

# 32 位高性能 DMA 控制器的 VLSI 实现

全正元, 戎蒙恬, 刘文江

(上海交通大学芯片与系统研究中心, 上海 200240)

**摘要:** 高性能的 DMA 控制器是诸如高端微处理器和媒体处理器等 SoC 片上通信架构的重要组成部分。该文提出了面向 AMBA 总线的高性能 32 位 DMA 控制器的 VLSI 实现, 引入了双时钟域、同步复位门控时钟和握手信号路由阵列的设计。在 SMIC 0.18 $\mu\text{m}$  的工艺下, 可以达到 180MHz 的工作频率。应用于 32 位 RISC 处理器 ALP3310 中, 与使用软件传输相比, AHB 总线和 APB 总线的数据传输速度分别提高 80.0% 和 26.7%。

**关键词:** DMA 控制器; AMBA 总线; SoC

## VLSI Design of 32-bit High Performance DMA Controller

QUAN Zhengyuan, RONG Mengtian, LIU Wenjiang

(VLSI and System Research Center, Shanghai Jiaotong University, Shanghai 200240)

**【Abstract】** A high performance DMA controller is an important component of the onchip communication architecture of SoCs such as advanced microprocessors and media processors. This paper presents a VLSI design of an AMBA-oriented 32-bit high performance DMA controller, involving dual clock domain, gated clock with synchronous reset and handshake signal routing array design. With 0.18 $\mu\text{m}$  library technology of SMIC, a working frequency of 180MHz is achieved. In RISC processor ALP3310, compared with transfer by software, DMA increases data transfer speed by 80.0% and 26.7% on AHB bus and APB bus respectively.

**【Key words】** DMA controller; AMBA bus; SoC

随着 SoC 芯片性能的提高, 各模块间的数据交换成为提高微处理器系统运行速度的瓶颈。DMA 控制器能够有效替代微处理器的加载/存储指令, 显著提高系统的并行能力。传统的 DMA 的结构专用性强而可扩展性不足<sup>[1-3]</sup>。在高速并行系统中, 单时钟域的设计已不能有效控制功耗的增长。面对 SoC 不断扩大的规模, 还必须设计一种动态配置 DMA 通道的结构以提高逻辑资源的利用率。本文介绍了在 0.18 $\mu\text{m}$  工艺下, 面向完整的 AMBA 总线结构的高性能 32 位 DMA 控制器的 ASIC 实现, 提出了使用双时钟域配合具有同步复位的门控时钟 (Gated Clock) 实现功耗控制的方法。握手信号路由阵列的设计使得有限数量的通道可以匹配更多的总线外设。

### 1 DMA 结构的比较

AMBA<sup>[4]</sup> 是 ARM 公司提出的用于微处理器片上通信的先进的总线结构。一种典型的 AMBA 总线由 AHB 和 APB 总线分段构成。总线上的设备可以分为能够主动读写的主设备 (Master) 与只能接收来自 Master 请求的从设备 (Slave)。

近来针对 DMA 控制器的研究引出了新的 SoC 架构, 如使用分布式 Flyby DMA 结构<sup>[1]</sup>; 为数据吞吐量大的模块预设专用通道<sup>[2,3]</sup>等。这些结构的共同点在于数据在搬移过程中不需要经过缓存就可以达到指定地址。然而缺乏缓存结构的 DMA 有一定的局限性。从应用的角度来讲, 在音视频编解码程序中普遍存在同端口内存数据块搬移的操作, 而非缓存结构的 DMA 无法在同端口上实现单周期的先读后写操作。此外, 在 APB 总线上只能存在一个 Master 设备, 所以既无法为每个外设配置独立 Master 以获取数据, 也不可能在低速总线上完成单周期传输。从功耗的角度看, 当系统中存在较多 Master 模块时, 总线仲裁器的负担就被加重, 而仲裁器正是 AMBA 总

线功耗的主要来源<sup>[5]</sup>。使用缓存结构的系统级 DMA 尽管存在数据传输的延时, 但能适用于 AMBA 不同速度的分段总线, 可扩展性远高于非缓存结构 DMA。

