仿真环境

Quartusii18.1+Modelsim-ase

SDRAM仿真模型

1. SDRAM仿真模型可从镁光官网下载

访问连接：https://www.micron.com/search-results?searchRequest=%7b%22Filters%22%3a%5b%7b%22Ids%22%3a%5b%22c414a34f-01cc-4847-b83e-a313f3d47748%22%5d%2c%22QueryToken%22%3a%22tch%22%2c%22UseLogicalOr%22%3afalse%7d%5d%7d

直接下载链接：https://media-www.micron.com/-/media/client/global/documents/products/sim-model/dram/sdram/sdr_sdram.zip?rev=2d8a2f168e654ac495819c520aa00da4

2. 修改仿真模型参数定义文件sdr_parameters.vh

增加如下宏定义：

```
28  ********************************************************************************
29  `define sg6a
30  `define den64Mb
31  `define x16
```

3. 调用仿真模型

下载的仿真模型包含两个sdram模型文件，我们使用sdr.v这个模型，sdr_module.v这个模型未使用，test.v文件也未使用。

调试记录

1. 时序报错：hold时间不满足

```
# ** Error: D:/develop/mygit/vip_sdram_sim/par/../src/sim_models/Micron/sdram_16bits_64m/sdr.v(1068): $hold( posedge Clk:337565 ns, Addr:337565 ns, 800 ps );
#    Time: 337565 ns  Iteration: 3  Instance: /Sdram_Control_4Port_vlg_tst/u_sdr_0
# Sdram_Control_4Port_vlg_tst.u_sdr_0 : at time  337575.0 ns ACT  : Bank = 0 Row =    0
# ** Error: D:/develop/mygit/vip_sdram_sim/par/../src/sim_models/Micron/sdram_16bits_64m/sdr.v(1068): $hold( posedge Clk:337575 ns, Addr:337575 ns, 800 ps );
#    Time: 337575 ns  Iteration: 4  Instance: /Sdram_Control_4Port_vlg_tst/u_sdr_0
# ** Error: D:/develop/mygit/vip_sdram_sim/par/../src/sim_models/Micron/sdram_16bits_64m/sdr.v(1066): $hold( posedge Clk:337575 ns, Ras_n:337575 ns, 800 ps );
#    Time: 337575 ns  Iteration: 4  Instance: /Sdram_Control_4Port_vlg_tst/u_sdr_0
# ** Error: D:/develop/mygit/vip_sdram_sim/par/../src/sim_models/Micron/sdram_16bits_64m/sdr.v(1067): $hold( posedge Clk:337595 ns, We_n:337595 ns, 800 ps );
#    Time: 337595 ns  Iteration: 4  Instance: /Sdram_Control_4Port_vlg_tst/u_sdr_0
# ** Error: D:/develop/mygit/vip_sdram_sim/par/../src/sim_models/Micron/sdram_16bits_64m/sdr.v(1065): $hold( posedge Clk:337595 ns, Cas_n:337595 ns, 800 ps );
#    Time: 337595 ns  Iteration: 4  Instance: /Sdram_Control_4Port_vlg_tst/u_sdr_0
# Sdram_Control_4Port_vlg_tst.u_sdr_0 : at time  337605.0 ns WRITE: Bank = 0 Row =    0, Col =  64, Data = 4040
# ** Error: D:/develop/mygit/vip_sdram_sim/par/../src/sim_models/Micron/sdram_16bits_64m/sdr.v(1071): $hold( posedge Dq_chk:337605 ns, Dq:337605 ns, 800 ps );
#    Time: 337605 ns  Iteration: 5  Instance: /Sdram_Control_4Port_vlg_tst/u_sdr_0
# ** Error: D:/develop/mygit/vip_sdram_sim/par/../src/sim_models/Micron/sdram_16bits_64m/sdr.v(1067): $hold( posedge Clk:337605 ns, We_n:337605 ns, 800 ps );
#    Time: 337605 ns  Iteration: 5  Instance: /Sdram_Control_4Port_vlg_tst/u_sdr_0
# ** Error: D:/develop/mygit/vip_sdram_sim/par/../src/sim_models/Micron/sdram_16bits_64m/sdr.v(1065): $hold( posedge Clk:337605 ns, Cas_n:337605 ns, 800 ps );
#    Time: 337605 ns  Iteration: 5  Instance: /Sdram_Control_4Port_vlg_tst/u_sdr_0
```

