# SHIFT REGISTER :

## VHDL Code –

-- Company:

-- Engineer:

--

-- Create Date: 13:25:11 11/6/2023

-- Design Name:

-- Module Name: shiftreg - Behavioral

-- Project Name:

-- Target Devices:

-- Tool versions:

-- Description:

--

-- Dependencies:

--

-- Revision:

-- Revision 0.01 - File Created

-- Additional Comments:

--

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 shiftreg is

Port ( S : in STD\_LOGIC\_VECTOR (1 downto 0); CLK : in STD\_LOGIC;

IL : in STD\_LOGIC; IR : in STD\_LOGIC;

Q : out STD\_LOGIC\_VECTOR (3 downto 0); I : in STD\_LOGIC\_VECTOR (3 downto 0));

end shiftreg;

architecture Behavioral of shiftreg is

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

begin process(CLK) begin

if rising\_edge(CLK)then case S is

when "00"=> Q<=qtemp;

when "01"=>

Q(3 downto 0)<=I(3 downto 0);

when "10"=>

qtemp(3 downto 1)<=qtemp(2 downto 0); qtemp(0)<=IR;

when "11"=>

qtemp(2 downto 0)<=qtemp(3 downto 1); qtemp(3)<=IL;

when others =>NULL;

end case;

end if; Q<=qtemp; end process;

end Behavioral;

## Testbench -

-- Company:

-- Engineer:

--

-- Create Date: 13:45:59 11/6/2023

-- Design Name: shiftreg

-- Module Name: C:/shiftreg/shifter\_vhd.vhd

-- Project Name: shiftreg

-- Target Device:

-- Tool versions:

-- Description:

--

-- VHDL Test Bench Created by ISE for module: shiftreg

--

-- Dependencies:

--

-- Revision:

-- Revision 0.01 - File Created

-- Additional Comments:

--

-- Notes:

-- This testbench has been automatically generated using types std\_logic and

-- std\_logic\_vector for the ports of the unit under test. Xilinx recommends

-- that these types always be used for the top-level I/O of a design in order

-- to guarantee that the testbench will bind correctly to the post-implementation

-- simulation model.

LIBRARY ieee;

USE ieee.std\_logic\_1164.ALL; USE ieee.std\_logic\_unsigned.all; USE ieee.numeric\_std.ALL;

ENTITY shifter\_vhd\_vhd IS END shifter\_vhd\_vhd;

ARCHITECTURE behavior OF shifter\_vhd\_vhd IS

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

PORT(

S : IN std\_logic\_vector(1 downto 0); CLK : IN std\_logic;

IL : IN std\_logic; IR : IN std\_logic;

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

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

);

END COMPONENT;

--Inputs

SIGNAL CLK : std\_logic := '1'; SIGNAL IL : std\_logic := '0'; SIGNAL IR : std\_logic := '0';

SIGNAL S : std\_logic\_vector(1 downto 0) := (others=>'0'); SIGNAL I : std\_logic\_vector(3 downto 0) := (others=>'0');

--Outputs

SIGNAL Q : std\_logic\_vector(3 downto 0);

BEGIN

-- Instantiate the Unit Under Test (UUT) uut: shiftreg PORT MAP(

S => S,

CLK => CLK, IL => IL,

IR => IR, Q => Q, I => I

);

tb : PROCESS BEGIN

-- Wait 100 ns for global reset to finish

--wait for 100 ns;

s<="00"; I<="1101";

wait for 100 ns;

s<="01"; I<="1101";

wait for 100 ns;

s<="10"; I<="1101";

wait for 100 ns;

s<="11"; I<="1101";

wait for 100 ns;

-- Place stimulus here

--wait; -- will wait forever END PROCESS;

END;

# FIFO MEMORY :

## VHDL Code -

-- Company:

-- Engineer:

--

-- Create Date: 10:50:22 11/6/2023

-- Design Name:

-- Module Name: fifo - Behavioral

-- Project Name:

-- Target Devices:

-- Tool versions:

-- Description:

--

-- Dependencies:

--

-- Revision:

-- Revision 0.01 - File Created

-- Additional Comments:

--

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;

---- Uncomment the following library declaration if instantiating

---- any Xilinx primitives in this code.

