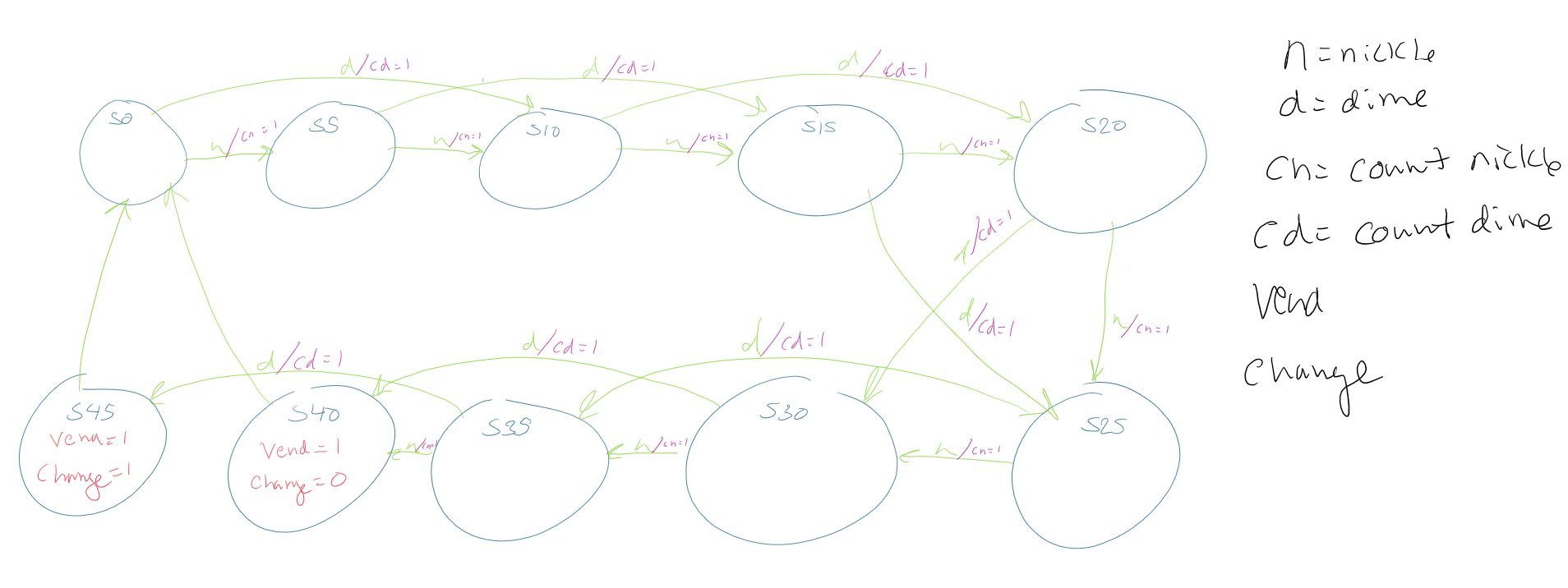
Ian Parker

Homework 10

State Diagram



library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity TopLevel is

port(clk, reset, nickel, dime: in std\_logic;

purchased, remaining : out std\_logic\_vector(3 downto 0);

cash: out std\_logic\_vector(5 downto 0);

change, food : out std\_logic);

end TopLevel;

architecture behavioral of TopLevel is

    signal vend, nickleIN, dimeIN : std\_logic;

    component Counter is port(reset, vended, nickleIN, dimeIN: in std\_logic;

                              purchased, remaining : out std\_logic\_vector(3 downto 0);

                              cash: out std\_logic\_vector(5 downto 0));

    end component;

    component VendingMachine is port(clk, reset, nickel, dime : in std\_logic;

                                          vend, change, countnickle, countdime : out std\_logic);

    end component;

    begin

    VM:VendingMachine port map(clk => clk,

                                reset => reset,

                                nickel => nickel,

                                dime => dime,

                                vend => vend,

                                change => change,

                                countnickle=>nickleIN,

                                countdime=>dimeIN);

