

高级练习5 用户端实现 DDR3 SDRAM 写、读控制

笔记本: FPGA练习

创建时间: 2020/2/3 19:40

更新时间: 2020/2/4 14:59

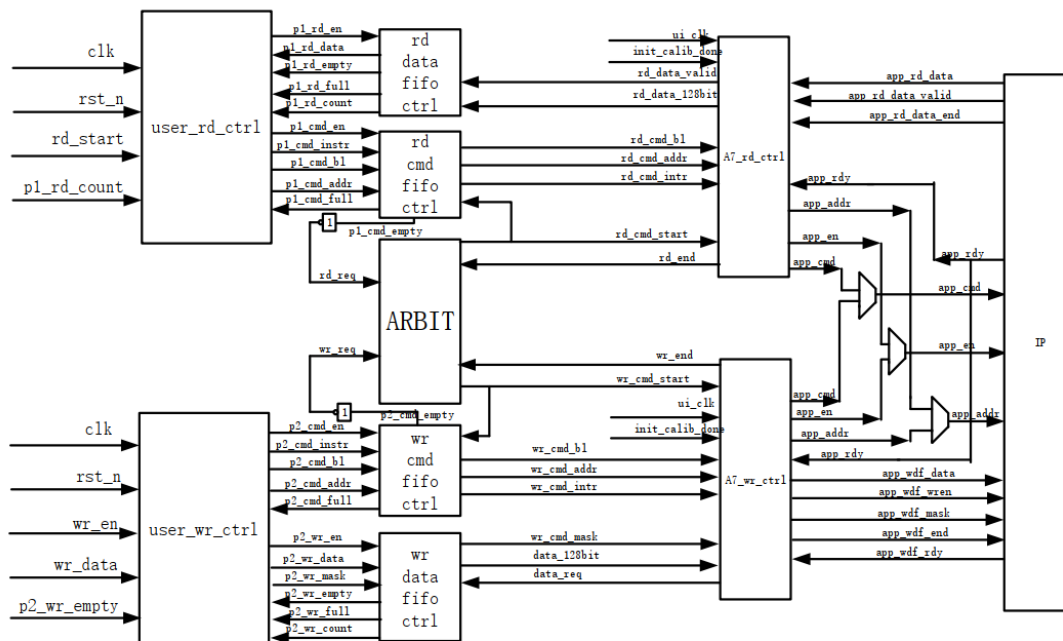
作者: 2322900041@qq.com

高级练习5 用户端实现 DDR3 SDRAM 写、读控制

一、练习内容

在已经实现的DDR3控制器的基础上, 实现用户的读写控制。

二、系统框图



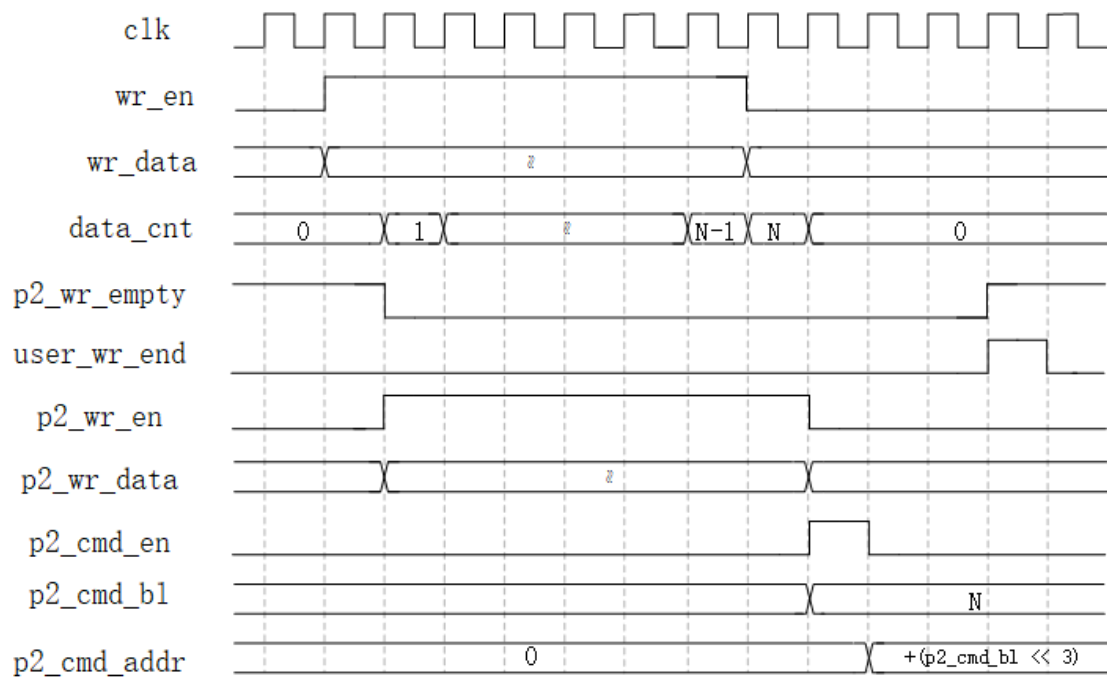
三、设计分析

1. 写时序分析

