

基于开放源代码的硬件设计方法研究

麦宋平^① 张春^② 杨昆^① 王志华^②

^①(清华大学电子工程系 北京 100084)

^②(清华大学微电子所 北京 100084)

摘要: 可重用性是当今超大规模集成电路设计的必要元素。与传统的封闭源代码付费 IP 相比, 开源硬件以共享设计文档和 IP 模块的方式为硬件设计的重用提供了更加彻底有效的解决办法。基于开源硬件的 SoC 设计方法以其开放性和灵活性正被越来越多的设计者所接受并付诸实用。该文对开源硬件的相关概念、意义、面临的问题及发展前景进行了较为详细的介绍, 并以开源处理器的设计作为实例, 对基于开源硬件的设计流程进行了深入的探讨。

关键词: 开放源代码; 开源硬件; 处理器设计; 设计流程; 设计方法学

中图分类号: TN43

文献标识码: A

文章编号: 1009-5896(2007)07-1761-04

Research on Open-Source Hardware Based Design Method

Mai Song-ping^① Zhang Chun^② Yang Kun^① Wang Zhi-hua^②

^①(Department of Electronic Engineering, Tsinghua University, Beijing 10084, China)

^②(Institute of Microelectronics, Tsinghua University, Beijing 10084, China)

Abstract: Reusability is the key element in today's VLSI design. Open-source hardware provides a more thorough and effective way for design reusability than traditional charged closed-source IP by sharing its design documents and IP modules. And open-source hardware based SoC design method is being accepted and practiced by more and more designers. This paper introduces relative concepts, benefits, problems and perspectives of open-source hardware in detail, and discusses open-source hardware based design flow in depth by exempling an open-source processor design.

Key words: Open-source code; Open-source hardware; Processor design; Design flow; Design methodology

1 引言

随着半导体制造技术的高速发展, 集成电路的复杂度呈指数型增长, 设计难度越来越大; 另一方面, 集成电路行业的激烈竞争使得产品设计周期缩短, 验证和测试成本倍增。在这样的背景下, 设计经验的共享和积累成了提高硬件设计效率的必要条件, 新的商业模式——付费 IP 应运而生。然而, 这种商业模式下的资源共享有着其固有的缺点。首先, 多数 IP 的设计细节并不对使用者公开, 使用者很难根据自身需求对其进行修改; 其次, IP 附带的高昂专利和版权费用通常使个人、中小科研机构或企业难以承受。开放源代码硬件(简称开源硬件: Open-Source Hardware, OSH)的出现, 为缓解这些问题提供了有效的途径。

开源硬件的出现, 得益于自由软件模式的成功和硬件描述手段的进步。上世纪 80 年代初, 由美国人 Stallman 发起的自由软件运动, 催生了包括 Linux 操作系统在内的一系列开放源代码软件, 并且以 GNU 通用公共许可证的形式提出“反版权”(或“版权属左”, Copyleft)的概念^[1]。经过近 20 年的

发展, 以 GNU 许可证形式传播的开源软件已经成为软件发展进程中的重要组成部分, 甚至被认为是软件开发的一种基本方式^[2], 从而为开源硬件的发展提供了重要的参考模式。另一方面则是硬件描述语言(Hardware Description Language, HDL)如 VHDL 和 Verilog HDL 的发展, 使得硬件设计模型能够以软件的形式存在并传播。目前绝大部分开源硬件都是以硬件描述语言的形式出现的。

本文首先在第 2 节中介绍开源硬件及 GNU 许可证的概念, 接着在第 3 节中讨论基于开源硬件的设计流程并在第 4 节中以 OR1200 处理器的开发实例作为说明, 第 5 节列举开源硬件的意义, 第 6 节讨论开源硬件设计面临的问题及发展前景, 第 7 节给出结束语。

2 相关概念

2.1 开源硬件

开源硬件所指的源代码公开, 具有丰富的内涵。开源组织对开源硬件的定义主要基于以下 3 点^[3]:

