

QSPI

```

module qspi #()(
input          i_qspi_req_vld,
output         i_qspi_req_rdy,
input         i_qspi_req_read,
input [7:0]    i_qspi_dat,
input         i_qspi_dummy,
input [1:0]    i_qspi_type,

output         o_qspi_rsp_vld,
input         o_qspi_rsp_rdy,
output[7:0]    o_qspi_rsp_dat,

input [1:0]    qspi_param_mode,
input [3:0]    qspi_param_div,
input         qspi_param_duxen,

output         qspi_sck,
output         qspi_dq0_en,
input         qspi_dq0_i,
output         qspi_dq0_o,
output         qspi_dq1_en,
input         qspi_dq1_i,
output         qspi_dq1_o,
output         qspi_dq2_en,
input         qspi_dq2_i,
output         qspi_dq2_o,
output         qspi_dq3_en,
input         qspi_dq3_i,
output         qspi_dq3_o,

input         clk,
input         rst_n
);

wire    qspi_i_hsked    =    i_qspi_req_vld & i_qspi_req_rdy;

// wire    qspi_cpol    =    qspi_param_mode[1];
// wire    qspi_cpha    =    qspi_param_mode[0];

// //CPOL & CPHA
// localparam    POSEDGE    =    1'b0;
// localparam    NEGEDGE    =    1'b1;

```

```

// wire    qspi_buf_vld_r;
// wire    qspi_buf_vld_nxt;
// wire    qspi_buf_vld_set;
// wire    qspi_buf_vld_clr;
// wire    qspi_buf_vld_ena;

// assign  qspi_buf_vld_set    =  qspi_i_hsked;
// assign  qspi_buf_vld_nxt    =  qspi_buf_vld_set | ~qspi_buf_vld_clr;
// assign  qspi_buf_vld_ena    =  qspi_buf_vld_set | qspi_buf_vld_clr;
// dfflr #(1) qspi_buf_vld_dfflr(qspi_buf_vld_ena, qspi_buf_vld_nxt,
qspi_buf_vld_r, clk, rst_n);

wire [7:0]  qspi_buf_dat_r;
wire [7:0]  qspi_buf_dat_nxt;

wire [1:0]  qspi_buf_type_r;
wire [1:0]  qspi_buf_type_nxt;

wire        qspi_buf_we_r;
wire        qspi_buf_we_nxt;

assign      qspi_buf_type_nxt    =  i_qspi_type;
assign      qspi_buf_dat_nxt     =  i_qspi_dat;
assign      qspi_buf_we_nxt      =  ~i_qspi_req_read;

dfflr #(1)  qspi_buf_we_dfflr(qspi_i_hsked, qspi_buf_we_nxt, qspi_buf_we_r,
clk, rst_n);
dfflr #(8)  qspi_buf_dat_dfflr(qspi_i_hsked, qspi_buf_dat_nxt,
qspi_buf_dat_r, clk, rst_n);
dfflr #(2)  qspi_buf_type_dfflr(qspi_i_hsked, qspi_buf_type_nxt,
qspi_buf_type_r, clk, rst_n);

reg [10:0]  sck_div_count;
always@(*) begin
    case (qspi_param_div)
        4'b0000 :      sck_div_count    = 11'h000; //div 2
        4'b0001 :      sck_div_count    = 11'h001; //div 4
        4'b0010 :      sck_div_count    = 11'h007; //div 16
        4'b0011 :      sck_div_count    = 11'h00f; //div 32
        4'b0100 :      sck_div_count    = 11'h003;
        4'b0101 :      sck_div_count    = 11'h01f;
        4'b0110 :      sck_div_count    = 11'h03f;
        4'b0111 :      sck_div_count    = 11'h07f;
        4'b1000 :      sck_div_count    = 11'h0ff;
    endcase
end

```

```

        4'b1001 :      sck_div_count    = 11'h1ff;
        4'b1010 :      sck_div_count    = 11'h3ff;
        4'b1011 :      sck_div_count    = 11'h7ff;
        default :      sck_div_count    = 11'h000;
    endcase
end

wire [10:0] sck_div_count_r;
wire [10:0] sck_div_count_nxt;
wire      sck_div_count_ena;

wire      div_counting_complete    =    (sck_div_count_r == sck_div_count);

assign      sck_div_count_nxt      =    (sck_div_count_r == sck_div_count)
? 0 : sck_div_count_r + 1;

dfflr #(11) sck_div_count_dfflr(sck_div_count_ena, sck_div_count_nxt,
sck_div_count_r, clk, rst_n);


// -----
// -- qspi state
// -----
localparam QSPI_STATE_IDLE      =    2'b00;
localparam QSPI_STATE_DUMMY     =    2'b01;
localparam QSPI_STATE_TRANSF    =    2'b10;

wire [1:0] qspi_state_r;
wire [1:0] qspi_state_nxt;
wire      qspi_state_ena;

dfflr #(2) qspi_state_dfflr(qspi_state_ena, qspi_state_nxt, qspi_state_r,
clk, rst_n);

wire      qspi_state_is_idle     =    (qspi_state_r == QSPI_STATE_IDLE);
wire      qspi_state_is_transfer =    (qspi_state_r ==
QSPI_STATE_TRANSF);
wire      qspi_state_is_dummy    =    (qspi_state_r == QSPI_STATE_DUMMY);


// -----
// -- bit transfer state
// -----

assign      sck_div_count_ena      =    qspi_state_is_transfer |
qspi_state_is_dummy;