(1) 关键点时序

- 数据FIFO写入完毕后, 从数据FIFO中读取数据, 即此时开始执行写入数据的命令, 这部分的控制采用*wr_en*来把控;
- 数据FIFO数据完毕后, 一帧数据读取完毕, 拉起写完成标志, 即*wr_end*拉高;

(2) 时序波形



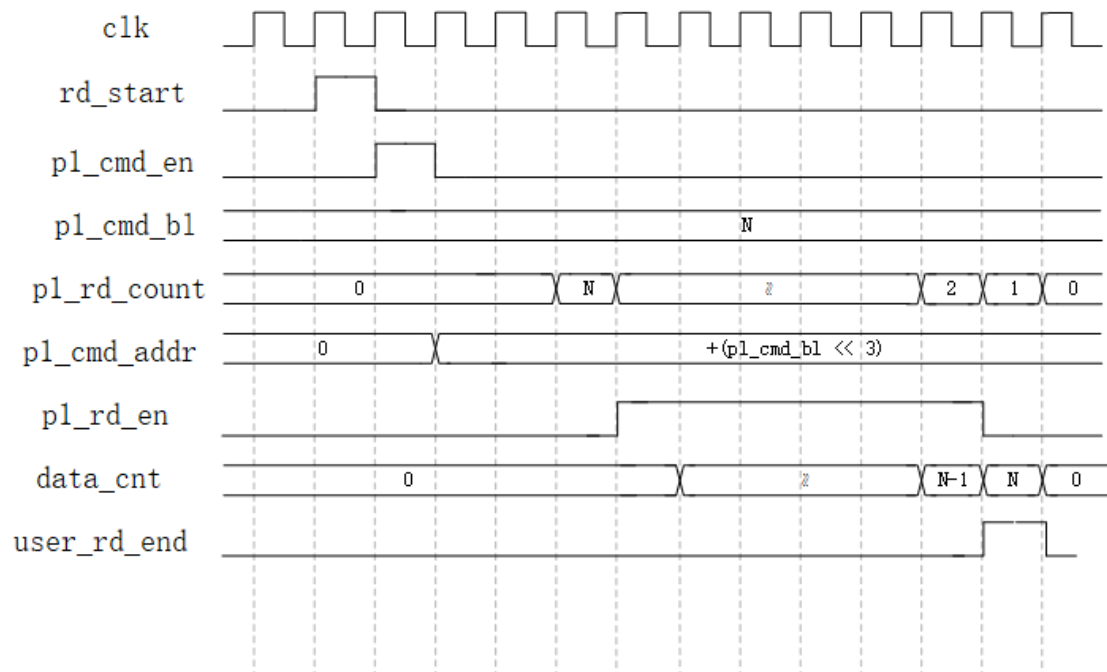
写时序控制

2. 读时序分析

(1) 关键点时序

- rd_start后，立即写入读取数据的命令，命令中cmd_b1是固定的；
- 等待数据FIFO中存储的数据和要求读取的数据一致时，开始读取数据，即pl_rd_en拉高；

