

一类基于 AMBA 总线的 SDRAM 控制器设计

孟中峰, 娄 晃, 张洵颖, 龚龙庆, 李振辉

(西安微电子技术研究所, 陕西 西安 710054)

摘 要: 提出一类 SDRAM 控制器的设计方法, 针对 SDR SDRAM 的突发特性, 采用数据预读取机制提高 SDRAM 的读取效率, 同时又加入了写指令 FIFO 消除读缓存带来的负面影响. 并且以该控制器在 AHB 总线的集成为例, 实现了控制器在 SoC 中的应用. 功能仿真和 FPGA 验证均表明本设计能够准确高效地实现对 SDR SDRAM 的访问控制.

关键词: SDR SDRAM; SDRAM 控制器; AHB

中图分类号: TN402

文献标识码: A

文章编号: 1000-7180(2014)09-0113-05

A Design for a Controller of SDRAM Based on AMBA

MENG Zhong-feng, LOU Mian, ZHANG Xun-ying, GONG Long-qing, LI Zhen-hui

(Xi'an Microelectronics Technology Institute, Xi'an 710054, China)

Abstract: This paper puts forward a data prefetching mechanism to improve SDRAM reading efficiency, also joined a write instruction FIFO to eliminate the negative impact brought about by the read buffer. And then applies the controller into an SoC as an AHB slave. Both functional simulation and FPGA verification show that this design can realize the efficient access to SDRAM.

Key words: SDR SDRAM; SDRAM Controller; AHB

1 引言

随着片上系统(System on Chip, SoC)应用领域的不断扩大,对系统的要求也越来越高.尤其在芯片速度日益提高的今天,慢速存储访问逐渐成为影响系统性能提升的一个重要瓶颈.在嵌入式系统中,SDRAM是最常用的大容量存储器,所以 SDRAM 控制器的性能高低对整个系统至关重要.

由于 SDRAM 存在访问延时大,读写速度较慢的问题,研究者提出诸多有效的方案对 SDRAM 控制器进行优化.比如文献[1]中提出了一种数据顺序预读取机制以提高访问效率,文献[2]中针对 SDRAM 的多 Bank 结构提出了 Bank 交错控制,文献[3]中提出了数据写缓存和双通道预取指令缓冲器,文献[4]中提出时间隐藏技术缩短了数据块的读写访问延时,文献[5]中则提出使用 Bank 访问历史

表来指导预充电策略.文献[6]中则是针对 HDTV 应用中运动估计算法的特点,采用相邻宏块存储于不同 Bank 等方法提高 SDRAM 访问速度,文献[7]中则提出如何在代码运行中最大限度的保证 SDRAM 不换页,以减少页缺失带来的等待时间,文献[8]中则进一步考虑 SDRAM 的换页特征,在软件上进行干预以减少换页情况的出现.

本文根据 SDRAM 器件特点,提出一类 SDR SDRAM 控制器的设计方案.然后以该设计在 AHB 总线上的集成为例,说明了在具体应用时控制器中读写缓存大小的设置方法.

2 SDR SDRAM 特点

SDR SDRAM 作为第一代 SDRAM,与 SRAM 相比,不仅实现了数据的同步存储,还具有以下特点^[9]:

(1)需要定时刷新. SDRAM 的基本存储单元是由一个晶体管与一个电容组成,通过电容电荷保存数据,由于电容不可避免地存在漏电,所以需要定时进行刷新防止数据丢失,刷新的最大间隔为 64 ms.

(2)使用前需要初始化. SDRAM 支持多种工作方式,比如是否猝发访问,在猝发时的 burst 长度、burst 方式以及工作时的列命令延迟(CL, CAS Latency)等参数都是可以改变的,以适应多种应用需要.但这也导致在使用 SDRAM 前必须先将这些参数配置好,否则 SDRAM 无法正常工作.

(3)支持 burst 传输. SDRAM 将以 READ 命令中的列地址为首地址,按照初始化时配置的 burst 方式以及长度,突发读出后续的数据,而不需要发新的列命令,这样便可以减轻控制器负担,提高访问效率.