(1)所有的硬件设计文档都是可以自由获取的, 包括原理图, HDL 源代码和版图文件等。

(2)基于硬件及其接口的所有软件都是可以自由获取的, 包括接口驱动、指令集、软件编译器等。

2005-12-05 收到, 2006-03-21 改回

国家自然科学基金(60372021) 和国家高技术(863)SoC 片上系统重大专项(2003AA1Z1100)资助课题

(3)与硬件使用相关的所有文档都是可以自由获取的,包括软件使用说明、硬件端口连接信息等。

总之,只有在修改、实现和使用该硬件所需的全部信息都可以自由获取情况下,该硬件才算是符合真正开放源代码的规范。

2.2 GNU 公共通用许可证

GNU公共通用许可证(General Public License, GPL)是由自由软件运动的发起者Stallman等人起草的具有法律效应的、旨在保证GNU软件可以自由地“使用、复制、修改和发布”的协议条款。对于使用GPL发布的软件,不管是免费或者收费的(自由软件或开源软件仅指开放源代码,并不意味着免费),使用者都有权利得到所有的源程序,并且可以自由地使用或修改直至重新发布^[1]。由GPL许可的软件派生出来的软件同样也是GPL许可的软件。

GPL许可证还派生了类似的LGPL许可证。除此之外,常见的开源软件许可证还有MIT许可证、BSD许可证和Artistic许可证等^[4]。在众多的许可证当中,以(L)GPL许可证条文最为严格,影响最为广泛。虽然(L)GPL许可证最初只适用于开源软件,但由于条文上具有普遍的适用性,现在也被广泛地应用到开源硬件上。当前绝大部分开源硬件都是以基于(L)GPL许可证的HDL程序形式发布的。

3 基于开源硬件的设计流程

基于开源硬件的 SoC 设计指的是在已有开源 IP 模块的基础上按照需要进行单片系统的集成。它的设计流程和基于传统 IP 的 SoC 设计流程相比并没有本质上的差别,但开源硬件允许用户自由修改源代码,使得整个设计流程显得灵活和高效。如图 1 所示,基于开源硬件的 SoC 设计流程由自然语言描述的系统规范开始,最终得到符合规范的电路芯片,整个流程分为系统设计和常规设计两个部分。其中,系统设计流程大致可分为以下 4 个步骤:

(1)功能定义 根据系统的自然语言描述,采用高级编程语言(如 C 或 HDL)对系统的功能进行完整的定义。把系统划分成若干功能模块,指定模块间的接口,明确系统的整体行为特征,最终得到符合规范的系统功能模型。

(2)开源 IP 模块映射 在允许的范围内尽可能多地搜索开源 IP 模块,并根据系统的功能模型对得到的 IP 模块进行评估,选出能够最大程度满足系统各模块功能要求的 IP 模块,最终得到一组开源的系统构件。

(3)开源 IP 模块修改 如果开源模块不能完全满足系统的要求,则需要对其进行修改。开源模块的开放性使得用户可以根据需要自由地进行个性化设计,这是开源硬件的一个根本优势。修改工作以完全满足功能模型中各模块的功能要求为准,最终得到优化的系统构件。

(4)系统集成 把所有的系统构件组合在一起,进行全局仿真,从系统角度对各构件进行验证,确保各模块间接口时序的正确性。系统集成和验证以步骤(1)定义的功能模型为准,最终得到完全满足功能要求的可综合的系统网表。

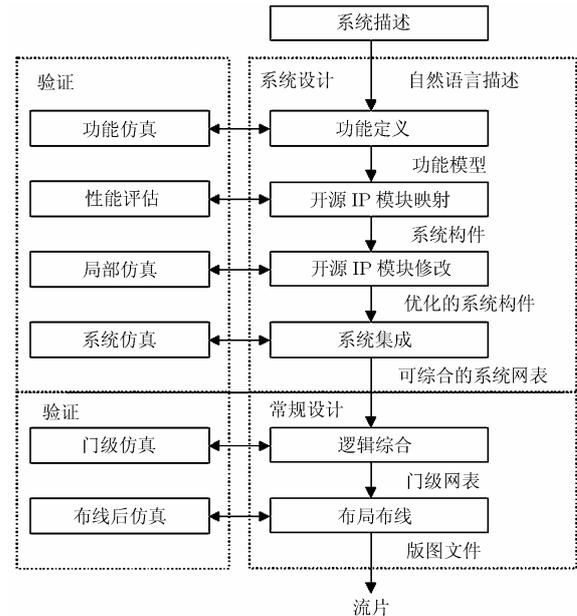


图 1 基于开源硬件的 SoC 设计流程

上述步骤是可以反复的。当发现某个步骤的最终输出结果无法满足要求时,设计者必须返回到前面的步骤重新设计,直至每一步都能得到符合要求的结果。

系统设计完成以后就进入常规设计阶段,这是标准的 ASIC 设计流程,包括逻辑综合、版图设计、芯片生产以及封装测试等环节^[5]。

4 设计实例

下面介绍的设计实例,是笔者采用上述流程设计的一个 32 位定点多媒体处理器。该处理器是在开源的通用处理器 IP 核 OpenRISC1200^[6](以下简称 OR1200)的基础上,通过修改原有的数据路径得到的。新处理器的最主要特点是具备非对齐数据存取和二维 DMA 传送能力,以及大容量片上数据存储。加上 OR1200 原有的强大的处理能力,使得它非常符合嵌入式多媒体处理的应用要求。下面简单介绍对 OR1200 开源模块的评估和改进,以及最后的物理实现结果。