如图 1 所示, 本文提出的 DMA 控制器在整个系统中可分为 3 个部分。DMA Master 端解析来自当前通道的命令包并发出 AHB 总线请求, 完成缓存区与 AHB 总线的数据交换。DMA 的引擎 (DMA Engine) 包含各个通道的逻辑资源与双口缓存结构 (Dual Port FIFO) 及相应的控制逻辑。DMA 控制器的 APB Master 发出 APB 的总线请求, 这些信号与来自 AHB 的转换信号一起在两段总线的桥接 (Bridge) 部分进行仲裁选择, 实现 APB 总线与 DMA 缓存区的数据交换。

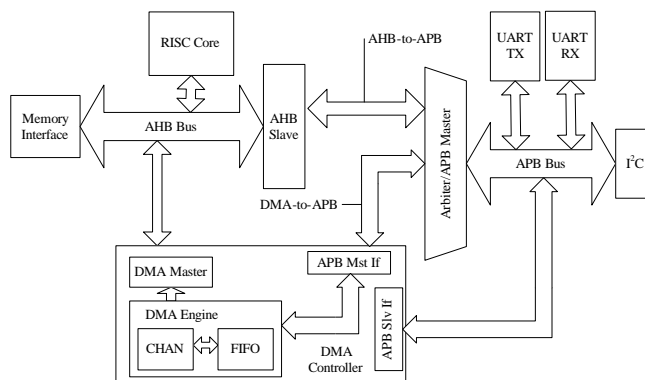


图 1 DMA 控制器的 AMBA 总线接口

**基金项目:** 上海应用材料研究与发展基金资助项目 (0501)

**作者简介:** 全正元 (1982 -), 男, 硕士生, 主研方向: 数字集成电路设计和 SoC 验证; 戎蒙恬, 博导; 刘文江, 讲师

**收稿日期:** 2006-05-25 **E-mail:** andyquan@sjtu.edu.cn

## 2 DMA 控制器的 VLSI 实现

### 2.1 DMA 的时钟域

在不具备 DMA 的 AMBA 总线中, APB 唯一的 Master 在 AHB 总线的驱动下进行数据传输。两段总线以串行的方式工作。从表 1 中可知, 由于在 AMBA 结构中微处理器不提供给 APB 的 Burst 操作, 完成对 APB 的 4 个 Word 的传输, 至少需要 20 个 AHB 的时钟周期。而当存在 DMA 控制器时, APB 总线能缓存需要和 AHB 交换的数据, 并且可以独立地驱动数据传输。不仅使传输周期减少了 50%, 而且实现了 APB 与 AHB 的并行运行。

表 1 各种总线/DMA 传输性能

总线时钟域	DMA	1 word 传输最小周期数	4 word 传输最小周期数
单时钟域总线	无	4	20
单时钟域总线	有	2	8
双时钟域总线	有	4	16

由于并行的实现, APB 总线上某些时间段中数据传输更加频繁。DMA 的 APB 信号进入总线以前必须经过仲裁, 当使用类似于高速总线上的 Burst 模式时, APB 可以多个连续两周期的传输来运行, 这增加了互连线的翻转频率。尤其在 DMA 多个通道相互交替占有 APB 时, 这种翻转间隔便被缩小。因此, 依据功耗  $P = CV^2f$ , AHB 总线的二分频时钟被用来驱动 APB 总线。在这种配置下, 相对于使用指令集进行的传输依然持有速度上的优势。

### 2.2 DMA 的内部结构

DMA 控制器的核心部分围绕通道状态机和双口缓存数据的 FIFO 结构来设计。如图 2 所示, 功能模块包括非整字地址对齐逻辑(Read/Write Aligner)、64×32bit 双口 FIFO、FIFO 控制逻辑、DMA 引擎、DMA 握手信号路由器(Handshake Router)和功耗管理电路。其中 DMA 引擎又包括 DMA 通道状态机逻辑和通道仲裁器(Chan Arb)。

