

一种面向应用优化的片上总线调度策略

李德贤 彭剑英 严晓浪

(浙江大学超大规模集成电路设计研究所 杭州 310027)

摘要: 该文提出了一种面向应用优化的片上总线调度策略。以系统通信事件信息为基础,使用最小任务松弛时间与最小总线空闲时间相结合的调度策略,在优先保证满足任务实时要求的基础上,最大限度利用总线空闲时间,提高调度效率。并提供了可配置的权重参数 α 用于总线时间消耗与片上缓冲区容量之间的设计折衷。在双核 SoC 平台上实现了本文调度方法,并以最新视频编解码标准 H.264/AVC 为目标应用,与 FP(Fixed Priority), SBA(Slack Based Arbitration), RR(Round Robin)等调度方法进行了性能对比。实验结果表明, $\alpha = 0.5$ 时,较以上 3 种策略,分别平均可减少 16.6%、13.2%与 9.7%的总线时间,在实时性能方面,较最接近的 SBA 方法,未实时完成的任务数量减少了 59.4%。额外缓冲区开销随 α 的变化关系表明,在最坏情况下($\alpha = 0$)仅需 435 字节。

关键词: 片上系统; 通信建模; 总线调度

中图分类号: TN47; TP336

文献标识码: A

文章编号: 1009-5896(2009)05-1205-05

An Optimized Bus Scheduling Scheme Based on Specific Application

Li De-xian Peng Jian-ying Yan Xiao-lang

(Institute of VLSI Design, Zhejiang University, Hangzhou 310027, China)

Abstract: An application-specific bus scheduling scheme was proposed in this paper. Two-fold optimization was considered in this scheme based on the communication events collected by system modeling and simulation. The first one, which had higher priority, was real time constraints of tasks while the other was making use of bus idle time to transfer data as much as possible. A configurable optimization parameter α was also proposed for the tradeoff between the total bus time consumed and the extra on-chip buffer requirements. This scheme was implemented in a dual-core SoC (System on Chip) for the H.264/AVC decoder and compared with RR (Round Robin), FP (Fixed Priority) and SBA (Slack Based Arbitration) schemes. The results showed that the proposed scheme had an average 16.6%, 13.2% and 9.7% less bus time when α was set to 0.5. The number of missed real time constraints tasks was 59.4% less than the SBA scheme, which was the closest to our scheme. The relationship between α and the extra on-chip buffer cost showed that under worst condition ($\alpha = 0$), it was only 435 bytes.

Key words: SoC; Communication model; Bus scheduling

1 引言

随着片上系统(System on Chip, SoC)规模的不断增大,设备间的数据通信逐渐成为性能瓶颈^[1]。通信架构在复杂 SoC 系统的设计中占据重要地位,其核心就是确定总线调度策略。常用调度策略有 RR(Round Robin)、FP(Fixed Priority)等^[2]。在实时性要求高、总线冲突频繁的系统(如实时视频编解码)中,以上几种通用策略往往不能胜任。因此针对特定目标应用进行系统总线调度的优化设计,已成为目前研究的热点。Sudeep 等^[3]研究了通信架构的自动生成技术,其设计流程考虑了总线调度策略的确定,但仅针对上述通用策略进行了设计空间搜索,无法得到优化的结果。陈科明等在文献^[4]中提出了通过调整任务的可执行窗以提高多媒体

SoC 系统总线利用效率的方法,并考虑了因任务开始时间变化而带来的片上缓冲区容量增大问题。但其方法仅适用于孤立的子任务而非完整应用,降低了实用性。Jun 等提出了基于任务松弛时间的总线调度策略^[5],是对最早截止时间优先(Earliest Deadline First, EDF)调度的改进,通过预先设定任务持续时间与截止时间保证实时性,但该方法对总线的利用率较低,导致任务完成时间较长。

本文提出了一种面向应用进行优化的片上总线调度策略。通过对目标应用进行通信建模及仿真得到系统通信事件的发生时间、截至时间、持续时间与通信量等信息。以此为基础,采用最小任务松弛时间(Minimum Slack Time, MST)与最小总线空闲时间(Minimum Idle Time, MIT)的两层调度策略,并给 MST 调度分配以高优先级。在确保任务实时性的同时,充分利用总线空闲时间进行数据传输,从而缩短系统运行时间,提高性能。并以最新视频编解码标准 H.264/

