分布式存储管理在多核设计中的高层建模

李　磊[①]　沈海斌[①*]　黄　凯[①]　严晓浪[①]　Han Sangil[②]　Ahmed A Jerraya[③]
[①](浙江大学超大规模集成电路设计研究所　杭州　310027)
[②](韩国汉城大学, Sillim-dong, Gwanak-gu, Seoul 151-742　Korea)
[③](CEA-LETI, MINATEC, 17 Rue des Martyrs 38054, GRENOBLE, France)

摘　要：该文提出了将分布式存储管理系统融入基于 Simulink 的多核设计流程的方法，改变了多核设计中数据通信部分采用全局存储器(GFIFO)来完成的做法，提高了系统间通信的效率。针对 Motion JPEG 解码器的实验结果表明，分布式存储管理系统使各个子系统之间的通信效率相对于 GFIFO 提高了 50%，解码时间降低了 30%。同时，分布式存储管理系统作为库单元加入到多核设计流程中，为高效、快速地完成高性能多核设计提供了便利。

关键词：分布式存储管理；多核设计；Motion-JPEG 解码器
中图分类号：TP336　　**文献标识码**：A　　**文章编号**：1009-5896(2008)11-2750-05

Distributed Memory Service Modeling in Multi-Processor Design

Li Lei[①]　Shen Hai-bin[①]　Huang Kai[①]　Yan Xiao-lang[①]　Han Sangil[②]　Ahmed A Jerraya[③]
[①](*Institute of VLSI Design*, *Zhejiang University*, *Hangzhou* 310027, *China*)
[②](*Seoul National University*, *Sillim-dong*, *Gwanak-gu*, *Seoul* 151-742, *Korea*)
[③](*CEA-LETI*, *MINATEC*, 17 *Rue des Martyrs* 38054, *GRENOBLE*, *France*)

Abstract: Distributed Memory Service is integrated in Simulink-based multiprocessor design flow to improve communication efficiency comparing to Global FIFO(GFIFO) method. In Motion-JPEG case study, DMS made the percentage cost on communication of total execution time decreased by 50%, and the execution time decreased by 30% for decoding 10 frames， comparing with GFIFO. This integration makes more opportunities in design exploration to improve the performance of Simulink-based multiprocessor design flow.

Key words: Distributed memory service; Multi-processor design; Motion-JPEG decoder

1　引言

目前多核设计方法主要向 3 个方面发展：系统级设计、基于平台设计和基于部件设计[1]。基于部件的设计是从底层到顶层的设计方法，因其能够从已经建立的处理器库单元和协议库单元中选取不同的处理器以及通信通道单元来搭建整个系统，降低了整个设计周期的时间，降低了人工设计出错的可能性。ROSES[2]多核设计流程正是利用了处理器库单元，通信通道单元以及线程生成工具，在此基础上发展出基于 Simulink 的多核设计方法[3]。在多核系统中，一个重要的参数是系统计算量与通信量的比例 r，式(1)中 Q_{comp} 和 Q_{comm} 分别表示计算量和通信量。

$$r = Q_{\text{comp}} / Q_{\text{comm}} \tag{1}$$

数据通信直接影响着整个系统的性能，r 越高，多核系统的性能越好。而对于固定的比例 r，高性能的存储器结构能够降低处理器的通信开销，从而提高整个系统的性能[4]。目前主要的存储器结构有两种，一是对称的共享存储器结构，主要是通过总线的方式来共享同一块存储器；另一种是分布式存储器结构，存储器分布在各个处理器节点上，所有的节点通过数据通信网络和控制网络连接。对于不同节点之间的通信，需要发送传输请求到远程节点。这样的消息传递机制为指令并行性提供了可能，更适合于异构核的多核设计[5]。

在原有的基于 Simulink 多核设计流程[3]中，通信库单元(Global FIFO，GFIFO)利用系统全局存储器，使用基本输入输出的方式进行数据传输，在数据通信量较小的应用中能够满足数据传输的要求。但是对于对数据通信依赖大的应用，且复杂的仲裁机制的场合，GFIFO 实现的通信通道成为系统整体性能的瓶颈。本文对分布式存储管理系统(Distributed Memory Service， DMS)[6]进行了系统建模，并将 DMS 应用到基于 Simulink 多核设计流程中，改变了原有设计中各个子系统之间的数据交换都通过GFIFO实现的做法，而是采用推荐的通信库单元 DMS 进行数据通信。通过对 Motion JPEG 解码器的应用分析，将 DMS 与 GFIFO 实现方法的性能作比较，证明了将 DMS 引入到高层建模的系统级多核设计流程的技术，降低了 JPEG 解码时间以及数据通信消

