

CA-CIG:基于 CIG 模型的通信体系结构设计

曲俊华¹, 王燕华¹, 牛亚文²

(1. 华北电力大学计算机科学与技术系, 北京 102206; 2. IBM 中国软件开发中心, 北京 100085)

摘要:提出了一种基于 CIG 模型的片上通信体系结构设计方法 CA-CIG。通过分析 CIG 模型中功能组件之间的通信关系, 并考虑了不兼容协议之间协议转换带来的通信开销和不同的总线宽度对系统通信性能的影响, 提出了一种快速有效的片上通信体系结构原型设计方法, 集成了通信拓扑体系结构设计、总线宽度配置和总线协议选择等过程, 提供了相应的算法实现其中的关键步骤。通过 JPEG 解码系统证明了 CA-CIG 设计方法的有效性。

关键词:通信体系结构; CIG; 片上通信

CA-CIG: Communication Architecture Design Based on CIG

QU Jun-hua¹, WANG Yan-hua¹, NIU Ya-wen²

(1. Dept. of Computer Science and Technology, North China Electrical Power University, Beijing 102206;

2. IBM China Software Development Lab, Beijing 100085)

【Abstract】 This paper proposes an efficient on-chip communication architecture design method (called CA-CIG) based on CIG. Analysing the communication between processing elements given in CIG, and taking into consideration of the communication cost from the transformation between incompatible protocols and the different bus width, CA-CIG presents a fast and effective architecture prototype design method, and integrates the communication topology generation, bus width configuration and the communication protocol selection processes. To automate the design, CA-CIG also provides the related algorithms for the key steps. This paper uses the JPEG decoder application to demonstrate the effectiveness of the method.

【Key words】 communication architecture; CIG; on-chip communication

1 概述

随着集成电路技术和工艺的不断进步, 系统芯片(system-on-chip, SoC)已经成为人们广泛研究的热点。根据系统级设计方法和模块复用的原则, 基于 IP 的设计方法已经成为系统设计的主要范式。设计人员首先将验证过的系统描述映射到一些预先设计好的 IP 模块上, 然后根据模块之间的通信信息来构造实现模块之间数据传输的通信体系结构。随着互连线的延迟超过门延迟成为整个系统性能的瓶颈, 传统的以“计算”为中心的设计逐渐转向以“通信”为中心的设计。如何设计高性能的系统级通信体系结构已经成为系统设计的一个关键问题。

从点到点的通信方式到全局总线的方式, 如何在有效的时间内从众多可能的实现方案中选取最适合系统需求的高效的片上通信体系结构(on-chip communication architecture, OCA), 是 SoC 设计面临的关键问题。一个好的 OCA 设计方法和设计工具能够帮助芯片设计人员在有效的设计时间内探索和评估大量的设计解空间, 并找到最优的解决方案。

作为软/硬件协同设计的一个关键步骤, OCA 的设计已经得到了广泛的研究。Gasteier 等人提出了一种共享通信通道的结群方法, 使得同一集合内部的进程之间的数据传输能够共享相同的通信通道。此外, 他们还实现了静态调度系统的通信拓扑结构自动生成方法。给定由多个通过抽象的 send/receive 函数进行通信的进程组成的行为级模型, 该方法能够自动生成一个高效得多总线(不考虑总线仲裁器)通信拓扑结构^[1]。Knudsen 等人提出了一种通信评估模型, 并以此

模型论述了集成通信协议选择过程和软/硬件划分过程的重要性。文献[2]提出了一种通信映射的设计流程, 并提供了相应的算法实现其中的关键步骤。Lahiri 等人提出了一种通信体系结构设计方法, 通过引入一个额外的电路层来监视系统不断变化的通信需求以提高系统性能。

2 CA-CIG 设计方法

2.1 系统设计框架

它以嵌入式系统功能描述为输入的OCA总体设计框架见图1, 说明CA-CIG在整个OCA设计过程中所起的作用。为了便于系统在不同的抽象级别进行建模, 引入两种中间表示模型: 层次化数据控制流图^[3](hierarchical control data flow graph, HCDFG)和CIG^[4]。

