

手机版

我们其它网站 ▼

您好 大灰灰 | 首页 | 我要开博 | 签到 ▼ | 个人信息中心 | 订阅电子快讯 | 退出

global sources

热搜排行，点一下又不会怎样。 3 文章 搜索 高级搜索


热门搜索： 无线路由器拆解 Allegro16.3 EXCEL常用技巧 特权同学 2015IIC China

首页 新闻 实例 技术 专题 参考设计 下载 视频 论坛 博客 小组 在线研讨会 热点专区


设计中心： 模拟 电源 嵌入式 测试与测量 通信 EDA 可编程器件 微处理器与DSP 消费 汽车 工业 医疗

博客大赛2nd，模拟、电源、EDA  
约会IIC2015，分享收获赢好礼  
2014年度博主评选—谁是MVP？  
每天登陆1秒签到，赚E币换礼！

EDN电子技术论坛>博客 >小墨同学博客 >零基础学FPGA（二十五）一路走来：SDR SDRAM（架构篇）



小墨同学的博客——零基础学FPGA学习日记




博主：小墨同学

介绍：主要是写一些自己从每天的学习中收获的一些东西，不一定很有价值，但是我觉得随着我的成长，这些东西将成为我的一段美好的回忆...而且我断定，在大学还剩下的时间里，FPGA将陪我一起走过...

文章(27) 访问(72648) 评论(140) 投票(122) 订阅本博

博文列表查看方式：

 摘要视图

原

零基础学FPGA（二十五）一路走来：SDR SDRAM（架构篇）

发布时间：2015-03-24 17:21:53

技术类别：CPLD/FPGA 个人分类：FPGA/CPLD

湖南最近终于也算是过了阴雨期，话说从开学到现在还没见过几个晴天。这几天还算好，阳光不错，学校里的樱花也开了，开得好是灿烂，但是面积不大，只有零零散散的几棵，每次到这樱花盛开的季节，总是有一个想法，就是去湖北武汉参加一下那里的樱花节。



虽然作为男生，但是也对樱花情有独钟。尤其面对那“仙云昨夜坠庭柯，化作翩跹万玉娥”般的樱花雨没有丝毫抵抗力。牵着自己喜欢人的手，漫步在樱花飘落的樱花道，是不是一幅令人陶醉的场景？

然而昨天还看新闻，说武汉大学的樱花节当日接待游客10W人，看樱花的人都在抱怨看得是人头，而不是樱花，这样说来，清明节假期去看樱花的人是不是更了不得了....本来打算清明节去一趟武汉，但是想起当时去杭州西湖的时候，瞬间就没了兴致，所谓西湖无断桥，只见人头啊...

话说小墨在大学这2年多也去了不少地方，读万卷书，不如行万里路嘛，什么北京，上海，杭州啊，桂林，凤凰，南岳啊以及周围的一些城市，长沙，武汉的就不用说了，最近同学又在组织去张家界了，而我更倾向于厦门和黄山，至于张家界，九寨沟等毕业之后再好好去看一下吧，要知道我可是个旅游达人~

博主资料

我是一名大二的在读本科生，认识FPGA没多久，但是我觉得挺吸引我的，希望能和爱好这一行的同学一起努力，让我们的大学不再迷茫，也希望网站的各位前辈多多指点，希望能与你们共同进步！也欢迎各位同行加我 小墨同学 QQ号 584642877 微信号 LYZ1023569

个人文章分类

FPGA/CPLD(27)

文章存档

2015年04月 (2)

2015年03月 (2)

2015年02月 (1)

2015年01月 (3)

2014年12月 (1)

2014年11月 (3)

展开

2015博客大赛第一季特别奖项NEW!

最新评论

● 雪龙523 21:30 03-29 好好学习一下，精辟

● xu疯子 10:37 03-26 继续加油，跟你一起学！

● wjfbblack 13:24 03-25 看来小墨同学的FPGA工程已经调好了，...

● 45du\_\_ 16:19 03-23 最近一直在搞SDRAM，翻看了你以前的...