3 SDRAM 控制器设计

基于以上对 SDRAM 工作特点的分析,设计 SDRAM 控制器如图 1 所示.

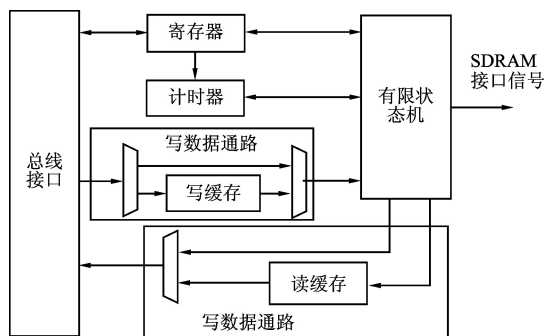


图 1 SDRAM 控制器结构框图

SDRAM 控制器的结构包括以下几个部分:总线接口、主状态机、读写缓存、计时器以及一些内部寄存器.

寄存器包括三类:时间参数寄存器、控制寄存器、状态寄存器. 时间参数寄存器用于配置 tRCD、tWL 等时间间隔的周期数,控制寄存器包括 SDRAM 的使能位、外接 SDRAM 的容量、列数、初始化时刷新次数以及工作时的 CL 周期数等参数. 这两个寄存器都是为了增加控制器的灵活性,使得控制器能够适应更多的工作频率以及更多型号的 SDR SDRAM. 状态寄存器只有一位,用于表征 SDRAM 的初始化是否完成,供用户查询使用.

总线接口部分需要根据控制器的具体应用环境进行设计,实现 SDRAM 控制器与目标系统的对接. 解析总线上对 SDRAM 的访问命令,从而启动主状态机进入相应的状态,完成对 SDRAM 中数据

的读写访问.

下面给出控制器的主状态机,并结合控制器的工作过程,从刷新策略以及预充电策略两个方面介绍本控制器的设计(图 2). 图 2 中浅灰色的为对 SDRAM 进行读操作的状态跳转过程,深灰色的为对 SDRAM 进行写操作时的状态跳转,其余为读写操作共用的状态.

传统的设计中为简化状态机的设计,通常将读和写的流程分别设计为两个相互独立的子状态机,对于读后写或者写后读的情况,必须经过预充电,主状态机回到 idle 状态后才能启动另一个子状态机,如果这两次访问的是同一行的地址,将有大量的时间浪费在 precharge 和 active 上. 所以,为提高控制器的效率,本设计中并没有设计独立的读/写子状态机,以便高效的在读写之间切换.

3.1 刷新策略

SDRAM 与 SRAM 最大的区别在于需要定时刷新,且当前业界的普遍标准为每 64 ms 至少完成一次刷新. 对于 SDRAM 而言刷新控制一般有两种策略:集中刷新和分布式刷新. 集中刷新是在每 64 ms 中专门留出一段时间连续发出 4 096 次(以验证中用到的 MT48LC4M32B2 为例)刷新命令. 而分布式刷新是将 4 096 次刷新命令平均分布到 64 ms 中,也就是每隔 15.625 μ s 发出一次刷新命令. 由于大多数 SoC 系统对实时性要求较高,无法空出大块的时间专门留给刷新操作,所以本方案采用分布式的定时刷新策略,通过在内部寄存器中配置刷新命令的时间间隔来实现. SDRAM 控制器主状态如图 2 所示.

对于控制器而言,每当其内部的计时器下溢后就会对 SDRAM 发出一次刷新命令. 发出刷新命令是其自发行为,不受总线命令的控制. 而且每次刷新操作的完成又需要 64 ns 的时间,在此期间总线上很可能出现新的访问请求. 所以必须建立合理的机制处理刷新与读写访问的冲突. 本设计中采用以下策略保证刷新能够按时完成.

(1)在刷新期间有新的访问命令时,锁存该次命令并阻塞总线,防止有新的访问命令到达,等待刷新结束后控制器马上执行该次访问,待访问结束之后才释放总线,以便接收新的总线命令.

