

基于 VHDL 的 SDRAM 控制器的实现

在高速实时或者非实时信号处理系统当中,使用大容量存储器实现数据缓存是一个必不可少的环节,也是系统实现中的重点和难点之一。SDRAM(同步动态随机访问存储器)具有价格低廉、密度高、数据读写速度快的优点,从而成为数据缓存的首选存储介制裁。但是 SDRAM 存储体结构与 RAM 有较大差异,其控制时序和机制也较复杂,限制了 SDRAM 的使用。目前,虽然一些能家长微处理器提供了和 SDRAM 的透明接口,但其可扩展性和灵活性不够,难以满足现实系统的要求,限制了 SDRAM 的使用。

在详细阅读 SDRAM 数据文档的前提下,参考 ALTERA 公司的 IP core,利用可编程器件(CPLD, FPGA)设计了一种通用的 SDRAM 控制器。它具有很高的灵活性,可以方便地和其它数据采集分析系统中,如图 1 所示。在该系统中,以 SDRAM 存储阵列缓存中频来的高速数据。存满后,数据被慢速读出至数据处理模块。下面将对 SDRAM 控制模块的设计进行详细的描述。

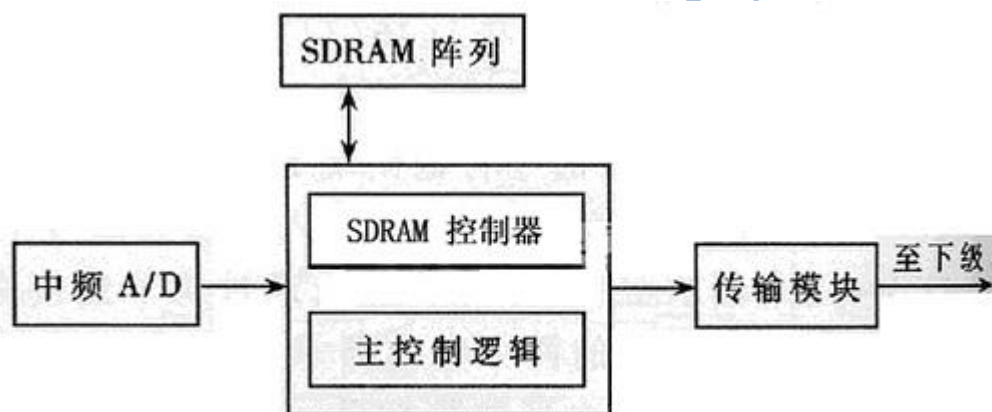


图 1 中频数据采集系统原理图

1 SDRAM 内存条的结构

SDRAM 内存条由 SDRAM 内存芯片构成,根据内存条的容量大小决定内存条上内存芯片的个数。现以 MICRON 公司生产的 TIM16LSDT6464A 型 SDRAM 内存条为例,简要介绍 SDRAM 的结构。

MIT16LSDT6464A 内存条容量为 512M Byte,由 16 片容量为 32M Byte 的内存芯片 MT46LC32M8A2 构成。16 片内存芯片被分为两级,每个芯片的数据位宽为 8bit。8 片一组,64bit 数据宽度。每个内存芯片的数据线和控制均是复用的。对内存条的读写操作,是以内存芯片组为单位的,通过内存条的片选信号 S0、S1、S2、S3 决定组号。S0、S2 控制芯片组 1, S1、S3 控制芯片组 2。

SDRAM 内存芯片的主要信号有控制信号、控制信号、数据信号，均为工作时钟同步输入、输出信号。

控制信号主要有：CS（片选信号），CKE（时钟使能信号），DQM（输入、输出使能信号），CAS、RAS、WE（读写控制命令字）。通过 CAS、RAS、WE 的各种逻辑组合，可产生各种控制命令。

地址信号有：BA0 和 BA1 页地址选择信号，A0~A12 地址信号，行、列地址选择信号。通过分时复用决定地址是行地址还是列地址。在读写操作中，在地线上依次给出页地址、行地址、列地址，最终确定存储单元地址。

