基于 trace 的仿真平台的调度研究

刘 俊[1],李战怀[1],Iuliana Bacivarov[2],黄 铠[2]
(1 西北工业大学 计算机学院,陕西 西安 710072;
2 苏黎世联邦理工学院 计算机工程与网络实验室,苏黎世 CH-8092)

摘 要:随着多理机 SoC 设计的复杂度和异构性不断增长,需要研究设计一种新的性能评估方法来缩短产品开发周期.提出并实现了一种基于 trace 的仿真平台,旨在为早期阶段系统的设计空间考察提供性能评估.为了保证准确性,仿真平台要求可以考虑计算资源和通信资源的共享.着重讨论了不同的调度策略的设计,其中包括可抢先的机制.最后通过一个实验对仿真平台的调度机制进行了验证.

关键词:性能评估;仿真;调度策略;可抢先
中图分类号:TP302.7 **文献标识码**:A **文章编号**:1000-7180(2008)08-0064-05

Scheduling Mechanisms in High-Level Simulation Framework

LIU Jun[1], LI Zhan-huai[1], Iuliana Bacivarov[2], HUANG Kai[2]
(1 School of Computer Science and Engineering, Northwestern Polytechnical University, Xi'an 710072, China;
2 Computer Engineering and Networks Lab, Swiss Federal Institute of Technology Zurich, Zurich CH-8092, Swiss)

Abstract: The increasing complexity and heterogeneity in multiprocessor SoCs call for new performance evaluation methodologies in order to meet the severe time-to-market constraints. This paper presents a modular trace-based simulation framework to overcome this challenge. To maintain accuracy, the framework should take into consideration the sharing of computation resources and communication resources. This paper lays emphasis on discussing the scheduling mechanisms in the framework, which are used to providing mutually exclusive access to the resources. Finally an experiment is given to verify the correctness of the scheduling mechanisms in the framework.

Key words: performance evaluation; simulation; scheduling mechanism; preemption

1 引言

随着应用程序变得越来越复杂,传统的单处理机 SoC 架构已经不能满足苛刻的性能要求.现在的嵌入式系统设计正在从单处理机架构向异构的多处理机架构转变.在多处机系统设计的早期阶段,由于设计空间很大,为了减少模拟的开销,缩短开发周期,需要提升性能仿真的抽象层次,例如系统级.

文中提出并实现了一种基于 trace 的仿真平台来解决嵌入式设计中的系统级仿真问题.为了保证考察设计空间时的速度和高效,这个仿真平台主要是基于应用程序的执行 trace.在这个仿真平台中,应用程序的功能被一系列的 trace 所表示.而应用程序的计算行为和通信行为则被分别抽象为计算事件(computation event)和通信事件(communication event).以应用程序、硬件平台,以及从应用程序到硬件平台的映射作为输入,从而得到性能估计结果,例如估计的执行时间,处理机负载和总线负载等.

2 相关研究

首先探讨系统级的性能估计方法的相关研究,其中着重讨论时控功能仿真(timed functional simulation)和基于 trace 的仿真[1].

收稿日期:2007-09-12

万方数据

2.1 时控功能仿真

时控功能仿真是形式性能估计方法和时钟级仿真的折衷.在时控功能仿真中,应用程序的运行效应由时间段来表示.它的优点是比时钟级的仿真快而比形式的性能估计方法准确.缺点是当考察设计空间时,应用程序的功能代码被反复地执行,从而降低了它的效率.文献[2-3]中给出了常用的方法.

2.2 基于 trace 的性能估计方法

基于 trace 的性能估计方法对时控功能仿真进行了改进.它将计算行为抽象为一些高层次的执行 trace 从而获得速度上的提升.当模拟应用程序的计算行为时,目标系统不需要运行实际的代码.另外,在基于 trace 的性能方法估计中,程序的执行过程已经完全被展开,所以不再需要对原来代码的分支进行繁琐的分析.

Sesame[3] 支持对硬件平台的逐渐细化.在 Sesame 中,KPN[4] 进程网络规定了应用程序的建模方式.Trace 是通过 UNIX 中基于 IPC 的接口动态传递给硬件模型.这种方式决定了在对多个映射进行考察时,程序代码将被反复地执行.此外,代码也需要手动地加入注释以跟踪计算行为和通信行为.