4.1 对 OR1200 的评估

OR1200 是由 OpenCores 组织^[7]开发的一个基于 MIPS 架构的 32 位标量 RISC,具有哈佛结构、5 级流水线、虚拟内存管理以及基本的数字信号处理功能,可用于中高端的嵌入式网络、移动电话及汽车电子等领域。其结构框图如图 2 所示。

OR1200 具有很多优良的开源特征。除了其强大的硬件处理能力外,OpenCores 组织还专门提供了用于 OR1200 的 GCC 编译器、OR1KSIM 仿真器以及基于 NCVerilog 的仿真环境,并且将标准的 Linux 和 μ CLinux 操作系统移植到了 OR1200 上。所有这些软硬件源代码均完全公开,从而为设计者进行 IP 内核验证、优化设计甚至修改指令集提供了很大的自由空间。

因为处理器是最核心也是最复杂的数字系统,所以成为

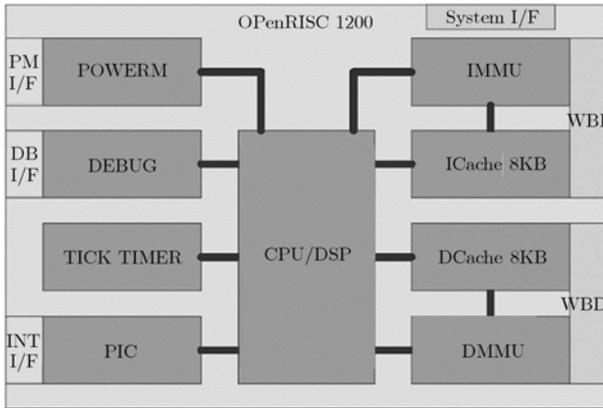


图 2 OR1200 处理器结构框图

了很多开源硬件组织竞相开发的对象。除OR1200 外，还有 LEOX 组织开发的 Leon2^[8]、查尔姆斯理工大学开发的 JAM CPU^[9] 以及自由 CPU 联盟开发的 F-CPU^[10] 等等，它们都是具有通用架构的开源处理器核。从硬件角度看，这些处理器核的处理能力与 OR1200 相当甚至更高，但它们的软件支持及开放程度却远远不及 OR1200。在满足性能要求的前提下，OR1200 是开源处理器设计的最佳选择。

4.2 对 OR1200 的改进

为了使 OR1200 能够满足多媒体处理对数据吞吐能力的需求，必须对其数据路径进行改进。图 3 和图 4 分别显示了 OR1200 和新处理器的数据路径。

针对多媒体数据处理的特点，新处理器对 OR1200 的数据路径进行了以下 3 个方面的改进：

(1) 多媒体尤其是图像处理，数据量大，通常要求处理器有较大的片上数据缓冲能力以加快处理速度。为此，OR1200 原有的单独一块指令和数据共享的片上快速存储单元 (QMEM) 被分成了两块，指令快速存储器 (IQMEM) 和数据快速存储器 (DQMEM) 分开，这在加快数据存取速度的同时

