

```

--
-- VHDL Architecture Bachelor.ADC_TO_FPGA.STUDENT
--
-- Created:
--         by - christop.grobety.UNKNOWN (WE2332207)
--         at - 11:23:31 31.05.2023
--
-- using Mentor Graphics HDL Designer(TM) 2019.2 (Build 5)
--
LIBRARY Common;
USE Common.CommonLib.all;

ARCHITECTURE STUDENT OF ADC_TO_FPGA IS
    signal done16 : std_uLogic ;
    signal down_ncs : std_uLogic ;
    signal down_ncs_old : std_uLogic ;
    signal is_up : std_uLogic ;
    signal finish_d : std_uLogic ;
    signal count: unsigned(requiredBitNb(counter_length)-1 downto 0);
BEGIN
    adc_to_fpga: process(clock,reset)
    begin
        if reset = '1' then
            fram_clock <= (others => '1');
            adc_clock <= '1';
            finish_d <= '0';
            --sdi <= (others => '0');
            ncs_fram <= (others => '0');
            ncs_adc <= '1';
            done16 <= '0';
            count <= (others => '0');
            is_up <= '1';
            count_go <= '0';
            down_ncs_old <= '1';
        elsif rising_edge(clock) then
            --if m = "0010" and down_ncs = '0' then
            count_go <= '0';
            if down_ncs /= down_ncs_old and m = "1010" and finish_d = '0' then
                down_ncs_old <= down_ncs;
                is_up <= '0';
            elsif is_up = '0' and finish_d = '0' and m = "1010" then
                ncs_fram <= (others => '0');
                ncs_adc <= '0';
                --sdi <= sdo;
                if count >= 1 and count <= counter_length-3 then
                    if count = 8 and count_go = '0' then
                        count_go <= '1';
                    end if;

                    if adc_clock = '1' then
                        adc_clock <= '0';
                        fram_clock <= (others => '0');
                    else
                        adc_clock <= '1';

```

```

        fram_clock <= (others => '1');
        --sdi <= sdo;
        count <= count + 1;
    end if;
    elsif count = counter_length-2 then
        count_go <= '1';
        adc_clock <= '1';
        fram_clock <= (others => '1');
        count <= count+1 ;
    elsif count = counter_length-1 then
        is_up <= '1';
        ncs_adc <= '1';
        --sdi <= (others => '0');
        ncs_fram <= (others => '0');
    elsif count = 0 then
        count <= count + 1;
    end if ;
else
    count <= (others => '0');
end if;

if done_sig = '1' then
    finish_d <= '1';
    ncs_fram <= (others => '1');
    ncs_adc <= '1';
elsif m /= "1010" then
    finish_d <= '0';
end if ;

end if;
end process adc_to_fpga;

NCS_clock: process(clock_1m,reset)
begin
    if reset = '1' then
        down_ncs <= '1';
    elsif rising_edge(clock_1m) then
        --if m = "0010" then
        if down_ncs = '1' and m = "1010" then
            down_ncs <= '0';
        else
            down_ncs <= '1';
        end if;
    end if;
end process NCS_clock;

data_to_trans: process(sdo, m)
begin
    if m = "1010" then
        sdi <= sdo;
    else
        sdi <= (others => '0');
    end if;
end process data_to_trans;

```

END ARCHITECTURE STUDENT;