



基于 ERAM 接口实现的高性能 SDRAM 读写

APUG011 (v1.0.1) 2021 年 01 月

Confidential

目 录

| | |
|----------------------|----|
| 1 概要 | 2 |
| 2 功能描述 | 2 |
| 2.1 ERAM 介绍 | 2 |
| 2.2 功能框架 | 2 |
| 2.3 用户读写操作 | 3 |
| 2.3.1 用户写数据时序 | 4 |
| 2.3.2 户读数据时序 | 4 |
| 2.3.3 数据掩码 | 4 |
| 2.3.4 跳跃地址读写 | 5 |
| 2.4 读写效率与注意事项 | 5 |
| 3 模块接口介绍 | 6 |
| 4 资源消耗与时序 Fmax | 7 |
| 5 参考设计与仿真 | 8 |
| 5.1 调试 | 8 |
| 5.2 仿真 | 8 |
| 6 工程文件 | 9 |
| 7 移植与注意事项 | 9 |
| 8 文件参考 | 10 |
| 9 版本信息 | 11 |
| 免责声明 | 11 |

1 概要

FPGA 为可编程逻辑器件, 广泛应用于通信、视频图像处理等领域, 这些应用都需要大量使用 FPGA 内的 ERAM, 当 ERAM 容量不能满足要求时, 可能会考虑更换更大型号的 FPGA, 或者外面扩展 SDRAM 或者 DDR 等存储器件。但扩展类似的存储器件, 需要 FPGA 以及存储器 IP 的支持, 对于小型 FPGA 器件或者没有存储器硬 IP 的 FPGA 器件, 要增加存储, 对于 FPGA 人员以及系统成本、产品面市都提出了挑战。

本文档通过介绍一种 ERAM 容量扩展方案, 使用者可以以 FPGA 内部的 ERAM 的方式接口进行操作, 通过此种 RAM 容量扩展方式, 将 ERAM 扩展到 64Mb 以及更大的容量, 读写时钟可以支持 200Mhz, 读写带宽可以达到 6Gbps 以上, 效率可支持到 95%。

方案中使用参数化完成, 支持各种任意大小容量和不同位宽的 SDRAM 器件。

使用方式接口与 ERAM 操作一致, 通过此种方式, 可以实现数据的大缓存以及处理, 用户可利用此方案快速实现各种数据缓存应用。

2 功能描述

2.1 ERAM 介绍

ERAM 是 FPGA 内的宝贵资源, 通常大约在 500Kbit 上下, 视不同系列器件, ERAM 大小容量会有差别。

按照 ERAM 的读写操作方式, 可分为单口 RAM, 简单双口 RAM, 真双口 RAM。单口 RAM 只有一套地址总线, 其他 2 种形式的 RAM, 则有 2 个地址总线, 且读写时钟可以独立使用。

其形式可分为以下三种, 如图 2-1 所示。

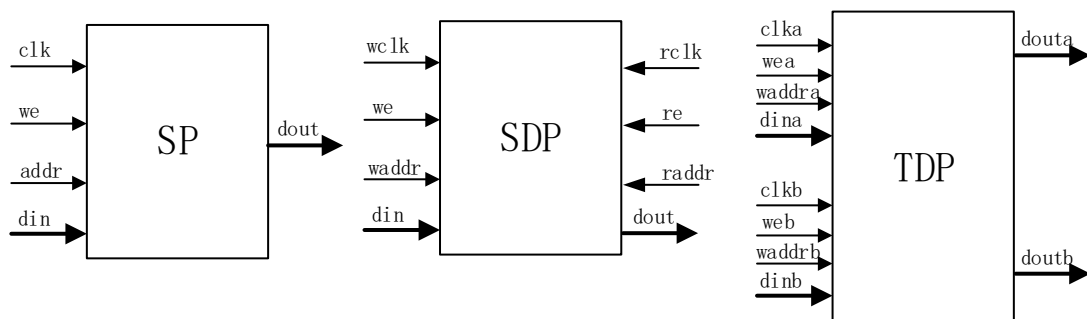


图 2-1 ERAM 的三种模式

2.2 功能框架

本例程，通过使用 Anlogic 的 EG4S 器件，该器件集成了 64Mb 的 SDRAM，顶层提供一种类似于 FPGA 内的 ERAM 读写接口给用户，方便用户对数据做读写保存，不需做额外多余的存储器件读写和调试工作。