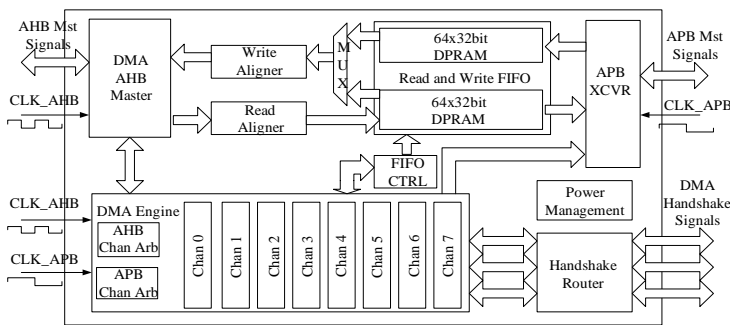


图 2 DMA 控制器内部结构

双时钟域设计的难点在于时钟域的划分以及不同时钟的相位要求。从图 2 中可以看到, DMA 的数据通路部分由 2 个总线接口及 2 块 64×32bit 的双口 SRAM 组成, 其中 DMA Master 接口与数据对齐逻辑由 CLK\_AHB 来驱动, 而 APB 端由 CLK\_APB 来驱动。DMA 的通道根据数据流方向共分为两种: 一种控制 AHB 上模块的数据交换; 另一种控制 AHB 与 APB 上模块的数据交换。

为了减小面积, 利用 APB 时钟是 AHB 时钟二分频的特点, 将两种不同类型的通道复用同一个存储端口。这样, 两种通道的状态机逻辑中使用各自的时钟产生 SRAM 读写地址与控制信号。为了支持多通道独立读写操作以提高并行度, 每个 AHB 与 APB 交换数据的通道必须使用读写独立的

SRAM。这样, 使用 2 块双口 SRAM 实现了不同时钟域缓存共享。每个通道拥有 2 块 8×32bit 的不同流向的缓存容量, 以匹配 DMA 的 AHB Master 端最高支持 4 个 Word 的 Burst 操作。

### 2.3 DMA 通道的状态机与仲裁

DMA 的通道是指能够控制数据从起点传输到终点的逻辑资源以及相应的缓存空间。图 3 展示了两种类型通道在传输数据过程中的状态机变化。其中虚线对应 AHB 到 AHB 的通道, 实线对应 AHB 与 APB 间数据交换的通道。

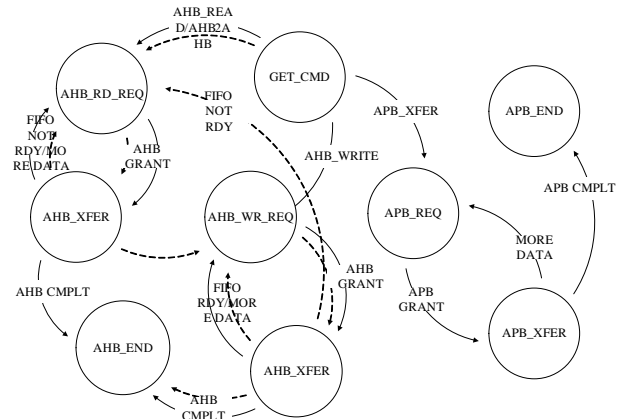


图 3 DMA 通道状态机

AHB 与 APB 的数据交换以接收来自 AHB 的命令符作为传输的开始。AHB 和 APB 能够根据当前的缓存充溢情况独立地发出总线请求。于是相应的逻辑各自进入 AHB\_RD(WR)\_REQ 和 APB\_REQ 状态, 并根据目的地址与当前剩余的传输字节数来调整单次传输的字节。DMA 的 AHB Master 和 APB Master 在状态机的控制下, 独立地对 Read FIFO 或 Write FIFO 的一个端口进行操作。

类似地, AHB 数据通道在接收命令符以后, 进入 AHB\_RD\_REQ 状态, 在 FIFO 不足半满的情况下, 从起始地址读出数据放入缓存; 在 FIFO 超过半满的情况下, AHB\_WR\_REQ 状态从缓存中取出数据送至目的地址。