2007-04-20 收到，2007-07-18 改回
*通讯联系人：沈海斌

耗的时间，为系统级任务划分，提供了更多的选择空间。

2　DMS 通信原理

由于在多核系统中，客户端子系统可以分为主动系统和被动系统两类[6]。DMS 针对不同的子系统类型，采取了不同的数据同步机制。图 1 表示 DMS 在整个多核设计架构中的位置。

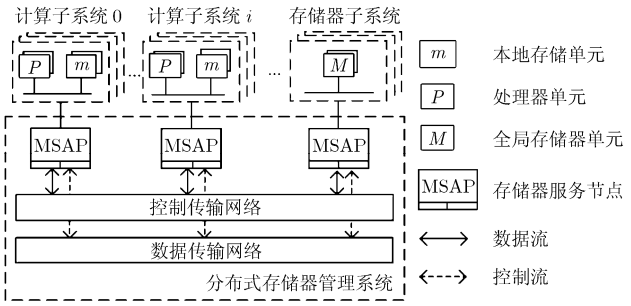


图 1　MPSoC 中的 DMS 结构

图 1 中多核系统由多个处理计算的子系统，分布式存储器服务系统 DMS，存储器子系统构成。计算子系统可以由一个或多个处理器单元(例如处理器，DSP 处理器，以及能够发出读写请求的 IP 模块)，本地存储器以及本地总线构成。DMS 在整个多核系统中充当了数据传输的服务器的作用。DMS 由存储器服务节点 (Memory Service Access Point, MSAP) ，控制信号网络和数据传输网络构成。MSAP 在接收到本地子系统发出的数据传输指令以后，通过控制信号网络和数据传输网络完成数据的传输和同步。当数据的接收者是被动系统，也就是图 1 所示存储器系统时，DMS 提供了条件传输以及传输建立的机制来保证数据的一致性。分布式存储器服务提供了两种传输机制：传输接收模式和远程 DMA 模式。传输接收模式同传统的消息传输机制类似[7] ，仅仅应用在两个主动系统之间。主动系统与被动系统之间的数据通信，使用远程 DMA 模式。另外，DMS 提供了条件数据传输的机制来解决数据一致性的问题[6]。

3　高层建模方法

3.1　基于 Simulink 的多核设计方法

基于 Simulink 模型的多核设计方法，延续了传统的系统设计方法[8]。其基本实现方式为：(1)首先由目标应用行为描述(C/C++描述)映射到 Simulink 模型中。(2)将 Simulink 应用模型转换为 Simulink 应用和体系结构的结合模型 Combined Algorithm and Architecture Model (CAAM)，将体系结构的模型同应用描述的模型结合起来。(3)由解析器 parser 将 CAAM 模型转换为基于 XML 描述的 Colif CAAM 模型[9]。(4)从处理器库和通信协议库中生成多核的硬件体系结构，包括处理器子系统，存储器子系统，以及各个子系统之间的通信网络。(5)生成多线程软件代码，以及生成用来发出线程、初始化通信通道[10]的顶层代码。

3.2　软硬件混合体系结构模型

软硬件混合体系结构[3]允许对于多线程的异构多核系统进行多种抽象层次的描述包括 (1)CAAM， (2)Virtual Architecture，(3)Transaction Accurate Model，(4)Virtual Prototype[11]4 种抽象级别。其中 CAAM 模型将 Simulink 模型同硬件体系结构相结合，VA 模型提供了相应的通信库单元及软件多线程，TA 模型包含了处理器的抽象描述，硬件处理系统通过 TLM 模型描述[12]，VP 模型采用了 ISS 仿真，软件部分包含了应用的多线程。

基于 Simulink 多核流程中主要采用 ARM7 和 Xtensa 处理器作为处理器的库单元。在通信库单元中，提供了全局存储单元的方式 Global FIFO(GFIFO)，软件 FIFO(SFIFO)。在结构单元中，基于 Simulink 多核设计流程提供了处理器的 ISS 仿真器，AMBA 总线、本地总线等硬件部分的 TLM 模型。另外，该多核设计流程提供了从 Colif CAAM 模型到 VA，TA，VP 模型的多线程生成工具。