(2) 时序波形



读时序控制

3. 仿真实现分析

(1) 写数据

写完64突发长度的数据后，需等待wr_end拉高后，再次写入64突发长度的数据。

(2) 读数据

读完64突发长度的数据后，需等待rd_end拉高后，再次读取64突发长度的数据。

四、练习步骤

1. 按照分析的时序编写代码
 2. 按照仿真分析编写测试代码
- (1) 写仿真

```
task gen_wr_data;
    integer i;
    begin
        @ (negedge rst);
        repeat (100) @ (posedge clk);
        wr_en = 1;
        for (i = 0; i < 64; i = i + 1)
            begin
                wr_data = i;
                @ (posedge clk);
            end
        wr_en = 0;
        @ (posedge clk);
        @ (negedge user_wr_end);
        repeat (100) @ (posedge clk);
        wr_en = 1;
        for (i = 0; i < 64; i = i + 1)
            begin
                wr_data = i + 64;
                @ (posedge clk);
            end
        wr_en = 0;
    end
endtask
```

(2) 读仿真

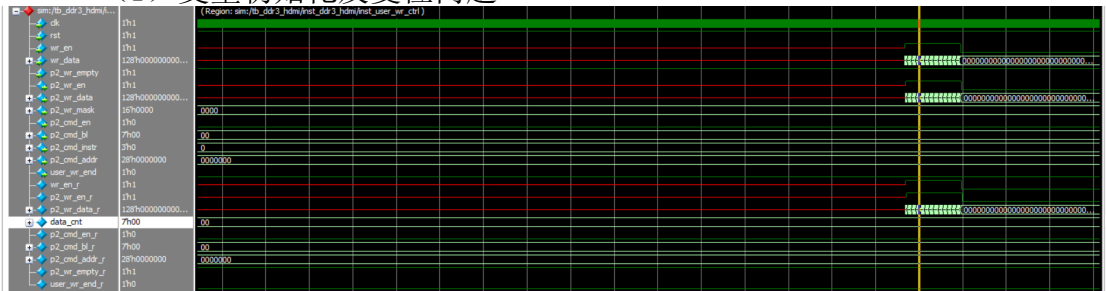
```
task gen_rd_data;
    begin
        @ (negedge rst);
        @ (negedge user_wr_end);
        @ (negedge user_wr_end);    //等待写入完毕
        rd_start = 1;
        @ (posedge clk);
        rd_start = 0;
        @ (posedge clk);
        @ (negedge user_rd_end);    //等待读取完毕1
        rd_start = 1;
        @ (posedge clk);
        rd_start = 0;
        @ (posedge clk);
    end
endtask
```

3. 查看仿真结果
- (1) 对比时序是否正确
 - (2) 对比数据是否正常

五、实际波形仿真

1. 写时序仿真测试

(1) 变量初始化及复位问题

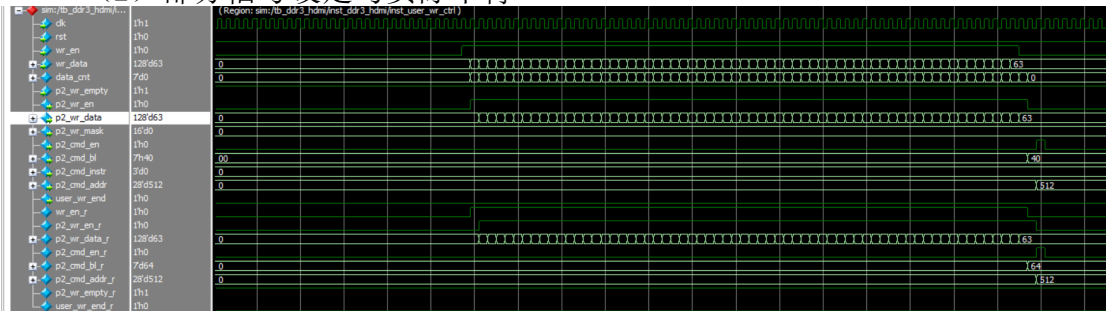


变量未正确的初始化，即某些变量并未发生改变，一致保持不变，这就是复位出现了错误，导致一致保持为零，这里需要进行简单的修改以达到要求。

```
85  
86  
87 wr_en = 0;  
88 wr_data = 0;  
rd_start = 0;
```

