Ian Parker

ECEN 423

4/22/19

Homework 6

**Part 1:** ALU without FeedBack

1) The inputs needed by the ALU are two 4-Bit numbers (**A,B**) and a select line(**SEL**)

2) The control signal that will be generated by the FSM are the outputs of the 4 different states **State0 (00), State1 (01), State2 (10), State3 (11)** which handle **AND, OR, ADD, SUBTRACT** of the two 4 bit inputs **A** and **B**.

**Part 2:** ALU with FeedBack

1) The inputs needed by the ALU are two 4-Bit numbers (**A,B**) and a select line(**SEL**)

2) The conditions that cause the FSM to perform one operation after another are as follows: Depending on the current state(**State0 (00), State1 (01), State2 (10), State3 (11)**) and the feedback from the ALU(**ALU\_OUT**) the FSM will decide to carry out 1 of 4 operations **AND, OR, ADD, SUBTRACT**.

3) The inputs needed by the FSM are **RESET,CLOCK,** and the output of the ALU (**ALU\_OUT**).

4) The control signal that will be generated by the FSM are the outputs of the 4 different states **State0 (00), State1 (01), State2 (10), State3 (11)** which handle **AND, OR, ADD, SUBTRACT** of the two 4 bit inputs **A** and **B**.

ALU (No Feedback)

library ieee;

use ieee.std\_logic\_1164.all;

use ieee.std\_logic\_unsigned.all;

use ieee.numeric\_std.all;

entity ALU is --Without Feedback

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;

architecture Behavioral of ALU is

begin

process(SEL)

begin

if(SEL = "00") then

OUTPUT <= A AND B; --AND

elsif(SEL = "01") then

OUTPUT <= A OR B; --OR

elsif(SEL = "10") then

OUTPUT <= A + B;--ADD

elsif(SEL = "11") then

OUTPUT <= A - B;--SUBTRACT

end if;

end process;

end Behavioral;

FSM(No Feedback)

library ieee;

use ieee. std\_logic\_1164.all;

use ieee. std\_logic\_arith.all;

use ieee. std\_logic\_unsigned.all;

Entity FSM is--Without Feedback

Port(RESET, CLK: in std\_logic;

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

End;

Architecture Behavioral of FSM is

Signal CS,NS : std\_logic\_vector(1 downto 0); --Current and Next State

Begin

Process(RESET,CLK)

Begin

if (RESET = '1')then-- When RESET is active

CS<= "00";--Reset State Machine

elsif(CLK='1' and CLK'EVENT) then

CS<=NS;--When Clock is active, move to next state

end if;

end process;

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";--Loop back to start

When others => NULL;

end case;

end process;

end Behavioral;

TOP LEVEL (No Feedback)

library ieee;

use ieee. std\_logic\_1164.all;

use ieee. std\_logic\_arith.all;

use ieee. std\_logic\_unsigned.all;

entity TopLevel 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 Behavioral of TopLevel is

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

--ALU Component

component ALU 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

component FSM port(RESET, CLK: in std\_logic;

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

end component;

begin

--Component Instantiation

BRAIN : FSM port map(RESET=> RESET,CLK => CLK, OUTPUT=> temp);

LEGS : ALU port map(A=> A, B=> B, SEL=>temp, OUTPUT => OUTPUT);

end;

TestBench (No Feedback)

library ieee;

use ieee.std\_logic\_1164.all;

use ieee.std\_logic\_unsigned.all;

use ieee.numeric\_std.all;

entity TestBench is--No Feedback

end TestBench;

architecture behavioral of TestBench 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

TL : 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-- simulation process

begin

--Set values for A and B

A <= "1001";

B <= "1111";

--Enable the reset for different cases

RESET <= '1';

wait for 10 ns;

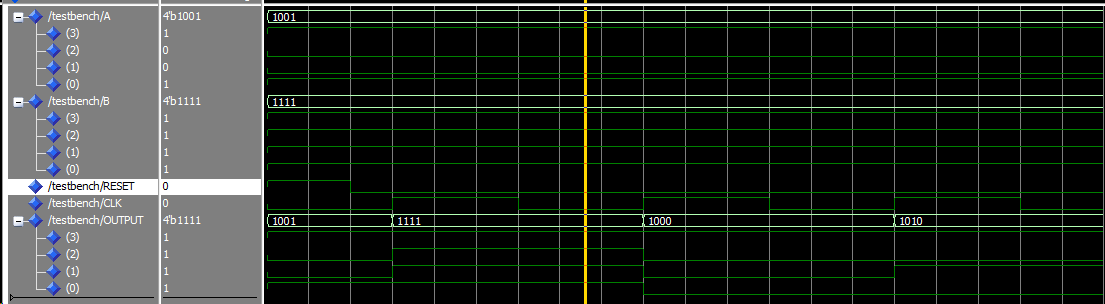
RESET <= '0';

wait for 200 ns;

end process;

end behavioral;