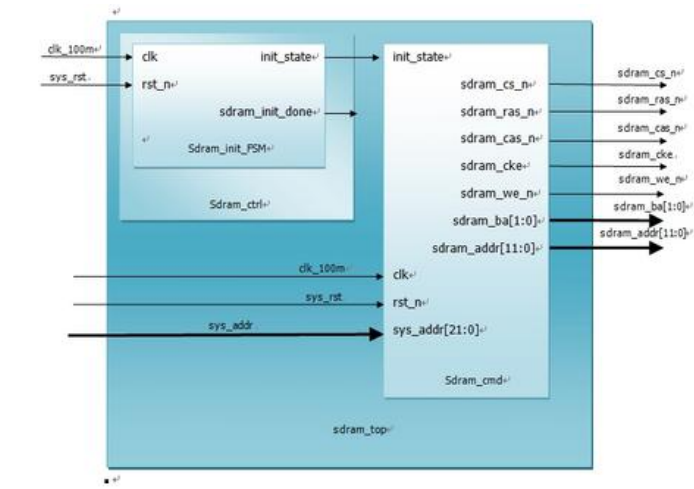
● 小学生来了 17:21 03-09 好，学习



旅游当然是要的，但是我们的专业知识也还是要好好学的，今天我们来讲的是SDRAM的架构以及设计，这也是小墨第一次接触架构，也谈不上给大家讲，就是把我理解的当做一个笔记分享给大家，我也试着做了一个SDRAM的架构word文档，在文章的后面，喜欢的朋友可以下载下来看一下，有什么错误也请积极指正，毕竟我也是没有老师教，也是自己摸索的，难免有些不合理的地方。

一、SDRAM工作部分

1、上电初始化



我们先来看第一部分，上电初始化。上电初始化我们知道，上电之后我们需要等待200us的稳定期，这段时间我们可以用一个定时器来计数，这没什么问题，然后进入的是预充电部分，这个时候，预充电的时候，sdrdram\_cmd模块会检测此时的初始化状态机的状态，若处于预充电状态，那么sdrdram\_cmd模块向SDRAM发命令，具体命令sdrdram的datasheet里面有，发的是一个precharge，即预充电，发完命令之后，需要等待一段时间，来确保这个命令被SDRAM捕获，这等待的时间，特权老师用的方式我觉得很好

```
//-----延时参数定义-----
`define end_trp cnt_clk_r == trp_clk //预充电后需等待trp个时钟周期才可以发送行有效
`define end_trfc cnt_clk_r == trfc_clk //自刷新周期
`define end_tmdr cnt_clk_r == tmdr_clk //模式寄存器设置等待周期
`define end_trcd cnt_clk_r == trcd_clk //行有效后延迟trcd个时钟周期后才可以发送读或写命令
`define end_tcl cnt_clk_r == tcl_clk //发送数据读取后会经过tcl个时钟周期的潜伏期数据才有
`define end_tread cnt_clk_r == tread_clk //数据读取时间
`define end_twrt cnt_clk_r == twrt_clk //数据写入时间
`define end_tdal cnt_clk_r == tdal_clk //写入数据后需要延时等待几个周期
`define end_trwait cnt_clk_r == trp_clk //读完成后的预充电等待
```

用一个宏定义，当计数器计数到相应的时间后，预充电完成参数置位 即end\_trp 置位，下面的用法也是一样，即当初始化状态机进入预充电状态i\_pre时，计数器开启，开始计数，计数完成，也就是预充电结束的时候，计数器复位，这个复位的控制，可以用case语句来检测初始化状态机的状态，在相应的状态给予相应的复位与置位

接下来是8个自刷新，操作和预充电一样

到了模式寄存器的配置阶段，我们需要选定L\_bank，以及SDRAM工作模式的配置，当sdrdram\_cmd模块检测到初始化状态机到达模式寄存器配置阶段时，我们的给SDRAM的sdrdram\_bank端口赋相应的值，并且设置地址总线

```
`i_mrs : begin
    sdrdram_cmd_r <= `cmd_lmr;
    sdrdram_ba_r <= 2'b00;
    sdrdram_addr_r <= {

        2'b00, //模式寄存器设置A11,A10
        1'b0, //A9, 设置为突发读/突发写
        2'b00, //A8, A7
        3'b011, //CAS潜伏期设置为个单位周期
        1'b0, //A3,突发传输方式设置为“顺序”
        3'b011 //突发长度设置，设置为8
    };
end
```

阅读排行

- 零基础学FPGA(一) 关于我和FPGA
- 零基础学FPGA（十四）新的开始
- [博客大赛]零基础学FPGA（十五）先来说V...
- 零基础学FPGA（二）关于触发器
- 零基础学FPGA（三）Verilog语法基础基...