首先，在testbench文件中添加相应的变量初始化代码；
然后，修改为高电平的复位已达到要求。
再重新查看仿真结果。

(2) 部分信号设定与实际不符



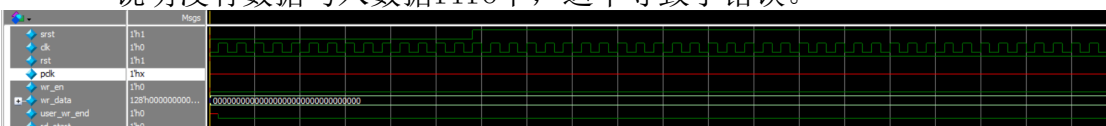
- `p2_cmd_en`与设计时序不符

```
86 //p2_cmd_en  
87 always @(posedge clk)  
88 begin  
89     if (rst == 1'b1)  
90         p2_cmd_en_r <= 1'b0;  
91     else if (wr_en == 1'b0 && wr_en_r == 1'b1) //下降沿检测  
92         p2_cmd_en_r <= 1'b1;  
93     else  
94         p2_cmd_en_r <= 1'b0;  
95 end
```

发现`p2_cmd_en`波形与实际设计波形相比，滞后了一个时钟周期，这里将其修改为上述代码，即可达标。

- 数据fifo一直为空的

说明没有数据写入数据fifo中，这个导致了错误。



经检查，是时钟信号出现了问题，一直没有时钟信号，故没有数据输入。

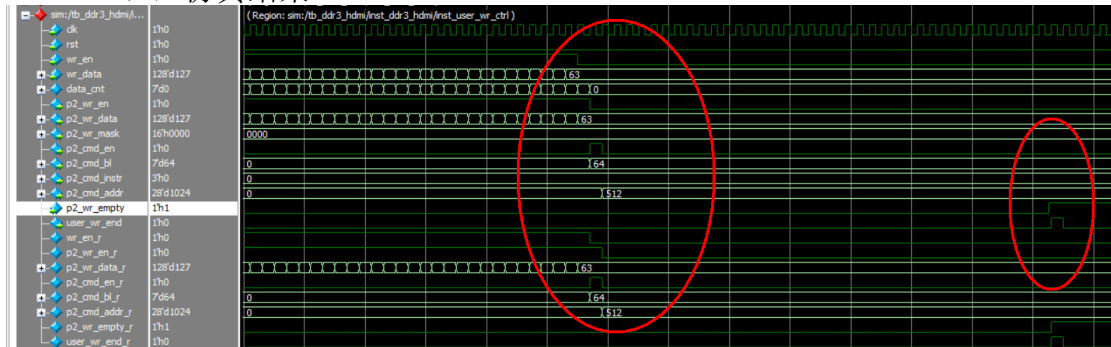
```

129 //fifo clock
130 initial begin
131     pclk = 0;
132     forever #(10) pclk = ~pclk;
133 end