(2)在读写访问的过程中到达刷新时间时,此时也将阻塞总线,防止总线上产生新的访问命令,从而使得控制器尽快的完成当前的读写操作,经过预充电之后回到 idle 状态,然后进行刷新,待刷新结束后

才释放总线,所以为了防止该次访问命令导致刷新过晚而丢失数据,在内部寄存器中刷新间隔要略小于 $15.625\ \mu\text{s}$ 。

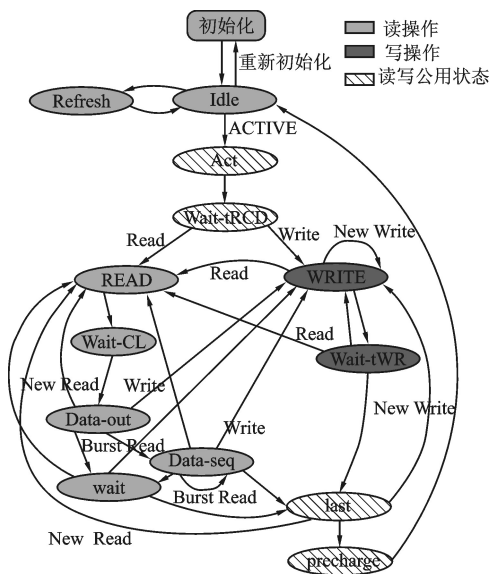


图2 SDRAM 控制器主状态机

3.2 预充电策略

由 SDRAM 的特性可知,在发出 READ 命令之后需要若干周期(根据 CL 配置值,为 2 或者 3 个周期)后 SDRAM 才能放出数据。如果是新的一行的访问,算上 ACTIVE 的时间,控制器需要更长的时间才能返回读数据,所以在进行完一次读写操作之后何时进行预充电,将成为影响控制器效率的主要因素。为提高效率,设计中需要在 READ/WRITE 周期对新的访问命令进行预判,并根据新的访问命令的不同情况实现灵活的预充电。

新的命令分为四种情况:

(1)该次访问为猝发传输,与上次访问的地址连续,控制器则无需再发出读命令。直接将 SDRAM 读出的第二个数据返回给总线,同时对下一条命令进行判断,直到猝发结束或者猝发长度到达设定的 BL(Burst Length)值。对于总线而言该次传输只需一个周期。

(2)与上一次访问的地址不连续但在同一行。这时也不需要预充电,在 Data_out 的下一个周期直接进入 READ 状态,发起一次新的读操作。SDRAM 最早在两个周期(CL 配置为 2 的时候)之后才能读出数据,而此时上一次突发读出的 4 个数据已经全部放出,数据端口不会存在冲突。

(3)与上一次的访问地址不连续且不在同一行。这时是最差情况,需要先通过预充电关闭当前行,然

后激活新的一行进行访问。

(4)新的命令为 IDLE,总线空闲,那么控制器则无需进行新的读写访问控制,这里我们选择了继续等待 BL 个周期,以等待 SDRAM 猝发读出的四个数据全部读出,同时监测总线命令,如果四个周期之后仍然没接收到总线上新的访问命令,则进行预充电,关闭当前行,主状态机回到 idle 状态。如果在此期间有新的访问,则同(2)、(3)中的情况,根据新的访问地址是否在同一行进行预充电。