解决方法：将给SDRAM模型的时钟相对于SDRAM控制器的时钟相移-72°

```
147    //线网连接
148    always@(*)
149    clk =  #2 OUT_CLK;  //SDRAM芯片的工作时钟相对于SDRAM控制器工作时钟相移-72°，本行意思是取当前OUT_CLK的值，2ns后赋给clk,波形上看就是clk滞后OUT_CLK 2ns
150    //OUT_CLK =  #2 clk;
```

2. LOAD_MODE命名未生效

解决办法：修改SDRAM控制器Sdram_Control_4Port.v文件

Line503：添加判断条件Sdram_Init_Done，防止SDRAM初始化阶段就开始产生读指令，从而导致读命令阻塞了LOAD_MODE命令的执行

```
491    // Auto Read/Write Control
492    always@(posedge CLK or negedge RESET_N)
493    begin
494        if(!RESET_N)
495        begin
496            mWR       <= 0;
497            mRD       <= 0;
498            mADDR <= 0;
499            mLENGTH   <= 0;
500            RD_MASK <= 0;
501            WR_MASK <= 0;
502        end
503        else if(Sdram_Init_Done)
504        begin
505            if( (mWR==0) && (mRD==0) && (ST==0) &&
506                (WR_MASK==0)    && (RD_MASK==0) &&
507                (WR1_LOAD==0)   && (RD1_LOAD==0) &&
508                (WR2_LOAD==0)   && (RD2_LOAD==0) )
509            begin
510                // Read Side 1
511                if( (read_side_fifo_wusedw1 < rRD1_LENGTH) )
512                begin
513                    mADDR <= rRD1_ADDR;
514                    mLENGTH   <= rRD1_LENGTH;
515                    WR_MASK   <= 2'b00;
516                    RD_MASK   <= 2'b01;
517                    mWR       <= 0;
518                    mRD       <= 1;
519                end
```

3. 其他改动

修改Sdram_Control_4Port.v文件：Line500~501 增加RD_MASK和WR_MASK复位赋值，防止X态

```
497            mRD       <= 0;
498            mADDR <= 0;
499            mLENGTH   <= 0;
500            RD_MASK <= 0;
501            WR_MASK <= 0;
502        end
503        else if(Sdram_Init_Done)
504        begin
505            if( (mWR==0) && (mRD==0) && (ST==0) &&
506                (WR_MASK==0)    && (RD_MASK==0) &&
```

