

# 基于 ESL 的华睿 2 号 SoC 系统级架构设计

刘 静, 周海斌

(南京电子技术研究所, 江苏 南京 210039)

**[摘要]** 华睿 2 号是我国自主研发的一款 8 核微处理器, 基于 CMOS 40nm 工艺设计, 片内集成 8 核 DSP、AXI 总线以及 PCIE/DDR3 等多种高速接口的大规模片上系统芯片(SoC)。鉴于华睿 2 号设计复杂, 需使用电子系统级(ESL)解决方案, 确定 SoC 级架构。ESL 设计方法学已被越来越多的复杂 SoC 设计所采纳, 利用可裁剪的 TLM 建模方法快速搭建系统, 进行系统级验证, 探索不同架构设计对系统性能的影响, 从而寻找最优方案。本文利用 Synopsys ESL 解决方案, 创建 SoC 不同的架构, 并在创建的架构上运行雷达信号处理典型应用, 分析处理时间、总线压力等架构性能, 通过优劣对比最终确定最优架构。

**[关键词]** 华睿 2 号, 片上系统芯片, 电子系统级, 架构设计

**[中图分类号]** TP391.9 **[文献标志码]** A **[文章编号]** 1672-1292(2016)04-0069-09

## Architecture Design for Huarui-2 SoC Based on ESL

Liu Jing, Zhou Haibin

(Nanjing Research Institute of Electronics Technology, Nanjing 210039, China)

**Abstract:** Huarui-2 is a 8 cores high-performance embedded processor (based on cmos 40nm) developed by NRIET. Huarui-2 is constructed by 8-core DSP, AXI bus, PCIE/DDR3 and other high-speed interface. Given the SoC complexity, we need to use the electronic system level(ESL) solution, and determine SoC-level architecture. ESL methodology has been adopted by more and more complex SoC designs. Designers can build and verify their SoC platforms rapidly, do architecture exploration to know the system performance with the generated Virtual Prototype with the scalable TLM modeling method. Via Synopsys ESL solution, we create different architectures and run typical radar signal processing applications based on the architectures. By analyzing the processing time and bus pressure, we ultimately determine the best architecture of Huarui-2.

**Key words:** Huarui-2, SoC, ESL, architecture design

SoC 芯片设计需要进行大规模、深入的系统级仿真, 以确保设计的体系架构合适均衡。传统的基于 RTL 的 SoC 架构及性能分析, 往往由于 RTL 设计实现的代码量大, 仿真速度慢, 一旦发现架构设计存在性能瓶颈时, 必须重新修改架构方案, 重新编写 RTL 代码, 导致迭代次数增加, 迭代周期过长, 因此已不可行。

为了让仿真具有较高的执行性能, SoC 结构设计正在向电子系统级(ESL)解决方案发展。一般认为 ESL 是“在满足系统约束的条件下, 为了增强对系统的理解和提高系统功能性成功实现的可能性所采取的一种适当的抽象建模方法”<sup>[1]</sup>。ESL 设计在 SoC 系统设计初期的架构定义和性能评估方面有着重要作用<sup>[2]</sup>, ESL 解决方案的目标在于提供让设计人员能够在一种更高抽象层次上对芯片进行描述和分析的工具和方法。在这种抽象层次上, 设计人员可以对芯片的 SPEC 进行功能性的描述, 而没有必要过早涉及硬件(RTL)实现的具体细节<sup>[3]</sup>。

华睿 2 号 SoC 系统级架构 ESL 模型开发, 目的是为华睿 2 号提供全芯片级建模, 通过对不同层次和不同总线结构的系统架构的分析, 寻找适合的最优架构和设计方案, 为后续的软件虚拟化平台开发提供基础, 使得软件早于 RTL 硬件完成前开发, 实现软硬件协同设计。

收稿日期: 2015-12-26.

基金项目: 核高基重大专项课题(2012ZX01034-001-002).

通讯联系人: 周海斌, 研究员级高级工程师, 研究方向: 数字芯片设计. E-mail: zhouhb\_863@163.com

## 1 华睿2号简介

### 1.1 多核 SoC 发展趋势

半导体工艺的迅速发展使芯片的速度和集成度越来越高,芯片设计者可用以实现芯片的晶体管资源数目和种类不断增加,芯片体系结构正面临着新的挑战和创新机遇。

目前国际主流的芯片厂商都将多核集成在芯片上,且将许多不同功能的核集成在一起。例如传统的 DSP 芯片厂家 TI、Freescale 的代表芯片 TMS320C6678、P4080,核数最多达到 8 核,片上集成了 DSP 核、无线通信算法加速核等,片上存储分为一级 RAM 32–64 kB、二级 RAM 128–512 kB、三级 RAM 2–4MB,集成高速串行接口 RapidIO、PCI-E、网络接口 GMAC 等。传统的 CPU/GPU 厂家 Intel、AMD、Nvidia 等,将 CPU 和 GPU 集成在同一芯片上,采用单指令多数据流(Single Instruction Multiple Data, SIMD),最新的 Intel 的 AVX(Advanced Vector Extensions)支持 256 位的向量指令。综上所述,目前主流处理器芯片发展趋势如下:

- (1) 异构多核,通用核(CPU/DSP)加上专用核(硬件加速核、GPU、FPGA);
- (2) SIMD 技术仍为 DSP 设计的主要方向之一, SIMD 将增加到 256 位宽,由此提供较大的并行处理;
- (3) 片上存储分为多级存储,一般为三级,总存储量达到 8MB;
- (4) 集成片内 DDR3 内存接口,高速 IO 采用 RapidIO、PCI-E 等标准接口。

### 1.2 华睿2号介绍

华睿2号是南京电子技术研究所研发的一款 8 核 DSP 芯片,采用异构簇式多核架构,集成面向灵活编程的向量 DSP 核、面向专用算法加速的可配置专用处理核,以及为了保证数据供应的多级交叉开关、多级存储和计算核分簇排布。专用处理核采用粗粒度静态可配置设计,将基本运算单元排列成计算阵列,增加存储阵列,通过数据通路的可配置,实现核心算法的加速。

#### 1.2.1 技术指标

华睿2号芯片采用 8 核架构,由 4 个向量 DSP 核和 4 个可配置专用处理核(Reconfigurable Application Specific Processor, RASP)构成,模块间通过交叉开关 XBar 互连。芯片主要包括以下模块:

- (1) DSP 核:具有 2 个 256 位向量运算部件,512 kB 二级 Cache/RAM,主频 1.0 GHz;
- (2) RASP 核:具有浮点运算阵列和 512 kB Local Memory,通过配置可实现专用信号处理算法的计算,主频 1.0 GHz;
- (3) DMA:负责数据搬运和矩阵转置等;
- (4) Share Memory:共享的片上 RAM,共 4 MB,500 MHz;
- (5) DDR3:内存控制器,64 位,1 066 MHz,峰值总带宽 256 Gbit/s;
- (6) RapidIO/PCI-E:复用 PHY,4 路,每路支持 4 lane,每 lane 支持 3.125 Gbps,峰值总带宽 100 Gbps。

#### 1.2.2 可行的架构

华睿2号架构中,1 个向量 DSP 核与 1 个 RASP 核紧耦合互连,组成一个计算簇。4 个计算簇与其他模块(如 DMA、Share Memory、DDR3 控制器)之间存在两种可行架构:

- (1) 架构方案 1:单个计算簇通过交叉开关 XBar0 与 DMA、Share Memory、DDR3 控制器及 RapidIO 等模块互连,组成基本结构单元;4 个结构单元组成顶层架构;
- (2) 架构方案 2:4 个计算簇、DMA、Share Memory、DDR3 控制器和 RapidIO 等模块均同等接入一级 Xbar,组成顶层架构。

本文将通过 ESL 方式对两种可行的架构展开分析,通过映射应用场景,仿真对比两种架构的性能。

### 1.3 华睿2号架构

架构探索采用数据流方式,完成以下两种方案的架构的数据流场景分析,根据评测分析,提出架构修改方案,迭代建模分析,最终得出最优架构。

构方案 1 如图 1 所示。处理器核与 IO 通过 XBar0 紧密相连成簇,簇间通过 XBar1 相连,分为两级交叉开关,每级的端口数相对较少,可保证簇内较高的传输效率。该架构对软件开发要求较高,若软件未规划分配好各核任务,将会导致传输延迟大,竞争严重,实际传输效率低。

架构方案 2 如图 2 所示。DSP、RASP 处理核组成簇,通过 Xbar0 共享一个 Xbar1 的端口,且共享完全

对等的 L3 和内存空间,方便软件编程. Xbar1 互连的交叉开关延迟较长,核与 IO 间的传输易互相竞争、互相干扰. L3 占用 Xbar1 两个端口,但在 L3 内部会分为 4 个 bank,以减小访问冲突. DDR3 的数量可选 2 组、3 组或 4 组,从物理实现角度看,2 组可实现 1 600 MHz,3 组可实现 1 333 MHz,4 组可实现 1 066 MHz.

