**Alu with no feedback**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

use IEEE.std\_logic\_unsigned.all;

entity AluFB is

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

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

sel: in std\_logic\_vector(1 downto 0);

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

end AluFB;

architecture bev of AluFB is

begin

process (a,b,sel)

begin

case sel is

when "00" => -- addition

output <= a + b;

when "01" => --subtract

output <= a - b;

when "10" => --AND

output <= a and b;

when "11" => --OR

output <= a or b;

when others=>

Null;

end case;

end process;

end;

**no feedback fsm**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

use IEEE.std\_logic\_unsigned.all;

entity Fsm is

port (reset, clk: in std\_logic;

--x: in std\_logic;

output: out std\_logic\_vector(1 downto 0));

end;

Architecture beh of Fsm is

signal cs, ns: std\_logic\_vector(1 downto 0);

begin

process(clk,reset)

begin

if(reset ='1') then

cs<="00";

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

cs<=ns;

end if;

end process;

--this process will loop through all the states

process ( cs)

begin

case cs is

when ("00") =>-- state 0

output <= "00";

ns<= ("01");

when "01" =>-- state 1

output <= "01";

ns<= "10";

when "10" =>--state 2

output <= "10";

ns<= "11";

when "11" =>-- state 3

output <= "11";

ns<="00";

when others =>

Null;

end case;

end process;

end;

**Top level of no feedback**

library ieee;

use ieee. std\_logic\_1164.all;

use ieee. std\_logic\_arith.all;

use ieee. std\_logic\_unsigned.all;

entity TopLevelWO is--Without Feedback

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

reset,clk: in std\_logic;

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

end;

architecture Beh of TopLevelWO is

signal temp : std\_logic\_vector(1 downto 0);

--ALU Component without feed back

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

sel : in std\_logic\_vector(1 downto 0);

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

end component;

--FSM Component without feed back

component Fsm port(reset, clk: in std\_logic;

output : out std\_logic\_vector(1 downto 0));

end component;

begin

--Component Instantiation will be complete here

Control: Fsm port map(reset=> reset,clk => clk, output=> temp);

Bod : ALUFB port map(a=> a, b=> b, sel=>temp, output => output);

end;

**Testbench feedback**

library ieee;

use ieee.std\_logic\_1164.all;

use ieee.std\_logic\_unsigned.all;

use ieee.numeric\_std.all;

entity TestBenchAluwofb is--No Feedback

end TestBenchAluwofb;

architecture beh of TestBenchAluwofb is

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

signal reset,clk: std\_logic;

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

-- Component Decleration of TopLevel

component TopLevel is

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

reset,clk : in std\_logic;

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

end component;

begin

--Component Instatiation

Testaluwofb : TopLevel port map(a=>a, b=>b, reset=>reset, clk=>clk, output=>output );

clock : process--clock process

begin

clk <= '0';

wait for 15 ns;

clk <= '1';

wait for 15 ns;

end process;

simulation : process

begin

--values for a and b

a <= "1001";

b <= "0111";

-- the reset will allow for different cases

reset <= '1';

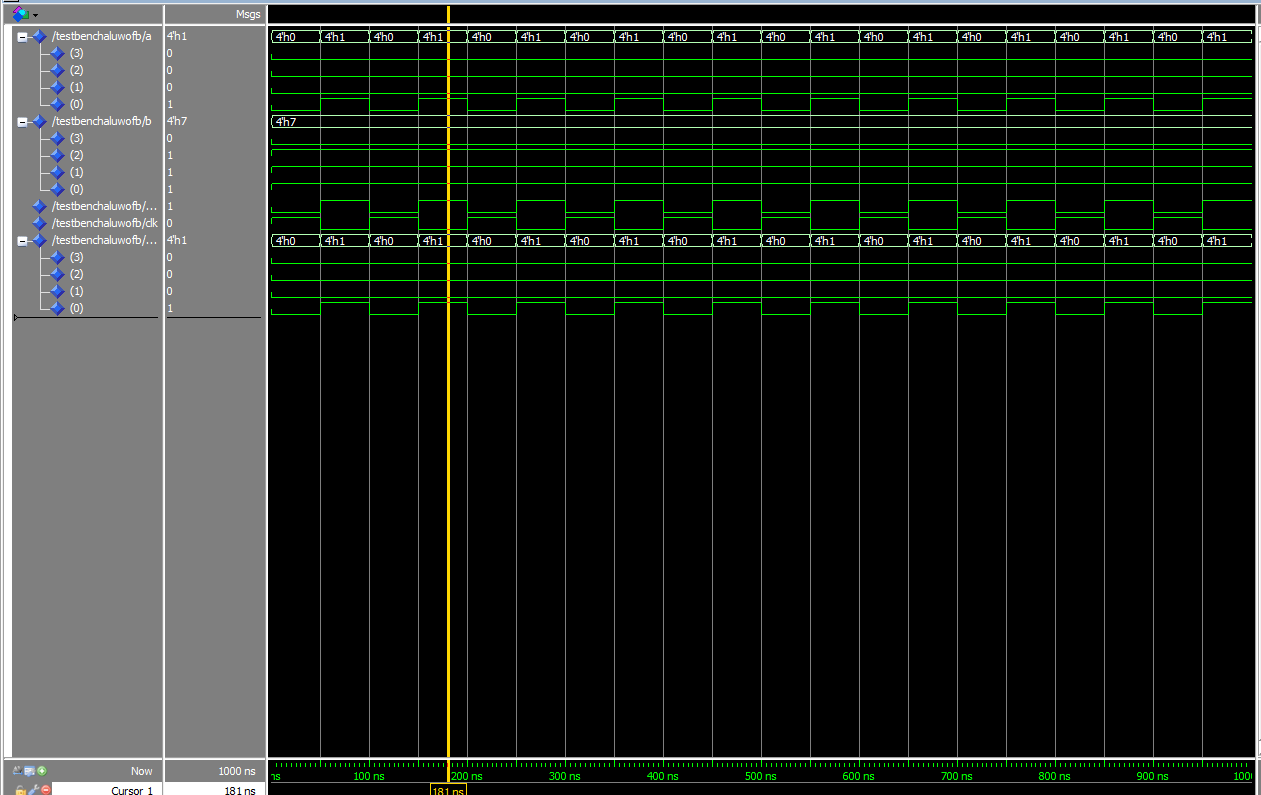
wait for 10 ns;