```

```

localparam SCK_LOW = 1'b0;
localparam SCK_HIGH= 1'b1;

wire      bit_state_r;
wire      bit_state_nxt;
wire      bit_state_ena;
wire      bit_state_is_sck_high  = (bit_state_r == SCK_HIGH);
wire      bit_state_is_sck_low   = (bit_state_r == SCK_LOW);

assign     bit_state_ena          = div_counting_complete |
qspi_state_is_idle;
assign     bit_state_nxt          = qspi_state_is_idle & (

((qspi_param_mode == 2'b00) | (qspi_param_mode == 2'b11)) & SCK_LOW
|
((qspi_param_mode == 2'b10) | (qspi_param_mode == 2'b01)) & SCK_HIGH)
|qspi_state_is_transfer &
~bit_state_r
|qspi_state_is_dummy &
~bit_state_r;

dfflr #(1) bit_state_dfflr(bit_state_ena, bit_state_nxt, bit_state_r, clk,
rst_n);

//bit tx/rx count
wire [7:0] bit_count_r;
wire [7:0] bit_count_nxt;
wire      bit_count_ena;

assign     bit_count_nxt          = {8{qspi_state_is_transfer}} & (

{8{~qspi_i_hsked}} & (

({8{qspi_buf_type_r == 2'b00}} & (bit_count_r + 1))

|   ({8{qspi_buf_type_r == 2'b01}} & (bit_count_r + 2))

|   ({8{qspi_buf_type_r == 2'b10}} & (bit_count_r + 4))

)

|

{8{qspi_i_hsked}} & 8'h00

)

|   {8{qspi_state_is_dummy}} & (

{8{~qspi_i_hsked}} & (bit_count_r + 1)

|

{8{qspi_i_hsked}} & 8'h00

)

|   {8{qspi_state_is_idle}} &

8'd0;

```

```

assign      bit_count_ena      =  qspi_state_is_idle
                                |  qspi_state_is_transfer & (

((qspi_param_mode == 2'b00) | (qspi_param_mode == 2'b11)) &
bit_state_is_sck_high& bit_state_ena
                                |

((qspi_param_mode == 2'b01) | (qspi_param_mode == 2'b10)) &
bit_state_is_sck_low & bit_state_ena
                                )

                                |  qspi_state_is_dummy      & (

((qspi_param_mode == 2'b00) | (qspi_param_mode == 2'b11)) &
bit_state_is_sck_high& bit_state_ena
                                |

((qspi_param_mode == 2'b01) | (qspi_param_mode == 2'b10)) &
bit_state_is_sck_low & bit_state_ena
                                );

dfflr #(8)  bit_count_dfflr(bit_count_ena, bit_count_nxt, bit_count_r, clk,
rst_n);

// -----
// -- bit tx/rx
// -----
wire [7:0]  qspi_bit_idx      =  8'b10000000 >> bit_count_r;

//rx
wire [7:0]  qspi_dat_rx_r;
wire [7:0]  qspi_dat_rx_nxt;
wire [7:0]  qspi_dat_rx_ena;

wire        qspi_dat_rx_vld_r;
wire        qspi_dat_rx_vld_nxt;
wire        qspi_dat_rx_vld_set;
wire        qspi_dat_rx_vld_clr;
wire        qspi_dat_rx_vld_ena;

assign      qspi_dat_rx_vld_set      =  qspi_state_is_transfer & (

(qspi_buf_type_r == 2'b00) & (//standard SPI

((qspi_param_mode == 2'b00) | (qspi_param_mode == 2'b11)) &
bit_state_is_sck_low & bit_state_ena & (bit_count_r == 7)

|  ((qspi_param_mode == 2'b01) | (qspi_param_mode == 2'b10)) &

