

# 一种基于总线控制器的 SoC 功耗分析方法

郑伟, 李东晓

(浙江大学信息与电子工程学系, 杭州 310027)

**摘要:** 总线是观测数据流行为从而进行媒体处理 SoC 芯片系统级功耗分析的较佳研究对象。Wishbone 总线具有简单、灵活、免费等特点, 是具有较强竞争力的系统芯片总线(SoC Bus)标准之一。在媒体处理 SoC 芯片的 Wishbone 总线控制器中增加具有功耗分析功能的专用模块, 可以在不改变正常集成电路 EDA 设计流程的情况下较好地完成任务, 在低功耗设计中具有广泛的应用前景。

**关键词:** 系统芯片总线; 功耗分析; 电子设计自动化; Wishbone 总线

## A System-level Power Analysis Method Based-on SoC Bus Controller

ZHENG Wei, LI Dongxiao

(Department of Information Science and Electronic Engineering, Zhejiang University, Hangzhou 310027)

**【Abstract】** SoC bus based power analysis technique is appropriate for system level power estimation. Wishbone is one of the most prospective SoC bus standards. A system level power analysis method based on Wishbone bus controller is purposed here. Through the technique, the power estimation at the early design stage is performed without modification of EDA design flow. The power consumption of main memory of a typical SoC platform is applied to testify the method and generate good results.

**【Key words】** SoC bus; Power analysis; EDA; Wishbone bus

随着半导体工艺技术以及 VLSI 芯片应用需求的发展, 集成多种功能甚至整机的系统芯片(System-on-a-Chip, SoC)设计成为超大规模集成电路设计领域的新挑战。将各种通用或者专用 IP 核模块连接起来构成完整的电子系统, 是系统芯片设计的主流趋势。完成这种模块连接任务的通常是具有一定规则的、可扩展性好、速度快的总线, 一般称为 SoC 总线。与其他外设总线、总线结构及其互联技术直接影响 SoC 芯片的整体性能, 是目前 SoC 设计方法学研究的重点和难点。

另一方面, 无论是应用于便携式电子产品的消费类芯片还是非电池供电的高性能 SoC 芯片, 低功耗设计都具有相当重要的意义。前者要求通过降低芯片功耗延长系统待机时间, 提高产品竞争力; 后者要求通过低功耗降低峰值功耗, 提高芯片的可靠性, 降低封装成本。因此, 将功耗作为速度、面积之外的第三维设计约束条件是 SoC 芯片设计的重要趋势。

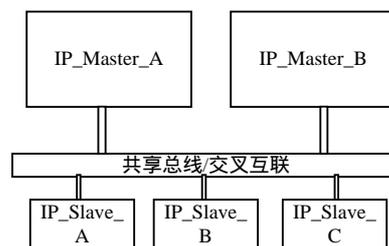
本文结合 SoC 总线和 SoC 芯片低功耗设计技术的研究, 提出了一种简单的高层次功耗分析技术和相应的低功耗设计流程。利用现有的设计仿真模型, 将功耗分析模块嵌入 SoC 总线控制器内, 在不影响正常 EDA 设计流程的前提下, 在完成系统模型仿真、验证工作的同时实现相应层次的功耗分析。在基于 Wishbone 总线规范搭建的多媒体处理 SoC 平台设计中使用了这项技术, 较好地实现了原有的设想, 并计划在应用于 IPTV 机顶盒的 H.264/AVC 解码 SoC 芯片的设计中普遍使用该方法。

### 1 Wishbone 技术特征

Wishbone 总线标准最早由 Silicore 公司提出, 现在由 OPENCORES 组织负责维护, 它具有结构简单、组织灵活、功能强大、便于移植、完全公开、完全免费等特点, 极有可能被 IEEE 采纳作为未来 SoC 片上总线的通用标准, 具有良好的市场应用前景。