Simulation(No Feedback)



ALU (With Feedback)

library ieee;

use ieee.std\_logic\_1164.all;

use ieee.std\_logic\_unsigned.all;

use ieee.numeric\_std.all;

entity ALU2 is --With Feedback (Same as without feedback)

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;

architecture Behavioral of ALU2 is

begin

process(SEL)

begin

if(SEL = "00") then

OUTPUT <= A AND B; --AND

elsif(SEL = "01") then

OUTPUT <= A OR B; --OR

elsif(SEL = "10") then

OUTPUT <= A + B;--ADD

elsif(SEL = "11") then

OUTPUT <= A - B;--SUBTRACT

end if;

end process;

end Behavioral;

FSM (With Feedback)

library ieee;

use ieee. std\_logic\_1164.all;

use ieee. std\_logic\_arith.all;

use ieee. std\_logic\_unsigned.all;

Entity FSM2 is--With Feedback

Port(RESET, CLK: in std\_logic;

ALU\_OUT: in std\_logic\_vector(3 downto 0);--Output of the ALU used for feedback into the FSM

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

End;

Architecture Behavioral of FSM2 is

Signal CS,NS: std\_logic\_vector(1 downto 0); --Current and Next State

Begin

Process(RESET,CLK)

Begin

if (RESET = '1')then-- When RESET is active

CS<= "00";--Reset State Machine

elsif(CLK='1' and CLK'EVENT) then

CS<=NS;--When Clock is active, move to next state

end if;

end process;

process (CS,ALU\_OUT)

begin

case CS is

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

if (ALU\_OUT = "0001") then

--Move to State 1

OUTPUT <= "01";

NS<= "01";

else --ALU\_OUT is "0000"

OUTPUT <= "00";

NS <= CS; --Stay in State

end if;

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

if (ALU\_OUT = "0001") then

OUTPUT <= "10";

NS<= "10";

else --ALU\_OUT is "0000"

OUTPUT <= "01";

NS <= CS; --Stay in State

end if;

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

if (ALU\_OUT = "0001") then

OUTPUT <= "11";

NS<= "11";

else --ALU\_OUT is "0000"

OUTPUT <= "10";

NS <= CS; --Stay in State

end if;

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

if (ALU\_OUT = "0001") then

OUTPUT <= "00";

NS<= "00";

else --ALU\_OUT is "0000"

OUTPUT <= "11";

NS <= CS; --Stay in State

end if;

When others => NULL;

end case;

end process;

end Behavioral;

TOPLEVEL (With Feedback)

library ieee;

use ieee. std\_logic\_1164.all;

use ieee. std\_logic\_arith.all;

use ieee. std\_logic\_unsigned.all;

entity TopLevel2 is--With Feedback

port(A,B, ALU\_OUT : in std\_logic\_vector(3 downto 0);

RESET,CLK: in std\_logic;

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

end;

architecture Behavioral of TopLevel2 is

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

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

--ALU Component

component ALU2 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

component FSM2 port(RESET, CLK: in std\_logic;

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

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

end component;

begin

--Component Instantiation

BRAIN : FSM2 port map(RESET=> RESET,CLK => CLK, ALU\_OUT=> ALU\_OUT, OUTPUT=> temp);

LEGS : ALU2 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 TestBench2 is--With Feedback

end TestBench2;

architecture behavioral of TestBench2 is

signal A,B, ALU\_OUT: 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, ALU\_OUT : 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

TL : TopLevel2 port map(A=>A, B=>B, ALU\_OUT => ALU\_OUT, 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-- simulation process

begin

--Set values for A and B and ALU\_OUT

A <= "1001";

B <= "1111";

ALU\_OUT <= "0001";

--Enable the reset for different cases

RESET <= '1';

wait for 10 ns;

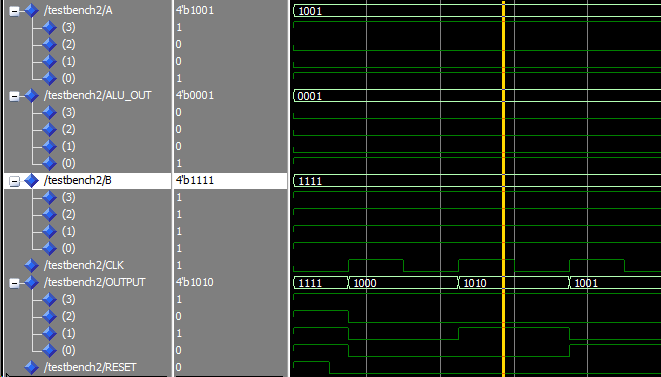
RESET <= '0';

wait for 200 ns;

end process;

end behavioral;

Simulation With Feedback



YOUTUBE

https://youtu.be/5\_P27NWqmL0