Lab 6 - M. D. B. Perera - 210465P

**Assigned Task:**

* Designing a 4-Bit Storage Register.
* Designing a 4-Bit Arithmetic Unit.

**VHDL Codes:**

1. StorageRegister.vhd

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity StorageRegister is

Port (

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

enableSignal : in STD\_LOGIC;

clockSignal : in STD\_LOGIC;

storedWord : out STD\_LOGIC\_VECTOR (3 downto 0)

);

end StorageRegister

architecture Behavioral of StorageRegister is begin

process (clockSignal) begin

if (rising\_edge(clockSignal)) then

if enableSignal = '1' then

storedWord <= inputWord;

end if;

end if;

end process;

end Behavioral;

1. ArithmeticUnit.vhd

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity ArithmeticUnit is

Port (

inputWord : in STD\_LOGIC\_VECTOR (3 downto 0); -- Binary word to be stored in a register

registerSelect : in STD\_LOGIC; -- Selector input to choose a register to store the binary word

clockSignal : in STD\_LOGIC; -- Clock signal

sum : out STD\_LOGIC\_VECTOR (3 downto 0); -- Result from the arithmetic unit

zeroFlag : out STD\_LOGIC; -- "Is the result zero?"

carryFlag : out STD\_LOGIC -- "Is there a carry bit?"

);

end ArithmeticUnit;

architecture Behavioral of ArithmeticUnit is

component RCA

Port (

firstWord, secondWord : in STD\_LOGIC\_VECTOR (3 downto 0);

carryIn : in STD\_LOGIC;

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

carryOut : out STD\_LOGIC

);

end component;

component Slow\_Clk

Port (

Clk\_in : in STD\_LOGIC;

Clk\_out : out STD\_LOGIC := '0'

);

end component;

component StorageRegister

Port (

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

enableSignal : in STD\_LOGIC;

clockSignal : in STD\_LOGIC;

storedWord : out STD\_LOGIC\_VECTOR (3 downto 0)

);

end component;

signal RegisterA\_out, RegisterB\_out, sum\_out : STD\_LOGIC\_VECTOR (3 downto 0);

signal enableA, Clk\_out, carry\_out : STD\_LOGIC;

begin

enableA <= NOT registerSelect;

Clock : Slow\_Clk

port map (

Clk\_in => clockSignal,

Clk\_out => Clk\_out

);

RegisterA : StorageRegister

port map (

inputWord => inputWord,

enableSignal => enableA,

clockSignal => Clk\_out,

storedWord => RegisterA\_out

);

RegisterB : StorageRegister

port map (

inputWord => inputWord,

enableSignal => registerSelect,

clockSignal => Clk\_out,

storedWord => RegisterB\_out

);

Adder : RCA

port map (

firstWord => RegisterA\_out,

secondWord => RegisterB\_out,

carryIn => '0',

sum => sum\_out,

carryOut => carry\_out

);

process (sum\_out) begin

if sum\_out = "0000" then

zeroFlag <= '1';

else

zeroFlag <= '0';

end if;

end process;

sum <= Sum\_out;

carryFlag <= carry\_out;

end Behavioral;

1. ArithmeticUnit\_Simulation.vhd

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

architecture Behavioral of ArithmeticUnit\_Simulation is

component ArithmeticUnit

Port (

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

clockSignal, registerSelect : in STD\_LOGIC;

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

carryFlag, zeroFlag : out STD\_LOGIC

);

end component;

signal inputWord, sum : STD\_LOGIC\_VECTOR (3 downto 0);

signal registerSelect, clockSignal, zeroFlag, carryFlag : STD\_LOGIC;

begin

UUT : ArithmeticUnit port map (

inputWord => inputWord,

clockSignal => clockSignal,

registerSelect => registerSelect,

sum => sum,

carryFlag => carryFlag,

zeroFlag => zeroFlag

);

process begin

registerSelect <= '0';

inputWord <= "0000";

for i in 0 to 9 loop

clockSignal <= '1';

wait for 5 ns;

clockSignal <= '0';

wait for 5 ns;

end loop;

registerSelect <= '1';

inputWord <= "0000";

for i in 0 to 9 loop

clockSignal <= '1';

wait for 5 ns;

clockSignal <= '0';

wait for 5 ns;

end loop;

registerSelect <= '0';

inputWord <= "0001";

for i in 0 to 9 loop

clockSignal <= '1';

wait for 5 ns;

clockSignal <= '0';

wait for 5 ns;

end loop;

registerSelect <= '1';

inputWord <= "0010";

for i in 0 to 9 loop

clockSignal <= '1';

wait for 5 ns;

clockSignal <= '0';

wait for 5 ns;

end loop;

registerSelect <= '0';

inputWord <= "0110";

for i in 0 to 9 loop

clockSignal <= '1';

wait for 5 ns;

clockSignal <= '0';

wait for 5 ns;

end loop;

registerSelect <= '1';

inputWord <= "0011";

for i in 0 to 9 loop

clockSignal <= '1';

wait for 5 ns;

clockSignal <= '0';

wait for 5 ns;

end loop;

registerSelect <= '0';

inputWord <= "1101";

for i in 0 to 9 loop

clockSignal <= '1';

wait for 5 ns;

clockSignal <= '0';

wait for 5 ns;

end loop;

registerSelect <= '1';

inputWord <= "1100";

for i in 0 to 9 loop

clockSignal <= '1';

wait for 5 ns;

clockSignal <= '0';

wait for 5 ns;

end loop;

registerSelect <= '0';

inputWord <= "1010";

for i in 0 to 9 loop

clockSignal <= '1';

wait for 5 ns;

clockSignal <= '0';

wait for 5 ns;

end loop;

registerSelect <= '1';

inputWord <= "0111";

for i in 0 to 9 loop

clockSignal <= '1';

wait for 5 ns;

clockSignal <= '0';

wait for 5 ns;

end loop;

registerSelect <= '0';

inputWord <= "1100";

for i in 0 to 9 loop

clockSignal <= '1';

wait for 5 ns;

clockSignal <= '0';

wait for 5 ns;

end loop;

registerSelect <= '1';

inputWord <= "0110";

for i in 0 to 9 loop

clockSignal <= '1';

wait for 5 ns;

clockSignal <= '0';

wait for 5 ns;

end loop;

registerSelect <= '0';

inputWord <= "1111";

for i in 0 to 9 loop

clockSignal <= '1';

wait for 5 ns;

clockSignal <= '0';

wait for 5 ns;

end loop;

registerSelect <= '1';

inputWord <= "0101";

for i in 0 to 9 loop

clockSignal <= '1';

wait for 5 ns;

clockSignal <= '0';

wait for 5 ns;

end loop;

wait;

end process;

end Behavioral;

* I created a GitHub repository for this lab.

**Timing Diagram of Arithmetic Unit:**

Graphical user interface

Description automatically generated