```

```

bit_state_is_sck_high& bit_state_ena & (bit_count_r == 7)

) & (~qspi_buf_we_r | qspi_buf_we_r & qspi_param_duxen)

(qspi_buf_type_r == 2'b01) & (//dual SPI

((qspi_param_mode == 2'b00) | (qspi_param_mode == 2'b11)) &
bit_state_is_sck_low & bit_state_ena & (bit_count_r == 6)

| ((qspi_param_mode == 2'b01) | (qspi_param_mode == 2'b10)) &
bit_state_is_sck_high& bit_state_ena & (bit_count_r == 6)

) & ~qspi_buf_we_r

(qspi_buf_type_r == 2'b10) & (//quad SPI

((qspi_param_mode == 2'b00) | (qspi_param_mode == 2'b11)) &
bit_state_is_sck_low & bit_state_ena & (bit_count_r == 4)

| ((qspi_param_mode == 2'b01) | (qspi_param_mode == 2'b10)) &
bit_state_is_sck_high& bit_state_ena & (bit_count_r == 4)

) & ~qspi_buf_we_r

);
wire          qspi_rsp_hsked          =  o_qspi_rsp_vld & o_qspi_rsp_rdy;

assign        qspi_dat_rx_vld_clr      =  qspi_rsp_hsked;
assign        qspi_dat_rx_vld_ena      =  qspi_dat_rx_vld_set |
qspi_dat_rx_vld_clr;
assign        qspi_dat_rx_vld_nxt      =  qspi_dat_rx_vld_set |
~qspi_dat_rx_vld_clr;
dfflr #(1)    qspi_dat_rx_vld_dfflr(qspi_dat_rx_vld_ena, qspi_dat_rx_vld_nxt,
qspi_dat_rx_vld_r, clk, rst_n);

assign        o_qspi_rsp_vld           =  qspi_dat_rx_vld_r;

assign        o_qspi_rsp_dat           =  qspi_dat_rx_r;

genvar i;
generate
    for(i=0; i<8; i=i+1) begin
        assign qspi_dat_rx_ena[i] =  qspi_state_is_transfer & (

(qspi_buf_type_r == 2'b00) & (//standard SPI

((qspi_param_mode == 2'b00) | (qspi_param_mode == 2'b11)) &
bit_state_is_sck_low & bit_state_ena

| ((qspi_param_mode == 2'b10) | (qspi_param_mode == 2'b01)) &
bit_state_is_sck_high& bit_state_ena

) & (qspi_param_duxen | ~qspi_buf_we_r) & (qspi_bit_idx == (8'b00000001 <<
i))

```

```

(qspi_buf_type_r == 2'b01) & (//Dual SPI

((qspi_param_mode == 2'b00) | (qspi_param_mode == 2'b11)) &
bit_state_is_sck_low & bit_state_ena

| ((qspi_param_mode == 2'b01) | (qspi_param_mode == 2'b10)) &
bit_state_is_sck_high& bit_state_ena

) & ~qspi_buf_we_r & ((qspi_bit_idx == (8'b00000001 << i)) | ((qspi_bit_idx
>> 1) == (8'b00000001 << i)))

(qspi_buf_type_r == 2'b10) & (//Quad SPI

((qspi_param_mode == 2'b00) | (qspi_param_mode == 2'b11)) &
bit_state_is_sck_low & bit_state_ena

| ((qspi_param_mode == 2'b01) | (qspi_param_mode == 2'b10)) &
bit_state_is_sck_high& bit_state_ena

) & ~qspi_buf_we_r & (

((qspi_bit_idx) == (8'b00000001 << i))

| ((qspi_bit_idx >> 1) == (8'b00000001 << i))

| ((qspi_bit_idx >> 2) == (8'b00000001 << i))

| ((qspi_bit_idx >> 3) == (8'b00000001 << i))

)

);
assign qspi_dat_rx_nxt[i] = (qspi_buf_type_r == 2'b00) &
qspi_dq0_i
| (qspi_buf_type_r == 2'b01) & (
((8'b00000001 << i) == (qspi_bit_idx)) & qspi_dq1_i
|
((8'b00000001 << i) == (qspi_bit_idx >> 1)) & qspi_dq0_i
)
| (qspi_buf_type_r == 2'b10) & (
((8'b00000001 << i) == (qspi_bit_idx)) & qspi_dq3_i //7
|
((8'b00000001 << i) == (qspi_bit_idx >> 1)) & qspi_dq2_i //6
|
((8'b00000001 << i) == (qspi_bit_idx >> 2)) & qspi_dq1_i //5
|
((8'b00000001 << i) == (qspi_bit_idx >> 3)) & qspi_dq0_i //4
);
dfflr #(1) qspi_dat_rx_dfflr(qspi_dat_rx_ena[i],
qspi_dat_rx_nxt[i], qspi_dat_rx_r[i], clk, rst_n);
end
endgenerate