AVC^[6]为应用举例,与上述几种调度策略进行了性能对比。

2 改进事务级模型

针对特定应用进行总线调度设计需要以系统运行时产生的通信事件为依据,为此要求对应用的数据通信部分进行高层建模与仿真。基于 SystemC 类库的事务级模型(Transaction Level Model, TLM)^[7]支持计算与通信分离建模及系统设计的逐层细化,为软硬件协同设计与验证提供了统一的平台。然而数据通信在 TLM 中以抽象函数调用实现,无法获得其准确信息。本文针对通信建模的要求,对 TLM 进行了如下改进:

(1)使用时序标记技术仿真处理器。嵌入式系统中的数据通信大多由处理器发起,为保证通信事件的时序正确,要求处理器模型以时钟精确的方式运行。在 TLM 中,处理器模型以时钟精确的指令集仿真器实现,其上运行目标应用的二进制代码。而媒体处理器由于其特殊的架构^[8],往往缺乏高性能编译系统的支持,手工编写整个应用的汇编代码,费时费力。本文采用时序标记的方法,只需将应用中各算法核心部分以汇编代码实现,得到其运行所需的时钟周期数后,再反标到 C 语言参考代码相应位置中。在尽可能少的改动原参考代码的前提下,保证了系统通信事件的时序正确。同时,由于避免了使用指令集仿真器,也极大提高了仿真速度。以下给出了 H.264/AVC 中 4×4 像素块插值算法反标的例子:

```

fifo_in_port->burst_read(m5[0]...m5[3]);//读入所需数据
for (j = 0; j < 4; j++)
{
m6[0]=m5[0]+m5[2];m6[1]=m5[0]-m5[2];m6[2]=(m5[1]
>>1)-m5[3];
m6[3] = m5[1] + (m5[3] >>1);m7[0] = m6[0] + m6[3];m7[1]
= m6[1] + m6[2];
m7[2]=m6[1] - m6[2];m7[3]=m6[0]-m6[3];
}
wait(14*t_CYCLE);//时序标记,以上代码共需 14 个时钟
周期
fifo_in_port->nb_write(m7[0]...m7[3]);//结果写回存储

```

(2)以 Kahn 进程网络(Kahn Process Network, KPN)中的无限深度 FIFO 做为通信介质。片上系统各功能单元使用 FIFO 的方式进行通信可保证正确的任务调度次序^[9],同时由于在进行通信建模时,系统的通信协议尚未确定,将通信资源设置为无限可避免因资源竞争而导致的通信事件丢失。通过修改 SystemC 内建通道类型 `sc_fifo` 实现了此无限 FIFO。由于 `sc_fifo` 类型仅提供静态端口检查,为保证仿真模型运行时刻读写进程同步,对于同一 FIFO 对象的访问设置了以下限制:(a)同时刻最多只允许一个进程读写;(b)对于已被标记为读操作的进程,不能再发起写请求;(c)对于已被标记为写的进程,不能再发起读请求。图 1 是一个简单的

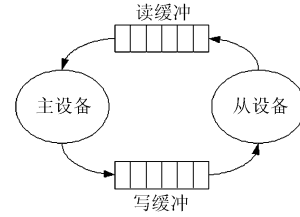


图 1 FIFO 通信示例

主从设备通过无限 FIFO 进行通信的例子。

(3)保持接口一致性。各种调度策略以一致的接口 `arbitrate()` 嵌入到原通信模型中进行仿真,从而实现了通信建模与性能验证平台的统一,提高了代码的复用性。

3 总线调度策略

一个总线传输模型 τ 可以使用其传输时间 c 、截止时间 d 与任务周期 t 来表示,定义为 $\tau = (c, d, t)$ 。嵌入式系统中,截止时间反映了任务实时性的要求。在实际应用中, t 一般与 d 相同。而一系列周期性的总线任务集合可定义为 $\Gamma_n = \{\tau_1, \dots, \tau_n\}$ 。在考虑任务的启动开销 Ov 、系统任务开销 Ov_{sys} 与从设备延时 Blk 的前提下,将 Γ_n 中的每一个元素按照优先级高低排序后,可以得到 FP 模型中总线任务 τ_i 在截止时间 d_i 之前完成的条件:

$$\left(\sum_{j=1}^i \frac{C_j + Ov_j}{t} \left\lfloor \frac{t}{T_j} \right\rfloor \right) + \frac{Ov_{sys}}{t} \left\lfloor \frac{t}{T_{sys}} \right\rfloor + \frac{Blk_i}{t} \leq 1, \quad 0 < t < d_i \quad (1)$$