Wishbone 总线规范对开发工具和目标硬件没有特殊规定, 只是定义了一种 IP 核之间的公共逻辑接口, 包括一套标准的握手控制信号、数据信号及其总线周期, 减轻了 SoC 系统的模块集成难度, 提高了 IP 核的可重用性、可移植性和可靠性, 从一定意义上可以大大缩短 SoC 芯片产品设计的 Time-to-Market 时间。

Wishbone 总线的主要技术特征可以概括如下: (1) 以一种体系结构解决所有 SoC 中 IP 核模块的集成应用; (2) 硬件接口信号逻辑简单、耗费硬件资源少; (3) IP 核之间的通信采用主从机制, 采用握手协议, 时序简单; (4) 数据总线支持 8~64 位可扩展; (5) 支持使用者定义标志位(TAG)确定数据传输类型; (6) 由终端用户定义仲裁方式。



共享总线在某个时刻只有一条通路存在; 交叉互联则允许许多个连接通路同时存在。

图1 共享总线连接方式和交叉开关连接方式

Wishbone 总线结构方便简单, 其应用领域较为广泛。它可以很方便地应用于嵌入式控制系统和高性能系统中, 在

**基金项目:** 国家自然科学基金资助项目(90307002)

**作者简介:** 郑伟(1974—), 男, 博士、讲师, 主研方向: 集成电路设计, 多媒体信号处理以及低功耗系统设计; 李东晓, 博士、讲师, 收稿日期: 2006-02-16 E-mail: zhengw@isee.zju.edu.cn

实现高性能系统时,可以在这个总线结构上预留一些其他接口。Wishbone 总线具有较强的灵活性,由于 IP 核种类多样,其间并没有一种统一的连接方式,为满足不同系统的需要,它提供了 4 种不同的 IP 核互联方式:(1)点到点方式,用于两个 IP 核直接互联;(2)数据流方式,用于多个串行 IP 核之间的数据并发传输;(3)共享总线方式,支持多个 IP 核共享一条总线;(4)交叉开关方式,支持同时连接多个主从部件,可以提高系统吞吐量。值得注意的一点是,Wishbone 总线的交叉开关互联方式是把通信领域的网络交换技术应用于系统芯片设计的范例,适应于片上网络(Network-on-Chip, NoC)技术的发展趋势。

## 2 基于总线的功耗分析技术

在目前主流的 EDA 设计流程中,一般采用图 2 所示的功耗分析方法,利用 Power Compiler 等商用功耗分析软件,根据一定的工艺库参数结合仿真或者其他途径得到的硬件翻转信息,计算功耗。这种方法的不足之处在于通常必须在完成逻辑综合之后才可进行,计算量相对较大。

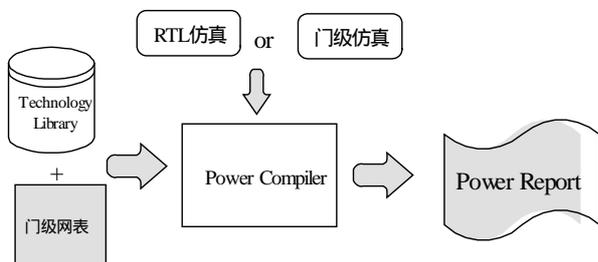


图 2 基于商用功耗分析工具的分析设计方法

一般认为,在较早的设计阶段采取的措施、作出的决定将对其后的设计工作产生更大的影响,功耗分析和低功耗设计也是如此。在 SoC 芯片系统层,总线结构以及由此确定的 IP 互联技术,是这一抽象设计层次最为显著的设计实体。将总线技术和功耗分析技术结合起来,是 SoC 设计方法学发展的必然趋势。

片上存储器是高性能芯片功耗的最大来源。超大规模集成电路的发展趋势之一就是片上存储器面积的急剧增加。SoC 芯片一般有多个独立的片上存储器模块,包括 CPU 核所需的片上 Cache 或者 DSP 核所需的片上 Memory,以及各个 ASIC 模块所需的片上缓存空间。增加片上 Cache、片上 Memory 和片上缓存是提高 SoC 芯片性能的最有效途径之一。据 OTRS 预测,在 2014 年高性能芯片中片上存储器面积将占据芯片面积的 90%,其相关功耗将成为芯片系统功耗的最主要来源。