数据信号有：DQ0~DQ7，双向数据。其使能受 DQM 控制。

SDRAM 的工作模式通过 LOAD MODE REGISTER 命令对工作模式寄存器进行设置来选择。设置参量有 Reserved（备用的人）Write Burst Mode（WB，写突发模式）、Operation Mode（Op Mode，工作模式）、CAS Latency（CAS 延迟）、Burst Type（BT，突发类型）、Burst Length（突发长度）。

2 SDRAM 的基本读写操作

SDRAM 的基本读操作需要控制线和地址线相配合地发出一系列命令来完成。先发出 BANK 激活命令（ACTIVE），并锁存相应的 BANK 地址（BA0、BA1 给出）和行地址（A0~A12 给出）。BANK 激活命令后必须等待大于 t_{RCD} （SDRAM 的 RAS 到 CAS 的延迟指标）时间后，发出读命令字。CL（CAS 延迟值）个工作时钟后，读出数据依次出现在数据总线上。在读操作的最后，要向 SDRAM 发出预充电（PRECHARGE）命令，以关闭已经激活的页。等待 t_{RP} 时间（PRECHARGE）命令，以关闭已经激活的页。等待 t_{RP} 时间（PRECHARGE 命令后，相隔 t_{RP} 时间，才可再次访问该行）后，可以开始下一次的读、写操作。SDRAM 的读操作只有突发模式（Burst Mode），突发长度为 1、2、4、8 可选。

SDRAM 的基本写操作也需要控制线和地址线相配合地发出一系列命令来完成。先发出 BANK 激活命令（ACTIVE），并锁存相应的 BANK 地址（BA0、BA1 给出）和行地址（A0~A12 给出）。BANK 激活命令后必须等待大于 t_{RCD} 的时间后，发出写命令字。写命令可以立即写入，需写入数据依次送到 DQ（数据线）上。在最后一个数据写入后延迟 t_{WR} 时间。发出预充电命令，关闭已经激活的页。等待 t_{RP} 时间后，可以展开下一次操作。写操作可以有突发写和非突发写两种。突发长度同读操作。

t_{RCD} 、 t_{RP} 、 t_{WR} 的具体要求，详见 SDRAM 厂家提供的数据手册。所等待的工作时钟个数由 t_{RCD} 、 t_{RP} 、 t_{WR} 的最小值和工作时钟周期共同决定。

由以上介绍可以得出，SDRAM 的读、写操作均由一系列命令组成，因此读、写操作是有时钟损耗的，工作时钟速率不等于 SDRAM 能达到的读、写速率。但是由于 SDRAM 有突发读、写模式，也就是说可以读出和写入一串地址连续的数据，

从而提高了效率。当突发长度为整页时，读、写速度达到最快。随机读、写速度的计算公式为：

$$\text{furite/read} = \text{工作时钟频率 (Hz)} \times \text{数据宽度 (bytes)} \times \text{突发读写长度/操作所需的时钟数}$$

为了提高存储密度，SDRAM 采用硅片电容存储信息。电容总会有漏电流流过，所以为了不使信息丢失，必须定期地给电容刷新充电。外部控制逻辑必须按要求定期向内存条发出刷新命令，保证在规定的时间内对每一个单元都进行刷新。