基于 trace 的性能分析[5]利用通信分析图(communication analysis graph)来模拟片上系统的性能.初始的仿真步骤用于产生 trace.之后对 trace 进行静态的分析而不是动态的仿真.这使得考察设计空间时的速度大大增加.但这种方法主要用于对通信架构进行考察,而不支持对计算映射和通信映射同时进行考察.另外,总线的共享策略是有限的(只实现了一种基于优先级的策略).

2.3 文中提出的基于 trace 的仿真平台

为了提高效率,提出的仿真平台也采用了基于 trace 的方法[6].它主要是在高的层次对系统的性能进行估计,可以用于早期对设计空间进行考察.和现有的基于 trace 的性能估计方法相比,文中所提出的基于 trace 的仿真平台具有以下特征:

(1) 产生 trace 的过程是完全自动的.这将嵌入式系统的设计者们从繁琐的手工注释中解放出来.

(2) 可以考察复杂的通信路径.这个通信路径有可能由一系列的多个通信资源组成.

(3) 可以考察不同通信缓冲区的实现位置,比如位于通信中发送端处理机或接收端处理机的内存,或者是共享内存.

(4) 可以考察不同的原子事务数据大小.原子事务数据代表了数据包的粒度.原子数据大小可以让设计者在精确度和效率之间进行权衡.

(5) 可以模拟计算资源和通信资源的共享.设计实现了多种嵌入式系统中常见的调度策略,包括 TDMA、FIFO 和 Fixed Priority(FP).其中,FP 允许中断.调度也是文中所论述的重点部分.

3 系统框架

基于 trace 的仿真平台的框架结构如图 1 所示.应用程序首先是以 KPN 进程网络来定义的.它定义了每个进程的功能行为以及各个进程之间是如何通过软件通道(software channel)进行通信的.仿真平台中所使用的硬件是真实硬件的一个抽象描述,只保留了跟硬件组件相关的一些主要参数.其中主要的组件有计算资源和通信资源.而仿真平台中的映射则定义了进程是如何被映射到计算资源上,以及软件通道如何被映射到通信资源上.

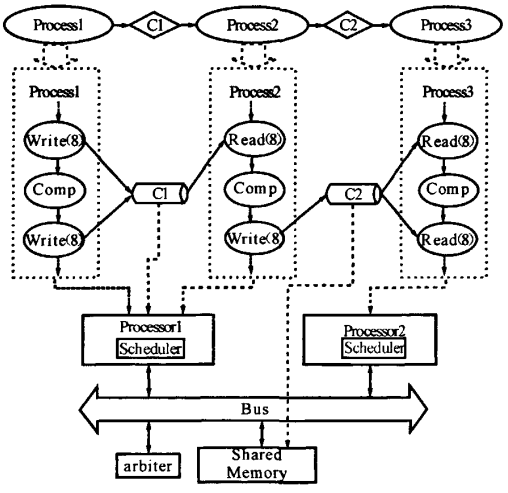

图 1　基于 trace 的仿真平台的框架结构

3.1 应用程序模型

用于表示应用程序的 trace 中包含三类事件:计算事件(computation event)、读事件(read event)和写事件(write event).

事件的具体含义如图 2 所示,最左边是某个进程中的代码片断.中间的是这段代码的有向图表示.有向图中的顶点对应代码中的一个读操作或者写操作,而边对应代码中的计算.边的权重代表了计算的延迟.最右边使用有向图的一个可能的展开方式来表示应用程序的一个执行路径,也就是 trace.如前所述,应用程序是根据 KPN 进程网络来规范的.由于 KPN 本身所固有的确定性,应用程序所产生的 trace 也是独立于实现平台和调度策略的.Trace 可

万方数据

以在考察设计空间的过程中被反复使用,而不需要再运行原先的具体代码,因而提高了效率.