下面我们将扩展的 ERAM 简称虚拟 ERAM。

工程方案框图如图 2-2 所示。

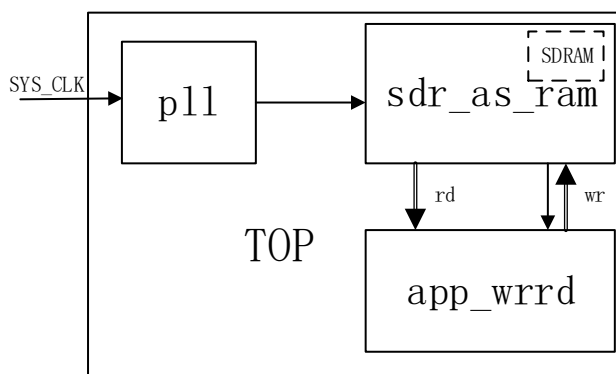


图 2-2 工程框图

2.3 用户读写操作

用户操作分 2 个通道，读通道和写通道。因硬件机制问题，应用方案中有一个限制，即不支持读写同时操作，读写需要独立排斥运行。

方案提供 2 种刷新机制，一种是由 IP 本身提供的，用户不需要参与，只需每次在读写前判断当前是否刷新。另外一种是由用户判断是否做刷新，用户通过刷新标志去自行刷新。正常读写操作中，用户不需要考虑刷新机制。

在用户对虚拟 ERAM 做操作前，如果使用方案提供的刷新机制，因此，请先判断当前虚拟 ERAM 是否忙，如果忙，请等待 Sdr_init_ref_vld 信号变为低电平后，再做读写操作。

虚拟 BRRAM 框图如图 2-3 所示。

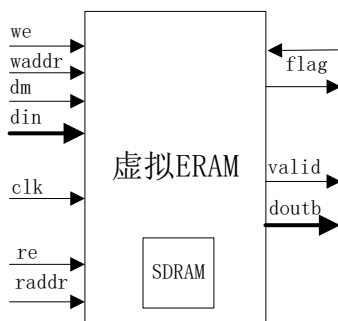


图 2-3 虚拟 ERAM

2.3.1 用户写数据时序

写通道中，分为写使能、写地址、写数据信号，同时也提供掩码信号，可以对虚拟 ERAM 进行字节掩码，实现数据的覆盖式读写。

每次写的长度没有限制，可持续进行写入，可一次写完整个芯片。如果连续写入，请将时钟频率工作在 64Mhz 以上。

类似 SDRAM 器件，为了不让数据丢掉，需要做数据刷新操作。本应用提供了 2 种方式，即方案中已经支持自动刷新，用户不需要干预，另外用户也可以选择自行发刷新标志，做刷新工作。

