WSTM.V

```
//////////////////////////// _WSTM ////////////////////////////////////
//  VERILOG source Code For TOP_STM BLOCK WRITTEN BY PROST 2010.06.17  //
//                                                                     //
/////////////////////////////////////////////////////////////////////////



module WSTM(
/* input signals */

    XRESET, //システムリセット
    SYSCLK, //システムクロック
    SDATA,  //DAIからのシリアルデータ
    LRCK,   //DAIからのLRCK
    BCK,    //DAIからのビットクロック
    DEPTH_SEL,

/*output signals  */

    XBUFOE, //双方向バスの出力イネーブル（負論理）
    XWCE,   //ライトのチップイネーブル
    XWE,    //ライトイネーブル
    XWBHE,  //ライトの？？
    RSTART, //リードシーケンスのスタート信号
    WADDRS, //ライトアドレス
    WDATA   //ライトデータ

    );


//////////////////////////* input signals *//////////////////////////////

input[1:0] DEPTH_SEL;

input

    XRESET,
    SYSCLK,
    SDATA,
    LRCK,
    BCK;

//////////////////////////* output signals *//////////////////////////////

output[15:0]WADDRS;
output[15:0]WDATA;

output
    XBUFOE,
    XWCE,
    XWE,
    XWBHE,
    RSTART;

wire[15:0] DEPTH;
wire
    WCNT15,
    WCNT31;

///////////////* make LRCK  1shot up edge *////////////////////

reg LRCK1,LRCK2;
wire LRCKU1S;

always@(negedge XRESET or posedge SYSCLK)
begin
    if (!XRESET)
    begin

        LRCK1 <= 1'b0;
        LRCK2 <= 1'b0;
    end

    else
    begin
        LRCK1 <= LRCK;
        LRCK2 <= LRCK1;
    end

end
```

```
assign LRCKU1S = (LRCK1 & !LRCK2);


///////////////* make BCK  1shot up edge and down edge *////////////////////

reg BCK1,BCK2;
wire BCKU1S;

always@(negedge XRESET or posedge SYSCLK)
begin
    if (!XRESET)
    begin
        BCK1 <= 1'b0;
        BCK2 <= 1'b0;
    end

    else
    begin
        BCK1 <= BCK;
        BCK2 <= BCK1;
    end

end

assign BCKU1S = (BCK1 & !BCK2);
assign BCKD1S = (!BCK1 & BCK2);

/////////////////////////// MAKE DEPTH DEC ///////////////////////////////

function  decdepth;

input [1:0] DEPTH_SEL;


    begin
    case({DEPTH_SEL[1],DEPTH_SEL[0]})
    2'b00 : decdepth = 16'h000f;
    2'b01 : decdepth = 16'h00ff;
    2'b10 : decdepth = 16'h0fff;
    2'b11 : decdepth = 16'hffff;
    default : decdepth = 16'h000f;

    endcase
    end

endfunction

assign DEPTH =  decdepth(
                        DEPTH_SEL);



//////////////////////////* wires *//////////////////////////////////////

wire

    ss0, // LRCKU1S
    ss1, // BCKU1S
    ss2, // BCKD1S
    ss3; // WCNTEND



////////////////* reg for wstm *////////////////////////////////////////////

reg[7:0] sto; //ステートマシーン用レジスタ


////////////////* conditions for state machine*//////////////////////////

assign ss0 = LRCKU1S;
assign ss1 = BCKU1S;
assign ss2 = BCKD1S;
assign ss3 = WCNT15;
assign ss4 = WCNT31;



//////////////////////* make state machines *//////////////////////////
```

```
parameter  s0   = 10'b0000111100; //XBUFOE=1 XWCE=1 XWE=1 XBHE=1 WCNTE=0 RSTART=0
parameter  s1   = 10'b0001111100; //XBUFOE=1 XWCE=1 XWE=1 XBHE=1 WCNTE=0 RSTART=0
parameter  s2   = 10'b0010111100; //XBUFOE=1 XWCE=1 XWE=1 XBHE=1 WCNTE=0 RSTART=0
parameter  s3   = 10'b0011111110; //XBUFOE=1 XWCE=1 XWE=1 XBHE=1 WCNTE=1 RSTART=0
parameter  s4   = 10'b0100011110; //XBUFOE=0 XWCE=1 XWE=1 XBHE=1 WCNTE=1 RSTART=0
parameter  s5   = 10'b0101000010; //XBUFOE=0 XWCE=0 XWE=0 XBHE=0 WCNTE=1 RSTART=0
parameter  s6   = 10'b0110000010; //XBUFOE=0 XWCE=0 XWE=0 XBHE=0 WCNTE=1 RSTART=0
parameter  s7   = 10'b0111111110; //XBUFOE=1 XWCE=1 XWE=1 XBHE=1 WCNTE=1 RSTART=0
parameter  s8   = 10'b1000011110; //XBUFOE=0 XWCE=1 XWE=1 XBHE=1 WCNTE=1 RSTART=0
parameter  s9   = 10'b1001000010; //XBUFOE=0 XWCE=0 XWE=0 XBHE=0 WCNTE=1 RSTART=0
parameter  s10  = 10'b1010000010; //XBUFOE=0 XWCE=0 XWE=0 XBHE=0 WCNTE=1 RSTART=0
parameter  s11  = 10'b1011111111; //XBUFOE=1 XWCE=1 XWE=1 XBHE=1 WCNTE=1 RSTART=1


always @(negedge XRESET or posedge SYSCLK)
begin
  if (XRESET == 1'b0)
    sto <= s0;
  else
