**FIFO**

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

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 fifo is

Port ( clk : in STD\_LOGIC;

rst : in STD\_LOGIC;

w : in STD\_LOGIC;

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

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

red : out STD\_LOGIC);

end fifo;

architecture Behavioral of fifo is

signal wptr,rptr:std\_logic\_vector(3 downto 0);

type fifomem is array (15 downto 0) of std\_logic\_vector(7 downto 0);

signal mem:fifomem;

begin

process(clk,rst,din,w)

begin

if rst='1' then

wptr<="0000"; rptr<="0000"; red<='0';

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

if(w='1') then

if(wptr<"1111") then

mem (conv\_integer(wptr))<=din;

wptr<=wptr+1;

else red<='1';

end if;

else

if (rptr<wptr) then

dout <= mem(conv\_integer(rptr));

rptr<=rptr+1;

else red<='1';

end if;

end if;

end if;

end process;

end Behavioral;

**Testbench**

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

LIBRARY ieee;

USE ieee.std\_logic\_1164.ALL;

ENTITY fifo\_tb IS

END fifo\_tb;

ARCHITECTURE behavior OF fifo\_tb IS

COMPONENT fifo

PORT(

clk : IN std\_logic;

rst : IN std\_logic;

w : IN std\_logic;

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

dout : OUT std\_logic\_vector(7 downto 0);

red : OUT std\_logic

);

END COMPONENT;

--Inputs

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

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

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

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

--Outputs

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

signal red : std\_logic;

-- Clock period definitions

constant clk\_period : time := 10 ns;

BEGIN

-- Instantiate the Unit Under Test (UUT)

uut: fifo PORT MAP (

clk => clk,

rst => rst,

w => w,

din => din,

dout => dout,

red => red

);

-- Clock process definitions

clk\_process :process

begin

clk <= '0';

wait for clk\_period/2;

clk <= '1';

wait for clk\_period/2;

end process;

-- Stimulus process

stim\_proc: process

begin

rst <= '1';

wait for 100 ns;

rst <= '0';

w <='1';

din <= "01010011";

wait for 10 ns;

din <= "01010101";

wait for 10 ns;

din <= "01011111";

wait for 10 ns;

din <= "01010000";

wait for 10 ns;

din <= "00010011";

wait for 10 ns;

w<='0';

wait for 100 ns;

din <= "01010011";

wait for 10 ns;

din <= "01010101";

wait for 10 ns;

din <= "01011111";

wait for 10 ns;

din <= "01010000";

wait for 10 ns;

din <= "00010011";

wait for 10 ns;

din <= "01010011";

wait for 10 ns;

din <= "01010101";

wait for 10 ns;

din <= "01011111";

wait for 10 ns;

din <= "01010000";

wait for 10 ns;

din <= "00010011";

wait for 10 ns;

din <= "00010011";

wait for 10 ns;

wait for clk\_period\*10;

-- insert stimulus here

wait;

end process;

END;