写入时序如图 2-4 所示。

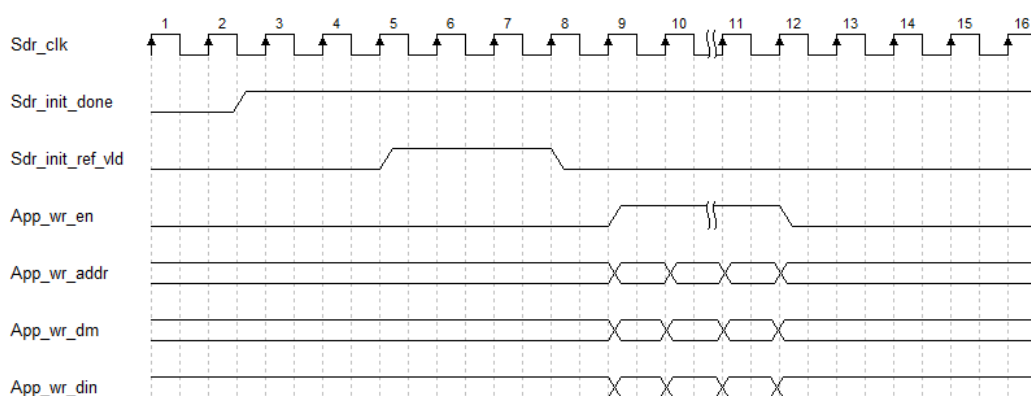


图 2-4 用户写入数据时序

2.3.2 户读数据时序

从虚拟 ERAM 中读出数据时，用户只需给出读使能信号和读地址，在给出读使能信号后的 10 个时钟，开始给出有效的数据输出使能和有效数据。

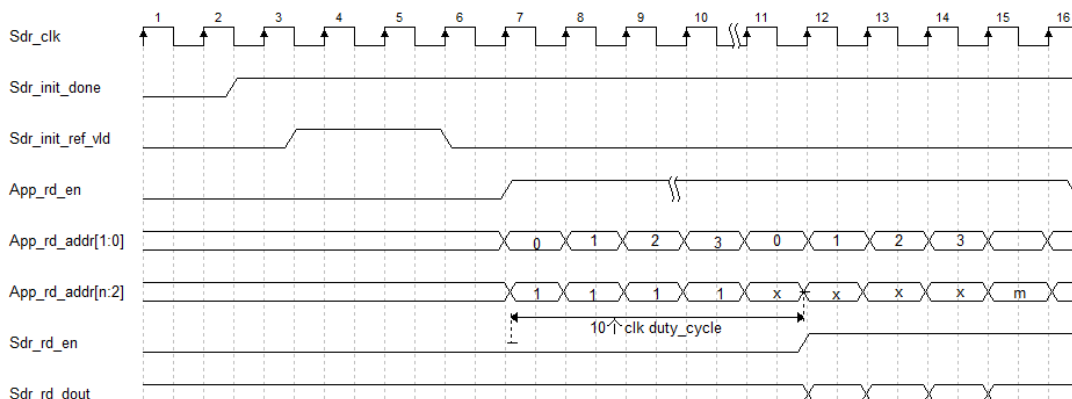


图 2-5 用户读出数据时序

2.3.3 数据掩码

方案中，可以对每个数据单元，如每个时钟的数据宽度是 32bit, 则对应掩码为 app_wr_dm[3:0], app_wr_dm[0] 对应数据 app_wr_din[7:0]。以此类推。

也可支持对多个数据单元进行掩码，此时，将 app_wr_din 设置为 4'b1111。数据写入掩码时序如图 2-6 所示。

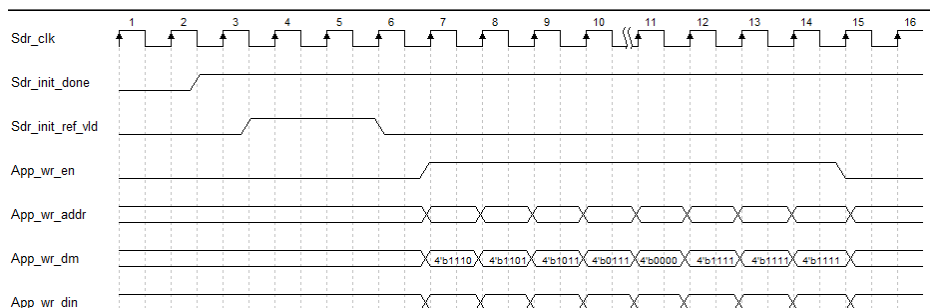


图 2-6 写入数据掩码时序

2.3.4 跳跃地址读写

FPGA 内的 ERAM 可以支持任意地址的跳跃，本方案理论上也可以支持任意地址，但数据是以 128bit 为基础（如本方案数据标准位宽为 32bit）进行任意地址跳跃。