4　DMS 的高层模型

基于 Simulink 的多核设计流程的优越性主要体现在实现自动化和基于部件的实现方法上。基于部件的设计方法，能够让设计者在设计过程中做出最初设计空间的探索，具体过程由基于 Simulink 多核设计流程来实现，这样大大提高了设计实现的效率，减小了设计反复的时间[10,13]。在现有的设计流程中，各个子系统之间数据通信采用的是 GFIFO 库单元，采用了基本的输入输出方式进行数据通信，能够实现简单的数据通信。

本文将 DMS 添加到基于 Simulink 的多核设计流程中，DMS 内部拥有相应的传输请求调度机制，同时每个传输端口都有相应的缓存空间，这样的结构提高了整个系统数据通信的效率，减少了应用程序的执行时间。从设计可重用性上面来看，在对通信性能高要求的场合中，采用 DMS 通信部件，可满足设计要求。本文在具体的 Motion-JPEG 解码器的应用中，将使用 DMS 作为通信部件的情况同原有流程中使用 GFIFO 作为通信部件的情况进行比较。

4.1 Simulink CAAM 模型

本文根据目前常用的 JPEG 解码方法，设计了基于 JPEG 顺序解码方法的 Motion-JPEG 解码器，用于比较 DMS 的高层建模多核设计与现有多核设计流程中使用的全局存储器的性能。实验从 Motion JPEG 解码器的源代码(C 语言描述)出发，经过修改转换为适合 Simulink 功能模型的代码，在 Simulink 里面建立相应的应用功能模型。建立的 Motion-JPEG 解码器模型包含了 7 个 S-Function(Simulink)，7 个延迟模块，26 个链接，4 个 If-Action 子系统(Simulink)。然后将软件硬件划分到不同的子系统中，形成 Simulink CAAM 模型，图 2 显示了模型框图。