  begin
    casex(sto)
    10'b0000111100 : begin //s0 ss4= WCNT31 ss3= WCNT15 ss2= BCKU1S ss1= BCKD1S ss0= LRCKU1S
        casex({ss4,ss3,ss2,ss1,ss0})
        5'b00xx1 : sto <=  s1;
        default : sto <=  s0;
        endcase
         end

    10'b0001111100 : begin //s1  ss4= WCNT31 ss3= WCNT15 ss2= BCKU1S ss1= BCKD1S ss0= LRCKU1S
        casex({ss4,ss3,ss2,ss1,ss0})
        5'b0010x : sto <=  s2;
        default : sto <=  s1;
        endcase
         end

    10'b0010111100 : begin //s2  ss4= WCNT31 ss3= WCNT15 ss2= BCKU1S ss1= BCKD1S ss0= LRCKU1S
        casex({ss4,ss3,ss2,ss1,ss0})
        5'b0001x : sto <=  s3;
        default : sto <=  s2;
        endcase
         end

    10'b0011111110 : begin //s3  ss4= WCNT31 ss3= WCNT15 ss2= BCKU1S ss1= BCKD1S ss0= LRCKU1S
        casex({ss4,ss3,ss2,ss1,ss0})
        5'b01xxx : sto <=  s4;
        5'b10xxx : sto <=  s8;
        default : sto <=  s3;
        endcase
         end

    10'b0100011110 : begin //s4  ss4= WCNT31 ss3= WCNT15 ss2= BCKU1S ss1= BCKD1S ss0= LRCKU1S
        casex({ss4,ss3,ss2,ss1,ss0})
        5'b0x10x : sto <=  s5;
        default : sto <=  s4;
        endcase
         end

    10'b0101000010 : begin //s5  ss4= WCNT31 ss3= WCNT15 ss2= BCKU1S ss1= BCKD1S ss0= LRCKU1S
        casex({ss4,ss3,ss2,ss1,ss0})
        5'bxxxxx : sto <=  s6;
        default : sto <=  s5;
        endcase
         end

    10'b0110000010 : begin //s6  ss4= WCNT31 ss3= WCNT15 ss2= BCKU1S ss1= BCKD1S ss0= LRCKU1S
        casex({ss4,ss3,ss2,ss1,ss0})
        5'bxxxxx : sto <=  s7;
        default : sto <=  s6;
        endcase
         end

    10'b0111111110 : begin //s7  ss4= WCNT31 ss3= WCNT15 ss2= BCKU1S ss1= BCKD1S ss0= LRCKU1S
        casex({ss4,ss3,ss2,ss1,ss0})
        5'bxxxxx : sto <=  s3;
        default : sto <=  s7;
        endcase
         end

    10'b1000011110 : begin //s8  ss4= WCNT31 ss3= WCNT15 ss2= BCKU1S ss1= BCKD1S ss0= LRCKU1S
        casex({ss4,ss3,ss2,ss1,ss0})
        5'b0x10x : sto <=  s9;
        default : sto <=  s8;
```

```
        endcase
         end

    10'b1001000010 : begin //s9  ss4= WCNT31 ss3= WCNT15 ss2= BCKU1S ss1= BCKD1S ss0= LRCKU1S
        casex({ss4,ss3,ss2,ss1,ss0})
        5'bxxxxx : sto <=  s10;
        default : sto <=  s9;
        endcase
         end

    10'b1010000010 : begin //s10  ss4= WCNT31 ss3= WCNT15 ss2= BCKU1S ss1= BCKD1S ss0= LRCKU1S
        casex({ss4,ss3,ss2,ss1,ss0})
        5'bxxxxx : sto <=  s11;
        default : sto <=  s10;
        endcase
         end

    10'b1011111111 : begin //s11  ss4= WCNT31 ss3= WCNT15 ss2= BCKU1S ss1= BCKD1S ss0= LRCKU1S
        casex({ss4,ss3,ss2,ss1,ss0})
        5'bxxxxx : sto <=  s3;
        default : sto <=  s11;
        endcase
         end

         endcase
    end
end



///////////////* make decorder for state machines *////////////////////


assign RSTART       = sto[0];
assign WCNTE        = sto[1];
assign XBHE         = sto[2];
assign XWE          = sto[3];
assign XWCE         = sto[4];
assign XBUFOE       = sto[5];


////////////////////////////* counters *//////////////////////////////

reg[15:0] WCNT;
wire[15:0] WADDRS;

always@(negedge XRESET or posedge SYSCLK)
begin

    if (!XRESET)

        WCNT <= 16'b0000;

    else if (!WCNTE | !BCKD1S)

        WCNT <= WCNT;

    else
        WCNT <=WCNT+1;

end

assign RSTART = (WCNT[15:0] == DEPTH) ? 1'b1 : 1'b0;
assign WADDRS = WCNT;


////////////////////////////* shift registers *//////////////////////////////

reg[15:0] WSREG;
wire[15:0] WDATA;

always@(negedge XRESET or posedge SYSCLK)
begin

    if (!XRESET)

        WSREG <= 16'h0000;

    else if (!BCKU1S)

        WSREG <= WSREG;
```

```
    else
        WSREG <= {WSREG[14:0], SDATA};

end

assign WDATA = WSREG;

assign WCNT15 = (WCNT[4:0] == 5'b01111) ? 1'b1 : 1'b0;
assign WCNT31 = (WCNT[4:0] == 5'b11111) ? 1'b1 : 1'b0;




endmodule
```