```

```

wire      qspi_state_idle_ena    = qspi_state_is_idle & qspi_i_hsked;
wire      qspi_state_dummy_ena   = qspi_state_is_dummy    & (

((qspi_param_mode == 2'b00) | (qspi_param_mode == 2'b11)) &
bit_state_is_sck_high & bit_state_ena

|

((qspi_param_mode == 2'b01) | (qspi_param_mode == 2'b10)) &
bit_state_is_sck_low  & bit_state_ena

) &

(bit_count_r == qspi_buf_dat_r);
wire      qspi_state_transfer_ena = qspi_state_is_transfer & (

(qspi_buf_type_r == 2'b00) & (//standard SPI

((qspi_param_mode == 2'b00) | (qspi_param_mode == 2'b11)) &
bit_state_is_sck_high& bit_state_ena & (bit_count_r == 7)

|

((qspi_param_mode == 2'b01) | (qspi_param_mode == 2'b10)) &
bit_state_is_sck_low & bit_state_ena & (bit_count_r == 7)

)

|

(qspi_buf_type_r == 2'b01) & (//dual SPI

((qspi_param_mode == 2'b00) | (qspi_param_mode == 2'b11)) &
bit_state_is_sck_high& bit_state_ena & (bit_count_r == 6)

|

((qspi_param_mode == 2'b01) | (qspi_param_mode == 2'b10)) &
bit_state_is_sck_low & bit_state_ena & (bit_count_r == 6)

)

|

(qspi_buf_type_r == 2'b10) & (//quad SPI

((qspi_param_mode == 2'b00) | (qspi_param_mode == 2'b11)) &
bit_state_is_sck_high& bit_state_ena & (bit_count_r == 4)

|

((qspi_param_mode == 2'b01) | (qspi_param_mode == 2'b10)) &
bit_state_is_sck_low & bit_state_ena & (bit_count_r == 4)

)

);

assign      qspi_state_ena        = qspi_state_idle_ena |
qspi_state_dummy_ena | qspi_state_transfer_ena;

wire [1:0]  qspi_state_idle_nxt    = qspi_dat_rx_vld_r & ~qspi_rsp_hsked
? QSPI_STATE_IDLE :

i_qspi_dummy ? QSPI_STATE_DUMMY :

QSPI_STATE_TRANSF;

```



```

wire [1:0]  qspi_state_dummy_nxt    =  {2{qspi_i_hsked & i_qspi_dummy}} &
QSPI_STATE_DUMMY
                                     |  {2{qspi_i_hsked & ~i_qspi_dummy}}&
QSPI_STATE_TRANSF
                                     |  {2{~qspi_i_hsked}}                &
QSPI_STATE_IDLE;

wire [1:0]  qspi_state_transfer_nxt =  {2{qspi_i_hsked & i_qspi_dummy}} &
QSPI_STATE_DUMMY
                                     |  {2{qspi_i_hsked & ~i_qspi_dummy}}&
QSPI_STATE_TRANSF
                                     |  {2{~qspi_i_hsked}}                &
QSPI_STATE_IDLE;

assign      qspi_state_nxt          =  {2{qspi_state_is_idle}} &
qspi_state_idle_nxt
                                     |  {2{qspi_state_is_dummy}}&
qspi_state_dummy_nxt
                                     |  {2{qspi_state_is_transfer}} &
qspi_state_transfer_nxt;

assign      i_qspi_req_rdy          =  qspi_state_is_idle &
(~qspi_dat_rx_vld_r | qspi_dat_rx_vld_clr)
                                     |  qspi_state_dummy_ena
                                     |  qspi_state_transfer_ena;

//tx
assign      qspi_sck                =  bit_state_is_sck_high;

assign      qspi_dq0_o              =  (qspi_buf_type_r == 2'b00) & |
(qspi_buf_dat_r & qspi_bit_idx)
                                     |  (qspi_buf_type_r == 2'b01) & |
(qspi_buf_dat_r & (qspi_bit_idx >> 1))
                                     |  (qspi_buf_type_r == 2'b10) & |
(qspi_buf_dat_r & (qspi_bit_idx >> 3));

assign      qspi_dq0_en             =  qspi_state_is_transfer;

assign      qspi_dq1_o              =  (qspi_buf_type_r == 2'b01) & |
(qspi_buf_dat_r & qspi_bit_idx)
                                     |  (qspi_buf_type_r == 2'b10) & |
(qspi_buf_dat_r & (qspi_bit_idx >> 2));

assign      qspi_dq1_en             =  qspi_state_is_transfer &
(qspi_buf_type_r != 2'b00);

assign      qspi_dq2_o              =  |(qspi_buf_dat_r & (qspi_bit_idx >>
1));

assign      qspi_dq2_en             =  qspi_state_is_transfer &

```

```
(qspi_buf_type_r == 2'b10);

assign      qspi_dq3_o          =  |(qspi_buf_dat_r & qspi_bit_idx);

assign      qspi_dq3_en        =  qspi_state_is_transfer &
(qspi_buf_type_r == 2'b10);

endmodule
```