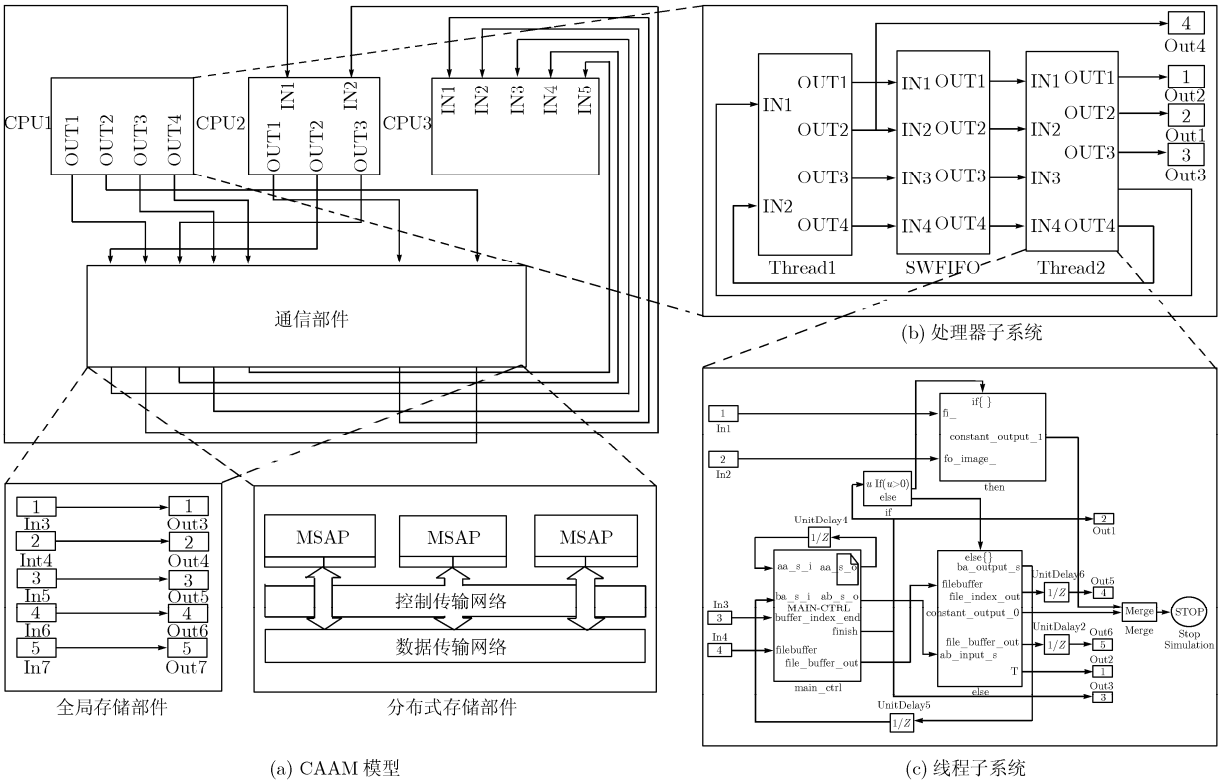


图 2 Motion-JPEG 解码器的 Simulink CAAM 模型

JPEG CAAM 模型中 CPU1 子系统承担控制和进行可变长编码 VLD 的功能，考虑到计算量离散余弦反变换 IDCT 拆开为 IDCTa 和 IDCTb 两部分并分别划分到 CPU2 和 CPU3 子系统中。CPU1 子系统中包含了两个线程，分别用于控制和 VLD 功能，两个线程之间通过 SWFIFO 进行数据通信，CPU 子系统间通过 DMS 通信。本文采用的 DMS 使用了 7 位配置 portID，最多可以支持 128 个 port，包括输入和输出端口。DMS 地址空间采用 32 位寻址，其中高 10 位作为处理器子系统寻址使用，低 14 位分别用于数据传输请求分类，内部端口号，以及端口配置寄存器寻址，其中最低 2 位做为字传输的保留位。

4.2 实验分析

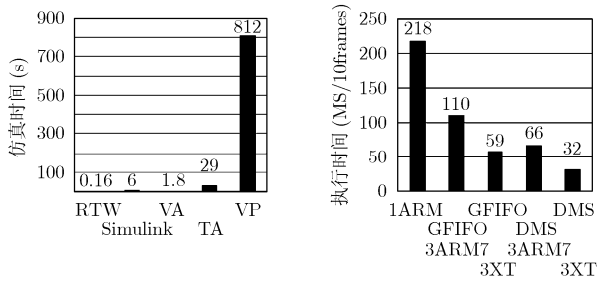
本文使用基于 Simulink 的多核设计流程中的 LESCEA[14]工具，由原始的 Simulink CAAM 模型转换成 Colif CAAM 模型，进而生成了 3 个不同抽象级别的仿真模型：VA，TA，VP 模型。对于图 2 中的通信通道模块部分，实验采用的策略是分别映射到 GFIFO 和 DMS 上，用 LESCEA 来生成基于不同通信通道的软件硬件代码。在基于部件的多核设计库单元中，包含了不同处理能力的处理器 ARM7 和 Xtensa 处理器，不同处理器构成的系统对于系统内部的计算能力以及对整个系统的通信能力的要求会有所变化，这里为了验证 DMS 对于不同处理器结构下同 GFIFO 的性能比较，在 JPEG 解码器应用中，本文分别对于 1 个 ARM7 处理器，3 个 ARM7 处理器，3 个 Xtensa 处理器 3 种不同的处理器体系结构来生成相应的相应的体系结构描述，其中 3 个 ARM7 处理器或者 3 个 Xtensa 处理器分别作为 3 个子系统中的计算单元。具体的处理器结构以及通信通道配置如下：JPEG_Single_ARM7，JPEG_GFIFO_3ARM7，JPEG_GFIFO_3XT，JPEG_DMS_3ARM7，JPEG_DMS_3XT。

5 种基于不同体系结构的测试情况中 Xtensa 模型协议的配置(XTMP)采取的配置是：一个 32 位乘法器，一个 16 位 MAC 控制器，没有指令/数据的缓存。这是没有指令扩展的 Xtensa 处理器的基本配置[15]，对 10 帧 QVGA(320×240) JPEG 的视频流进行解码。

(1)仿真时间 图 3(a)显示了使用 3 个 ARM7 处理器，通信通道配置使用 GFIFO 的情况下，不同抽象级别下的仿真时间。实时工作 RTW 情况下，仿真只需要 0.16s，Simulink 模型，VA 和 TA 模型分别花费了 6s，1.8s 和 29s 的仿真时间，而对于 VP 模型，由于其使用了 ISS 仿真和时钟精确的 TLM 模型，仿真时间为 812s，比 VA，TA 模型消耗了更多的时间。