由于在同一时刻只能有一个通道占有 AHB 或 APB 总线, DMA 多通道的总线请求必须经过仲裁器的仲裁。根据文献[6]的分析, Round Robin 策略在相应多设备的请求中能最有效地利用已知带宽。为了让 AHB 的传输减少延时, 在不同类型通道间使用固定优先级。在同类型通道间, 使用 Round Robin 策略使各设备有相同几率被服务。

### 2.4 DMA 握手信号路由阵列

当 AMBA 总线上的外设数量增加时, 数据源设备与目标设备之间的对应关系也随着数量线性增加。必须设计一种可以动态配置每个通道占有设备的方法, 以减少通道的数量。

本文提出了一种基于握手信号路由阵列动态配置通道的方法。DMA 的握手信号控制着通道起始和终止状态。DMA\_REQ 信号是目标设备对 DMA 控制器的输入, 标识可以开始传输数据。当传输完成后, 目标设备发出 DMA\_CMD\_END 信号以通知 DMA 结束传输过程。由于目标设备处理总线数据的速度可能低于总线的传输速度, 因此通常在总线接收端使用小规模的缓存。而握手信号即用来控制缓存数据不会在进入处理流程之前被来自总线的新数据覆盖。

图 4 展示了连接 8 个总线外设的 DMA 控制器的握手信号路由阵列的单元。2 个二级选择器阵列分别用作 2 个握手信号的路由。H1~H8 是来自目标设备的握手信号。每个通道使用 2 个 4 选 1 和 1 个 2 选 1 的选择器级联成一个单元来从 8 个握手信号中选出当前占有通道设备的信号。在命令字获取阶段得到的目标设备地址映射为一个 8bit 的信号（如 00000001, 00000010），输入 8/3 编码器来产生阵列的选择信号。当系统具有 16 个总线外设时，阵列单元为 4 个 MUX4、2 个 MUX2 和最后一个 MUX2 的三级级联，以此类推。

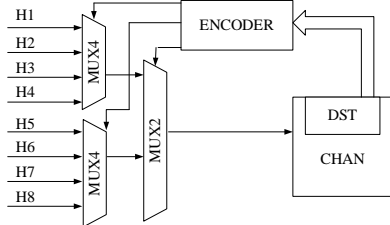


图 4 DMA 握手信号路由单元

### 2.5 功耗管理和同步复位的实现

在 0.18 $\mu\text{m}$ 的工艺下，由于时钟树的反转引起的电容充放电对功耗的贡献达到了整体功耗的 30%~40%<sup>[7]</sup>。一种有效的遏制此类功耗的方法是采用门控时钟。使用低电平有效的锁存器和与门串连的方法得到门控时钟，可以避免时钟毛刺的产生<sup>[8]</sup>。

由于时钟无效以后 DMA 模块进入状态锁存，在下一次时钟有效时，系统将以上一次时钟关闭时锁存的状态进入运行。为了避免一个不确定的状态在关闭时钟时被锁存，同步复位的电路将在时钟关闭之前先复位所有触发器电路。如图 5 所示，一个两级移位寄存器电路的输出用来产生门控时钟的使能信号 CLK\_DISABLE。当 Soft\_Rst 信号为低时，门控时钟的使能信号为 0，DFF3 输入为 0，时钟正常输入。当需要关闭时钟时，Soft\_Rst 先通过 APB 总线被写高，在第 1 个 APB 时钟周期后，模块同步复位成功且 Soft\_Rst 已经移位至 DFF1 的输出，随后组合逻辑把使能信号拉高，DFF3 的输入信号为 1。在第 2 个 APB 时钟上升沿，DFF3 输出为高，在随后到来的相应时钟域的低电平时段，LAT 通路，于是时钟被关闭。

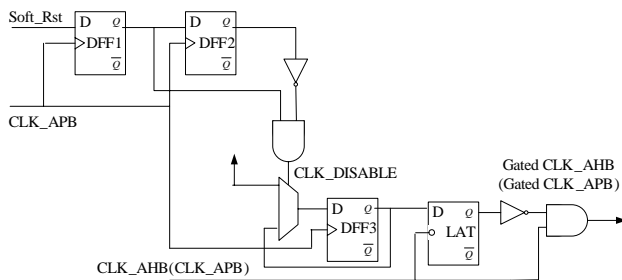


图 5 DMA 复位与门控时钟电路

### 3 仿真结果与性能

经过 Design Compiler 逻辑综合，DMA 控制器在 SMIC 0.18 $\mu\text{m}$ 的工艺下，能够达到 AHB 时钟域 180MHz，APB 时钟域 90MHz 的工作频率。