与此相适应的是芯片所需要处理的数据量的急剧增加。众所周知,数字信号处理系统都是数据流驱动的,媒体处理系统更是如此。以处理高清信号(1080i)的 H.264/AVC 视频解码器为例,每秒需要处理输出的数据量为 1.4Gbp。这些数据在片上流动、存储、计算、交换、输出,引起了芯片结构中绝大部分的硬件操作。从某种意义上讲,“数据流”是媒体处理 SoC 芯片系统功耗的最大来源。

上述所有的数据流动,在 SoC 芯片体系结构中都与总线结构密切相关,都要通过总线(或者说总线控制器)来连接、交换。因而总线是观测数据流动从而进行系统功耗分析的最佳研究对象。Wishbone 总线采用主从方式通信,信号时序简单,总线控制器结构也较为简单。例如,完成一次写操作只需要配合使用 Master 端的 STB 信号、CYC 信号、WE 信号,

以及 Slave 端的 ACK 信号。读操作控制信号除 WE 信号之外基本一致。因此,在 HDL 描述的总线控制器中只需加入简单的代码就可以实现对上述信号的观测,从而清晰地了解相关模块的相应总线行为。

根据  $P_{dyn} = \alpha \cdot f \cdot C_l \cdot V_{dd}^2$  可知,通过观测总线的数据流行为可以得到包括存储器在内的模块活动度 ( $\alpha \cdot f$ ),配合以相应模块一定精度的电气特性 ( $C_l \cdot V_{dd}^2$ ),就能够计算相应模块的动态功耗,由此就可以抓住系统功耗的最主要部分。基于上述功耗分析数据,可以实现图 3 所示的低功耗设计流程,在芯片设计的每个抽象设计层次在进行系统仿真验证的同时进行功耗分析和低功耗设计工作:通过合理地组织片上存储系统以及模块互联、数据搬运机制,有效地调度片内、片外数据的交换和片内数据的流动,提高片上数据传输效率,满足高性能、低功耗的系统设计要求。随着模块电气特性 ( $C_l \cdot V_{dd}^2$ ) 数据的逐步趋于精确,功耗分析也逐渐从最初的定性分析变为定量分析。

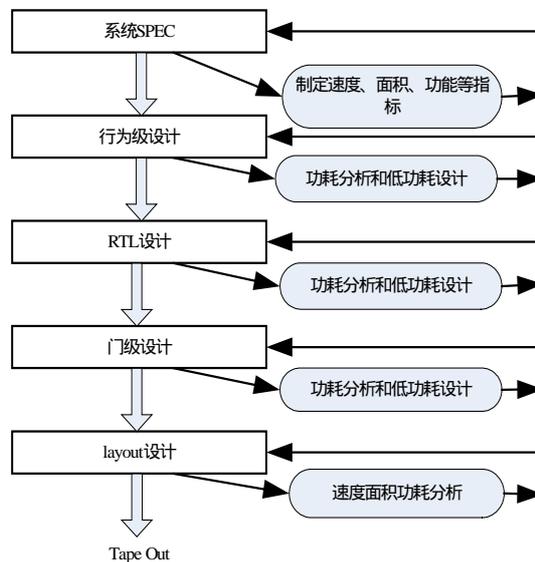


图 3 本文适用的低功耗设计流程

## 3 技术应用以及结论

在 H.264/AVC 视频解码芯片设计中,以开放源码的 32 位 OR1200(OpenRISC 系列处理器核中的一种)作为嵌入式处理器核,采用 Wishbone 总线构建了相应的 SoC 平台,如图 4 所示。其中,以 Wishbone 总线连接 CPU 核及其片上 Cache,运动补偿模块,熵解码模块, IDCT/IQ 模块,去块滤波模块以及相应的片上缓存,片外 Flash,片外 SDRAM 以及其他外设等。所有关键的模块都是通过 Wishbone 总线控制器联结,提供数据的交换和供给。因此,根据“数据流即功耗”的原理,统计和分析与 Wishbone 总线相关的“数据流”功耗,是得到 SoC 平台系统功耗的重要手段。