(2)执行时间 不同体系结构下 JPEG 解码执行时间如图 3(b)所示，将上述 5 种处理器结构以及通信通道的配置，在 VP 模型中进行执行时间比较：单个 ARM7 处理器需要 218 百万时钟，而 3 个 ARM7 处理器将处理速度提高了 1 倍，

只需要 110 百万时钟。3 个 Xtensa 处理器构成的系统，需要 59 百万时钟周期来完成解码，速度提高了 1 倍，这是因为 Xtensa 处理器所配置的乘法器，适用于 JPEG 解码过程中的 IDCT 变换，提高了乘法的效率。当使用 DMS 作为通信通道部件时，3 个 ARM7 处理器的处理器配置下需要 66 百万时钟解码，同比使用 GFIFO 的情况执行时间缩短了 40%；3 个 Xtensa 处理器配置下需要 32 百万时钟解码，同比 GFIFO 的情况执行时间缩短了 45%。



(a) 不同抽象级别仿真时间比较　　(b) 不同体系结构下执行时间比较

图 3　Motion-JPEG 解码器仿真时间和执行时间

(3)具体性能分析　为了更为详细地对加入 DMS 对于整个系统性能的影响进行分析，将处理器的所有操作根据功能划分为 3 种类别：计算时间，通信时间和空闲时间[3]。在这样的时间划分下，首先从纵向的角度对不同处理器结构性能比较，结果如图 4 所示。使用 ARM7 处理器的子系统，花费了大量的时间在数据计算上面，而使用 Xtensa 的子系统，计算时间所占的比例有所减小。这是由于 ARM7 需要进行大量的数据运算，而 Xtensa 因为提供了 32 位乘法器，提高了处理器的处理乘法的性能，因此对于 Xtensa 组成的子系统，在计算时间上占用的比例有所下降，更多的时间花费在了等待数据同步，即空闲时间上。在通信时间方面，在使用同种通信通道的不同处理器组成的情况下，通信时间占用的比例变化不大。

从横向比较来看，同样的处理器子系统配置，对于不同的通信通道的配置，DMS 为通信通道构成的系统从整个执行时间来看，只花费了将近 60%的时钟数。而从通信时间占总的执行时间的比例上面来看，使用 DMS 的系统花费在通信上面的时间比例都下降了 10%左右。这是由于 DMS 采用了自主的数据传输的模式，处理器在发出数据传输的指令以后，在不影响数据同步性的前提下，可以将下面的数据传输操作交给 DMS 处理，而处理器子系统可以进行其他的运算操作。这样的操作类似于系统内部的 DMA 操作，将数据搬运的工作由处理器子系统一端转交给 DMS 来完成。当然对于不同的应用，通信时间上面的优化效果不一定明显，特别是对于计算/通信比例较大的应用中，系统本身花费在通信上面的时间就比较少，不用花费更多的面积和功耗来增加通信通道的复杂程度，进而提高通信效率。

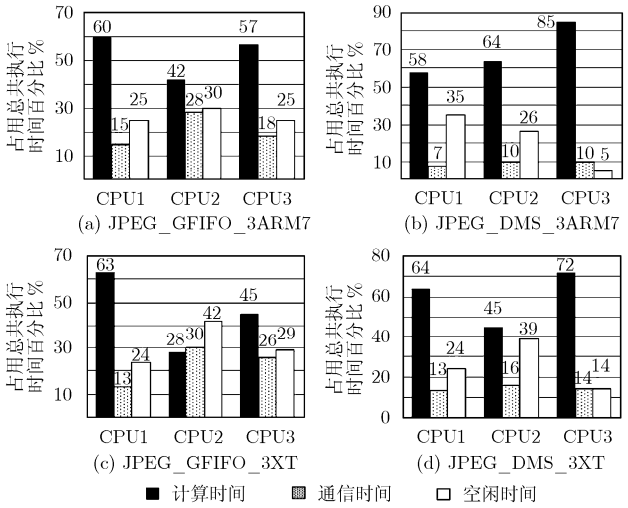


图 4　Motion-JPEG 解码器执行时间

5　结束语

如何针对不同的应用，采取不同的任务划分，对于整个系统的性能来说都是很重要的，所以设计初期的设计空间探索显得更为重要。本文将分布式存储服务系统 DMS 加入到多核设计流程中，为现有的设计流程添加了更为高效的通信库单元。在对数据通信性能要求更高的应用中，设计者可以选择 DMS 替代 GFIFO，满足设计要求。