仿真激励和结果

仿真激励代码：

```
1  `timescale 1 ns/ 100 ps
2  module Sdram_Control_4Port_vlg_tst();
3  // constants
4  // general purpose registers
5  // test vector input registers
6  reg clk;
```

```
reg wr_rd_clk;
reg OUT_CLK;
reg RD1;
reg RD2;
reg RESET_N;
reg WR1;
wire [15:0] WR1_DATA;
//reg WR2;
//reg [15:0] WR2_DATA;
// wires
wire [1:0]  BA;
wire CAS_N;
wire CKE;
wire [1:0]  CS_N;
wire [15:0]  DQ;
wire [1:0]  DQM;
wire RAS_N;
wire [15:0]  RD1_DATA;
wire RD1_EMPTY;
wire [9:0]  RD1_USE;
wire [15:0]  RD2_DATA;
wire RD2_EMPTY;
wire [9:0]  RD2_USE;
wire [11:0]  SA;
wire SDR_CLK;
wire Sdram_Init_Done;
wire WE_N;
wire WR1_FULL;
wire [9:0]  WR1_USE;
wire WR2_FULL;
wire [9:0]  WR2_USE;
// assign statements (if any)
Sdram_Control_4Port i1 (
// port map - connection between master ports and signals/registers
        .BA(BA),
        .CAS_N(CAS_N),
        .CKE(CKE),
        .CS_N(CS_N),
        .DQ(DQ),
        .DQM(DQM),
```

```
47          .OUT_CLK(OUT_CLK),
48          .RAS_N(RAS_N),
49          .RD1(RD1),
50          .RD1_ADDR(0),//起始地址
51          .RD1_CLK(wr_rd_clk),
52          .RD1_DATA(RD1_DATA),
53          .RD1_EMPTY(RD1_EMPTY),
54          .RD1_LENGTH(9'd32),//突发长度
55          .RD1_LOAD(~RESET_N),
56          .RD1_MAX_ADDR(256),
57          .RD1_USE(RD1_USE),
58          .RD2(RD2),
59          .RD2_ADDR(256),//起始地址
60          .RD2_CLK(wr_rd_clk),
61          .RD2_DATA(RD2_DATA),
62          .RD2_EMPTY(RD2_EMPTY),
63          .RD2_LENGTH(9'd32),//突发长度
64          .RD2_LOAD(~RESET_N),
65          .RD2_MAX_ADDR(512),//结束地址
66          .RD2_USE(RD2_USE),
67          .REF_CLK(clk),
68          .RESET_N(RESET_N),
69          .SA(SA),
70          .SDR_CLK(SDR_CLK),
71          .Sdram_Init_Done(Sdram_Init_Done),
72          .WE_N(WE_N),
73          .WR1(WR1),
74          .WR1_ADDR(0),
75          .WR1_CLK(wr_rd_clk),
76          .WR1_DATA(WR1_DATA),
77          .WR1_FULL(WR1_FULL),
78          .WR1_LENGTH(9'd32),
79          .WR1_LOAD(~RESET_N),
80          .WR1_MAX_ADDR(256),
81          .WR1_USE(WR1_USE),
82          .WR2(1'b0),//(WR2),
83          .WR2_ADDR(256),
84          .WR2_CLK(wr_rd_clk),
85          .WR2_DATA(16'd0),//(WR2_DATA),
86          .WR2_FULL(WR2_FULL),
```

```
        .WR2_LENGTH(9'd32),
        .WR2_LOAD(~RESET_N),
        .WR2_MAX_ADDR(512),
        .WR2_USE(WR2_USE)
);
sdr u_sdr_0 (
    .Dq(DQ),
         .Addr(SA),
         .Ba(BA),
         .Clk(SDR_CLK),
         .Cke(CKE),
         .Cs_n(CS_N),
         .Ras_n(RAS_N),
         .Cas_n(CAS_N),
         .We_n(WE_N),
         .Dqm(DQM)
         );
initial
begin
clk = 0;
RESET_N = 0;
RD1 = 0;
RD2 = 0;
WR1 = 0;
wr_rd_clk = 0;
OUT_CLK = 0;
//WR2 = 0;
#1000  RESET_N = 1;
end
//生成50Mhz时钟
//always #5 clk = ~clk;
always #5 OUT_CLK = ~OUT_CLK;
//生成读写慢速时钟
always #25 wr_rd_clk = ~wr_rd_clk;
//线网连接
always@(*)
clk =  #2 OUT_CLK;  //SDRAM芯片的工作时钟相对于SDRAM控制器工作时钟相移-72°，本行意思是取当前OUT_CLK的值，2ns后赋给clk,波形上看就是clk滞后OUT_CLK 2ns
//OUT_CLK =  #2 clk;
//产生写逻辑
```

```
126 reg [7:0] wr1_num;
127 initial
128 begin
129 wr1_num = 0;
130 end
131 assign WR1_DATA = {wr1_num,wr1_num};
132 always@(posedge wr_rd_clk or negedge RESET_N)
133 begin
134 if(!RESET_N)
135     begin
136          wr1_num <= 0;
137          WR1     <= 1'b0;
138          end
139 else
140     begin
141          WR1 <= ~WR1_FULL && Sdram_Init_Done;
142          wr1_num <= wr1_num+WR1;
143          end
144 end
145 //产生读逻辑
146 always@(*)
147 RD1 = ~RD1_EMPTY;
148 endmodule
```

如下图所示，循环写入256个数：0x0000-0x0101-0x0202-...-0xffff，同时循环读出相同地址的数据，可以看出循环读出的数据和写入的数据比对一致，说明仿真结果正确。