```

发现是时钟信号没有初始化，增加初始化代码，再次进行仿真。

(3) 仿真结果



实际波形与仿真波形完全一致，这部分时序基本上没有什么问题。

```

# tb_ddr3_hdm1.inst_ddr3_model.main: at time 115118314.0 ps INFO: Sync On Die Termination Rtt_NGM = 60 Ohm
# tb_ddr3_hdm1.inst_ddr3_model.data_task: at time 115127064.0 ps INFO: WRITE @ DQS= bank = 0 row = 0000 col = 000003f8 data = 007f
# tb_ddr3_hdm1.inst_ddr3_model.data_task: at time 115128314.0 ps INFO: WRITE @ DQS= bank = 0 row = 0000 col = 000003f9 data = 0000
# tb_ddr3_hdm1.inst_ddr3_model.data_task: at time 115129564.0 ps INFO: WRITE @ DQS= bank = 0 row = 0000 col = 000003fa data = 0000
# tb_ddr3_hdm1.inst_ddr3_model.data_task: at time 115130814.0 ps INFO: WRITE @ DQS= bank = 0 row = 0000 col = 000003fb data = 0000
# tb_ddr3_hdm1.inst_ddr3_model.data_task: at time 115132064.0 ps INFO: WRITE @ DQS= bank = 0 row = 0000 col = 000003fc data = 0000
# tb_ddr3_hdm1.inst_ddr3_model.data_task: at time 115133314.0 ps INFO: WRITE @ DQS= bank = 0 row = 0000 col = 000003fd data = 0000
# tb_ddr3_hdm1.inst_ddr3_model.data_task: at time 115134564.0 ps INFO: WRITE @ DQS= bank = 0 row = 0000 col = 000003fe data = 0000
# tb_ddr3_hdm1.inst_ddr3_model.data_task: at time 115135814.0 ps INFO: WRITE @ DQS= bank = 0 row = 0000 col = 000003ff data = 0000
# tb_ddr3_hdm1.inst_ddr3_model.data_task: at time 110267064.0 ps INFO: WRITE @ DQS= bank = 0 row = 0000 col = 00000000 data = 0000
# tb_ddr3_hdm1.inst_ddr3_model.data_task: at time 110268314.0 ps INFO: WRITE @ DQS= bank = 0 row = 0000 col = 00000001 data = 0000
# tb_ddr3_hdm1.inst_ddr3_model.data_task: at time 110269564.0 ps INFO: WRITE @ DQS= bank = 0 row = 0000 col = 00000002 data = 0000
# tb_ddr3_hdm1.inst_ddr3_model.data_task: at time 110270814.0 ps INFO: WRITE @ DQS= bank = 0 row = 0000 col = 00000003 data = 0000
# tb_ddr3_hdm1.inst_ddr3_model.data_task: at time 110272064.0 ps INFO: WRITE @ DQS= bank = 0 row = 0000 col = 00000004 data = 0000
# tb_ddr3_hdm1.inst_ddr3_model.data_task: at time 110273314.0 ps INFO: WRITE @ DQS= bank = 0 row = 0000 col = 00000005 data = 0000
# tb_ddr3_hdm1.inst_ddr3_model.cmd_task: at time 110273314.0 ps INFO: Write bank 0 col 010, auto precharge 0
# tb_ddr3_hdm1.inst_ddr3_model.data_task: at time 110274564.0 ps INFO: WRITE @ DQS= bank = 0 row = 0000 col = 00000006 data = 0000
# tb_ddr3_hdm1.inst_ddr3_model.data_task: at time 110275814.0 ps INFO: WRITE @ DQS= bank = 0 row = 0000 col = 00000007 data = 0000
# tb_ddr3_hdm1.inst_ddr3_model.data_task: at time 110277064.0 ps INFO: WRITE @ DQS= bank = 0 row = 0000 col = 00000008 data = 0001
# tb_ddr3_hdm1.inst_ddr3_model.data_task: at time 110278314.0 ps INFO: WRITE @ DQS= bank = 0 row = 0000 col = 00000009 data = 0000
# tb_ddr3_hdm1.inst_ddr3_model.data_task: at time 110279564.0 ps INFO: WRITE @ DQS= bank = 0 row = 0000 col = 0000000a data = 0000
# tb_ddr3_hdm1.inst_ddr3_model.data_task: at time 110280814.0 ps INFO: WRITE @ DQS= bank = 0 row = 0000 col = 0000000b data = 0000
# tb_ddr3_hdm1.inst_ddr3_model.data_task: at time 110282064.0 ps INFO: WRITE @ DQS= bank = 0 row = 0000 col = 0000000c data = 0000
# tb_ddr3_hdm1.inst_ddr3_model.data_task: at time 110283314.0 ps INFO: WRITE @ DQS= bank = 0 row = 0000 col = 0000000d data = 0000
# tb_ddr3_hdm1.inst_ddr3_model.cmd_task: at time 110283314.0 ps INFO: Write bank 0 col 018, auto precharge 0

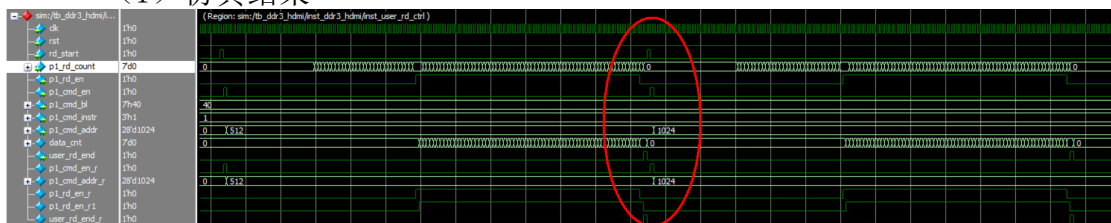
```