综上可知,对 SDRAM 的访问地址连续性越强,该预读取机制对控制器效率的提升越明显,而地址不连续时,该机制也并不会影响控制器效率。

3.2.1 读缓存的设置

根据不同目标系统的行为特点,其总线上产生的猝发访问长度 n 的值也会不同(n 的值一般取决于系统的 Cache 结构以及指令系统的特点,常见的值为 0、2、4、8 等),在 SDRAM 初始化的时候将其 BL 值也配置为 n 。这样每次读操作的时候 SDRAM 的 DQ 端口上都会连续放出 n 个数据,为充分利用这一条件,设计控制器时再其内部增加了一个深度只有 n 的读缓存。该缓存单元只在 cache 不使能或者系统中没有 cache 的情况下工作。在每次读操作后,将 SDRAM 猝发读出的 n 个数据放入缓存当中。那么当有新的读命令时,将首先判断缓存是否命中,如果命中则直接将缓存中的数据返回给总线,而无需再启动主状态机从 SDRAM 中读取数据。这样在 cache 不使能的情况下在一定程度上实现了 cache 的功能,提高了 SDRAM 的访问效率。当系统中 cache 使能时,则该模块与 cache 的功能重复,没必要继续使能读缓存,产生不必要的功耗。

3.2.2 写缓存的设置

增加读缓存并采用预读取机制,很好地改善了连续读时控制器的性能,但是会影响到读后写性能。如图 3 所示。

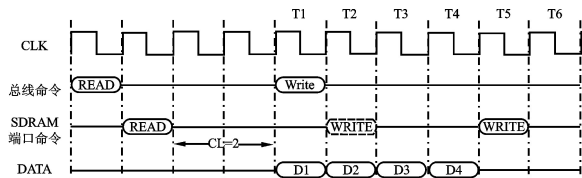


图3 读后写时序分析(以 $n=4$ 为例)

在 T_1 时刻,状态机进入 Data_out 状态,将读出的第一个数返回给总线,同时接收新的总线命令。假如此时的总线命令为写操作,正常情况下,控制器在下个周期就可以对 SDRAM 发出 WRITE 命令(如

图 3 中虚线部分所示),但由于采取了数据的预读取机制,在 $T_2 \sim T_4$ 时刻 SDRAM 读出的数据正在写入到读缓存当中,所以 WRITE 命令只能推迟到 T_5 时刻才发出,这样会阻塞总线,导致在 T_6 时刻才能够更新总线命令,制约了系统的运行速度.为改善这一情况,本设计中在控制器中又增加了写缓存区,通过由一组寄存器级联构成的 FIFO 保存写命令的数据以及地址,等到 SDRAM 空闲后执行写操作,同时不再阻塞总线的流水。

由图 3 可知,最差情况下写命令需要保存 n 个周期之后才能执行(在 T_1 时刻有写命令,但该命令在 T_{n+1} 时刻才能够执行),所以构成写缓存的 FIFO 只需要 n 级就可.其工作过程如图 4 所示。

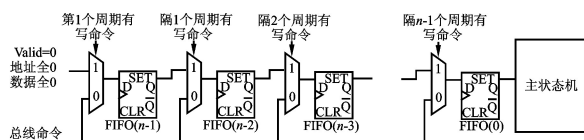


图 4 写缓存的工作过程

在没有读后写的情况出现时,该 FIFO 中打入的值为 0,相应的标志位 valid 也置为 0,表明 FIFO 中是无效值.对于读后写的第一条写指令, n 个 mux 的选择条件依次为,写指令在图 3 中的 T_1 时刻、 T_2 时刻、 T_3 时刻…… T_n (图 3 中 $n=4$)时刻到达.这样就保证了写指令能够在 SDRAM 空闲后第一时间被执行.如果是连续写的情况,每次收到写指令的时候,控制器将判断上一条有效的写指令是在 FIFO 的哪一级,然后将该次写指令写入相邻一级的 FIFO 中,以保证写指令能够连续的被执行.例如上一条有效的写指令在 FIFO(1)中时,当前的写指令将被写入到 FIFO(2)中。

通过增加写缓存,改善了设置读缓存后带来的不良影响.由读到写的切换不再需要等待 n 个周期,实现了读后写的无缝衔接。

4 控制器在 AHB 总线的集成

4.1 目标系统行为特点分析

本文中 SDRAM 控制器的应用环境如图 5 所示.作为总线从机,SDRAM 控制器对 SDRAM 的读写访问完全受控于 AHB 总线.而总线的访问命令又来自于作为总线主机的处理器。

