# SERIAL PERPHERAL INTERFACE (SPI)

# DAC (DIGITAL TO ANALOG) INTERFACING WITH FPGA SPARTEN 3E

Verilog code

`timescale 1ns / 1ps

//////////////////////////////////////////////////////////////////////////////////

// Company:

// Engineer:

//

// Create Date: 25.07.2023 16:41:29

// Design Name:

// Module Name: dac

// Project Name:

// Target Devices:

// Tool Versions:

// Description:

//

// Dependencies:

//

// Revision:

// Revision 0.01 - File Created

// Additional Comments:

//

//////////////////////////////////////////////////////////////////////////////////

module dac(clk,reset,data,dac\_cs,spi\_sck,spi\_mosi,spi\_miso,dac\_clr,

SPI\_SS\_B,AMP\_CS,AD\_CONV,SF\_CE0,FPGA\_INIT\_B,command,address,send);

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\_CE0,FPGA\_INIT\_B; // PERIPHERAL SIGNAL TO BE DISABLED

output reg [3:0]command; // COMMAND =4'B0011

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

reg [31:0]dac\_out; //DAC INPUT={8'b don'tcare,4'b command,4'b address,12'b data,4'b don't care} // usuall command =4'b0011

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 SO 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 ASSIGNING to DAC

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

dac\_state<=dac\_state+1;

end

2:begin // BIT ASSIGNING ON SPI\_MOSI LINE

dac\_cs<=0; // FPGA TRANSMITS DATA ON MOSI line when DAC\_CS<=0;

spi\_sck<=0;

spi\_mosi<=dac\_out[count-1]; // ASSIGNING DIGITAL BIT TO MOSI LINE , STARTING FROM MSB

count<=count-1;

dac\_state<=dac\_state+1;

end

3:begin // WAITING FOR COMPLETE 32 BIT INPUT TO MOSI

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; // ANALOG CONVERSION STARTS, WHEN DAC\_CS<=1;AFTER ASSIGNING 32 BIT TO MOSI line

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

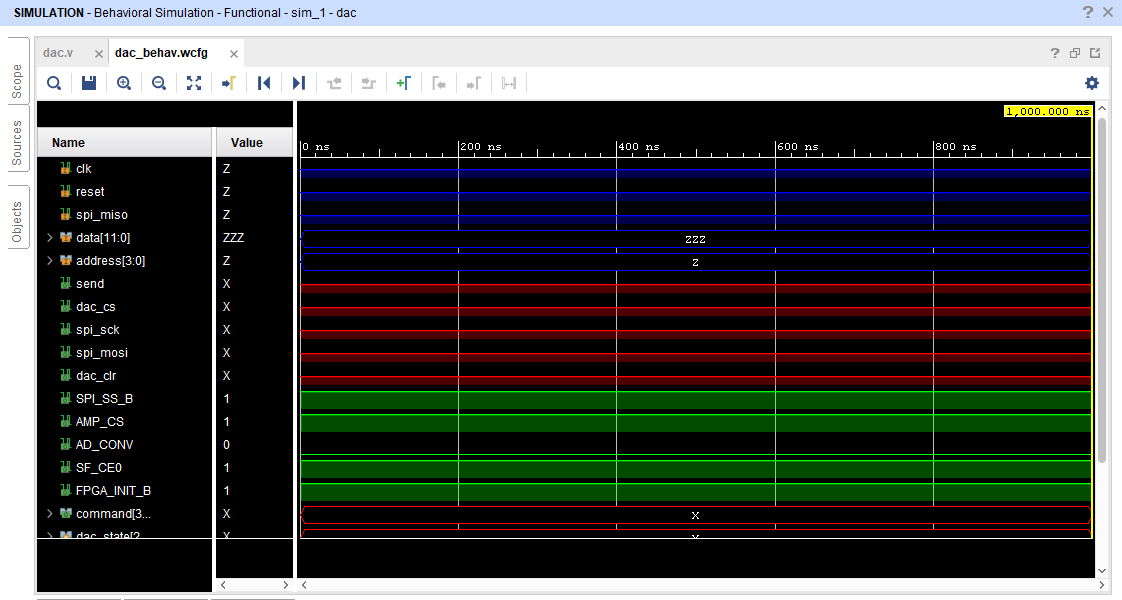
endcase

end

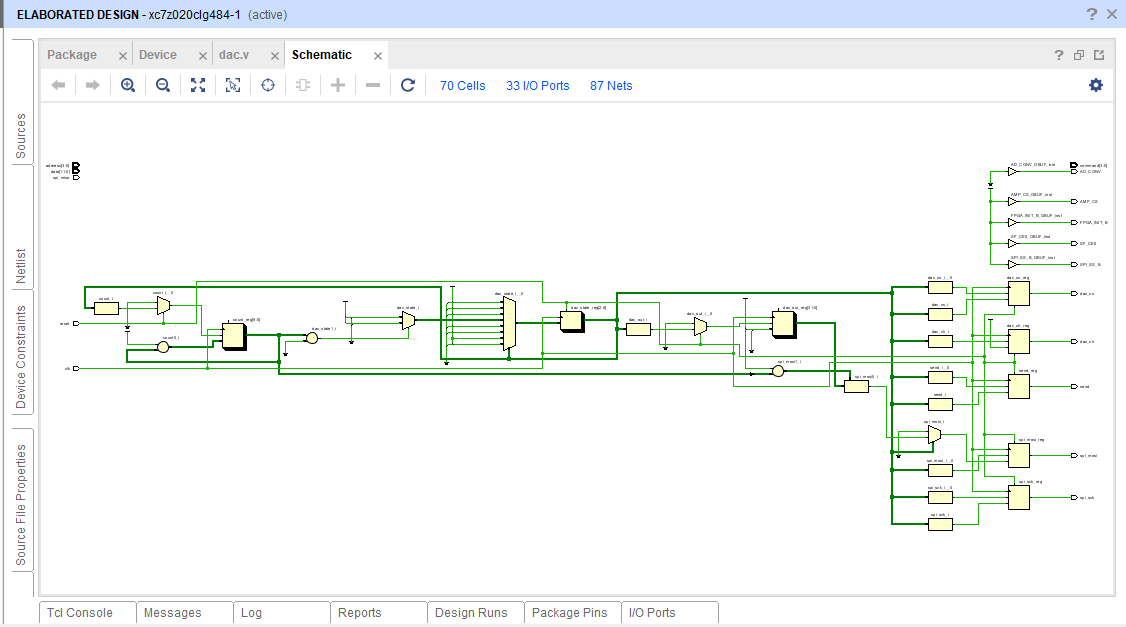
end

endmodule

SIMULATION



SCHEMATIC



IMPLEMENTATION DESIGN