小墨同学的好友



小梅哥



EDNChina



coyoo



MamBa24



ilove314

[更多好友>>](#)

小墨同学的小组 加入小组

**EDN助学—FPGA/CPLD**  
学习小组

成员17369名  
创建者: ilove314

**《深入理解Altera FPGA**  
应用设计》书友会

成员279名  
创建者: 胖螃蟹

精彩文章

推荐博客

- 人脸识别算法新进展：隐瞒年龄，不可能！
- 嵌入式开发：请不要屈服于质量，速度，成...
- LED小台灯终极升级
- 【博客大赛】零基础学FPGA（二十六）必...
- 【博客大赛】FPGA实战演练逻辑篇9：FPGA...
- 【博客大赛】板上系统逐渐过时，片上系统...
- 一个工程师说走就走的西藏之旅（五）
- 【博客大赛】FPGA实战演练逻辑篇4：Verilo...

最新网络博文

- Energia 发布 0101E0015 版
  - 苹果终于开始行动！
  - 电容测量注意事项
  - 【博客大赛】FPGA实战演练逻辑篇9：FP...
  - TimeQuest User Guide 翻译4
- [更多博文>>](#)

### • 1.1.2 SDRAM 上电初始化状态转移图

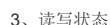


```

graph LR
    subgraph Sdr_ctrl [Sdr_ctrl]
        subgraph work_state [work_state]
            clk_100m_in[clk_100m] --> work_state
            rst_n_in[rst_n] --> work_state
            init_done_in[init_done] --> work_state
            work_state_out[work_state]
            sdr_ref_req[sdr_ref_req]
            sdr_ref_ack[sdr_ref_ack]
            sdr_work_fsm[sdr_work_fsm]
        end
        subgraph Sdr_cmd [Sdr_cmd]
            clk_100m_in2[clk_100m] --> Sdr_cmd
            rst_n_in2[rst_n] --> Sdr_cmd
            sdr_ref_req2[sdr_ref_req] --> Sdr_cmd
            Sdr_cmd_out[Sdr_cmd]
            sdr_cs_n[sdr_cs_n]
            sdr_ras_n[sdr_ras_n]
            sdr_cas_n[sdr_cas_n]
            sdr_cke[sdr_cke]
            sdr_we_n[sdr_we_n]
        end
        work_state -- sdr_ref_req --> Sdr_cmd
        Sdr_cmd -- sdr_ref_ack --> work_state
        Sdr_cmd -- clk_100m --> clk_100m_out[clk_100m]
    end

```

### 1.2.2 SDRAM 自刷新状态转移图



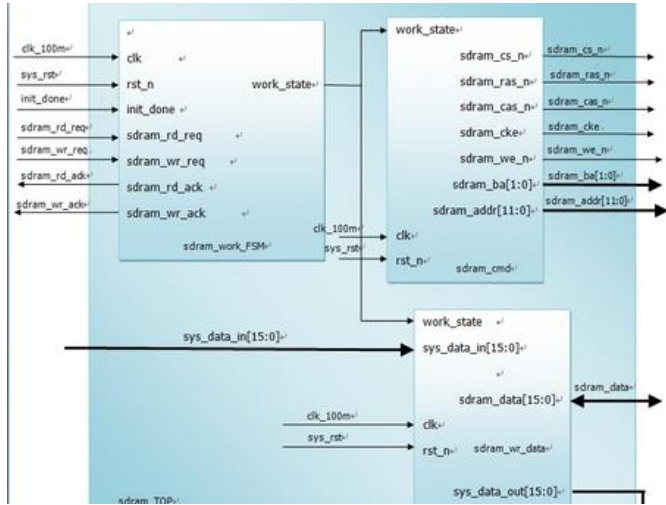
初始化结束以后，SDRAM就处于工作状态，每15us进行一次自刷新，这个时候，如果想要读或写数据的话，即向工作状态机sdram\_work\_FSM发送读写请求，如果是读请求，那么工作状态机进入行有效状态，也即激活状态w\_active，此时，sdram\_cmd模块会发送行有效命令，即CMD\_ACTIVA