reset <= '0';

wait for 200 ns;

end process;

end;



**ALU Feedback**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

use IEEE.std\_logic\_unsigned.all;

entity AluFBW is

--declaring everything i will use in my problem

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

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

sel: in std\_logic\_vector(1 downto 0);

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

end;

architecture bev of AluFBW is

begin

process (a,b,sel)

begin

case sel is

when "00" => -- addition

output <= a + b;

when "01" => --subtract

output <= a - b;

when "10" => --AND

output <= a and b;

when "11" => --OR

output <= a or b;

when others=>-- if anything else give null

Null;

end case;

end process;

end;

**Fsm with feedback**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

use IEEE.std\_logic\_unsigned.all;

entity Fsmfb is

port (reset, clk: in std\_logic;

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

output: out std\_logic\_vector(1 downto 0));

end;

Architecture beh of Fsmfb is

signal cs, ns: std\_logic\_vector(1 downto 0);

begin

process(clk,reset)

begin

if(reset ='1') then

cs<="00";

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

cs<=ns;

end if;

end process;

process (cs,x)

--this process will loop through all the states

begin

case cs is

when ("00")=> --State 0

if(x="0001") then

output <= "01";

ns<="01";

else

output<="00";

ns<=cs;

end if;

when ("01")=> --State 1

if(x="0001") then

output <= "10";

ns<="10";

else

output<="01";

ns<=cs;

end if;

when ("10")=> --State 2

if(x="0001") then

output <= "11";

ns<="11";

else

output<="10";

ns<=cs;

end if;

when ("11")=> --State 3

if(x="0001") then

output <= "00";

ns<="00";

else

output<="11";

ns<=cs;

end if;

when others=> Null;

end case;

end process;

end;

**Toplevel with feedback**

library ieee;

use ieee. std\_logic\_1164.all;

use ieee. std\_logic\_arith.all;

use ieee. std\_logic\_unsigned.all;

entity TopLevelW is--With Feedback

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

reset,clk: in std\_logic;

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

end;

architecture Beh of TopLevelW is

signal temp : std\_logic\_vector(1 downto 0);

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

--ALU Component with feed back

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

sel : in std\_logic\_vector(1 downto 0);

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

end component;

--FSM Component with feed back

component FsmW port(reset, clk: in std\_logic;

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

output : out std\_logic\_vector(1 downto 0));

end component;

begin

--Component Instantiation will be complete here

Control2: FsmW port map(reset=> reset,clk => clk, x=>x, output=> temp);

Bod2 : ALUFBW port map(a=> a, b=> b, sel=>temp, output => output);

end;

**Testbench with feedback**

library ieee;

use ieee.std\_logic\_1164.all;

use ieee.std\_logic\_unsigned.all;

use ieee.numeric\_std.all;

entity TestBenchAluwfb is--No Feedback

end TestBenchAluwfb;

architecture beh of TestBenchAluwfb is

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

signal reset,clk: std\_logic;

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

-- Component Decleration of TopLevel

component TopLevel2 is

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

reset,clk : in std\_logic;

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

end component;

begin

--Component Instatiation

Testaluwofb : TopLevel2 port map(a=>a, b=>b,x=>x, reset=>reset, clk=>clk, output=>output );

clock : process--clock process

begin

CLK <= '0';

wait for 15 ns;

CLK <= '1';

wait for 15 ns;

end process;

simulation : process

begin

--values for a and b

a <= "1001";

b <= "1111";

x<="0001";

-- the reset will allow for different cases

RESET <= '1';

wait for 10 ns;

RESET <= '0';

wait for 200 ns;

end process;

end;

