Major component is my timing unit for my design. My washing machine needs to be able to control the timing on the coins, lids and cycles. The machine should be able to switch cycles during the process and cleaning and the timer can controller this process. The machine should be able to return coins if it is not being used and that is what a timer can help with. The timer also can help with the alerting the user that the lids need to be closed in other to start the machine in the first place.

library ieee;

use ieee.std\_logic\_1164.all;

use ieee.std\_logic\_arith.all;

use ieee.std\_logic\_unsigned.all;

entity timingcontroller is

port (clk, coincontroller, lidcontroller, cyclecontroller: in std\_logic;--These are all signals that the fsm will send and start the counters

counter1: inout std\_logic\_vector(2 downto 0);--this counter will go back to fsm

counter2: inout std\_logic\_vector(2 downto 0));--This counter will display on the seven segments

end;

architecture beh of timingcontroller is

begin

process(clk,coincontroller,lidcontroller,cyclecontroller)

begin

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

if(coincontroller='1')then--if a coin is inserted it started a timer to reset after a certain amount of time

counter1<=counter1 +1;

counter2<=counter2 +1;

elsif(coincontroller='0') then --if the coins inserted arent used then put the timers back to zero

counter1<="000";

counter2<="000";

if(lidcontroller='1')then--if the lid of the machine is left open for a certain amount of time and alarm will go off

counter1<=counter1 +1;

counter2<=counter2 +1;

elsif(lidcontroller='0') then --once the lid is closed then the alarm will turn off as well

counter1<="000";

counter2<="000";

if(cyclecontroller='1')then--this will start counting during the process of washing

counter1<=counter1 +1;

counter2<=counter2 +1;

elsif(cyclecontroller='0') then --When a washing task is done it will turn off the timer again.

counter1<="000";

counter2<="000";

end if;

end if;

end if;

end if;

end process;

end;