3 初始化操作

SDRAM 在上电以后必须对其进行初始化操作，具体操作如下：

(1) 系统在上电后要等待 $100 \sim 200 \mu\text{s}$ 。在待时间到了以后至少执行一条空操作或者指令禁止操作。

(2) 对所有芯片执行 PRECHARGE 命令，完成预充电。

(3) 向每组内存芯片发出两条 AUTO REFRESH 命令，使 SDRAM 芯片内部的刷新计数器可以进入正常运行状态。

(4) 执行 LOAD MODE REGISTER 命令，完成对 SDRAM 工作模式的设定。

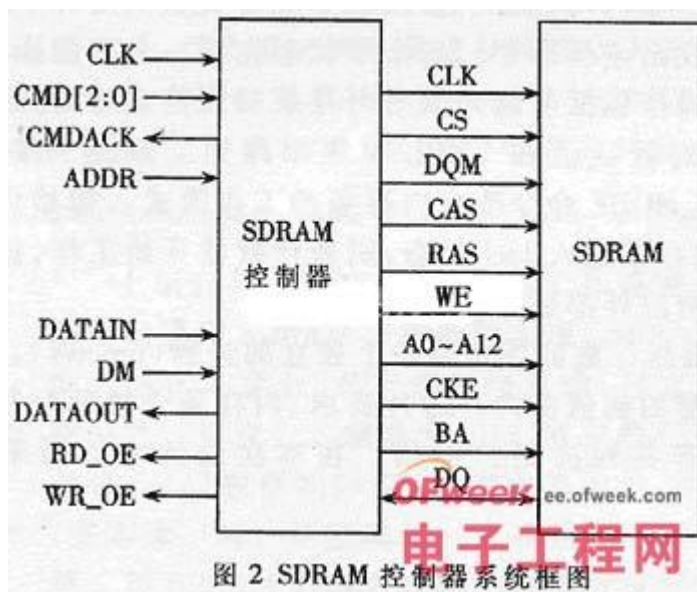
完成以上步骤后，SDRAM 进入正常工作状态，等待控制器对其进行读、写和刷新等操作。

4 SDRAM 控制器设计

4.1 功能说明

在以 SDRAM 作为缓存的系统中，使用可编程器件对其进行控制具有很强的灵活性。为了使设计具有模块化和可重复使用的优点，设计了一个简化的 SDRAM 接口电路。这样就屏蔽掉了 SDRAM 操作的复杂性，而其它逻辑模块可通过接口电路对 SDRAM 进行访问。此外，由于整个 SDRAM 控制器用 VHDL 语言编写，只要对其进行简单的修改就可以满足不同的需求，具有很强的灵活性。

参照图 2，SDRAM 控制器完成的主要功能是对 CMD[2:0] 的命令字和 ADDR 端的地址进行解析，产生相应的 SDRAM 的控制时序。



CLK 为输入的工作时钟端口。

ADDR 为输入地址端口。控制器将其解析为对应的片选、页以及行、列地址。以一条 MIT16LSDT6464A 内存条为例，其大小为 512Mbyte (2²⁹ byte)。数据位宽为 64bit (8byte)，则地址线 ADDR 应为 26 根。可以这样映射地址：ADDR[25] 对应内存芯片组号；ADDR[24:23] 对应页号；ADDR[22:10] 对应行号；ADDR[9:0] 对应列号。