--library UNISIM;

--use UNISIM.VComponents.all;

entity fifo is Generic (

constant DATA\_WIDTH : positive := 4; constant FIFO\_DEPTH : positive := 4);

port ( clk : in STD\_LOGIC; reset : in STD\_LOGIC; enr : in STD\_LOGIC; enw : in STD\_LOGIC;

data\_in : in STD\_LOGIC\_VECTOR (1 downto 0); data\_out : out STD\_LOGIC\_VECTOR (1 downto 0); fifo\_empty : out STD\_LOGIC;

fifo\_full : out STD\_LOGIC); end fifo;

architecture Behavioral of fifo is

type memory\_type is array(0 to FIFO\_DEPTH-1) of std\_logic\_vector(DATA\_WIDTH-1 downto 0); signal memory : memory\_type :=(others=>(others =>'0'));

signal readptr,writeptr : integer :=0; signal empty ,full: std\_logic :='0'; begin

fifo\_empty <= empty; fifo\_full<=full;

process(clk,reset)

variable num\_elem : integer :=0;

begin

if(reset = '1') then

memory <=(others =>(others =>'0')); data\_out<=(others =>'0');

empty <='0'; full<='0'; readptr<=0; writeptr<=0; num\_elem := 0;

elsif(clk'event and clk ='1') then if(enr ='1' and empty ='0')then data\_out<=memory(readptr); num\_elem:=num\_elem-1;

end if;

if (enw ='1' and full ='0') then memory(writeptr)<=data\_in; writeptr<=writeptr +1; num\_elem :=num\_elem+1; end if;

if(readptr = FIFO\_DEPTH) then readptr <= 0;

end if;

if(writeptr = FIFO\_DEPTH) then writeptr <= 0;

end if;

if(num\_elem = 0) then empty <='1';

else

empty <='0'; end if;

if(num\_elem = FIFO\_DEPTH ) then full <='1';

else

full <='0'; end if; end if;

end process; end Behavioral;

## FIFO Testbench -

-- Company:

-- Engineer:

--

-- Create Date: 12:11:25 11/6/2023

-- Design Name: fifo

-- Module Name: C:/fifo/fifo\_tb.vhd

-- Project Name: fifo

-- Target Device:

-- Tool versions:

-- Description:

--

-- VHDL Test Bench Created by ISE for module: fifo

--

-- Dependencies:

--

-- Revision:

-- Revision 0.01 - File Created

-- Additional Comments:

--

-- Notes:

-- This testbench has been automatically generated using types std\_logic and

-- std\_logic\_vector for the ports of the unit under test. Xilinx recommends

-- that these types always be used for the top-level I/O of a design in order

-- to guarantee that the testbench will bind correctly to the post-implementation

-- simulation model.

LIBRARY ieee;

USE ieee.std\_logic\_1164.ALL; USE ieee.std\_logic\_arith.ALL;

ENTITY fifo\_tb\_vhd IS END fifo\_tb\_vhd;

ARCHITECTURE behavior OF fifo\_tb\_vhd IS

--Inputs and outputs

constant DATA\_WIDTH : positive := 4; constant FIFO\_DEPTH : positive := 4 ;

signal Clk,reset,enr,enw,empty,full : std\_logic := '0';

signal data\_in,data\_out : std\_logic\_vector(DATA\_WIDTH - 1 downto 0) := (others => '0');

--temporary signals signal i : integer := 0;

-- Clock period definitions

constant Clk\_period : time := 10 ns;

BEGIN

UUT : entity work.fifo GENERIC MAP (

DATA\_WIDTH => DATA\_WIDTH, FIFO\_DEPTH => FIFO\_DEPTH) PORT MAP (

clk => clk, reset => reset, enr => enr, enw => enw,

data\_in => data\_in, data\_out => data\_out, fifo\_empty => empty, fifo\_full => full

);

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

reset <= '1'; --apply reset for one clock cycle. wait for clk\_period;

reset <= '0';

wait for clk\_period\*3; --wait for 3 clock periods(simply) enw <= '1'; enr<= '0'; --write 4 values to fifo.

for i in 1 to 4 loop

Data\_In <= conv\_std\_logic\_vector(i,DATA\_WIDTH); wait for clk\_period;

end loop;

enw <= '0'; enr <= '1'; --read 4 values from fifo.

wait for clk\_period\*4; enw <= '0'; enr <= '0';

wait for clk\_period\*10; --wait for some clock cycles.

wait; -- will wait forever END PROCESS;

END;

# LCD

## VHDL Code -

-- Company:

-- Engineer:

--

-- Create Date: 11:38:38 11/6/2023

-- Design Name:

-- Module Name: lcd - Behavioral

-- Project Name:

-- Target Devices:

-- Tool versions:

-- Description:

--

-- Dependencies:

--

-- Revision:

-- Revision 0.01 - File Created

-- Additional Comments:

--

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 lcd is

Port ( rst : in STD\_LOGIC; clk\_12Mhz : in STD\_LOGIC; lcd\_rs : out STD\_LOGIC; lcd\_en : out STD\_LOGIC;

lcd\_data : out STD\_LOGIC\_VECTOR (7 downto 0)); end lcd;

architecture Behavioral of lcd is

signal div : std\_logic\_vector(20 downto 0); --- delay timer 1 signal clk\_fsm,lcd\_rs\_s: std\_logic;