从系统层次考虑,根据某模块的功耗特征(单次操作的平均功耗数据)和一定时间内的翻转次数就可以计算得到该模块的功耗,包括平均功耗和动态功耗。具体而言,在与存储系统相关的 Wishbone 总线控制器模块中,通过嵌入信号访问统计模块,就可以通过统计系统对主要存储器的访问次数,结合存储器功耗模型,就能得到存储器相关功耗的动态分布,特别是可以得到峰值功耗。在此基础上,可以分析得到视频解码芯片中与存储器输入输出活动密切相关的其他重要模块

(如运动补偿、IDCT 变换、去块滤波等模块)的功耗,进而完成系统层次的功耗分析工作。

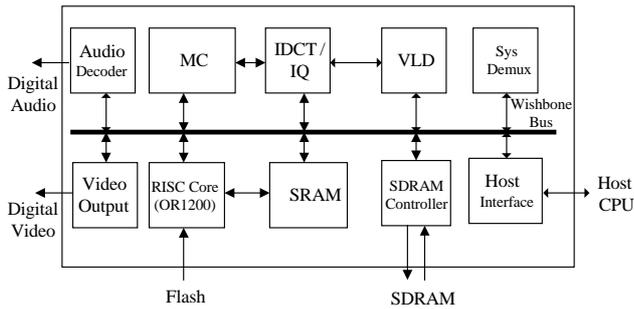


图 4 H.264/AVC 解码器 SoC 平台

在芯片设计工作中,分析了 uClinux 初始化、MPEG-2 系统层 TS 流解复用、MP2/MP3 音频解码以及 H.264/AVC 标准 CAVLC 解码过程等任务的 SRAM 存储器数据访问情况。在本文中,只以嵌入式 CPU 所需要的存储器系统为研究对象,通过统计作为系统引导程序存储器的 flash 和 CPU 主存空间的 SRAM 在实时操作系统 uClinux 的初始化过程中的数据访问情况,并由此初步分析了 SRAM 部分的功耗,得到了相对精度的分析数据。这些定性分析的数据对于我们接下来进行系统层次低功耗设计具有相当重大的价值。

设计工作中,每 65 536 个时钟周期统计一次存储器访问次数,以下是嵌入总线控制器模块内的完成功耗分析工作的部分 verilog 代码:

```
//synopsys translate_off
wire sram_rd_i, sram_wr_i;
assign sram_rd_i = !wb_ss_we_i & wb_ss_ack_o;
assign sram_wr_i = wb_ss_we_i & wb_ss_ack_o;
power_analysis pa_sram(.clk_i(clk),
    .rst_n(rstn),
    .rd_i(sram_rd_i),
    .wr_i(sram_wr_i));
//create the log files
integer fsram;
initial fsram = $fopen("sram_act.log");
always @(posedge wb_clk)
    if (period_cnt == 16'b0)
        $fdisplay(fsram, "%t [read]= %d\n [write]= %d", $time, sram_rd, sram_wr);
// synopsys translate_on
```

