10 ns;

p<='1';

q<='1';

r<='0';

wait for 10 ns;

p<='1';

q<='1';

r<='1';

wait for 10 ns;

end process;

END;

**DEMUX38**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity decoder38 is

Port ( a : in STD\_LOGIC\_VECTOR (2 downto 0);

e : in STD\_LOGIC;

y : out STD\_LOGIC\_VECTOR (7 downto 0));

end decoder38;

architecture Behavioral of decoder38 is

begin

process(a,e)

begin

if e='1' then

if a="000" then y <= "00000001";

elsif a="001" then y <= "00000010";

elsif a="010" then y <= "00000100";

elsif a="011" then y <= "00001000";

elsif a="100" then y <= "00010000";

elsif a="101" then y <= "00100000";

elsif a="110" then y <= "01000000";

elsif a="111" then y <= "10000000";

else null;

end if;

end if;

end process;

end Behavioral;

**TB**

LIBRARY ieee;

USE ieee.std\_logic\_1164.ALL;

ENTITY decoder38\_tb IS

END decoder38\_tb;

ARCHITECTURE behavior OF decoder38\_tb IS

-- Component Declaration for the Unit Under Test (UUT)

COMPONENT decoder38

PORT(

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

e : IN std\_logic;

y : OUT std\_logic\_vector(7 downto 0)

);

END COMPONENT;

--Inputs

signal a : std\_logic\_vector(2 downto 0) := (others => '0');

signal e : std\_logic := '0';

--Outputs

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

-- No clocks detected in port list. Replace <clock> below with

-- appropriate port name

BEGIN

-- Instantiate the Unit Under Test (UUT)

uut: decoder38 PORT MAP (

a => a,

e => e,

y => y

);

-- Stimulus process

stim\_proc: process

begin

a<="000";

wait for 10 ns;

a<="001";

wait for 10 ns;

a<="010";

wait for 10 ns;

a<="011";

wait for 10 ns;

a<="100";

wait for 10 ns;

a<="101";

wait for 10 ns;

a<="110";

wait for 10 ns;

a<="111";

wait for 10 ns;

end process;

END;

**MUX81**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity mux81 is

Port ( I : in STD\_LOGIC\_VECTOR (7 downto 0);

S : in STD\_LOGIC\_VECTOR (2 downto 0);

Y : out STD\_LOGIC);

end mux81;

architecture Behavioral of mux81 is

begin

with s select

y<= I(0) when "000",

I(1) when "001",

I(2) when "010",

I(3) when "011",

I(4) when "100",

I(5) when "101",

I(6) when "110",

I(7) when others;

end Behavioral;

**TB**

LIBRARY ieee;

USE ieee.std\_logic\_1164.ALL;

ENTITY mux81\_tb IS

END mux81\_tb;

ARCHITECTURE behavior OF mux81\_tb IS

-- Component Declaration for the Unit Under Test (UUT)

COMPONENT mux81

PORT(

I : IN std\_logic\_vector(7 downto 0);

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

Y : OUT std\_logic

);

END COMPONENT;

--Inputs

signal I : std\_logic\_vector(7 downto 0) := (others => '0');

signal S : std\_logic\_vector(2 downto 0) := (others => '0');

--Outputs

signal Y : std\_logic;

-- No clocks detected in port list. Replace <clock> below with

-- appropriate port name

BEGIN

-- Instantiate the Unit Under Test (UUT)

uut: mux81 PORT MAP (

I => I,

S => S,

Y => Y

);

-- Stimulus process

stim\_proc: process

begin

I(0)<='1';

I(1)<='0';

I(2)<='0';

I(3)<='1';

I(4)<='1';

I(5)<='0';

I(6)<='1';

I(7)<='0';

S<="000";

wait for 100 ns;

S<="001";

wait for 100 ns;

S<="010";

wait for 100 ns;

S<="011";

wait for 100 ns;

S<="100";

wait for 100 ns;

S<="101";

wait for 100 ns;

S<="110";

wait for 100 ns;

S<="111";

wait for 100 ns;

wait;

end process;

END;

**FULL ADDER**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity fulladder is

Port ( A : in STD\_LOGIC;

B : in STD\_LOGIC;

Cin : in STD\_LOGIC;

Sum : out STD\_LOGIC;

Carry : out STD\_LOGIC);

end fulladder;

architecture Behavioral of fulladder is

begin

Sum<= A xor B xor Cin;

Carry<= (A and B) or (A and Cin) or (B and Cin);

end Behavioral;

**TB**

LIBRARY ieee;

USE ieee.std\_logic\_1164.ALL;

ENTITY fulladder\_tb IS

END fulladder\_tb;

ARCHITECTURE behavior OF fulladder\_tb IS

-- Component Declaration for the Unit Under Test (UUT)

COMPONENT fulladder

PORT(

A : IN std\_logic;

B : IN std\_logic;

Cin : IN std\_logic;

Sum : OUT std\_logic;

Carry : OUT std\_logic

);

END COMPONENT;

--Inputs

signal A : std\_logic := '0';

signal B : std\_logic := '0';

signal Cin : std\_logic := '0';

--Outputs

signal Sum : std\_logic;

signal Carry : std\_logic;

-- No clocks detected in port list. Replace <clock> below with

-- appropriate port name

BEGIN

-- Instantiate the Unit Under Test (UUT)

uut: fulladder PORT MAP (

A => A,

B => B,

Cin => Cin,

Sum => Sum,

Carry => Carry

);