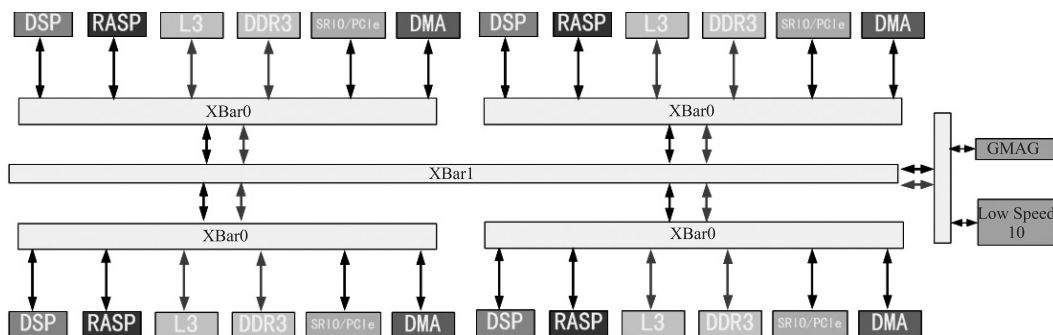


图 1 架构方案 1

Fig. 1 Architecture 1

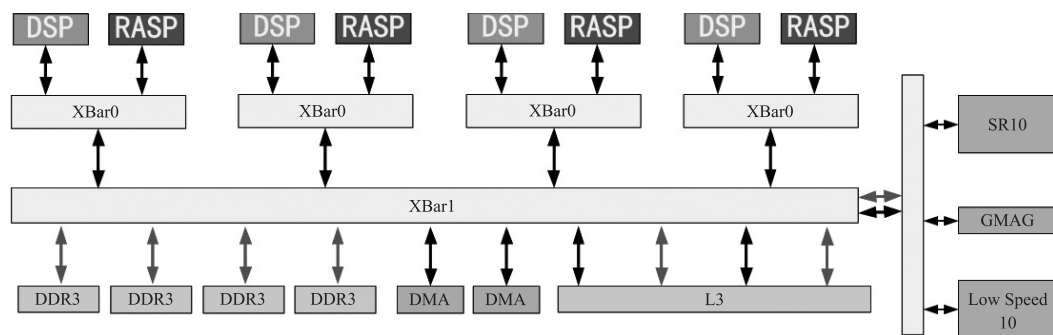


图 2 架构方案 2

Fig. 2 Architecture 2

#### 1.4 建模需求

本文采用 ESL 模型来进行架构探索、软件开发和性能分析. 按模型的构建从易到难、仿真速度从快到慢、仿真精准度从低到高,ESL 模型相应分为 3 种:数据流 ESL 模型、指令精确 ESL 模型和周期精确 ESL 模型. 其中,架构探索主要基于数据流 ESL 模型完成,软件开发基于指令精确 ESL 模型完成,性能分析基于周期精确 ESL 模型完成. 流程如图 3 所示.

本文仅介绍数据流 ESL 分析. 数据流 ESL 模型是 ESL 建模的第一个阶段,目的是快速探索出适合信号处理的系统架构,其建模速度和仿真速度快,但精准度低.

由于数据流 ESL 模型仅对 AXI 交叉开关、DDR3 控制器等进行建模,对于向量 DSP 核、RASP 核、RapidIO 等采用虚拟处理器单元(VPU)替代,因此可获得较快的建模速度和仿真速度.

在数据流 ESL 模型中,通过施加信号处理领域典型算法和产品领域典型信号处理测试任务(TaskGraph). Task-Graph 中包含信号处理流程、任务分解、参数选取及性能指标要求等,同时编写了 Mapping 文件,将 TaskGraph 映射到架构后,重点对系统架构中 AXI 和 DDR3 的带宽、访问延迟和访问冲突等性能进行分析,进行合理的软硬件划分,探索出适合信号处理的系统架构.

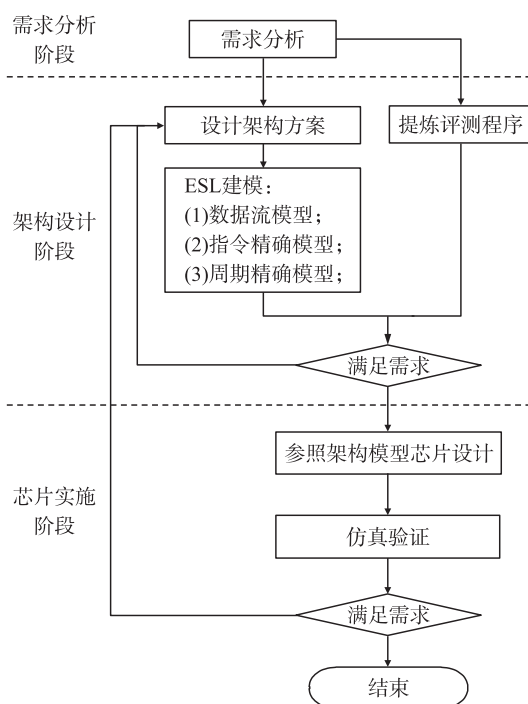


图 3 模型架构及性能分析流程图

Fig. 3 Taskgraph of architecture analysis

## 2 ESL 建模

### 2.1 Platform Architect

Platform Architect<sup>[4]</sup>(PA)是 Synopsys 公司发布的一款 SoC 系统级架构设计的 ESL 软件工具,提供基于 SystemC TLM 设计方法学的多核 SoC 高效设计、性能分析等开发平台. 利用 PA 工具,可以探索和优化 SoC 软硬划分、架构组织结构,特别是总线互联结构、Memory 组成等子系统,从而设计出最优系统架构. 由于 PA 具有多种 IP 模型以及良好的可视化分析界面,已成为业界 SoC 架构设计领域(特别是 ARM AMBA-based 的 SoC)主流的架构设计平台.

### 2.2 PA 架构分析

#### 2.2.1 架构运行算法

数字脉压技术基于匹配滤波信号处理理论,通过对回波信号作匹配滤波卷积运算,在输出端获得最大信噪比完成信号检测<sup>[5]</sup>. 数字脉压通常可采用时域 FIR 和频域 FFT 两种方法实现.