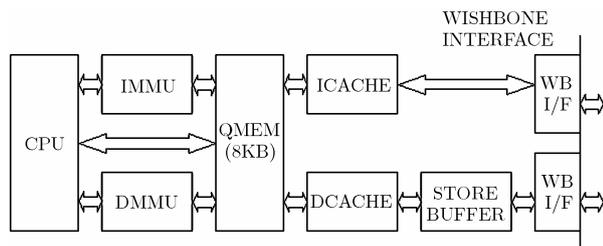


图 3 OR1200 处理器的数据路径

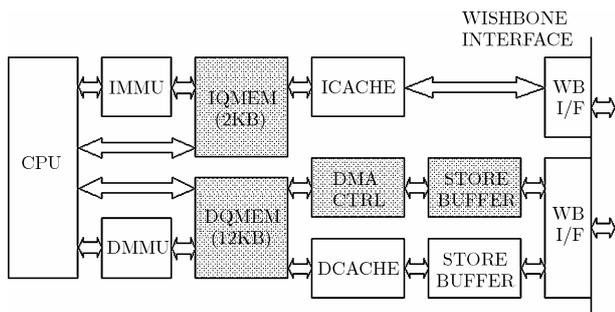


图 4 改进后的处理器数据路径

也加大了存储器的容量。

(2) 对于 32 位的通用处理器来说，每次存取数据都必须以 4 个字节为单位，并且要求低位地址是 32 的整数倍。但许多多媒体数据格式如 MPEG2、MPEG4 或 H.264 都是以单字节为基本单位，操作上会经常出现地址不与 32 的整数倍对齐的情况。为了提高数据的存取效率，在片上数据存储单元上加入了非对齐数据存取机制，使得每次 32 位数据存取只要求低位地址是 8 的整数倍，而不必与 32 的整数倍对齐。

(3) 多媒体处理往往比较规则，数据的可预知性强，而可重用性差，所以在处理器与外部存储器交换数据的时候采用 DMA 方式往往比采用 Cache 效率要高。另外，多媒体变换 (如 DCT, VLC 等) 都是基于数据块 (如 8×8 的像素块) 的矩阵操作，通用处理器在此类数据块的存取上效率很低。在 OR1200 原有的 Cache 结构上增加了具有二维数据存取能力的 DMA 通道，可以一次性存取预定大小的数据块，很好地解决了上述问题。

4.3 处理器的实现

改进后的 OR1200 最终流片并实现了原定的所有功能，各项指标见表 1。

表 1 处理器芯片各项指标

制造工艺	UMC 0.18- μm CMOS 1P/6M
片上存储器	8k 字节指令 Cache
	8k 字节数据 Cache
	2k 字节指令快速存储器 12k 字节双端口数据快速存储器
性能	80MIPS
多媒体增强	单指令乘累加 (MAC)
	非对齐数据存取 二维 DMA 数据传送
芯片面积	4.8mm × 4.8mm
晶体管数	1.36 × 10 ⁶ 个
电源电压	1.8V (内部逻辑), 3.3V (I/O)

5 开源硬件的意义

开源硬件是应技术和市场的发展出现的，它作为一种新的设计理念，能够有效地解决现有设计上和市场上存在的问题^[3, 11]。它的以下几点好处是显而易见的。

(1) 开源使新系统的开发时间减少，成本降低

传统的商业运作模式把技术保密当成理所当然的竞争手段。由于不能有效地共享资源，重复开发、反向设计在行业中盛行，造成了极大的资源浪费。而开源硬件则通过共享技术文档和 IP 模块的方式，使设计者可以在已有的技术基础上设计出新的系统，避免了不必要的重复劳动，既缩短开发时间，又降低成本。

(2) 开源使硬件设计和开发变得更加规范可靠

集成电路规模的飞速增长对独立于物理实现的设计单元库有着巨大的需求。但是，传统的封闭源代码硬件设计使得单元库设计上十分混乱，出现了大量互不兼容的、未经足

够验证的商业库,阻碍了 SoC 设计向规模更大、性能更优的方向发展。与此不同的是,开源硬件的设计文档可以在开放平等的基础上供公众审查,在相互借鉴和比较的基础上,硬件单元的设计能够变得有序和规范。而且,开源设计由于不存在技术保密问题而可以随时获得众多的支持,不断的改进和充分的验证必然使开源硬件变得更加可靠。

(3) 开源硬件使其他设计者可以从中学习并相互交流

数量众多的开源硬件文档包含了设计者的风格和经验,可以为不同层次的使用者提供丰富的学习材料。设计文档的广泛共享,使使用者可以从中获取毫无保留的技术知识,还可以通过对不同设计的比较来提高学习效率。同时,使用者还可以通过修改开源硬件把自己的设计经验也附加上去,反馈给原设计者。这种相互学习和交流的方式,不仅增加了设计者和使用者的经验,也使硬件设计的总体水平不断地得到提升。

6 面临的问题及发展前景

虽然开源硬件有许多优点,并且这种思想已经吸引了大量的设计人员,但开源系统设计模型的实现在范围上和实用性上仍面临着很大的挑战^[3, 11]。开源硬件设计发展上的最大障碍在于其硬件物理实现上的困难,这也是开源硬件与开源软件之间最明显的差别,本质上则是硬件设计和软件设计的不同。