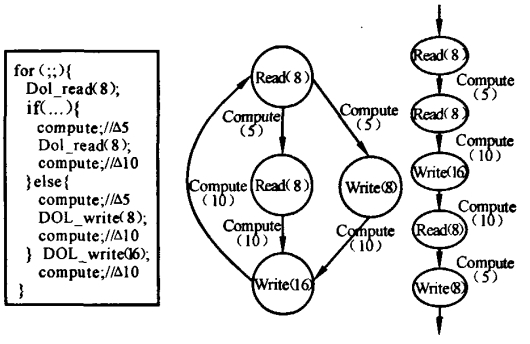

图 2 Trace 中事件的表示

进程网络中的每一个进程都有它自己的 trace. 这个 trace 中的事件集构成一个全序. 由于阻塞读和阻塞写的存在,来自不同进程的事件可能存在依赖关系. 因此,应用程序中的所有事件形成偏序. 如果两个事件之间不存在先后顺序,那么在硬件平台中他们便可以并发地执行.

3.2 硬件模型

基于 trace 的仿真中的硬件模型主要用来对事件的性能效果进行模拟. 硬件模型中主要包括两类资源:计算资源和通信资源. 在仿真的初始化阶段,所有的资源从硬件设计文件中提取出来. 文件中的参数定义了资源的类型、时钟频率和调度策略. 计算资源,如 RISC 或者 DSP,用于模拟计算事件. 由于应用程序的功能行为已经在产生 trace 的过程中被捕获了,计算资源只需要衡量在这个处理机上执行这个计算所需要的延迟.

对于通信事件,计算资源和通信资源都可以模拟其中的数据传输. 例如,内部通信是在计算资源上进行的. 另一方面,有时为了完成一个数据传输,一个通信事件需要被一系列的资源处理. 实际情况中,数据传输有时从源处理机需要经过好几个总线才能到达目的处理机. 另外,为了实现对这些资源的互斥访问,仿真平台提供了各种调度策略,如 TDMA、FIFO 和 FP.

4 调度策略

硬件模型中的每个资源都有它自己的调度器(scheduler),以对派遣到它上面的事件根据某种策略进行调度. 调度器中的事件队列(event queue)是各种调度策略的基础,也是被调度的对象. 对于不同的调度策略,调度器都有与之对应的算法. 但事件队列的结构对于所有类型的调度器都是一致的.

4.1 事件队列

每一个事件队列都是一个无限长的 FIFO 缓冲区,用来存放被派遣到这个资源上的事件. 对于一个处理机,一个事件队列对应一个映射到这个资源上的一个进程. 如果一个进程的 trace 中的下一个事件需要被执行时,这个事件会被分发到目标处理机并且插入到相应事件队列的末尾. 如果在事件队列中排在这个事件之前的所有事件都已经完成,并且属于该进程的时间片到达,那么这个事件将会被调度器调度执行. 对于通信资源,如总线,一个事件队列对应于连接到这个总线上的一个资源. 这个资源可能是一个处理机,也可能是另外一条总线. 事件队列用于组织来自于那个资源的读写请求. 在所提出的仿真平台中,每一个事件队列都可以被相应的调度器进行调度.

每个事件队列有三个状态,即 ready、running 和 waiting. 这几个状态对应于进程在调度中的各个状态. 例如图 3 的 TDMA 调度中,如果某个事件队列有事件等待被调度而时间片没有到达,这意味着进程处于等待状态并且没有阻塞在 I/O 上,这时事件队列的状态被设置为 ready. 当时间片到达后,调度器从事件队列中取出最早到达的事件,然后通过等待相应的延迟模拟这个事件的执行. 这时候事件队列的状态被设置为 running. 如果事件队列的事件都已经完成,并且没有新的事件来临而使事件队列变为空,那么事件队列的状态被设置为 waiting.

对于所实现的三种调度策略,即 TDMA、FIFO 和 FP,各自的设计细节如下文所述.

4.2 TDMA

在 TDMA 调度策略中,时间轴被划分为一些有着固定长度的时间片. 每个事件队列被分配一些时间片集合. 当这个事件队列的时间片来临时,该事件队列便可使用资源. 于是不同的事件队列可以共享处理机或总线的带宽. 这种策略在某种程度上可看作是公平的. 但是相对于其他的调度策略,用于上下文转换的消耗要大些.

在实现的 TDMA 调度策略中,如果某个事件队列的当前时间片不足以完成该事件队列中的所有事件,调度器将会在时间片结束时切换到另一个事件队列. 这可以看成是一种中断行为,调度器将正在执行的进程停止而将处理机交给另外一个进程. 事件队列在 TDMA 调度器中的状态转换如图 3 所示.