脉冲压缩时域处理是直接对雷达回波信号进行卷积运算,其公式为:

$$y(n)=s(n)*h(n), \quad (1)$$

式中, $s(n)$ 为 A/D 采样之后的回波信号; $h(n)$ 为匹配滤波器的冲激响应; $y(n)$ 为脉压输出.

根据卷积定理,2 个信号时域的卷积相当于频域的乘积,因此式(1)可变换为:

$$y(n)=\text{IFFT}[S(w)H(w)]=\text{IFFT}\{\text{FFT}[s(n)]\text{FFT}[h(n)]\}. \quad (2)$$

频域脉冲压缩方法图 4 所示. 脉冲压缩时域处理比较直观、简单,适用于处理点数较小的情况. 当处理点数较大时,时域卷积运算量迅速增大,此时宜采用频域 FFT 方法实现,以减少运算量<sup>[6]</sup>. 频域脉压基于华睿 2 号架构的多核处理结构图如图 5 所示.

#### 2.2.2 架构方案 1

##### (1)架构创建

方案 1 的系统架构为 4 个异构簇结构,每个异构簇包含 1 个 DSP 核和 1 个 RASP 核,以及 L3、DMA 等部件,采用 Synopsys PA 的 Virtual Processing Unit (VPU) 数据流<sup>[7]</sup>方式实现,架构和单簇结构图如图 6、图 7 所示.

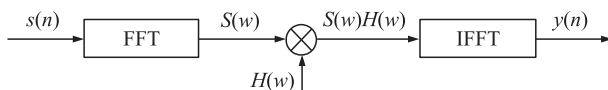


图 4 脉压流程图

Fig. 4 Taskgraph of DPC

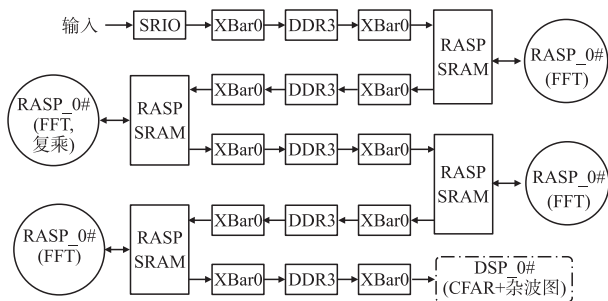


图 5 脉压数据传输、存储和处理流程图

Fig. 5 Taskgraph of DPC dataflow/store/disposal

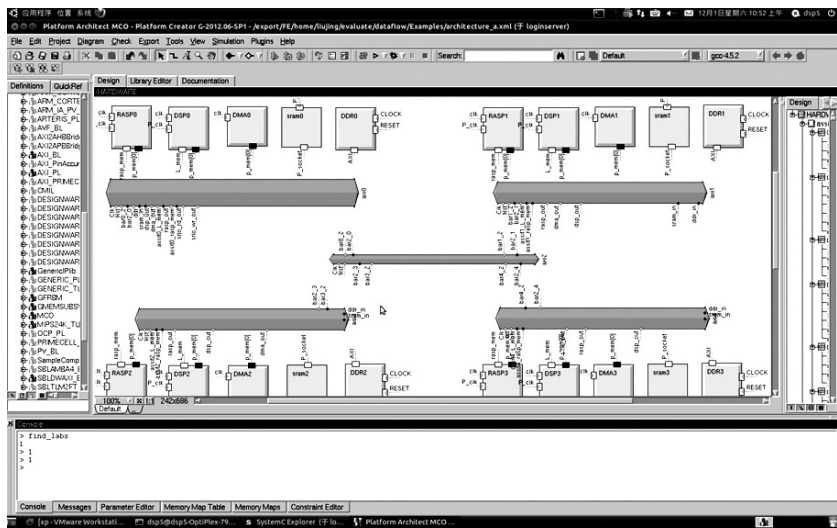


图 6 架构 1 截图

Fig. 6 Screenshot of SoC architecture 1

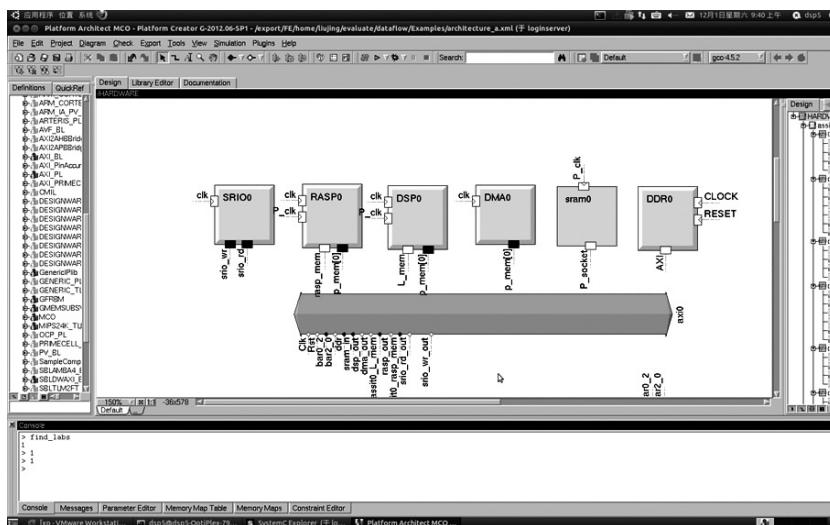


图 7 架构 1 单簇模型截图

Fig. 7 Screenshot of Cluster architecture 1

各主要模块性能参数如表 1 所示。

表 1 架构 1 模块性能参数

Table 1 Performance parameters of modules in architecture 1

名称	DSP	RASP	DMA	AXI	SRAM	DDR
频率	1G	1G	1G	500M	1G	1.06G
接口位宽	256	256	256	256	256	64
ESL 库模型	VPU	VPU	VPU	AXI3	MEMORY	MEMCTRL

