

```

// *****
// **
// ** AD7837.v - AD7837 12-BIT PARALLEL MDAC (A/B/S GRADES)
// **
// *****
// **
// **          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 : 02/28/2000
// ** Revision History:
// **
// ** 02/28/2000: Initial design
// **
// *****
// **          TABLE OF CONTENTS
// *****
// **-----**
// **  DECLARATIONS
// **-----**
// **-----**
// **  INITIALIZATION
// **-----**
// **-----**
// **  CORE LOGIC
// **-----**
// **-----**
// **  TIMING CHECKS
// **-----**
// **
// *****

```

```
`timescale 1ns/10ps
```

```

module AD7837      ( VOUTA,
                   VOUTB,

                   DB00,
                   DB01,
                   DB02,
                   DB03,
                   DB04,
                   DB05,
                   DB06,
                   DB07,

                   A0,
                   A1,

                   WR_N,
                   CS_N,
                   LDAC_N,

                   RESET
                 );

input  DB00;           // parallel data bus in
input  DB01;           // parallel data bus in
input  DB02;           // parallel data bus in
input  DB03;           // parallel data bus in
input  DB04;           // parallel data bus in
input  DB05;           // parallel data bus in
input  DB06;           // parallel data bus in
input  DB07;           // parallel data bus in

input  A0;             // input latch address
input  A1;             // input latch address

input  WR_N;           // data write strobe
input  CS_N;           // chip select
input  LDAC_N;         // DAC update strobe

```

```

input                RESET;                // system reset

output [11:00]       VOUTA;                // DAC A voltage output
output [11:00]       VOUTB;                // DAC B voltage output

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

reg [11:00]          DACA_InputLatch;      // DAC A input data latch
reg [11:00]          DACB_InputLatch;      // DAC B input data latch

reg [11:00]          VOUTA;                // DAC A output voltage
reg [11:00]          VOUTB;                // DAC B output voltage

wire [07:00]         DataDI;              // parallel data bus input

// *****
// **  INITIALIZATION                                **
// *****

initial begin
    DACA_InputLatch= 12'hXXX;
    DACB_InputLatch= 12'hXXX;
end

initial begin
    VOUTA = 12'hXXX;
    VOUTB = 12'hXXX;
end

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

assign DataDI[07:00] = {DB07,DB06,DB05,DB04,DB03,DB02,DB01,DB00};

always @(DataDI or A0 or A1 or WR_N or CS_N) begin
    case ({A1,A0, WR_N,CS_N})
        4'b00_00 :    DACA_InputLatch[07:00] = DataDI[07:00];
        4'b01_00 :    DACA_InputLatch[11:08] = DataDI[03:00];
        4'b10_00 :    DACB_InputLatch[07:00] = DataDI[07:00];
        4'b11_00 :    DACB_InputLatch[11:08] = DataDI[03:00];
    endcase
end

always @(LDAC_N or DACA_InputLatch or DACB_InputLatch) begin
    if (LDAC_N == 0) begin
        VOUTA = DACA_InputLatch[11:00];
        VOUTB = DACB_InputLatch[11:00];
    end
end

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

specify
    specparam
        t3 = 30,                // WR_N pulse width - low
        t4 = 80,                // DB to WR_N setup time
        t8 = 50;                // LDAC_N pulse width - low

    $width (negedge WR_N, t3);
    $width (negedge LDAC_N, t8);

    $setup (DB00, posedge WR_N &&& (RESET==0), t4);
    $setup (DB01, posedge WR_N &&& (RESET==0), t4);
    $setup (DB02, posedge WR_N &&& (RESET==0), t4);
    $setup (DB03, posedge WR_N &&& (RESET==0), t4);
    $setup (DB04, posedge WR_N &&& (RESET==0), t4);
    $setup (DB05, posedge WR_N &&& (RESET==0), t4);
    $setup (DB06, posedge WR_N &&& (RESET==0), t4);
    $setup (DB07, posedge WR_N &&& (RESET==0), t4);

```

endspecify

endmodule