                                food<=vend;

    C:Counter port map(reset => reset,

                        vended => vend,

                        nickleIN => nickleIN,

                        dimeIN => dimeIN,

                        purchased => purchased,

                        remaining => remaining,

                        cash => cash);

end behavioral;

Library IEEE;

USE IEEE.Std\_logic\_1164.all;

entity FullTB is

end FullTB;

architecture behavioral of FullTB is

--input signals

signal clk, reset, nickel, dime: std\_logic;

--output signals

signal purchased, remaining: std\_logic\_vector(3 downto 0);

signal cash: std\_logic\_vector(5 downto 0);

signal change, food : std\_logic;

component TopLevel

    port(clk, reset, nickel, dime: in std\_logic;

            purchased, remaining : out std\_logic\_vector(3 downto 0);

            cash: out std\_logic\_vector(5 downto 0);

            change, food : out std\_logic);

end component;

begin

Top:TopLevel

port map(clk, reset, nickel, dime, purchased, remaining, cash, change, food);

clockpulse :process --process that oscillates the clock

begin

        clk <= '0';

        wait for 100 ns;

        clk <= '1';

        wait for 100 ns;

end process;

sim: process --simulation of the vending machine process

       begin

       wait for 100ns;--clk low

       nickel<='1';--5

       dime<='0';

       wait for 100ns;

       wait for 100ns;--clk low

       nickel<='0';

       dime<='1';--15

       wait for 100ns;

       wait for 100ns;--clk low

       nickel<='0';

       dime<='1';--25

       wait for 100ns;

       wait for 100ns;--clk low

       nickel<='0';

       dime<='1';--35

       wait for 100ns;

       wait for 100ns;--clk low

       nickel<='0';

       dime<='1';--45

       wait for 100ns;

       wait for 100ns;--clk low

end process;

end behavioral;

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity VendingMachine is

    port(clk, reset, nickel, dime : in std\_logic;

            vend, change, countnickle, countdime: out std\_logic);

end VendingMachine;

architecture behavioral of VendingMachine is

    type state\_type is (S0, S5, S10, S15, S20, S25, S30, S35, S40, S45);

    signal cs, ns: state\_type;

    begin

    process(clk, reset)

    begin

        if(reset='1') then

            cs<=S0;

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

            cs<=ns;

        end if;

    end process;

    process(cs, nickel, dime)

    begin

    case cs is

        when S0 => --0 cents

        vend<='0';

        change<='0';

        countnickle<='0';

        countdime<='0';

            if(nickel = '1') then

                countnickle<='1';

                ns<= S5;

            elsif(dime='1') then

                countdime<='1';

                ns<=S10;

            end if;

        when S5 => --5 cents

        countnickle<='0';

        countdime<='0';

        vend<='0';

        change<='0';

            if(nickel = '1') then

            countnickle<='1';

            ns<= S10;

            elsif(dime='1') then

            countdime<='1';

            ns<=S15;

            end if;

        when S10 => --10 cents

        vend<='0';

        change<='0';

        countnickle<='0';

        countdime<='0';

            if(nickel = '1') then

            countnickle<='1';

            ns<= S15;

            elsif(dime='1') then

            countdime<='1';

            ns<=S20;

            end if;

        when S15 => --15 cents

        vend<='0';

        change<='0';

        countnickle<='0';

        countdime<='0';

            if(nickel = '1') then

            countnickle<='1';

            ns<= S20;

            elsif(dime='1') then

            countdime<='1';

            ns<=S25;

            end if;

        when S20 => --20 cents

        vend<='0';

        change<='0';

        countnickle<='0';

        countdime<='0';

            if(nickel = '1') then

            countnickle<='1';

            ns<= S25;

            elsif(dime='1') then

            countdime<='1';

            ns<=S30;

            end if;