CIG模型是对系统描述的高级抽象, 能够表示PE之间的整体通信关系。图2给出了一个简单的CIG模型示例。其中, 结点表示PE, 边表示PE之间的通信关系。例如, PE1 与PE2 之间的边 80(16)表示PE1 与PE2 之间共有80次数据传输, 最大的数据传输宽度是16。基于HCDFG模型, 可以方便地实现CIG模型的提取^[4]。CA-CIG设计方法在CIG模型的基础上实现了PE之间的通信拓扑体系结构生成, 总线协议选择和总线宽度配置等过程, 从而帮助芯片设计人员在系统设计的早期设计高效的OCA原型。

作者简介:曲俊华(1952-), 女, 教授, 主研方向: 计算机网络及应用; 王燕华, 硕士; 牛亚文, 硕士、工程师

收稿日期:2006-10-15 **E-mail:** qju_beijing@126.com

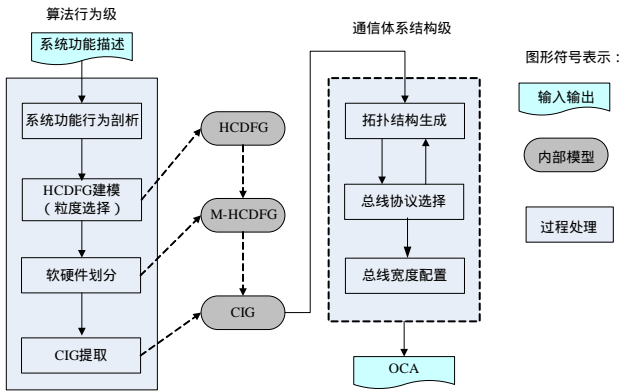


图1 系统设计流程

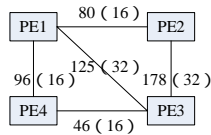


图2 CIG示例

(1)拓扑结构生成：是通信体系结构级设计的一个关键步骤。该步骤决定了PE之间的拓扑连接方式，并没有详细地确定总线的参数和接口电路。

(2)总线协议选择：决定了每条总线的通信协议。当一条总线连接了多个不兼容协议的PE/总线时，需要为该总线选择合理的通信协议使得系统通信开销最小。

(3)总线宽度配置：在生成的系统通信拓扑结构的基础上进一步确定了每条总线的数据传输宽度。

2.2 拓扑结构生成

图3给出了几种典型的OCA拓扑结构。其中，图3(a)表示点对点的连接方式。这种拓扑结构可以实现任意两点之间的高效通信，然而总线结构复杂，很难适应多PE组成的复杂系统；图3(b)表示全局总线拓扑结构，其总线结构简单，易于实现，然而当PE之间频繁通信时会造成严重的总线冲突，导致通信性能下降。图3(c)给出了前两种方案的折中，当PE1与PE4，PE2与PE3之间的通信比较频繁时可以达到性能与代价的合理折中。

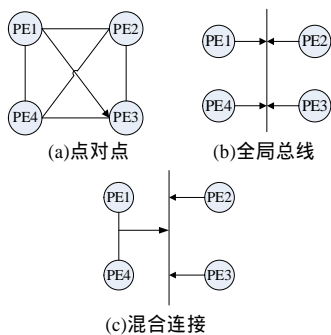


图3 OCA中几种典型的总线拓扑结构

CA-CIG设计方法通过权衡PE之间的通信量与单次通信的时间开销来构造系统的OCA拓扑结构。为了减少设计解空间，本文采用通信拓扑结构树表示OCA的拓扑体系结构。其中，叶子结点（原子结点）表示PE，非叶子结点（复合结点）表示总线。

假设每个PE都有自己的通信协议Protocol(PE)，不兼容通信协议的PE之间通信需要通过协议转换器进行协议转换，单次的通信时间开销较大；相反，相同通信协议的PE之间可以直接通信，单次通信开销较小。