在 Motion JPEG 解码应用中，使用基于 Simulink 的多核设计流程来生成不同抽象级别的多核体系，并根据不同的处理器及通信通道的映射策略来生成相应的系统。实验结果表明，将 DMS 加入到现有的设计流程中，在 JPEG 解码器的应用中，降低了整个系统解码消耗的时间，同时也降低了系统在通信时间上的消耗。这一方法证明了本文的 DMS 通信库单元，相比较 GFIFO 而言，在数据通信上更具有竞争力，为不同系统级别设计策略的系统建模提供了更多的选择空间。

在未来的工作中，将会对分布式存储器系统进行进一步的优化，包括对于多个数据传输请求的调度机制进行进一步的优化，对于输入输出端口的划分，从而更进一步提高两个 MSAP 之间的数据传输效率。

参 考 文 献

[1] Cesário W and Jerraya A, *et al*.. Multiprocessor SoC platforms: A component-based design approach[J]. *IEEE, Design & Test of Computers*, 2002, 19(6): 52–63.

[2] TIMA laboratory, SLS group. A component based approach for hardware and software integration [EB/OL]. http://tima.imag.fr/sls/.

[3] Huang K and Li L, *et al*.. Simulink-based MPSoC design flow: case study of Motion-JPEG and H.264[C]. Proc. of 44th Design Automation Conf., San Diego, California, Jun. 4-8, 2007: 39–42.

[4] Hennessy J and Patterson D. Computer Architecture: A Quantitative Approach[M]. Morgan Kaufmann, 2006: 528–234.

[5] Benini L and De Micheli G. Networks on Chips: A New Paradigm for Component-Based MPSoC Design Multiprocessor Systems on Chips [M]. Morgan Kaufmann, 2004: 49–80.

[6] Han S, *et al.*. An efficient scalable and flexible data transfer architecture for multiprocessor SoC with massive distributed memory[C]. Proc. 41st Design Automation Conf., San Diego, 2004: 250–255.

[7] Dunning D, *et al.*. The virtual interface architecture[J], *IEEE, Micro,* 1998, 18(2): 66–76.

[8] Keutzer K, *et al.*. A system-level design: orthogonalization of concerns and platform-based design[J]. *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, 2000, 19(12): 1523–1543.

[9] Cesario W, *et al.*. Colif: A design representation for application specific multiprocessor SOCs[J]. *IEEE, Design & Test of Computers,* 2001, 18(5): 8–20.

[10] Jerraya A, *et al.*. Application-specific multiprocessor systems-on-chip [J]. *Microelectronics Journal, Elsevier Science*, 2002, 33(11): 891–898.

[11] Jerraya A and Wolf W. Hardware/software interface codesign for embedded systems [J]. *IEEE, Computer*, 2005, 38(2): 63–69.

[12] Synopsys, Inc., SystemC[EB/OL], Version 2.0. http://www.systemc. org.

[13] Cesario W, *et al.*. Component-based design approach for multicore SoCs [C]. Proc. IEEE/ACM 39th Design Automation Conf, New Orleans, 2002: 789–794.

[14] Han S, *et al.*. Buffer memory optimization for video codec application modeled in Simulink[C]. Proc. of 43rd Design Automation Conf., New York, 2006: 689–694.

[15] Tensilica instruction extensions languages user guide[CP/OL]. http://www.tensilica.com/.

李 磊： 男，1980 年生，博士生，从事片上网络(NoC)的通信研究.

沈海斌： 男，1967 年生，副教授，从事信息安全、SoC 设计、可重构技术研究.

黄 凯： 男，1981 年生，博士生，从事处理器体系结构、图像算法研究.

严晓浪： 男，1947 年生，教授、博士生导师，浙江大学信息科学与工程学院院长，研究领域主要包括集成电路 CAD 与集成电路设计、信息安全技术等.

Han Sangil： 男，1976 年生，博士生，从事多核系统体系结构研究.

Ahmed A Jerraya：男，教授，博士生导师，研究领域包括多核系统体系结构、EDA 自动化工具支持等.