-- LCD controller FSM states

type state is (reset,func,mode,cur,clear,d0,d1,d2,d3,d4,hold,start); signal ps,nx : state;

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

begin

process(rst,clk\_12Mhz)

begin

if(rst = '1')then

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

elsif( clk\_12Mhz' event and clk\_12Mhz ='1')then div <= div + 1;

end if;

end process;

clk\_fsm <= div(15);

----- Presetn state Register -----------------------

process(rst,clk\_fsm)

begin

if(rst = '1')then ps <= reset;

elsif(clk\_fsm' event and clk\_fsm ='1')then ps <= nx;

end if;

end process;

- ----state and output decoding process process(ps)

begin case(ps) is when reset =>

nx <= func; lcd\_rs\_s <= '0';

dataout\_s <= "00111000"; -- 38h

when func =>

nx <= mode; lcd\_rs\_s <= '0';

dataout\_s <= "00111000"; -- 38h

when mode =>

nx <= cur; lcd\_rs\_s <= '0';

dataout\_s <= "00000110"; -- 06h

when cur =>

nx <= clear; lcd\_rs\_s <= '0';

dataout\_s <= "00001100";

when clear=>

nx <= start; lcd\_rs\_s <= '0';

dataout\_s <= "00000001"; -- 01h

when start=>

nx <= d0; lcd\_rs\_s <= '0';

dataout\_s <="11000100"; -- c4

when d0 =>

lcd\_rs\_s <= '1';

dataout\_s <= "01000001"; -- A

nx <= d1;

when d1 =>

lcd\_rs\_s <= '1';

dataout\_s <= "01000010"; -- B nx <= d2;

when d2 =>

lcd\_rs\_s <= '1';

dataout\_s <= "01000011"; -- C nx <= d3;

when d3 =>

lcd\_rs\_s <= '1';

dataout\_s <="01000100"; -- D

nx <= d4;

when d4 =>

lcd\_rs\_s <= '1';

dataout\_s <="00100000"; -- space nx <= hold;

when hold =>

lcd\_rs\_s <= '0';

dataout\_s<= "00000000"; -- hold nx <= hold;

when others=>

nx <= reset; lcd\_rs\_s <= '0';

dataout\_s<="00000001"; -- CLEAR

end case; end process;

lcd\_en <= clk\_fsm; lcd\_rs <= lcd\_rs\_s; lcd\_data <= dataout\_s;

end Behavioral;

## Testbench Code -

-- Company:

-- Engineer:

--

-- Create Date: 12:32:34 11/6/2023

-- Design Name: lcd

-- Module Name: C:/lcd/lcd\_tb.vhd

-- Project Name: lcd

-- Target Device:

-- Tool versions:

-- Description:

--

-- VHDL Test Bench Created by ISE for module: lcd

--

-- Dependencies:

--

-- Revision:

-- Revision 0.01 - File Created

-- Additional Comments:

--

-- Notes:

-- This testbench has been automatically generated using types std\_logic and

-- std\_logic\_vector for the ports of the unit under test. Xilinx recommends

-- that these types always be used for the top-level I/O of a design in order

-- to guarantee that the testbench will bind correctly to the post-implementation

-- simulation model.

LIBRARY ieee;

USE ieee.std\_logic\_1164.ALL; USE ieee.std\_logic\_unsigned.all; USE ieee.numeric\_std.ALL;

ENTITY lcd\_tb\_vhd IS END lcd\_tb\_vhd;

ARCHITECTURE behavior OF lcd\_tb\_vhd IS

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

PORT(

rst : IN std\_logic; clk\_12Mhz : IN std\_logic; lcd\_rs : OUT std\_logic; lcd\_en : OUT std\_logic;

lcd\_data : OUT std\_logic\_vector(7 downto 0)

);

END COMPONENT;

--Inputs

SIGNAL rst : std\_logic := '1';

SIGNAL clk\_12Mhz : std\_logic := '1';

--Outputs

SIGNAL lcd\_rs : std\_logic; SIGNAL lcd\_en : std\_logic;

SIGNAL lcd\_data : std\_logic\_vector(7 downto 0); constant clk\_12Mhz\_period : time :=10 ns;

BEGIN

-- Instantiate the Unit Under Test (UUT) uut: lcd PORT MAP(

rst => rst,

clk\_12Mhz => clk\_12Mhz, lcd\_rs => lcd\_rs,

lcd\_en => lcd\_en, lcd\_data => lcd\_data

);

clk\_12Mhz\_process : PROCESS BEGIN

clk\_12Mhz <='0';

wait for clk\_12Mhz\_period/2; clk\_12Mhz <='1';

wait for clk\_12Mhz\_period/2; end process ;

stim\_proc:process begin

rst <='1';

wait for 20 ns;

rst <='0';

wait; -- will wait forever END PROCESS;

END;