## (2) 脉压分析

算法流程图按照 PA 工具数据流输入要求进行加工,写出 CSV 文件,并将 CSV 文件导入 PA 中,转换得到 TaskGraph,如图 8 所示。

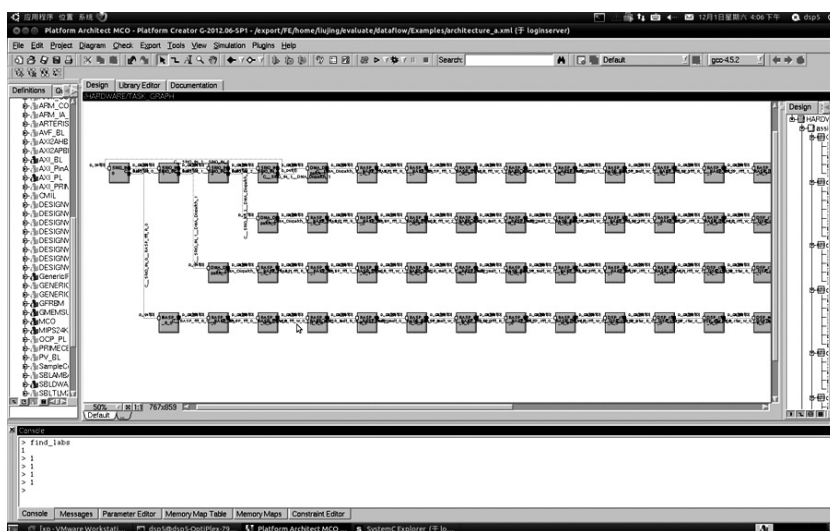


图 8 脉压 TaskGraph 图

Fig. 8 Taskgraph of DPC

根据脉压算法流程图,将 4 通道的任务映射到 4 核处理分为 3 步:数据分配、处理及汇总. 具体过程为:

(a)外部数据通过 srio0 导入 core0 的 ddr;

(b)通道 0 数据进入 core0 的 ddr 后,启动 core0 的 rasp、dsp 开始处理数据;同时,通道 0 数据读入后,立即开始通道 1 的数据输入;



(c)通道 1 数据包读入后,启动 core0 的 dma 将数据从 core0 的 ddr 通过两级 AXI 总线搬至 core1 的 ddr,然后启动 core1 的 rasp、dsp 开始处理;

(d)通道 2、3 数据同上步骤进行搬运,分别由 core2、3 进行处理;

(e)通道 0 的处理结果在 core0 的 ddr 中,直接由 srio 将结果送出;core1、core2 和 core3 的 ddr 中通道处理结果需要用本 core 中的 dma 将数据搬运到 core0 的 ddr 中,再由 srio 送出;

(f)在整个处理过程中,各 core 间对总线、srio 以及 core 内 dma、ddr 等资源使用采用抢占原则,模拟竞争。

任务运行后,采用 Synopsys PA 的 VPEXplorer Debug and Analysis<sup>[8]</sup> 分析处理结果,如图 9 所示。由仿真结果看出,4 核总处理时间为 6.94 ms。

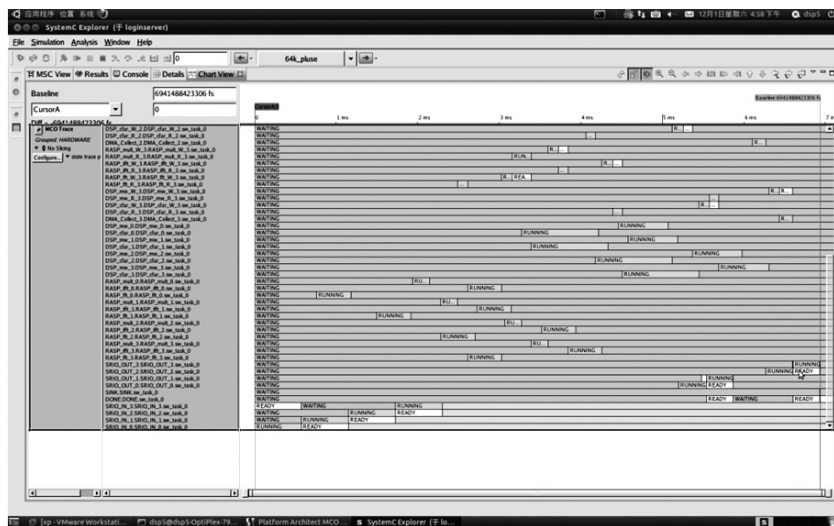


图 9 脉压 TaskGraph 分析图

Fig. 9 Analysis graph of DPC

在此算法中,core0 的 dma 负责数据分发。同时,dma 还负责本 core 数据在 dsp 与 ddr 之间的搬运,导致 core0 的 dma 利用率高。core0 的 dma 在进行数据分发、替 dsp 读写数据时,srio 也往 ddr 中发送数据,造成 core0 的 ddr 数据吞吐繁忙,dma 处理时间增长,延迟了其他 core 的数据读取及处理。core0 的 dma 与 core1 的 dma 对比如图 10 所示。core0 的 dma 冲突率最高达到 103%;core1 的 dma 冲突最高 35%。