在实现地址跳跃时，app_wr_addr[1:0] 为 2'b00，任意更改 app_wr_addr[22:2]，简单理解为每 4 个时钟，可以任意跳跃 1 次地址。

下图 2-7 中的 x 代表任意地址。

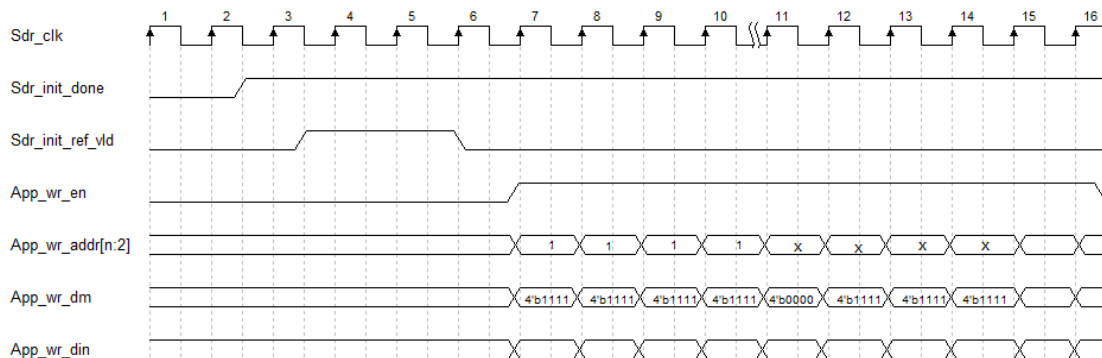


图 2-7 数据地址跳跃操作时序

2.4 读写效率与注意事项

SDRAM 工作频率可以支持 200Mhz，按照 32bit 位宽，理论上可以支持 6.4Gbps 有效带宽，如果持续写入或者持续读出，实际带宽与理论带宽一致。如写入和读出有停顿，带宽可以支持到 6Gbps，带宽利用率在 95% 以上。

每次写入或读出持续的长度，可随机，为了效率，尽量长度是 4 的倍数。长度为非 4 的倍数时，请

配合数据掩码使用。

每次写入或读出的数据以 4 个为单位，每 4 个种第 1 个地址即为首地址，第 2~4 地址是依据首地址做累加，如下图 2-8 所示。

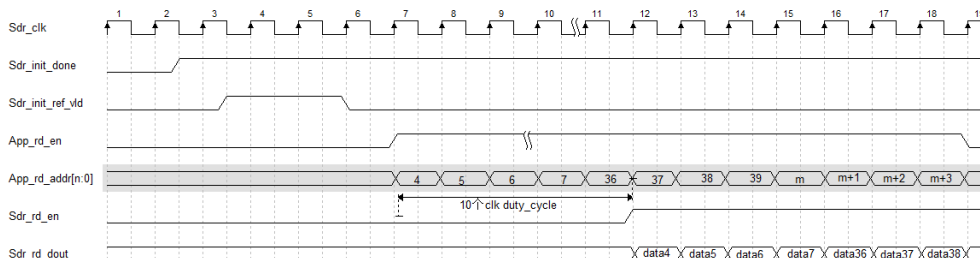


图 2-8 地址规则

3 模块接口介绍

方案顶层接口中，只提供用户的读写通道信号，扩展 SDRAM 器件使用的是 Anlogic EG4S 内部自带的 SDRAM SIP。如用户需要更换成 Anlogic 其他器件，将此部分接口引出到工程顶层，分配 IO 即可。