实际写入数据也是如此，这样，写时序部分，就没有什么问题啦。

2. 读时序仿真测试

这部分，没什么问题，可以直接查看测试结果。

(1) 仿真结果



波形与时序分析时一致，程序实现数据读取。

```

# tb_ddr3_hdm1.inst_ddr3_model.data_task: at time 115167064.0 ps INFO: READ @ DQS= bank = 0 row = 0000 col = 00000000 data = 0000
# tb_ddr3_hdm1.inst_ddr3_model.data_task: at time 115168314.0 ps INFO: READ @ DQS= bank = 0 row = 0000 col = 00000001 data = 0000
# tb_ddr3_hdm1.inst_ddr3_model.data_task: at time 115169564.0 ps INFO: READ @ DQS= bank = 0 row = 0000 col = 00000002 data = 0000
# tb_ddr3_hdm1.inst_ddr3_model.data_task: at time 115170814.0 ps INFO: READ @ DQS= bank = 0 row = 0000 col = 00000003 data = 0000
# tb_ddr3_hdm1.inst_ddr3_model.data_task: at time 115172064.0 ps INFO: READ @ DQS= bank = 0 row = 0000 col = 00000004 data = 0000
# tb_ddr3_hdm1.inst_ddr3_model.data_task: at time 115173314.0 ps INFO: READ @ DQS= bank = 0 row = 0000 col = 00000005 data = 0000
# tb_ddr3_hdm1.inst_ddr3_model.cmd_task: at time 115173314.0 ps INFO: Read bank 0 col 010, auto precharge 0
# tb_ddr3_hdm1.inst_ddr3_model.data_task: at time 115174564.0 ps INFO: READ @ DQS= bank = 0 row = 0000 col = 00000006 data = 0000
# tb_ddr3_hdm1.inst_ddr3_model.data_task: at time 115175814.0 ps INFO: READ @ DQS= bank = 0 row = 0000 col = 00000007 data = 0000
# tb_ddr3_hdm1.inst_ddr3_model.data_task: at time 115177064.0 ps INFO: READ @ DQS= bank = 0 row = 0000 col = 00000008 data = 0001
# tb_ddr3_hdm1.inst_ddr3_model.data_task: at time 115178314.0 ps INFO: READ @ DQS= bank = 0 row = 0000 col = 00000009 data = 0000
# tb_ddr3_hdm1.inst_ddr3_model.data_task: at time 115179564.0 ps INFO: READ @ DQS= bank = 0 row = 0000 col = 0000000a data = 0000
# tb_ddr3_hdm1.inst_ddr3_model.data_task: at time 115180814.0 ps INFO: READ @ DQS= bank = 0 row = 0000 col = 0000000b data = 0000
# tb_ddr3_hdm1.inst_ddr3_model.data_task: at time 115182064.0 ps INFO: READ @ DQS= bank = 0 row = 0000 col = 0000000c data = 0000
# tb_ddr3_hdm1.inst_ddr3_model.data_task: at time 115183314.0 ps INFO: READ @ DQS= bank = 0 row = 0000 col = 0000000d data = 0000
# tb_ddr3_hdm1.inst_ddr3_model.cmd_task: at time 115183314.0 ps INFO: Read bank 0 col 018, auto precharge 0