万方数据

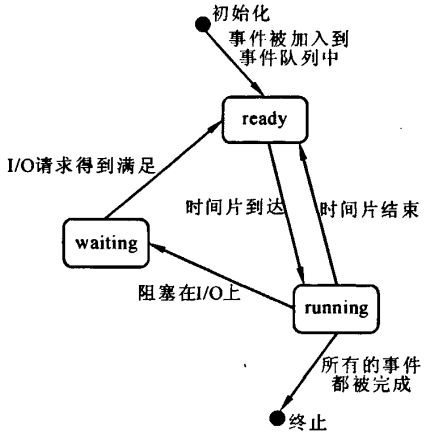

图 3　TDMA 调度器中的事件队列的状态转换

4.3　FIFO

FIFO 调度器是基于先到先服务的原则来实现的.事件队列将根据它们到达的先后顺序被选择执行.这是一种非中断的调度算法,因此每一个事件队列将被选择执行直到它因为 I/O 阻塞.当一个事件队列的状态变成 ready 时,这个事件队列被加入到 FIFO 调度器的就绪队列(ready queue)中.相对于其他调度策略而言,FIFO 调度策略的平均等待时间将被延长.如果有事件存在于事件队列中等待被执行,事件队列将被加入到就绪队列的末尾.如果之前所有的事件队列被执行完或者变为阻塞状态,下一个事件队列将会被调度器选择执行并且它的状态变为 running.之后这个事件队列将占有处理机直到该队列中的所有事件被执行完,然后它阻塞在 I/O 上.如果 I/O 请求得到了满足,这个事件队列的状态将再次变为 ready.在 FIFO 调度器中,事件队列不存在从 running 到 ready 的转换,原因是 FIFO 中不存在中断.

4.4　FIXED PRIORITY

FP 调度策略原来被称作实时调度.FP 调度器保证了在任何时候,当前具有最高优先级的事件队列能够得到处理机,即便此时有其他的进程正在执行.在仿真平台中,每一个事件队列被赋予一个固定的值来表示各自的优先级.当事件被加入到具有最高优先级的事件队列中时,这个事件队列马上被选择执行,因此它有可能中断其他事件队列的执行.在 FP 调度器中的事件队列的状态转换中,中断发生在当具有更高优先级的事件队列变成 ready 时,而当前的事件队列的状态将则从 running 变成 ready.当具有更高优先级的队列都被完成或者阻塞时,之前被中断的事件队列将继续执行.

5　实验

5.1　实验配置

生产者消费者程序是按照 KPN 进程网络的规范来写的.它包括三个进程,即 generator、square 和 consumer,以及两个连接这些进程的软件通道,即 C1 和 C2.图 4 描述了该生产者消费者进程网络的结构.

图 4　生产者消费者的 KPN 进程网络

如图 5 所示,实验中仿真平台中的硬件平台由两个处理机 processor1 和 processor2,一根总线 in_tile_link,以及一块共享存储组成.

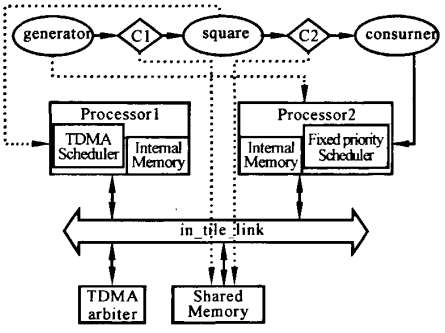

图 5　仿真中所使用的映射

5.2　实验结果

限于篇幅,以下仅对调度策略为 FP 的一种映射的实验结果进行详细分析.具体的映射如图 5 所示.在这个映射中,将 square 映射到 processor1,generator 和 consumer 映射到 processor2.其中在 processor2 中所采用的调度策略为 FP,并且 consumer 具有更高的优先级.

使用 VCD 波形来追踪仿真的执行过程,输出的波形如图 6 所示.其中针对 FP 的一些特性在波形中得到了体现.