方案接口信号定义如表 3-1 所示。

表 3-1 方案接口信号

| 信号 | 方向 | 功能说明 |
|-------------------|-----|--------------------|
| Sdr_clk | in | 应用方案 SDRAM 工作时钟 |
| Sdr_clk_sft | in | 同频可能不同相位的 SDRAM 时钟 |
| Rst | in | 高有效 |
| 用户读写接口 | | |
| App_wr_en | in | 写使能，高有效 |
| App_wr_addr[20:0] | in | 写地址 |
| App_wr_din[31:0] | in | 写数据 |
| App_wr_dm[3:0] | in | 写入数据掩码 |
| App_rd_en | in | 读使能 |
| App_rd_addr[20:0] | in | 读地址 |
| Sdr_rd_en | out | 有效读数据输出使能 |
| Sdr_rd_dout[31:0] | out | 有效读出数据 |

| 控制信号 | | |
|------------------|-------|------------------|
| Sdr_init_done | out | SDRAM 初始化完成 |
| Sdr_init_ref_vld | out | SDRAM 刷新有效标志 |
| Sdr_busy | out | SDRAM 忙状态，暂不使用 |
| App_ref_req | in | 用户自行请求刷新操作 |
| SDRAM 信号 | | |
| SDR_CLK | in | SDRAM 时钟 |
| SDR_CKE | in | SDRAM 的 CKE 时钟使能 |
| SDR_RAS | in | SDARM 行有效 |
| SDR_CAS | in | SDRAM 片选 |
| SDR_WE | in | SDRAM 写使能 |
| SDR_BA[1:0] | in | SDRAM Bank |
| SDR_ADDR[10:0] | in | SDRAM 的操作地址 |
| SDR_DM[3:0] | in | SDRAM 的数据掩码 |
| SDR_DQ[31:0] | inout | SDRAM 数据总线 |

操作该虚拟 ERAM 时，请先等待初始化完成后，可进行操作。使用前，请先判断是否在刷新，如没有刷新，方可进行任意长度的读写操作。

为了不造成有效带宽的利用，写入和读出的长度，尽量是 4 的倍数。如长度不是 4 的倍数，请使用掩码功能，屏蔽无效数据的写入。如写入长度为 5，那么在第 5~8 个数据时，第 5 个数据正常写入，第 6~8 个的数据写入时，将 app_wr_dm[3:0]设置位 3'b1111。

4 资源消耗与时序 Fmax

使用 TD 软件版本 TD5.0.2 23393,在 EG4S20BGA256-MINI-DEV3 Demo 板上进行验证。消耗的资源及时序结果如表 4-1 所示：

表 4-1 资源与时序结果统计

| 资源 | 类型 | 数值 |
|------|----|--------|
| 最大频率 | c0 | 219Mhz |

| | | |
|------|----------------------------------|-----|
| 使用资源 | LUT | 242 |
| | REG | 533 |
| | PLL | 1 |
| | IO | 2 |
| | ERAM | N/A |
| | DSP | N/A |
| | SERDES | N/A |
| 适用器件 | Anlgoc 全系器件（除 EG4S 外，需要外挂 SDRAM） | |