DATAIN 为写入数据端口，64bit 位宽。

DATAOUT 为读出数据端口，64bit 位宽。

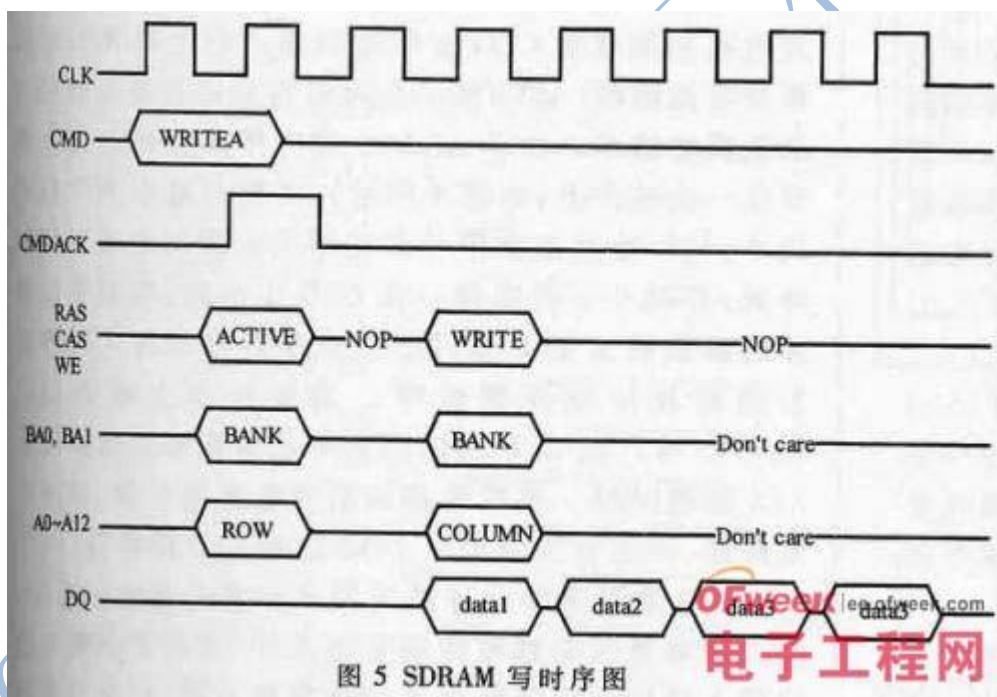
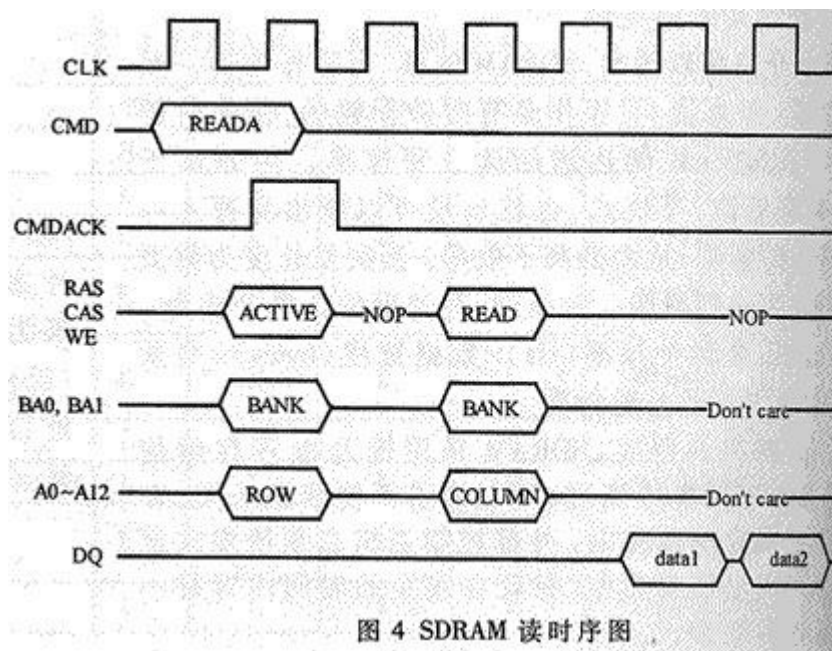
RD_OE 为读出数据使能端口，当其为 1 时，表示从下一个时钟起，数据将依次出现在 DATAOUT 口上。

WR_OE 为写入数据使能端口，当其为 1 时，写入数据应该依次出现在 DATAIN 口上。

CMD[2:0] 为命令输入端口，分别表示读、写内存等待操作。其中，CMD=“000”表示无操作，内存条交给控制器管理，定其完成刷新工作；REFRESH 命令由外部逻辑指定特刷新的内存芯片信号，组号由 ADDR 的低位给出；LOAD_MODE 命令执行内存条工作寄存器初始化工作，初始化值由 DATAIN 的低 13 位决定，内存芯片组号同样由 ADDR 的低位给出；同理，ADDR 的低位也决定了预充电操作所对应的内存芯片组号。

CMDACK 为命令应答端口，表示命令已经被执行，使外部逻辑可以向控制器发出下一个动作。

4.2 状态机



该 SDRAM 控制器在中频数据海量存储系统中已得到应用。数据接收逻辑将接收到的中频采样数据整理后（拼接成 64bit），通过 SDRAM 控制器存入 SDRAM 阵列。存满后，数据输出逻辑将中频数据通过 SDRAM 控制器从内存条中取出，传输至上位机。其 VHDL 代码在 ATERA 公司的 FPGA—EP1C6Q240 中通过了 Quartus II 的仿真、综合和布局、布线。占用 499 个 logic cellk, 消耗了 8% 的逻辑资源。留有丰富的资源可提供给其它逻辑单元使用。

上面介绍了 SDRAM 的基本工作原理和一种简单的通用 SDRAM 控制器的实现。SDRAM 的控制机制比较复杂，具有多种突发读、写方式和工作模式（详细内容请参考 SDRAM 的数据手册）。但是，可以根据实现应用，实现其中的一个子集（基

本读、写、刷新操作)来满足实际系统的需要。用 SDRAM 实现大容量的高速数据缓存具有明显的优势,使用可编程器件实现 SDRAM 控制器则使之具有更高的灵活性,其应用前景广阔。

OFweek电子工程网