```

```

# tb_ddr3_hdmi.inst_ddr3_model.data_task: at time 118215814.0 ps INFO: READ @ DQS= bank = 0 row = 0000 col = 000003ef data = 0000
# tb_ddr3_hdmi.inst_ddr3_model.data_task: at time 118217064.0 ps INFO: READ @ DQS= bank = 0 row = 0000 col = 000003f0 data = 007e
# tb_ddr3_hdmi.inst_ddr3_model.data_task: at time 118218314.0 ps INFO: READ @ DQS= bank = 0 row = 0000 col = 000003f1 data = 0000
# tb_ddr3_hdmi.inst_ddr3_model.data_task: at time 118219564.0 ps INFO: READ @ DQS= bank = 0 row = 0000 col = 000003f2 data = 0000
# tb_ddr3_hdmi.inst_ddr3_model.data_task: at time 118220814.0 ps INFO: READ @ DQS= bank = 0 row = 0000 col = 000003f3 data = 0000
# tb_ddr3_hdmi.inst_ddr3_model.data_task: at time 118222064.0 ps INFO: READ @ DQS= bank = 0 row = 0000 col = 000003f4 data = 0000
# tb_ddr3_hdmi.inst_ddr3_model.data_task: at time 118223314.0 ps INFO: READ @ DQS= bank = 0 row = 0000 col = 000003f5 data = 0000
# tb_ddr3_hdmi.inst_ddr3_model.data_task: at time 118224564.0 ps INFO: READ @ DQS= bank = 0 row = 0000 col = 000003f6 data = 0000
# tb_ddr3_hdmi.inst_ddr3_model.data_task: at time 118225814.0 ps INFO: READ @ DQS= bank = 0 row = 0000 col = 000003f7 data = 0000
# tb_ddr3_hdmi.inst_ddr3_model.data_task: at time 118227064.0 ps INFO: READ @ DQS= bank = 0 row = 0000 col = 000003f8 data = 007f
# tb_ddr3_hdmi.inst_ddr3_model.data_task: at time 118228314.0 ps INFO: READ @ DQS= bank = 0 row = 0000 col = 000003f9 data = 0000
# tb_ddr3_hdmi.inst_ddr3_model.data_task: at time 118229564.0 ps INFO: READ @ DQS= bank = 0 row = 0000 col = 000003fa data = 0000
# tb_ddr3_hdmi.inst_ddr3_model.data_task: at time 118230814.0 ps INFO: READ @ DQS= bank = 0 row = 0000 col = 000003fb data = 0000
# tb_ddr3_hdmi.inst_ddr3_model.data_task: at time 118232064.0 ps INFO: READ @ DQS= bank = 0 row = 0000 col = 000003fc data = 0000
# tb_ddr3_hdmi.inst_ddr3_model.data_task: at time 118233314.0 ps INFO: READ @ DQS= bank = 0 row = 0000 col = 000003fd data = 0000
# tb_ddr3_hdmi.inst_ddr3_model.data_task: at time 118234564.0 ps INFO: READ @ DQS= bank = 0 row = 0000 col = 000003fe data = 0000
# tb_ddr3_hdmi.inst_ddr3_model.data_task: at time 118235814.0 ps INFO: READ @ DQS= bank = 0 row = 0000 col = 000003ff data = 0000
# tb_ddr3_hdmi.inst_ddr3_model.cmd_task: at time 118255814.0 ps INFO: Precharge bank 0

```

读取数据与写入数据一致，总的来说，成功。

六、总结与讨论

1. 仿真时长为120us
2. 尽可能简化的话，有些变量是必须是常量的，这是不可避免的。