        when S25 => --25 cents

        vend<='0';

        change<='0';

        countnickle<='0';

        countdime<='0';

            if(nickel = '1') then

            countnickle<='1';

            ns<= S30;

            elsif(dime='1') then

            countdime<='1';

            ns<=S35;

            end if;

        when S30 => --30 cents

        vend<='0';

        change<='0';

        countnickle<='0';

        countdime<='0';

            if(nickel = '1') then

            countnickle<='1';

            ns<= S35;

            elsif(dime='1') then

            countdime<='1';

            ns<=S40;

            end if;

        when S35 => --35 cents

        vend<='0';

        change<='0';

        countnickle<='0';

        countdime<='0';

            if(nickel = '1') then

            countnickle<='1';

            ns<= S40;

            elsif(dime='1') then

            countdime<='1';

            ns<=S45;

            end if;

        when S40 => --represents 40 cents in the machine for the current user

        vend<='1'; --spit out snack

        change<='0';

        ns<=S0;

        when S45=> --45 cents

        vend<='1'; --spit out snack

        change<='1'; --spit out change

        ns<=S0;

        end case;

    end process;

end behavioral;

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

use IEEE.STD\_LOGIC\_UNSIGNED.ALL;

entity Counter is

    port(reset, vended, nickleIN, dimeIN: in std\_logic;

        purchased, remaining : out std\_logic\_vector(3 downto 0);--purchased items, remaining items

        cash: out std\_logic\_vector(5 downto 0));--amount of money added

end Counter;

architecture behavioral of Counter is

signal ptmp, rtmp: std\_logic\_vector(3 downto 0);--purchased items, remaining items

signal ctmp: std\_logic\_vector(5 downto 0);--cash

begin

process(reset,nickleIN,dimeIN,vended)

begin

    if(reset='1') then

        ctmp<="000000";

        ptmp<="0000";

        rtmp<="1010";--start with 10 items remaining

        cash<=ctmp;

        purchased<=ptmp;

        remaining<= rtmp;

    elsif(nickleIN='1') then

        ctmp <= ctmp + "000101";--add 5

        cash<=ctmp;

    elsif(dimeIN='1') then

        ctmp<=ctmp + "001010";--add 10

        cash<=ctmp;

    elsif(vended='1') then--if item is vended

        ptmp<= ptmp + "0001";--purchased items add 1

        rtmp <= rtmp - "0001";--remaining items minus 1

        purchased<=ptmp;

        remaining<= rtmp;

    end if;

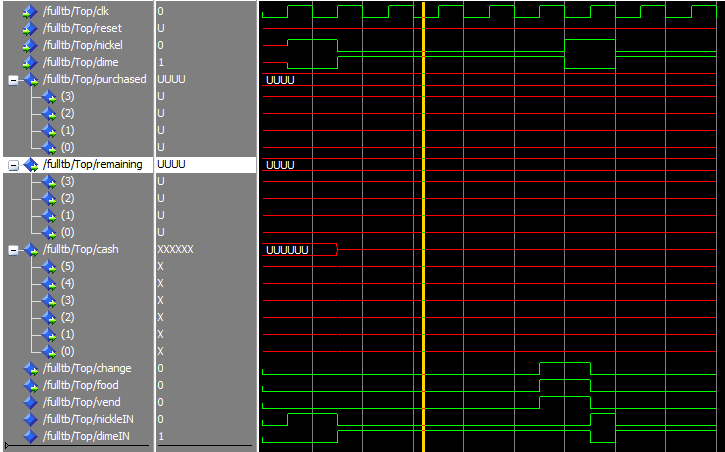
        cash<=ctmp;

        purchased<=ptmp;

        remaining<= rtmp;

    end process;

end behavioral;



In the simulation above I got everything working other than the counting. I’m pretty sure my code for it is correct but I think I messed up in my port maps on my top-level design.

Video

https://youtu.be/-Dx57SJ2y58