同时，我们需要将我们的12位行地址送给我们的sdram地址总线sdram\_addr，然后便是一个等待时间段，TRCD，这个时间段里我们是不需要做什么工作的，只需等待TRCD结束

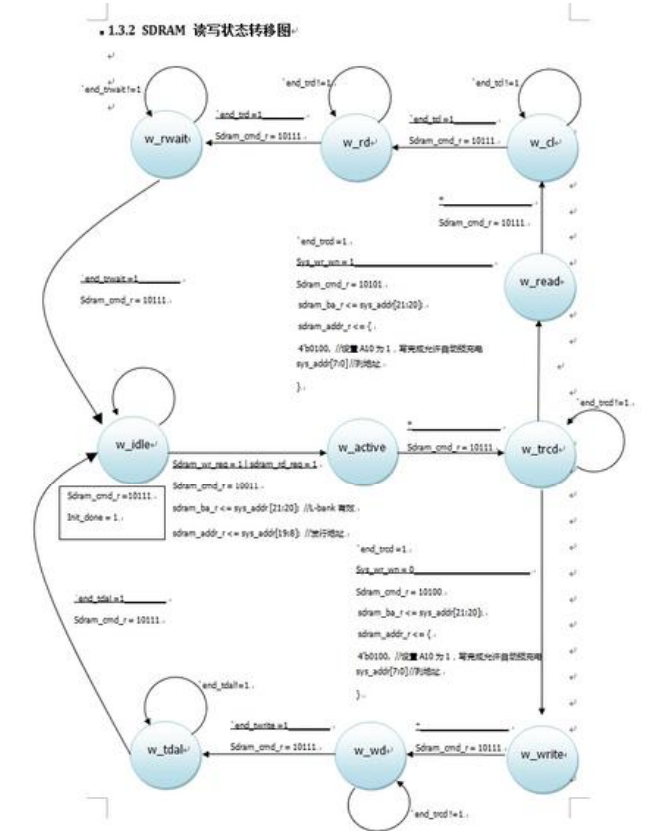
TRCD结束的时候，工作状态机进入读状态，此时，我们需要发送列地址选中我们的存储单元，并告诉sdram\_cmd模块发送读命令，这个过程是读命令和列地址同时发送的。我们知道，读命令发出之后，会进入潜伏期TCL。

这个状态我们也不需要做什么，因为我们在模式寄存器配置阶段已经将各种参数配置好了，包括什么潜伏期长，突发读写长度之类的，等到潜伏期结束之后，SDRAM会自动进行读数据，而且读的数据长度会跟我们之前设置的一样长，这个读数据的状态需要时间，即我们的工作状态机工作在w\_rd状态，等待8个时钟周期之后，所有的数据已经被送到数据总线。

这个时候我们需要等一会，因为我们在发送读命令的时候，A10是置1的，也就是说，每次读写完之后要自动进行预充电，从而才可以打开新的工作行，这时，我们的工作状态机即进入了w\_rwait状态，等到预充电结束，才完成一次读操作，返回初始状态，等待下一个请求的到来



下面是读写状态转移图，由于写状态只是没有潜伏期，写完之后有一个写回延时，其他地方与读状态相同，我不再介绍了





我们先来想一下我们的要想测试我们的SDRAM控制器是否正确，都需要什么测试参数吧

1、首先我们需要发送读写地址对吧，但是我们需要有一个间隔，即每隔一段时间发送一次写地址，为什么呢？因为我们之前在模式寄存器配置的时候，定义了突发读写的长度为8，也就是我们发送一个地址，那么sdram会在连续的8个地址中连续写8次数，我们如果要再发送地址，需要等到这8次地址写完之后再发送下一个地址这个延时大概为640ns，也就是计数器为什么要计到3f的原因，下一个地址的发送就需要比之前的地址大8了，比如第一次我发送的是0地址，那隔一段时间之后，我就需要发送8地址了。