簇内总线 axi 及簇间 axi 总线利用率如图 11 所示。core0 的 axi 最高利用率为 25.1%,其他 core0 最高利用率为 15%左右,核间 axi 总线最高利用率为 6.5%,均在正常范围内。

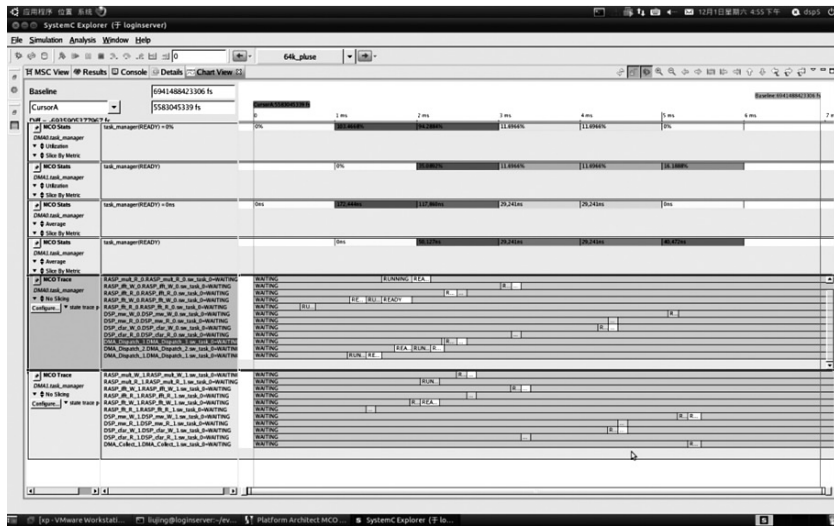


图 10 脉压 DMA 利用对比图

Fig. 10 Utilization ratio contrast of DPC's DMA

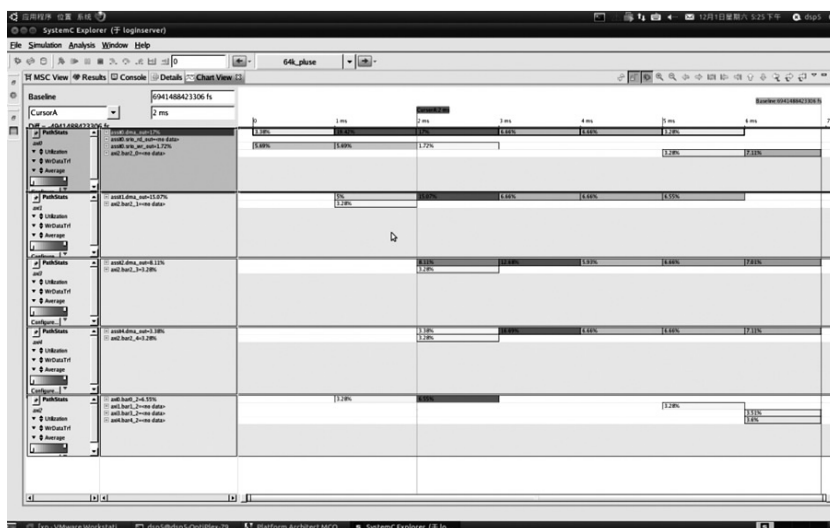


图 11 AXI 总线利用率

**Fig. 11** Utilization ratio of AXI bus

### 2.2.3 架构方案 2

### (1) 架构创建

方案 2 的系统架构为 4 个异构簇结构,每个异构簇仅包含 1 个 DSP 核和 1 个 RASP 核. 4 簇共享 L3、DMA 等部件. 具体架构图如图 12 所示.

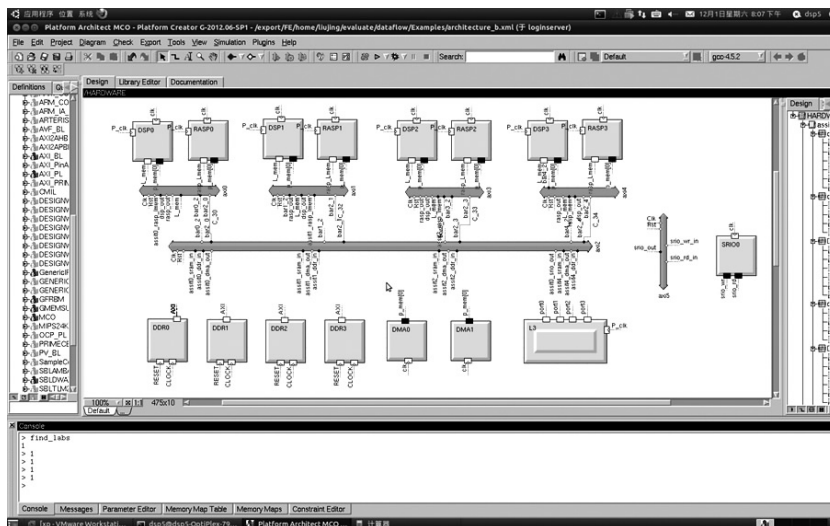


图 12 架构 2 截图

**Fig. 12 Screenshot of SoC architecture 2**

各主要模块性能参数如表 2 所示.

