CODES: Interfacing of ADC with FPGA using SPI Protocol

`timescale 1ns / 1ps

module SPI\_DAC(

input clk, //50 MHz FPGA CLOCK

input reset,

input spi\_miso, //MASTER IN,SLAVE OUT

input [11:0]data, //Digital to be given to DAC Module

input [3:0]address, //DAC Address for A,B,C,D Pin

output reg send,

output reg dac\_cs,spi\_sck,spi\_mosi,dac\_clr, //Signal on DAC

output SPI\_SS\_B,AMP\_CS,AD\_CONV,SF\_CEO,FPGA\_INIT\_B, //Perpheral Signal to be disabled

output reg [3:0]command, // Command = 4'B0011

output reg [2:0] dac\_state, //DAC STATES

output reg [31:0] dac\_out, //DAC INPUT=(8'b don't care, 4'b command, 4'b address, 12'b data, 4'

output reg [5:0]count=32);// 32-bit counter

assign SPI\_SS\_B=1; //SPI FLASH

assign AMP\_CS=1; //AMPLIFIER SELECT

assign AD\_CONV=0; //ADC CONVERSION

assign SF\_CE0=1; //STRATA FLASH

assign FPGA\_INIT\_B=1; //PLATFORM FLASH

always@(posedge clk or posedge reset)

begin

if (reset==1)

//Disabling other peripherals connected to SPI BUS

// WILL BE INTERFERED WITH DAC OPERATION

begin

dac\_cs<=1;

spi\_sck<=0;

spi\_mosi<=0;

dac\_clr<=1;

send<=0;

dac\_state<=0;

end

else begin

case(dac\_state) //DAC States

0:begin //IDLE

dac\_cs<=1;

spi\_sck<=0;

spi\_mosi<=0;

dac\_clr<=1;

send<=0;

dac\_state<=dac\_state+1;

end

1:begin //32 Bit Assigningto DAC

dac\_out<={8'b00000000,4'b0011,4'b0000,12'b101100000000,4'b0000};

dac\_state<=dac\_state+1;

end

2:begin

dac\_cs<=0;

spi\_sck<=0;

spi\_mosi<=dac\_out[count-1];

count<=count-1;

dac\_state<=dac\_state+1;

end

3:begin

if(count>0)

begin

spi\_sck<=1;

dac\_state<=2;

end

else

begin

spi\_sck<=1;

dac\_state<=dac\_state+1;

end

end

4:begin

spi\_sck<=0;

dac\_state<=dac\_state+1;

end

5:begin

dac\_cs<=1;

dac\_state<=dac\_state+1;

end

6:begin

send<=1;

dac\_state<=dac\_state+1;

end

7:begin

send<=0;

dac\_state<=1;

end

default:begin

dac\_cs<=1;

spi\_mosi<=0;

spi\_sck<=0;

dac\_clr<=1;

send<=0;

dac\_state<=0;

count<=32;

end

endcase

end

end

endmodule