为了权衡PE之间的通信量和单次通信的时间开销，CA-CIG通过式(1)来计算两个PE之间的引力，引力越大的PE之间具有越高的通信优先级。式(1)中， $Delay(PE_i, PE_j)$ 表示PE_i与PE_j之间进行一次数据传输所需单位时间。假设兼容协议之间PE_i与PE_j的 $Delay(PE_i, PE_j) = 1$ ，不兼容的PE_i与PE_j的 $Delay(PE_i, PE_j) > 1$ ，取决于两种协议之间协议转换器的接口电路。 $Delay(PE_i, PE_j)$ 越大表明PE_i与PE_j之间单位数据传输时间越长； $Times(PE_i, PE_j)$ 表示PE_i与PE_j之间数据的传输次数。

$$f(PE_i, PE_j) = \frac{Times(PE_i, PE_j)}{Delay(PE_i, PE_j)} \quad (1)$$

基于式(1)的PE之间引力计算方法，CA-CIG设计方法采用图4所示的结群算法构造系统的通信拓扑结构。其中，Cluster表示当前所有PE的集合。当Cluster中的元素多于一个时(第1行)，选择引力最大的两个节点结群(3、4行)。结群之后，从集合Cluster中删除结群的两个结点(5、6行)，增加一个新的结群结点(第7行)，并根据结群结果修改CIG模型(第8行)。

```

1: While (Account(Cluster)>1)
2:   {
3:      $\forall E_i, E_j \in Cluster, arc(E_i, E_j) : f(E_i, E_j)$ ;
4:      $Max(f(E_i, E_j)) : Merge(E_i, E_j)$ ;
5:     Remove(Cluster,  $E_i$ );
6:     Remove(Cluster,  $E_j$ );
7:     Add(Cluster,  $E_{ij}$ );
8:     ChangeCIG();
9:   }

```

图4 拓扑结构生成算法

由于结群的结点可能是原子结点或复合结点，不同的结点类型结群可能生成不同的拓扑结构。本文根据结群的类型将结群方案进行如下归类：

(1)两个原子结点结群

当两个结群结点均为原子结点时，可以直接将其合并成一个复合结点。合并后的复合结点作为这两个原子结点的父结点。如图5(a)。

(2)一个原子结点，一个复合结点结群

当两个结群结点中有且仅有一个结点为原子结点时，结群方案有以下两种：

1)将复合结点的根结点作为合并后的根结点，将原子结点作为该根结点的一个子结点，如图5(b)。

2)与类型1)类似，生成一个新的结点作为这两个结群结点的父结点，如图5(c)。

(3)两个复合结点结群

当两个结群结点均为复合结点时，结群方案有以下几种：

1)合并两个结群结点的根结点作为结群之后的根结点，如图5(d)。

2)将其中一个复合结点的根结点作为结群后的根结点，另一个复合结点作为该根结点的子结点，如图5(e)。

由于两个结点可以有以上多种结群方案，因此需要一个结群策略选择合适的结群类型。本文根据结群结点的通信协议决定结群类型：如果两个结群结点的通信协议兼容，选择图5(a)、图5(b)或者图5(d)；否则，选择图5(c)或者图5(e)。图6给出了5个PE之间的结群过程示例，图7表示该结群过程中相应的拓扑结构生成过程。按照这种策略进行结群，需要知道每个结群结点的通信协议，即通信协议的选择能够