表 2 架构 2 模块性能参数

Table 2 Performance parameters of modules in architecture 2

名称	DSP	RASP	DMA	AXI	SRAM	DDR
频率	1G	1G	1G	500M	1G	1.06G
接口位宽	256	256	256	256	256	64
ESL 库模型	VPU	VPU	VPU	AXI3	MEMORY	MEMCTRL

## (2) 脉压分析

根据脉压算法流程图,将4通道的任务映射到4核处理.不同于架构1,架构2处理时不包含数据分发和汇总,具体过程为:

(a)数据通过 srio0 导入;

(b) srio 将通道 0 数据写入 ddr\_0,启动 core0 的 rasp、dsp 开始处理数据,处理结果由 srio 从 ddr\_0 中送出;

(c) srio 写完通道 0 数据后,开始通道 1 的数据往 ddr\_1 写入,并启动 core1 的 rasp、dsp 开始处理,处理结果由 srio 从 ddr\_1 中送出;

(d) 通道 2、3 数据同步骤 3,分别由 core2、3 进行处理;

(e) 在整个处理过程中,dma0 负责 core0/2 数据搬运;dma1 负责 core1/3 数据搬运. 各 core 间对总线、srio 以及 dma、ddr 等资源使用采用抢占原则,模拟竞争.

任务处理结果如图 13 所示. 由仿真结果看出,4 核总处理时间为 7.64 ms. 由于任务划分合理,各通道处理时间均匀. 2 个 dma 负责 4 个簇数据搬运,由于搬运数据次数较多,出现了的冲突现象,dma0 冲突率最高达 86%,dma1 冲突最高达 111%.

簇内总线 axi 及簇间 axi 总线利用率如图 15 所示. Core0、Core1 等核内 axi 最高利用率为 10%以下,核间 axi 总线最高利用率为 42%.

#### 2.2.4 两架构方案分析对比

由以上分析可知,架构 1 由于总线上数据传输效率高,处理性能优于架构 2,但架构 2 对应的算法划分均匀,没有数据分发汇总过程,资源冲突少,见表 3.