首先,硬件设计的流程比软件设计要复杂得多,尤其是前者还包括了后端物理设计的环节。硬件设计最终目的是制造出芯片或者电路来,而对于同一个设计,需要根据不同的工艺进行特殊的后端(版图)设计。整个后端设计就是对生产实际的迭代逼近过程,但由于实际制造过程中存在一定的不确定因素(如温度和压力的偏差),导致后端设计过程十分复杂,而且很难使设计结果和生产结果达到完全的吻合。软件设计则不同,只需在计算机上通过仿真得到正确的结果,而且结果可重复,不存在偏差的问题。其次,硬件实体无法像软件一样通过拷贝的方式来流通。硬件的“拷贝”就是一次复杂的再制造过程,而且这种制造还必须基于同样的生产工艺,其复杂程度根本不是软件拷贝所能比拟的。这一点严重限制了开源硬件的传播。

然而最为重要的还是,整个后端设计所需要的EDA工具价格十分昂贵,硬件的制造成本更是惊人。尤其随着工艺尺寸推进到深亚微米级别,这种经济上的问题就显得更加突出。其最终的结果就是造成许多开源硬件只停留在设计仿真阶段而无法进行实际生产,同时促使开源硬件设计人员转向可编程逻辑设备上的 IP 内核设计,因为这些设计不但可以较好地避免实际生产中的工艺偏差问题,还可以降低成本,并且设计本身具有较好的延展性,设计源码也具有可拷贝性。另外,为了进一步充实开源硬件的概念,降低设计成本,部分开源社区(如gEDA组织^[12])也正在努力尝试建立开源的EDA软件来替代商用EDA工具。

除了上述问题外,与开源软件的起步阶段类似,开源硬

件也存在开源硬件来源少、许可制度不完善、开源硬件组织松散以及缺乏有效的商业盈利模式等诸多问题。尽管如此,开源硬件设计以其良好的开放性将不可避免地对硬件技术产生重大影响,就像开放源代码为软件开发带来革命一样,最终使整个硬件设计行业受益。

7 结束语

本文对开源硬件的相关概念、意义、面临的问题及发展前景进行了较为详细的介绍,并且以 OR1200 处理器作为设计实例深入探讨了基于开源硬件的 SoC 设计方法。对开源硬件的调研和实践表明,开源硬件以及基于其上的设计方法,以开放的设计理念和灵活的设计流程,正日益成为硬件设计者相互学习、共享设计经验和提高设计效率的重要手段。

参考文献

- [1] GNU Project. Licenses. <http://www.gnu.org/licenses/licenses.html>. 2005.
- [2] Mockus A, Fielding R T, and Herbsleb J. A case study of open source software development: the Apache server. Proceedings of the 2000 International Conference on Software Engineering. Limerick, Ireland. 2000: 263-272.
- [3] Jamil K. Free chips for all. <http://www-128.ibm.com/developerworks/library/os-openhw.html>. 2005.
- [4] Open Source Initiative. License Index. <http://www.opensource.org/licenses/>. 2005.
- [5] Neil W and David H. CMOS VLSI design: a circuit and system perspective. Boston: Addison-Wesley, third edition. 2005: 520-535.
- [6] OPENCORES-Project. OpenRISC 1000: OpenRISC 1200. http://www.opencores.org/projects.cgi/web/or1k/openrisc_1200. 2005.
- [7] Opencores.org. Home page. <http://www.opencores.org/>. 2005.
- [8] LEOX project. Free Resources for System on Chip. <http://www.leox.org/>. 2005.
- [9] JAM CPU. A RISC CPU core written in VHDL. <http://www.etek.chalmers.se/~e8mn/web/jam/index.html>. 2005.
- [10] Freedom CPU project. F-CPU. <http://www.f-cpu.org/f-cpu.org/>. 2005.
- [11] Jamil K. Open hardware design trend. <http://www.opencores.org/articles.cgi/view/12>. 2005.
- [12] GEDA project. Home page. <http://www.geda.seul.org/>. 2005.

麦宋平: 男, 1980年生, 博士生, 研究方向为集成电路设计、数字信号处理等。

张春: 男, 1972年生, 副教授, 研究方向为数模混合电路设计、生物医学电子等。

杨昆: 男, 1979年生, 博士生, 研究方向为集成电路设计、多媒体信号处理等。

王志华: 男, 1960年生, 教授, 研究方向为电路与系统, 包括集成电路设计、数模混合信号集成电路的设计和测试、集成电路的计算机辅助设计、最优化设计等。