影响拓扑结构的生成过程。通信协议选择算法会在下一节详细介绍。

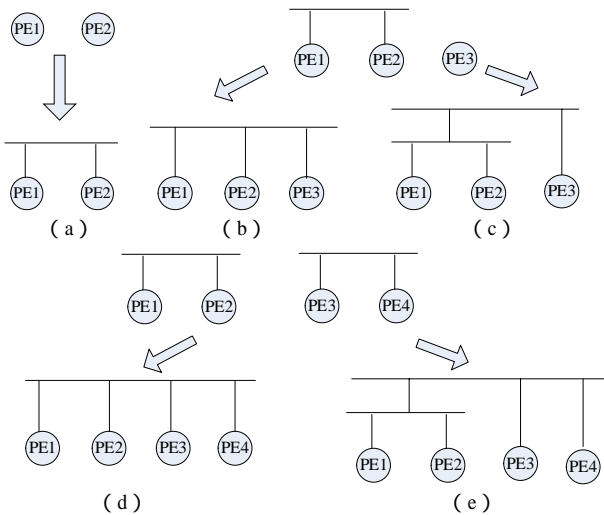


图5 结群类型

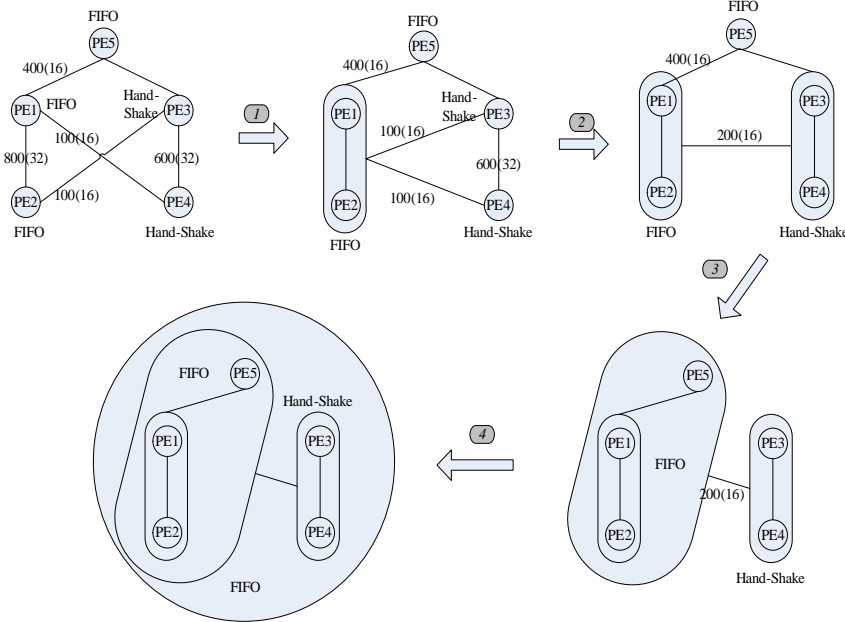


图6 结群过程

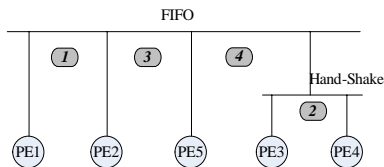


图7 拓扑结构

2.3 总线协议选择

总线协议选择即确定每一条总线的通信协议，使得整个系统达到较优的通信性能和较少的系统实现代价。由于不兼容的通信协议的 PE/bus 之间进行通信需要通过协议转换器，增加了系统的通信开销和实现代价。一种高性能的总线协议配置方案能够尽可能地减少协议转换的次数，提高系统的通信性能。

CA-CIG 设计方法中，总线协议的选择过程与拓扑结构生成过程是交替进行的。每当拓扑结构生成过程中合并两个结点时，需要配置合并后的根结点的通信协议。2.2 节介绍了

拓扑结构的生成与结群结点的类型有关，因此，需要为每一种可能的结群方法配置一种相应的通信协议配置方法。

(1) 两个原子结点结群

如图 5(a)所示的两个原子结点结群时，可以选择其中任意一个结点的通信协议作为新生成结点的通信协议。

(2) 一个原子结点，一个复合结点结群

1) 当一个原子结点和一个复合结点采用图 5(b)所示的方案生成拓扑结构时，将复合结点的通信协议作为合并后根结点的通信协议，这样可以保证当前的合并方案需要增加一个协议转换器，从而保证增加的协议转换器的数量最少。

2) 当一个原子结点和一个复合结点采用图 5(c)所示的方案生成拓扑结构时，可以选取任意一个结群结点的通信协议作为结群后根结点的通信协议，无论选取哪个都只需要增加一个协议转换器。