读地址也一样，当写地址写满了之后，让地址清零，从零地址读起即可

```
//-----读写地址产生-----
//由于一次写入八个数据，这里只定义19位，这样addr自加的时候就一次加8了
//每640ns写入8个16位的数据到SDRAM，那么将所有的地址写完需要2*19*640ns=336us
reg [18:0]addr;

always @ (posedge clk or negedge rst_n)
    if(!rst_n) addr <= 19'b0;
    else if(!wr_done) && (cnt_wr==6'h3f) addr <= addr +1'b1; //写地址加1
    else if(wr_done == 1'b1 && neg_rdack == 1'b1) addr <= addr+1'b1; // 读地址从0自加1
    else if(addr == 19'h000ff) addr <= 19'd0;

assign moni_addr ={addr,3'b000}; //构成22位地址，每次自加8
```

还有再解释一下，特权老师为什么定义22位的模拟地址，22位的模拟地址包括2位的L\_BANK的地址，12位行地址和8位列地址。有人会问，为什么是行地址和列地址分开呢，不是行列地址共用吗？注意这里只是模拟地址，到时候我们给地址总线赋值的时候是分开赋值的

2、然后我们需要产生递增数据，每640ns产生8个数即可

```
//-----递增数据产生模块-----
reg [15:0] wr_data;
always @ (posedge clk or negedge rst_n)
    if(!rst_n) wr_data <= 16'b0;
    else if(!wr_done && (cnt_wr >6'h05) && (cnt_wr <=6'h0d)) wr_data <= wr_data +1'b1;
//如果SDRAM还没写满，那么在8个时钟周期内数据连续递增

assign wrf_din = wr_data;
//-----
```

3、在发送递增数据之前，我们需要发送写请求信号，这个写请求信号是要发送到我们接下来要讲的FIFO里面，发送写请求之后，数据进入FIFO，注意，这里的写请求是发送给FIFO的，不是发送到SDRAM的，SDRAM的写请求是由FIFO发出去的

```
//-----写请求信号产生模块-----

reg wrf_wreq_r;
always @ (posedge clk or negedge rst_n)
    if(!rst_n) wrf_wreq_r <=1'b0;
    else if(!wr_done && (cnt_wr ==6'h05)) wrf_wreq_r<= 1'b1;
    else if(!wr_done && (cnt_wr ==6'h0d) | wr_done == 1)wrf_wreq_r<= 1'b0;

//如果SDRAM没写满，那么当数据开始递增的时候发送写请求，当数据递增结束，写请求撤销
```

有人会问，写请求有了，是不是还该有读请求，要知道，我们的数据是先写到FIFO里面，再有SDRAM对FIFO进行读操作，将写进FIFO的数据再送到SDRAM，而我们从SDRAM中读回的数据，同样要写到FIFO里面，然后我们再从FIFO里面读。是不是有人会问，这不是多此一举吗？其实不然，要知道我们的SDRAM是工作在100M的时钟频率下，而且是有相位偏移的，对我们FPGA来说是一个异步时钟，如果我们不采用FIFO的形式，那很容易发生亚稳态问题，导致系统不稳定

三、数据缓存部分

1、写FIFO

```
//-----要写入SDRAM的数据缓存fifo-----
huut_wrfifo uut_wrfifo_inst (
    .data ( wrf_din ),
    .rdclk ( clk_100m),
    .rdreq ( sdram_wrf_ack ),
    .wrclk ( clk_50m ),
    .wreq ( wrf_wreq ),
    .q ( sys_data_in),
    .wusedw ( wrusedw_r )
);
```

我来解释一下，写入时钟，也就是我们FPGA的时钟，50MHZ，在这个时钟频率下，我们需要给FIFO发送写请求和递增数据，wrusedw反映了我写入FIFO的占用量，当sdram工作状态机检测到写请求之后（FIFO发出的写请求），会产生一个响应，告诉FIFO要读数据了，这个响应作为FIFO的读信号，在100MHZ的时钟频率下，将我们写入的数据读走，送到SDRAM数据总线上

