```
//
// Master timing, ADC and DAC interfaces
//
// Mark Atherton, New Zealand
// 11 Nov 2012
//
// Generate master clocks for the system
// with 50.000MHz xtal, FS = 48.828kHz
// with 49.152MHz xtal, FS = 48.000kHz
//
// This is nearly, but not quite I2S, so please exuse incorrect names
//

module i2s_master_clock(clk, m_clk, b_clk, lr_clk, start);

output m_clk, b_clk, lr_clk, start;
input clk;

reg m_clk, b_clk, lr_clk, start;
reg [9:0]cnt;                                   // 50MHz / 1024 = 48.828kHz

always @(posedge clk)
begin
   m_clk <= cnt[1];                             // 12.288MHz master clock
   b_clk <= cnt[3];                             // 3.072MHz bit clock
   lr_clk <= cnt[9];                            // 48kHz LR clock

    if((cnt & 10'h1FF) == 0)                    // start bit at posn 0 and 512
        start = 1'b1;
    else
        start = 1'b0;

    cnt <= cnt + 1;                             // advance main counter
end
endmodule
```

```
// start bit is high clock after a LR transition
// 1024 clocks per frame of data, 16 clocks per data cell

module i2s_in(clk, start, lr_clk, data_in, odd_data_out, even_data_out, marker);

input clk, start, lr_clk, data_in;
output [17:0]odd_data_out, even_data_out;
output marker;

reg [17:0]odd_data_out, even_data_out, sr;
reg [8:0]ctr;

reg marker;

always @(posedge clk)
begin
    if(start)
    begin
        ctr <= 0;
    end
    else if(ctr < 288)                          // gate open ?
    begin
        if((ctr & 6'hF) == 6)                   // check for centre of data cell
        begin
            sr <= sr << 1;                      // shift data up
            sr[0] <= data_in;                   // move data into shift reg
            marker <= 1;                        // mark sample point
        end
        else
            marker <= 0;                        // mark sample point
        begin
        end
        ctr <= ctr + 1;
    end
    else
```

```
    begin
        if(lr_clk)
            even_data_out <= sr;                // transfer even data
        else
            odd_data_out <= sr;                 // transfer odd data
    end
end
endmodule

// start bit is high clock after a LR transition
// 1024 clocks per frame of data, 16 clocks per data cell

module i2s_out(clk, start, lr_clk, data_out, odd_data_in, even_data_in, marker);

input clk, start, lr_clk;
input [17:0]odd_data_in, even_data_in;
output marker, data_out;

wire [17:0]odd_data_in, even_data_in;
reg [17:0]sr;
reg [8:0]ctr;

reg data_out, marker;

always @(posedge clk)
begin
    ctr <= ctr + 1;                             // advance main counter
    if(start)                                   // start of L or R frame
    begin
        ctr <= 0;                               // reset this counter
        if(lr_clk)                              // load shift register
            sr <= odd_data_in;
        else
            sr <= even_data_in;
    end
    else if(ctr < 288)                          // gate open ?
```

```
    begin
        if((ctr & 15) == 0)                 // test for bit clock
        begin
            data_out <= sr[17];             // move data out
            sr <= sr << 1;                  // shift the register
            marker <= 1;                    // mark sample point
        end
        else
        begin
            marker <= 0;
        end
    end
end
endmodule
```