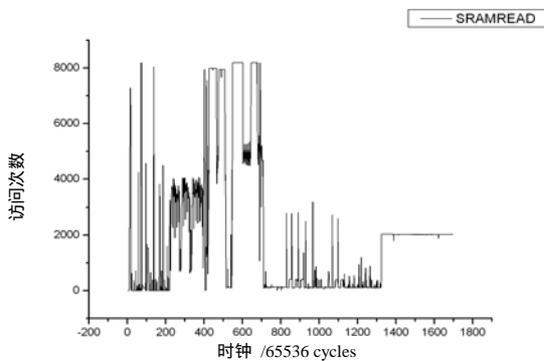
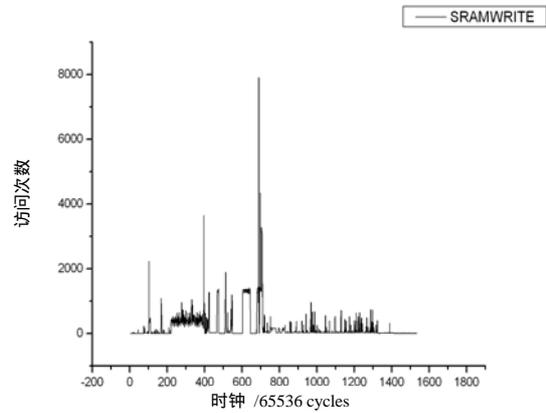


图 5 uClinux 初始化过程 SRAM 读操作统计

相应的 SRAM 数据访问统计如图 5、图 6 所示,根据某类型片上 SRAM 工作在 1.8V 电压下,平均读、写电流分别是 3.5mA 和 4.0mA 的实验数据,计算得到其动态功耗,如图 7 所示。观察其分析结果,可清楚看到操作系统初始化过

程中作为 CPU 主存的 SRAM 存储器的动态功耗和峰值功耗。

这种基于 SoC 总线总线控制器的功耗分析方法简单易行。它充分利用了现有的系统仿真模型,在完成系统仿真、功能验证的同时,快捷、方便地完成了相应层次功耗分析的工作,既在较早的设计阶段得到了系统功耗数据,又节省了芯片开发时间。接下来的工作将利用这一技术进一步分析 RTOS + SoC 系统平台的功耗,并利用功耗分析结果改进 SoC 系统结构以及实现实时操作系统的低功耗研究,最终实现低功耗、高性能的设计目的。



(每 65536 个时钟周期统计一次,时钟频率为 50MHz)

图 6 uClinux 初始化过程 SRAM 写操作统计

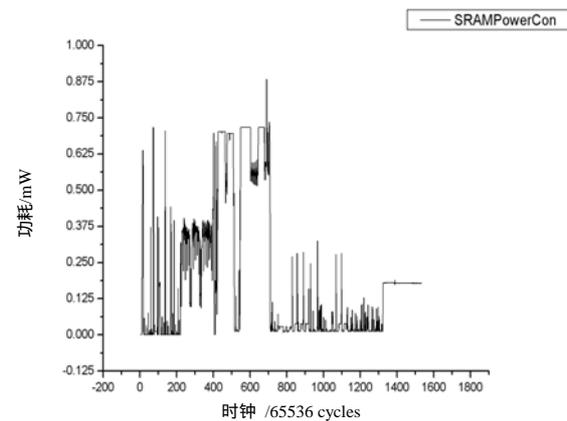


图 7 uClinux 初始化过程 SRAM 功耗的动态分布

## 参考文献

- 1 Ryu K K, Shin E, et al. A Comparison of Five Different Multiprocessor SoC Bus Architectures[C]. Proc. of Euromicro Symposium on Digital Systems Design, 2001-09-04: 202-209.
- 2 Li Dongxiao, Yao Qingdong, Liu Peng, et al. A Bus arbitration Scheme for HDTV Decoder SoC[C]. Proc. of APCCAS '02, 2002, 2: 79-83.
- 3 田 泽, 张怡浩, 于敦山等. WISHBONE IP 核互联总线[J]. 半导体技术, 2005, 30(1): 28-31.
- 4 Brooks D. Power-aware Micro Architecture: Design and Modeling Challenges for Next Generation Microprocessors[J]. IEEE Micro, 2000, 20(6).
- 5 Bhaumik B, Pradhan P, Visweswaran G S, et al. A Low Power 256 kB SRAM Design[C]. Proc. of Twelfth International Conference on VLSI Design, 1999: 67-70.
- 6 International Technology Roadmap[Z]. <http://public.itrs.net/>, 2003.