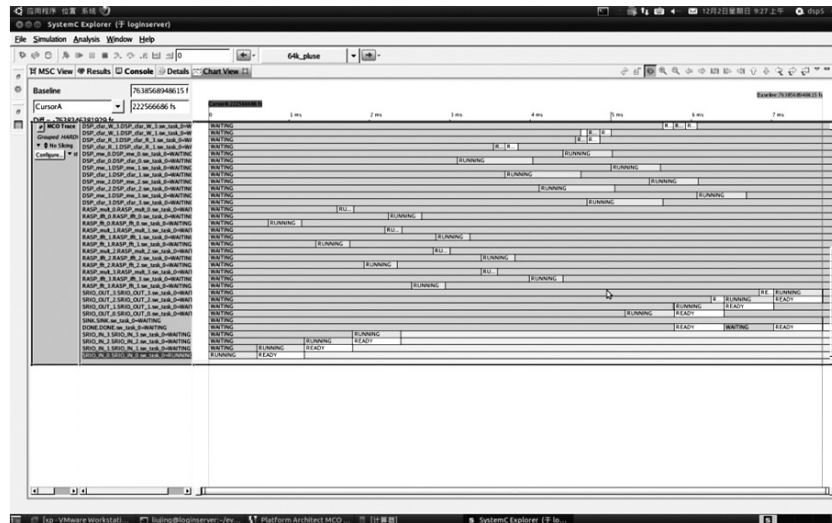


图 13 脉压 TaskGraph 分析图

Fig. 13 Analysis graph of DPC

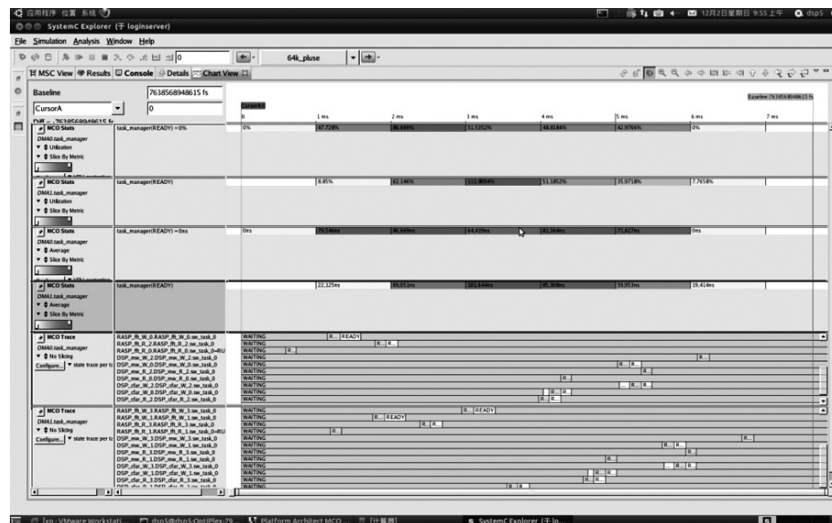


图 14 脉压 DMA 利用对比图

Fig. 14 Utilization ratio contrast of DPC's DMA



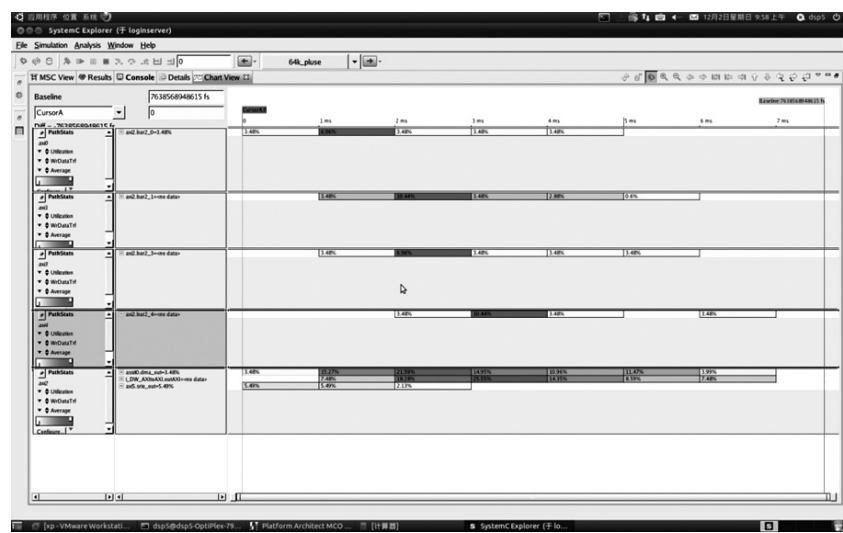


图 15 AXI 总线利用率

Fig. 15 Utilization ratio of AXI bus

表 3 架构分析对比

Table 3 Contrast result of two architectures

内容	处理时间	簇内 AXI		簇间 AXI 利用率	簇 0 的 DMA 冲突率	簇 1 的 DMA 冲突率
		簇 0 的 AXI 利用率	簇 1 的 AXI 利用率			
架构 1	6.94 ms	25.10%	15%	6.50%	103%	35%
架构 2	7.64 ms	10%	10%	42%	86%	111%

3 结语

本文以华睿 2 号 SoC 架构设计为例,借助 Synopsys 的 Platform Architect 软件描述了 ESL 设计方法学所论述的建模、验证分析、架构探索和虚拟原型. 通过建立系统级模型,利用仿真分析得到总线、模块吞吐率的信息,尝试了不同架构对信号处理算法运行性能的影响,并借助 ESL 系统级建模方法,完成不同架构的性能分析. 采用 ESL 设计方法学的复杂 SoC 设计能够极大缩短由于直接进行 RTL 级别的设计调试所带来的开发周期延长问题,满足设计复杂度持续增长和上市时间的要求.

[参考文献] (References)

[1] BRIAN B,GRANT M,ANDREW P.ESL design and verification[M]. Burlington,USA:Morgan Kaufmann Publishers,2007:1-9.

[2] SU A P,CHEN R. Applying ESL in a Dual-Core SoC platform designing[C]//IEEE International SOC Conference, USA, 2006:171-174.

[3] 郭炜,郭箬,谢懋,等. SOC 设计方法与实现[M]. 北京:电子工业出版社,2007:42-48.

GUO W,GUO Z,XIE J,et al. SOC design method and implementation[M]. Beijing:Publishing House of Electronics Industry, 2007:42-48.(in Chinese)

[4] Synopsys Inc. Welcome to Platform Architect MCO[EB/OL]. [2012-09-01]http://www.synopsys.com.

[5] 苏涛,强生斌,吴顺君. 数字正交采样和脉压的高效算法及实现[J]. 现代雷达,2001,23(1):39-41,48.

SU T,QIANG S B,WU S J,et al. An efficient algorithm realizing the quadrature sampling and pulse compression[J]. Modern radar,2001,23(1):39-41,48.(in Chinese)

[6] 贺知明,黄巍,向敬成. 数字脉压时域与频域处理方法的对比研究[J]. 电子科技大学学报,2002,31(2):120-124.

HE Z M,HUANG W,XIANG J C,et al. Contradistinction research between the digital pulse compression time-domain and frequency-domain processing method[J]. Journal of university of electronic science and technology of China,2002,31(2):120-124.(in Chinese)

[7] Synopsys Inc. Task modeling and virtual processing unit user's guide[EB/OL]. [2012-09-01]http://www.synopsys.com.

[8] Synopsys Inc. VPEXplorer debug and analysis user's guide[EB/OL]. [2012-09-01]www.synopsys.com.

[责任编辑:严海琳]