所以目标系统的行为特点取决于 cache 以及处理器的行为特点.本系统中 cache 采用哈佛结构,数据 cache 和指令 cache 均为四路组相连,每路的容量是 8 kB,每行的容量是 16 B.cache 的使能可以通过

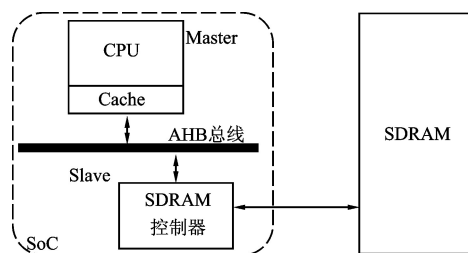


图 5 SDRAM 控制器应用环境

特权指令进行控制。

对于指令 cache,如果使能 cache,从缺失字到该行的最后一字都会被取回,进行对应的更新,同时取回的指令被送到处理器,如果由于处理器内部相关或者执行多周期指令的译码,导致处理器不能接受猝发取回的指令,那么处理器的流水线将被停顿,直到这次猝发取指完成.如果通过特权指令将 cache 置为 disable 状态,cache 仍将进行猝发取指,但取回的指令将被直接返回给处理器,并不更新 cache 本身的内容.在 cache 猝发取指期间,总线上产生的是 4 次渐增的猝发.综上可知,当 SDRAM 用于存放指令的时候,对其的访问总为渐增猝发,且最大的猝发长度为 4。

对于数据 cache 而言则没有猝发取数的行为,所以 SDRAM 用于存放数据的话,系统对其访问特点则取决于处理器的指令集.本系统中采用的是 Sparc V8 的指令集,该指令集中对 memory 访问一般为单次的访问,只有 ldd 和 std 指令^[10]才能在总线上产生对 memory 的猝发访问,且生的最大猝发长度为 2。

4.2 SDRAM 控制器在目标系统中的集成

由 4.1 节的分析可知目标系统中对于 SDRAM 访问的最大猝发长度为 4,且猝发方式只有渐增.所以在对 SDRAM 初始化时,将其工作方式配置为 BL=4 的渐增猝发即可.同时读写缓存的深度 n 也配置为 4。

同时设计 AHB 从机接口,实现控制器与 AHB 总线的对接.在 HSEL 信号有效时解析总线上的访问命令.启动主状态机进入相应的状态,完成对 SDRAM 中数据的读写访问.当控制器无法在一个周期内完成该次访问命令时,总线接口将会以拉低返回给总线的 HREADY 信号的方式阻塞总线的流水操作,直到该次访问完成。

5 SDRAM 控制器的实现以及验证

本设计通过 VHDL 语言实现,通过 Synplify 进

行综合,最大工作频率可达到 169.9 MHz,用到的 LUT 为 1 998 个.为检验控制器的实际应用效果,将控制器作为从机集成在某 SoC 芯片的 AHB 总线上,综合后在 FPGA 平台上进行验证,验证 SoC 环境以及实验平台的组成分别如图 6 (a)、(b)所示.验证所用 FPGA 型号为 Virtex-5 XC5VLX 330T,且依次外接了不同型号的 SDR SDRAM 进行验证.所用到的 SDRAM 型号分别为为美光的 MT48LC4M32B2、三星的 K4S641632C、现代的 HY57V641620,基本覆盖了 SDR SDRAM 的几大厂商,验证结果具有一定代表性.