5 参考设计与仿真

该设计提供仿真文件，用户只需添加相应的 do 文件即可进行在 Modelsim 环境下进行仿真。

5.1 调试

app_wrrd 模块产生的写入数据时序是递增数据，读出来后再与递增数进行判断，如果数据正确，则 Check_OK 信号为低电平，否则为高电平。

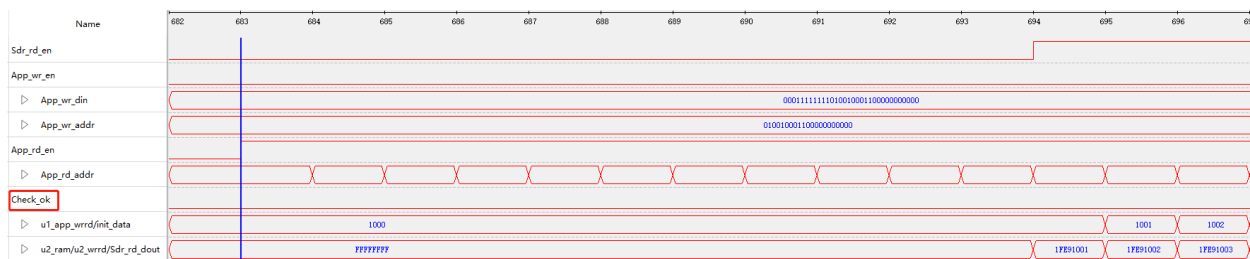


图 5-1 CWC 调试

5.2 仿真

本方案中的 Test bench 仿真代码支持综合，最顶层仿真文件提供时钟 25Mhz，系统可以工作起来。如果出错，仿真工具会给出错误提示。指定好库后，运行 sim.do 仿真文件，可一键仿真。

```
vlog -work work +incdir+$INCLUDE_PATH +define+SIMULATION ../source_code/rtl/*.v
```

```
vlog -incr +incdir+$INCLUDE_PATH ../source_code/model/*.v
```

```
vlog -incr +incdir+$INCLUDE_PATH +define+SIMULATION ../source_code/tb/*.v
```

```
vlog -incr +incdir+$INCLUDE_PATH +define+SIMULATION ../prj/al_ip/*_sim.v
```

6 工程文件

工程中有如下关键代码文件，如表 7-1 所示。

表 7-1 工程文件

| 参数 | 说明 |
|-------------------|------------------|
| Developers | xuguo |
| Reference Design | Yes |
| RTL Language | Verilog/VHDL |
| Test bench | YES |
| Test bench Format | YES |
| Simulation | Modelsim |
| C | N/A |
| IP Model | Yes |
| Project Platform | EG4S20-MINI-DEV3 |

7 移植与注意事项

1. 使用前请注意等待 SDRAM 初始化完成，再开始工作；
2. 地址跳跃和掩码，请按照第二章中数据掩码规则进行；
3. 每次读写操作前，请判断 Sdr_busy 是否忙；
4. 读写操作长度可以任意，尽量是 4 的倍数；
5. 地址可以任意跳跃，为 4 的倍数，倍数内不支持跳，倍数间可以任意跳；
6. 读数据时，SDRAM 与不同硬件，使用不同的频率和相位，可能出现时钟和数据错一拍的情形，请根据实际情况，在应用中做调整；
7. 可支持 Anlogic 其他系列 FPGA；
8. 方案中使用的是 64Mb、宽度为 32bit 的 SDRAM，更换其他型号的器件时，请更改文件中的全局宏定义参数；

```
`define DATA_WIDTH
```

```
32
```

| | | |
|---------|------------|----|
| `define | ADDR_WIDTH | 21 |
| `define | DM_WIDTH | 4 |
| `define | ROW_WIDTH | 11 |
| `define | BA_WIDTH | 2 |

8 文件参考

[AN EG4 DataSheet DS300](#)

[AN EG4S20 DataSheet DS301](#)

9 版本信息

| 日期 | 版本 | 修订记录 |
|------------|-------|----------------|
| 2020.10.28 | 1.0 | 首次发布中文版 |
| 2021.01.29 | 1.0.1 | 修改 BRAM 为 ERAM |
| | | |

版权所有©2021 上海安路信息科技有限公司

未经本公司书面许可，任何单位和个人都不得擅自摘抄、复制、翻译本档内容的部分或全部，并不得以任何形式传播。

免责声明

本文档并未授予任何知识产权的许可，并未以明示或暗示，或以禁止发言或其它方式授予任何知识产权许可。除安路科技在其产品的销售条款和条件中声明的责任之外，安路科技概不承担任何法律或非法律责任。安路科技对安路科技产品的销售和/或使用不作任何明示或暗示的担保，包括对产品的特定用途适用性、适销性或对任何专利权、版权或其它知识产权的侵权责任等，均不作担保。安路科技对文档中包含的文字、图片及其它内容的准确性和完整性不承担任何法律或非法律责任，安路科技保留修改文档中任何内容的权利，恕不另行通知。安路科技不承诺对这些文档进行适时的更新。