ALP3310 是基于 AMBA 总线开发并采用了该 DMA 控制器的 32 位 RISC 处理器，含有 UART、I<sup>2</sup>C 等 16 个总线外设。DMA 控制器具有 2 个 AHB 数据通道，6 个 AHB 与 APB

的数据通道。

不同的内部结构和系统仲裁的实现都会影响 DMA 的总线占有率。仿真表明，忽略 AHB 总线竞争以及 DMA 内部通道竞争的影响，DMA 的 AHB 端口平均每 10 个 AHB 周期可以完成 4 个 Word 的传输，达到约 280MB/s 的传输速度。

在 AHB 和 APB 上对比 DMA 与软件传输数据的速度。在 AHB 上，分别用微处理器和 DMA 将 SRAM 中的 128 个字节的数搬移到 SRAM 的另一段地址中。如表 2 所示，DMA 的 AHB 传输对比软件传输具有 80.0% 的速度优势。

表 2 DMA 与指令传输比较

传输方法	传输类型	平均周期数	总周期数
指令传输	AHB 至 AHB	30	N/A
DMA 传输	AHB 至 AHB	6	N/A
指令传输	AHB 至 APB	N/A	7 100
DMA 传输	AHB 至 APB	N/A	5 400

对于 APB 的传输，通过微处理器直接写 APB 外设花费了 7 100 个 AHB 周期。而写入 SRAM 仅花费了 5 400 个 AHB 周期，并且微处理器随即被释放，剩余的工作由 DMA 完成，从而减少了 26.7% 的传输时间。

### 4 结束语

本文提出了面向 AMBA 总线的 32 位高性能 DMA 控制器的设计方法，引入了双时钟域、同步复位门控时钟和握手信号路由阵列的设计，并根据仿真结果进行了性能分析。实验结果表明，采用了该 DMA 控制器的 32 位 RISC 处理器 ALP3310 的总线传输速度得到大幅提高。

### 参考文献

- 1 Olugbon A, Khawam S, Arslan T, et al. An AMBA AHB-based Reconfigurable SoC Architecture Using Multiplicity of Dedicated Flyby DMA Blocks[C]//Proc. of the Design Automation Conference. 2005: 1256.
- 2 Yuan Hang, Chen Hongyi, Bai Guoqiang. An Improved DMA Controller for High Speed Data Transfer in MPU Based SoC[C]//Proc. of the 7<sup>th</sup> International Conference on Solid-state and Integrated Circuits Technology. 2004: 1372.
- 3 Shi Xinlei, Yang Jun, Lu Shengli. The Design and Optimization of DMA Controller in Embedded System-on-a-Chip[J]. Electronic Engineer, 2004, 30(1): 5.
- 4 ARM Corp. AMBA 2.0 Specification[Z]. 2006-04. [http://www.arm.com/products/solutions/AMBA\\_Spec.html](http://www.arm.com/products/solutions/AMBA_Spec.html).
- 5 Lahiri K, Raghunathan A. Power Analysis of System-level On-chip Communication[C]//Proc. of International Conference on Hardware/Software Codesign and System Synthesis. 2004, 1: 236.
- 6 Polmi F, Bntazzi D, Bmini L, et al. Performance Analysis of Arbitration Policies for SoC Communication Architectures[J]. Kluwer Journal on Design Automation for Embedded System, 2003, 8(2): 189.
- 7 Hodges D A, Jackson H G, Saleh R A. Analysis and Design of Digital Integrated Circuits: in Deep Submicron Technology[M]. McGraw-Hill Companies, Inc., 2003.
- 8 Emmett F, Biegel M. Power Reduction Through RTL Clock Gating[C]//Proc. of Synopsys Users Group Conference, San Jose. 2002.