```
// ***************************************************************************************************************
// **                                                                                                           **
// **   AD8801.v - AD8801 8-BIT SERIAL TDAC                                                                     **
// **                                                                                                           **
// ***************************************************************************************************************
// **                                                                                                           **
// **                              COPYRIGHT (c) 2000 YOUNG ENGINEERING                                         **
// **                                      ALL RIGHTS RESERVED                                                  **
// **                                                                                                           **
// **   THIS PROGRAM IS CONFIDENTIAL AND  A  TRADE SECRET  OF  YOUNG  ENGINEERING.  THE RECEIPT OR              **
// **   POSSESSION  OF THIS PROGRAM  DOES NOT  CONVEY  ANY  RIGHTS TO  REPRODUCE  OR  DISCLOSE ITS             **
// **   CONTENTS,  OR TO MANUFACTURE, USE, OR SELL  ANYTHING  THAT IT MAY DESCRIBE, IN WHOLE OR IN             **
// **   PART, WITHOUT THE SPECIFIC WRITTEN CONSENT OF YOUNG ENGINEERING.                                        **
// **                                                                                                           **
// ***************************************************************************************************************
// **   Revision       : 1.0                                                                                    **
// **   Modified Date  : 02/01/2000                                                                             **
// **   Revision History:                                                                                       **
// **                                                                                                           **
// **   02/01/2000:  Initial design                                                                             **
// **                                                                                                           **
// ***************************************************************************************************************
// **                                        TABLE OF CONTENTS                                                  **
// ***************************************************************************************************************
// **-------------------------------------------------------------------------------------------------------------**
// **   DECLARATIONS                                                                                            **
// **-------------------------------------------------------------------------------------------------------------**
// **-------------------------------------------------------------------------------------------------------------**
// **   INITIALIZATION                                                                                          **
// **-------------------------------------------------------------------------------------------------------------**
// **-------------------------------------------------------------------------------------------------------------**
// **   CORE LOGIC                                                                                              **
// **-------------------------------------------------------------------------------------------------------------**
// **-------------------------------------------------------------------------------------------------------------**
// **   TIMING CHECKS                                                                                           **
// **-------------------------------------------------------------------------------------------------------------**
// **                                                                                                           **
// ***************************************************************************************************************


`timescale 1ns/10ps

module AD8801 (O1, O2, O3, O4, O5, O6, O7, O8, SDI, CLK, CS_N, RS_N, RESET);

   input               SDI;                        // serial data in
   input               CLK;                        // serial data clock

   input               CS_N;                       // chip select
   input               RS_N;                       // midscale preset

   input               RESET;                      // system reset

   output [07:00]      O1;                         // DAC 1 voltage output
   output [07:00]      O2;                         // DAC 2 voltage output
   output [07:00]      O3;                         // DAC 3 voltage output
   output [07:00]      O4;                         // DAC 4 voltage output
   output [07:00]      O5;                         // DAC 5 voltage output
   output [07:00]      O6;                         // DAC 6 voltage output
   output [07:00]      O7;                         // DAC 7 voltage output
   output [07:00]      O8;                         // DAC 8 voltage output


// ***************************************************************************************************************
// **   DECLARATIONS                                                                                            **
// ***************************************************************************************************************

   reg  [10:00]        ShiftRegister;              // serial shift register
   wire                ShiftClock;                 // shift register clock

   reg  [07:00]        O1;                         // DAC 1 output voltage
   reg  [07:00]        O2;                         // DAC 2 output voltage
   reg  [07:00]        O3;                         // DAC 3 output voltage
   reg  [07:00]        O4;                         // DAC 4 output voltage
   reg  [07:00]        O5;                         // DAC 5 output voltage
   reg  [07:00]        O6;                         // DAC 6 output voltage
   reg  [07:00]        O7;                         // DAC 7 output voltage
   reg  [07:00]        O8;                         // DAC 8 output voltage
```

```
// ***************************************************************************************************************
// **   INITIALIZATION                                                                                          **
// ***************************************************************************************************************

   initial begin
      O1 = 8'hXX;
      O2 = 8'hXX;
      O3 = 8'hXX;
      O4 = 8'hXX;
      O5 = 8'hXX;
      O6 = 8'hXX;
      O7 = 8'hXX;
      O8 = 8'hXX;
   end


// ***************************************************************************************************************
// **   CORE LOGIC                                                                                              **
// ***************************************************************************************************************

   assign ShiftClock = CLK & !CS_N;

   always @(posedge ShiftClock) begin
      ShiftRegister[00] <= SDI;
      ShiftRegister[01] <= ShiftRegister[00];
      ShiftRegister[02] <= ShiftRegister[01];
      ShiftRegister[03] <= ShiftRegister[02];
      ShiftRegister[04] <= ShiftRegister[03];
      ShiftRegister[05] <= ShiftRegister[04];
      ShiftRegister[06] <= ShiftRegister[05];
      ShiftRegister[07] <= ShiftRegister[06];
      ShiftRegister[08] <= ShiftRegister[07];
      ShiftRegister[09] <= ShiftRegister[08];
      ShiftRegister[10] <= ShiftRegister[09];
   end

   always @(posedge CS_N or negedge RS_N) begin
      if (RS_N == 0) begin
         O1 <= 8'h80;
         O2 <= 8'h80;
         O3 <= 8'h80;
         O4 <= 8'h80;
         O5 <= 8'h80;
         O6 <= 8'h80;
         O7 <= 8'h80;
         O8 <= 8'h80;
      end
      else begin
         case (ShiftRegister[10:08])
            3'h0 : O1 <= ShiftRegister[07:00];
            3'h1 : O2 <= ShiftRegister[07:00];
            3'h2 : O3 <= ShiftRegister[07:00];
            3'h3 : O4 <= ShiftRegister[07:00];
            3'h4 : O5 <= ShiftRegister[07:00];
            3'h5 : O6 <= ShiftRegister[07:00];
            3'h6 : O7 <= ShiftRegister[07:00];
            3'h7 : O8 <= ShiftRegister[07:00];
         endcase
      end
   end


// ***************************************************************************************************************
// **   TIMING CHECKS                                                                                           **
// ***************************************************************************************************************

   specify
      specparam
         tCH  = 15,                                    // CLK pulse width - high
         tCL  = 15,                                    // CLK pulse width - low
         tDS  = 5,                                     // SDI to CLK setup time
         tDH  = 5,                                     // SDI to CLK hold  time
         tRS  = 60,                                    // RS_N pulse width - low
         tCSS = 10,                                    // CS_N to CLK setup time
         tCSH = 15,                                    // CS_N to CLK hold  time
         tCSW = 10;                                    // CS_N pulse width - high

      $width (posedge CLK,  tCH);
      $width (negedge CLK,  tCL);
```

```
      $width (negedge RS_N, tRS);
      $width (posedge CS_N, tCSW);

      $setup (SDI,  posedge CLK &&& (RESET==0), tDS);
      $setup (CS_N, posedge CLK &&& (RESET==0), tCSS);

      $hold  (posedge CLK &&& (RESET==0), SDI,  tDH);
      $hold  (posedge CLK &&& (RESET==0), CS_N, tCSH);
   endspecify

endmodule
```