(3) 两个复合结点结群

1) 当两个复合结点按照图 5(d)所示的方案结群时，根结点采用的通信协议选择方法为：假设结群结点 1 和结群结点 2 分别有 m 和 n 个子结点。如果 $m > n$ ，则选择结点 1 的通信协议作为结群后根结点的通信协议，否则选取结点 2 的通信协议。

2) 当两个复合结点按照图 5(e)所示的方案结群时，选取作为结群后根结点的原结群子结点的通信协议作为根结点的通信协议，从而保证结群后只增加一个协议转换器。

按照以上的协议选择策略，可以保证每次结群插入的协议转换器数目最少，从而优化系统的实现代价和性能开销。具体的总线协议选择算法如图 8。

```

1: Case (MergeType)
2: {
3:   T1, T3: P(root)=P(Ei); //(1),(2,2)
4:   T2: If (Child(Ei)!=null) //(2.1)
5:     P(root)=P(Ei);
6:   else
7:     P(root)=P(Ej);
8:   T4: If (ChildNum(Ei)>ChildNum(Ej))
9:     //(3.1)
10:    P(root)=P(Ei);
11:   else
12:    P(root)=P(Ej);
13:   T5: If (Father(Ei)!=NULL) //(3.2)
14:     P(root)=P(Ej);
15:   else
16:     P(root)=P(Ei);
17: }

```

图8 总线协议选择算法

2.4 总线宽度配置

总线宽度配置即确定系统中每一条总线的数据宽度。对于相同的通信拓扑体系结构，不同的总线参数配置方案会导致不同的通信开销和系统实现代价。例如，如果采用如图 9 所示拓扑结构实现图 2 中 CIG 模型中 PE 之间的通信。图 9 中经过 bus₁ 的数据通信有： $\langle PE_1, PE_2 \rangle$ ， $\langle PE_1, PE_3 \rangle$ ， $\langle PE_2, PE_3 \rangle$ ， $\langle PE_2, PE_4 \rangle$ ， $\langle PE_3, PE_4 \rangle$ ，其数据传输位数分别是 $width(PE_1, PE_2) = width(PE_1, PE_4) = width(PE_3, PE_4) = 16$ ， $width(PE_1, PE_3) = width(PE_2, PE_3) = 32$ 。如果将 bus₁ 的总线宽度配置为 $width(bus_1) = 16$ ，则 $\langle PE_1, PE_3 \rangle$ ， $\langle PE_2, PE_3 \rangle$ 之间的一次数据传输都需要分成两次才能完成。

为了提高系统的性能，尽可能使所有的数据传输都能够在一个传输周期完成，CA-CIG选择通过每一条总线的最大数据传输类型作为该总线的数据宽度。由于本文采用树形拓扑结构，任意两个PE之间的数据传输路径都是唯一的，因此可以方便地确定出经过每条总线的所有数据传输类型，确定每条总线宽度。按照这种配置方案，图9表示的拓扑结构中，bus₁和bus₂的总线宽度分别为width(bus₁)=width(bus₂)=32。

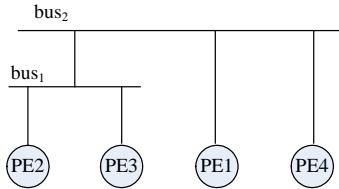


图9 总线宽度配置示例

总线宽度配置算法如下：

```

1: busi = pop(Bus);
2: While (busi != NULL)
3:   {
4:     Width(busi) = 0;
5:     for each arcj In Arc
6:       If (Path(busi, arcj) &&
           (Width(arcj) > Width(busi)))
7:         Width(busi) = Width(arcj);
8:     busi = pop(Bus);
9:   }

```

3 实验及其结果

本节以JPEG解码系统为例证明CA-CIG设计方法的正确性和有效性。JPEG解码系统的CIG模型可以由12个互相通信的功能单元(包括一个全局存储器)和16条通信边组成(图10)^[4]。假设兼容协议之间单位通信延迟均为1，不兼容协议之间的单位通信延迟均为6。