2、读FIFO

```
//-----读出SDRAM的数据缓存fifo-----
uut_rdfifo uut_rdfifo_inst (
    .data ( sys_data_out ),
    .rdclk ( clk_50m ),
    .rdreq ( rdf_rdreq),
    .wrclk ( clk_100m ),
    .wreq ( sdram_rd_ack ),
    .q ( rdf_dout ),
    .wusedw ( rdusedw_r )
);
```

读FIFO跟写FIFO相反，写FIFO时钟为100MHZ，在这个时钟频率下，我们将从SDRAM中读回的数据写到FIFO里面，注意这里我们还用了wrusedw，来反应写FIFO的占用量，也就是我们从SDRAM里读了多少数据，然后在50MHZ的时钟下，将从SDRAM读回的数据采集下来，送到显示模块来验证是否是递增数据，这个显示模块我们用数码管来显示，比较方便，为了观察清楚，我们可以在数码管显示模块，每隔1S种发送一次读FIFO请求，这样在数码管上，会每隔1秒钟显示一个递增的

大体上就是这么多了，整个SDRAM的工作过程就介绍完了，还有些细节没讲到的，大家可以自行消化，有不懂的可以给我留言，或者加我QQ微信，欢迎交流

下一篇文章我们讲时序，真正在板子上让它跑起来才是王道，下面附上我做的一个word，还请各位大神指正~

以后小墨同学的文章将在EDNchina和北京至芯科技官方技术论坛同时更新，至芯科技的官方技术论坛也有很多的资料供大家下载，里面也有小墨的专题板块，希望广大网友和爱好者的大力支持~

[url]http://www.fpgaw.com/[url]

小墨同学《零基础学FPGA》学习专版 一个大三学生学习之路的真实记录

[url]http://www.fpgaw.com/thread-78527-1-1.html[url]

下面是至芯科技官方网站，夏宇闻教授和那里的老师都超好的~有意愿接受FPGA培训的可以找工作人员咨询

[url]http://www.zxopen.com/[url]

以后小墨同学的教程会跟这款开发板配套，想跟小墨同学一起学习，一起进步的可以考虑一下购买一块开发板，毕竟学习要舍得投资嘛，下面是开发板淘宝链接

[url]http://item.taobao.com/item.htm?spm=a230r.1.14.13.NZBxw3&id=42366952129&ns=1&abbucket=13#detail[url]

附件：  
一路走来：SDR SDRAM（架构篇）.zip  
下载次数：28

标签：  
fpga

分享到：

我喜欢

有[0]名读者喜欢此文

阅读(423)

评论(1)

收藏

|

打印

|

推荐到小组


|

转发到我的博客(0)


上一篇：【博客大赛】零基础学FPGA（二十七）必会！从静态时序分析到SD...

下一篇：零基础学FPGA（二十四）一路走来：SDR SDRAM（理论篇）

最近到访博主

  
KPUHD

  
szm3

  
EDNChina

有[1]名读者发表评论 - 零基础学FPGA（二十五）一路走来：SDR SDRAM（架构篇）

网友：  
wjfblack

踩(0)

赞(0)

快速回复

引用回复(0)

2015-03-25 13:24

评论：  
看来小墨同学的FPGA工程已经调好了，能共享出来学习一下吧？

--- 现有 1 个主题，共 1 页 ---

我来评论 - 零基础学FPGA（二十五）一路走来：SDR SDRAM（架构篇）

昵称：  
大灰灰

评论：

你还可以输入1000字

分享：



2015 EDN China -- Use of this website is subject to its terms of use.

京 ICP证090436号 | 京ICP备14018902号-1 | 京公网安备110105019137 | 经营性网站备案信息  
[EDN.com](#) | [EDN Asia](#) | [EDN Japan](#) | [EDN Europe](#) | [EDN Australia](#) | [EDN Taiwan](#) | [电子工程专辑](#) | [国际电子商情](#) | [DatasheetsChina](#)  
[机器人网](#) | [Global Sources Direct](#)