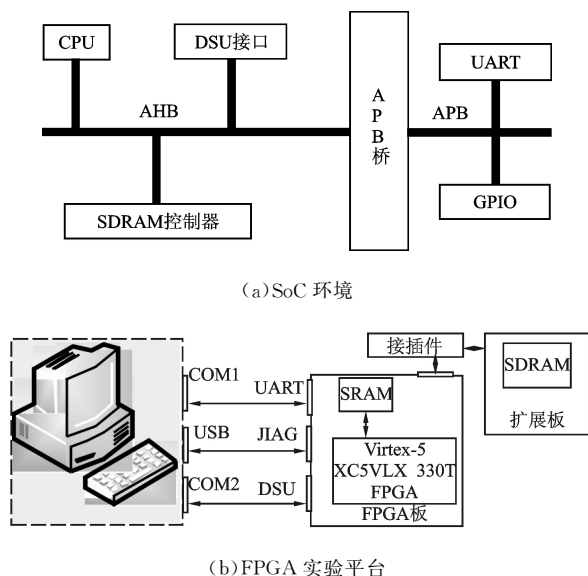


图 6 SDRAM 控制器验证环境

验证过程中,首先通过 JTAG 将硬件设计逻辑文件下载到 FPGA 中,测试程序通过 DSU 接口写入,利用 UART 在 PC 机上打印显示程序的执行结果.验证表明对任意地址的单次读写、连续读写,各种读写之间的交叉访问,以及整块地址的遍历访问,本文设计的 SDRAM 控制器都能够快速无误的完成.

6 结束语

本设计根据 SDRAM 行为特点,在控制器中加入读缓存区,实现了数据的预读取.同时又增加了写缓存机制,以消除预读取机制对读后写性能的影响.同时将 SDRAM 的时间参数、块容量、位宽等参数作为位寄存器可配交由用户根据不同情况进行配置,扩大了控制器的应用范围,具有一定的通用性.然后以控制器在 AHB 总线上的应用为例对控制器的应用方法进行说明:首先分析目标系统的行为特点,合理设置读写缓存的大小,采用最少的资源实现

对 SDRAM 的高效访问.FPGA 验证表明该方案切实可行,具有一定的通用性以及启发意义.

参考文献:

- [1] 张凯,李云岗.基于 AMBA 总线的 DDR2 SDRAM 控制器研究与实现[J].微电子学与计算机,2005,22(9):117-122.
- [2] Lee K B, Lin T C, Jen C W. An efficient quality-aware memory controller for multimedia platform SoC [J]. Circuits and Systems for Video Technology, IEEE Transactions on, 2005, 15(5): 620-633.
- [3] 张宇,时龙兴,王学香,等.面向片上系统的高性能 SDRAM 控制器设计[J].固体电子学研究与进展,2007,27(3):38-42.
- [4] 王斌,熊志辉,陈立栋,等.具有时间隐藏特性的数据块读写 SDRAM 控制器[J].Computer Engineering, 2009, 35(4):47-51.
- [5] 李文,唐志敏.一种减少内存访问延时的方法[J].Computer Engineering, 2006, 32(3):138-146.
- [6] 赵强,罗嵘,汪蕙,等.用于 HDTV 视频解码器的高性能 SDRAM 控制器[J].电子与信息学报,2007,29(6):1332-1337.
- [7] Watanabe T, Ayukawa K, Miura S, et al. Access optimizer to overcome the "future walls of embedded DRAMs" in the era of systems on silicon[C]// Solid-State Circuits Conference, 1999. Digest of Technical Papers ISSCC 1999 IEEE International. [s.l.]. IEEE, 1999: 370-371.
- [8] Miura S, Ayukawa K, Watanabe T. A dynamic - SDRAM - mode - control scheme for low - power systems with a 32 - bit RISC CPU[C]// Proceedings of the 2001 international symposium on Low power electronics and design. [s.l.]. ACM, 2001: 358-363.
- [9] Micro SDR SDRAM MT48LC32B2 Date Sheet [DB]. Micro, 2001.
- [10] The SPARC architecture manual[M]. Englewood Cliffs, NJ 07632: PTR Prentice Hall, 1994.

作者简介:

- 孟中峰 男,(1988-),硕士研究生.研究方向为 SOC 设计.
 娄 冕 男,(1987-),博士研究生.研究方向为多核处理器.
 张洵颖 男,(1981-),博士,高级工程师.研究方向为微处理器设计.
 龚龙庆 男,(1962-),硕士,研究员.研究方向为空间嵌入式计算机研究与应用.
 李振辉 男,(1988-),硕士研究生.研究方向为 SOC 的设计应用.