-- Stimulus process

stim\_proc: process

begin

A<='0';

B<='0';

cin<='0';

wait for 10 ns;

A<='0';

B<='0';

cin<='1';

wait for 10 ns;

A<='0';

B<='1';

cin<='0';

wait for 10 ns;

A<='0';

B<='1';

cin<='1';

wait for 10 ns;

A<='1';

B<='0';

cin<='0';

wait for 10 ns;

A<='1';

B<='0';

cin<='1';

wait for 10 ns;

A<='1';

B<='1';

cin<='0';

wait for 10 ns;

A<='1';

B<='1';

cin<='1';

wait for 10 ns;

end process;

END;

**FULL SUB**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity fullsub is

Port ( A : in STD\_LOGIC;

B : in STD\_LOGIC;

Cin : in STD\_LOGIC;

diff : out STD\_LOGIC;

bor : out STD\_LOGIC);

end fullsub;

architecture Behavioral of fullsub is

begin

diff<= A xor B xor Cin;

bor<= (not(A)and(B or Cin)) or(B and Cin);

end Behavioral;

**TB**

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;

ENTITY fullsub\_tb IS

END fullsub\_tb;

ARCHITECTURE behavior OF fullsub\_tb IS

-- Component Declaration for the Unit Under Test (UUT)

COMPONENT fullsub

PORT(

A : IN std\_logic;

B : IN std\_logic;

Cin : IN std\_logic;

diff : OUT std\_logic;

bor : OUT std\_logic

);

END COMPONENT;

--Inputs

signal A : std\_logic := '0';

signal B : std\_logic := '0';

signal Cin : std\_logic := '0';

--Outputs

signal diff : std\_logic;

signal bor : std\_logic;

-- No clocks detected in port list. Replace <clock> below with

-- appropriate port name

BEGIN

-- Instantiate the Unit Under Test (UUT)

uut: fullsub PORT MAP (

A => A,

B => B,

Cin => Cin,

diff => diff,

bor => bor

);

-- Stimulus process

stim\_proc: process

begin

A<='0';

B<='0';

cin<='0';

wait for 10 ns;

A<='0';

B<='0';

cin<='1';

wait for 10 ns;

A<='0';

B<='1';

cin<='0';

wait for 10 ns;

A<='0';

B<='1';

cin<='1';

wait for 10 ns;

A<='1';

B<='0';

cin<='0';

wait for 10 ns;

A<='1';

B<='0';

cin<='1';

wait for 10 ns;

A<='1';

B<='1';

cin<='0';

wait for 10 ns;

A<='1';

B<='1';

cin<='1';

wait for 10 ns;

end process;

END;

**BINARY ADDER**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity binaryadder is

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

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

Cin : in STD\_LOGIC;

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

Carryout: out STD\_LOGIC);

end binaryadder;

architecture Behavioral of binaryadder is

--declare signals

signal temp1,temp2,temp3:std\_logic;

--write components

component fa is

port(

a,b,c:in std\_logic;

co:out std\_logic;

s:out std\_logic

);

end component;

begin

fa0:fa port map (a=>a(0),b=>b(0),c=>Cin,co=>temp1,s=>s(0));

fa1:fa port map (a=>a(1),b=>b(1),c=>temp1,co=>temp2,s=>s(1));

fa2:fa port map (a=>a(2),b=>b(2),c=>temp2,co=>temp3,s=>s(2));

fa3:fa port map (a=>a(3),b=>b(3),c=>temp3,co=>Carryout,s=>s(3));

end Behavioral;

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

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity fa is

port(

a,b,c:in std\_logic;

co:out std\_logic;

s:out std\_logic

);

end fa;

architecture fa\_arch of fa is

begin

s<=a xor b xor c;

co<=(a and b) or (c and (A or B));--(a and b)or(a and c)or(b and c);

end fa\_arch;

**TB**

LIBRARY ieee;

USE ieee.std\_logic\_1164.ALL;

ENTITY binaryadder\_tb IS

END binaryadder\_tb;

ARCHITECTURE behavior OF binaryadder\_tb IS

-- Component Declaration for the Unit Under Test (UUT)

COMPONENT binaryadder

PORT(

A : IN std\_logic\_vector(3 downto 0);

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

Cin : IN std\_logic;

S : OUT std\_logic\_vector(3 downto 0);

Carryout : OUT std\_logic

);

END COMPONENT;

--Inputs

signal A : std\_logic\_vector(3 downto 0) := (others => '0');

signal B : std\_logic\_vector(3 downto 0) := (others => '0');

signal Cin : std\_logic := '0';

--Outputs

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

signal Carryout : std\_logic;

BEGIN

-- Instantiate the Unit Under Test (UUT)

uut: binaryadder PORT MAP (

A => A,

B => B,

Cin => Cin,

S => S,

Carryout => Carryout

);

-- Stimulus process

stim\_proc\_A: process

begin

A<="0100";

wait for 100 ns;

A<="0111";

wait for 100 ns;

end process;

stim\_proc\_B: process

begin

B<="1111";

wait for 100 ns;

B<="0011";

wait for 100 ns;

end process;

stim\_proc\_Cin: process

begin

Cin<='0';

wait for 100 ns;

wait;

end process;

END;