6　结束语

文中提出了一种基于 trace 的仿真平台,用来指导设计者在设计空间考察的早期阶段对多处理机 SoC 的功能进行评估.描述了平台的基本组件和基本机制,如应用程序模型,硬件模型等.然后着重讨论了硬件模型中的调度问题.对于每种调度策略,即 TDMA、FIFO 和 Fixed Priority,为之设计了相应的

万方数据

算法.最后,通过用所实现的仿真平台对生产者消费者程序进行了实验,波形的输出验证了所设计的调度机制的正确性.

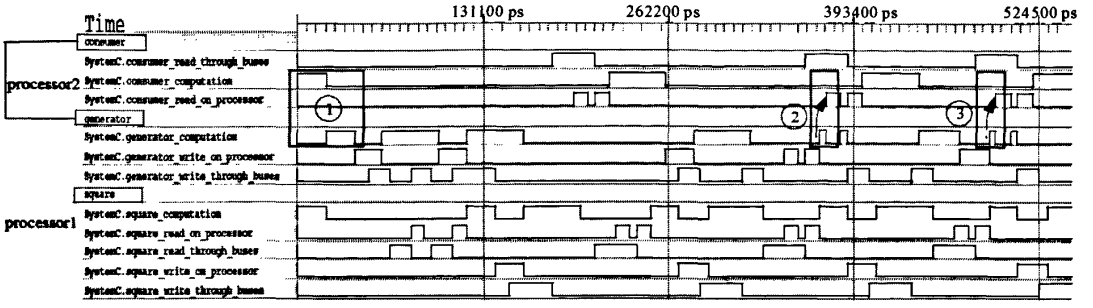

图 6 基于 trace 的仿真的波形输出

参考文献:

[1] Baghdadi A, Zergainoh N E, Cesario W O, et al. Combining a performance estimation methodology with a hardware/software codesign flow supporting multiprocessor systems[J]. IEEE Trans. Softw. Eng., 2002,28(9):822-831.

[2] Bacivarov I, Bouchhima A, Yoo S, et al. Chronosym: a new approach for fast and accurate soc cosimulation[J]. International Journal of Embedded Systems, 2005(1):103-111.

[3] Pimentel A D, Erbas C, Polstra S. A systematic approach to exploring embedded system architectures at multiple abstraction levels[J]. IEEE Trans. Comput., 2006,55(2):99-112.

[4] Kahn G. The semantics of a simple language for parallel programming[C]// Proc. IFIP Congress. Netherlands, North Holland Publishing Co, 1974.

[5] Lahiri K, Raghunathan A, Dey S. System-level performance analysis for designing on-chip communication architectures[J]. IEEE Trans. Computer-Aided Design Integr. Circuits Syst., 2001,20(6):768-783.

[6] 沈绪榜, 梁政. 一种嵌入式协处理器的设计[J]. 微电子学与计算机,2001,18(5):21-24.

作者简介:

刘 俊 男,(1982-),硕士研究生.研究方向为存储管理、嵌入式设计等.

李战怀 男,(1961-),教授,博士生导师.研究方向为数据库理论与技术、网络存储等.

(上接第 63 页)

动性能,在所有工艺角及温度范围(-40~125℃)下芯片都能正常工作,达到预定设计目标.

参考文献:

[1] Xu Jiangping, Cao Xiaohong, Luo Qiancho. The effects of control techniques on the transient response switching DC-DC converter[C]// IEEE PEDS' 99. Hong Kong, 1999: 794-796.

[2] Holland B. Modeling, analysis and compensation of the current-mode converter[R]. USA: Texas Instruments Incorporated, 1999.

[3] 童诗白, 华成英. 模拟电子技术基础[M]. 北京:高等教育出版社, 1980: 115-116.

[4] Raymond B Ridley, Bo H Cho, Fred C Y Lee. Analysis and interpretation of loop gains of multiloop-controlled switching regulators[J]. Power Electronics, IEEE Transactions on power electronics, 1988, 3(4): 489-498.

[5] Yuvarajan S. Performance analysis and signal processing in a current sensing MOSFET (SENSEFET)[C]// Industry Applications Society Annual Meeting. USA, 1991: 1445-1450.

作者简介:

陈晓飞 女,(1968-),博士后.研究方向为模拟及数模混合集成电路设计.

邹雪城 男,(1964-),博士,教授,IEEE 会员.研究方向为数模混合大规模集成电路设计.

万方数据