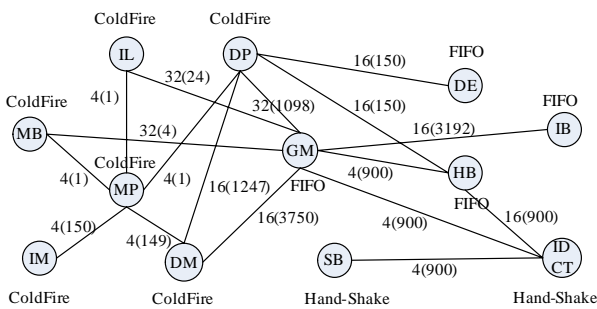


图10 JPEG解码系统的CIG模型

根据本文给出的CA-CIG设计方法，生成的JPEG系统

的OCA通信体系结构如图11所示，其中包括4条通信总线和3个协议转换器(菱形)。由图11可以看出，CA-CIG生成的OCA具有以下优点：

- (1)使通信频繁的PE经过较少的通信路径，减少通信开销。
- (2)生成高效的层次化总线拓扑结构以及较少的协议转换器。
- (3)配置高性能的总线通信协议。
- (4)设计合理的总线数据宽度。

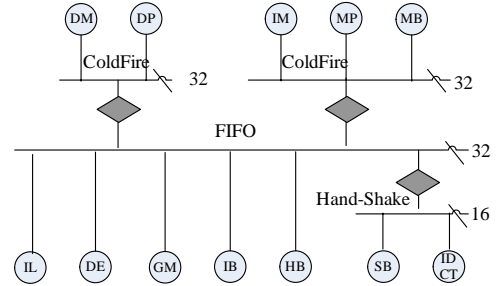


图11 JPEG解码系统的OCA通信体系结构

4 结论

CA-CIG系统级通信体系结构设计方法提出了一种基于CIG模型的OCA通信体系结构快速设计方法，旨在帮助芯片设计人员在系统设计的早期设计高性能的片上通信体系结构。CA-CIG提出了从CIG模型出发设计OCA的总体设计框架，并集成了通信拓扑体系结构生成、总线宽度配置和总线协议选择等过程。为了实现OCA的自动化设计，本文还提供了相应的算法实现了OCA设计过程中的关键步骤。通过JPEG解码实验表明，CA-CIG设计方法能够有效地分析CIG模型并设计高性能的OCA体系结构。

参考文献

- 1 Gasteier M, Glesner M. Bus-based Communication Synthesis in System Level[J]. ACM Transactions on Design Automation of Electric Systems, 1999, 4(1): 1-11.
- 2 Cai Lukai, Gajski D. Channel Mapping in System Level Design[R]. University of California, Irvine, Technical Report: ICS-TR-00-03, 2003.
- 3 Wu Qiang, Bian Jinian. A Hierarchical CDFG as Intermediate Representation for Hardware/Software Co-design[C]//Proc. of the International Conference on Communications, Circuits and Systems, Chengdu, China. 2002: 1429-1432.
- 4 Niu Yawen, Bian Jinian. CGEM: A Communication Graph Extraction Methodology Based on HCDFG for Channel Mapping in System Level Design[C]//Proc. of the 6th International Conference on Computer-aided Industrial Design & Conceptual Design, Delft. 2005.

(上接第114页)

参考文献

- 1 刘乃安. 无线局域网——原理、技术与应用[M]. 西安: 西安电子科技大学出版社, 2004.
- 2 金纯, 陈林星, 杨吉云. IEEE 802.11 无线局域网[M]. 北京: 电子工业出版社, 2004.

- 3 IEEE. IEEE Standard Part 11. Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Medium Access Control (MAC) Quality of Service (QoS) Enhancements[S]. 2004.
- 4 徐恪, 吴建平, 徐明伟. 高等计算机网络——体系结构、协议机制、算法设计与路由器技术[M]. 北京: 机械工业出版社, 2003.