不等式左边第一部分表示所有优先级高于或等于 τ_i 的传输时间包括启动开销之和,第二部分为系统任务开销,而第三部分为从任务的延时开销。

对于 RR 模型,可进行如下的调度可行性分析:假设 p 为任务 τ_i 在 RR 转轮中的位置,对于包含 n 个任务的 Γ_n 而言,有 $1 \leq p \leq n$ 。要使 τ_i 在其截止时间 d_i 完成,需满足:

$$(p-1)S + C_i + \left(\left\lfloor \frac{C_i}{S} \right\rfloor - 1 \right) (n-1)S \leq d_i \quad (2)$$

其中 S 代表 RR 模型中每个时间片的大小。不等式左边第一部分表示在 τ_i 第一次获得总线之前其余设备的时间片之和,第二部分为 τ_i 的实际传输时间,第三部分为完成 τ_i 传输的过程中,由其他设备占据总线的时间总和。显然当 $p = n$ 时,是任务 τ_i 的最差情况。此时式(2)可改写为

$$(n-1)S + C_i + \left(\left\lfloor \frac{C_i}{S} \right\rfloor - 1 \right) (n-1)S \leq d_i \quad (3)$$

经过整理后可得:

$$C_i + \left\lfloor \frac{C_i}{S} \right\rfloor (n-1)S \leq d_i \quad (4)$$

式(4)中假设每个设备的 S 均相同,在不同 S 的情况下,可改写为以下更一般的形式:

$$C_i + \left\lfloor \frac{C_i}{S} \right\rfloor \sum_{j=1, j \neq i}^n S_j \leq d_i \quad (5)$$

与式(1)类似，在考虑任务启动开销的情况下，使用 $C_i + Ov_i$ 取代式(5)中的 C_i 即可。

在前述 τ 模型的基础上，本文增加了任务开始参数 st ，修改后的模型可使用 $e(st, lat, dl, t)$ 来表示，其中 st 代表此次通信事件的起始时间， lat 代表通信持续时间， dl 代表其截止时间， t 代表任务周期。忽略了总线传输的初始化延迟。其中 $dl - st$ 的值反映了任务实时性要求的高低。 $dl - st$ 越大时，任务实时性要求越低，同时调度的余地也越大；反之则调度余地越小。 $dl - st = 0$ 时，对该任务进行调度必然违反实时性要求。以 st' 代表经过调度后任务的开始时间，则调度后的任务集合可表示为 $\Gamma'_n = \{\tau'_1, \dots, \tau'_n\}$ ，其中， $\tau' = (st', lat, dl, t)$ 。而在改变任务开始时间的同时，为保证数据完整性及各任务间的数据依赖关系，需要相应增加片上缓冲区的容量。时间由 st 延迟到 st' 时，需要增加的缓冲区容量为 $m = (st' - st) \cdot W$ 。 W 代表总线宽度，单位为 byte。则系统总的缓冲区容量增加可表示为 $M = \sum_{i=0}^N (st'_i - st_i) \cdot W$ 。任务完成总时间为 $T = st_i + lat_i$ 。而当 $st_i > dl_i$ 时，经调度后的任务不满足实时要求。

定义总线空闲时间 $IT = st_{i+1} - (st_i + lat_i)$ ，松弛时间 $ST = st_{i+1} - (dl_i + lat_i)$ 。各变量定义如图 2 所示。本文提出的 MST 与 MIT 两层总线调度策略可以描述如下：当系统中发生多个通信任务竞争总线时，首先计算出各任务的 ST ，若存在 $ST < 0$ 的任务，则该任务获得总线所有权，进行数据传输；若存在多个任务满足 $ST < 0$ 的条件，则 $|ST|$ 最大的任务占据总线，以尽可能保证任务的实时性。当所有并发任务均满足 $ST > 0$ 时，说明当前时刻并发任务的实时性要求不高，则按照总线空闲时间最小的原则进行调度，各并发任务中的 IT 值最小的获得总线所有权。

为分析本文提出模型的调度可行性，首先将 Γ_n 分为两部分重新组织：第一部分所有任务的 $ST < 0$ ，且按照 $|ST|$ 以从大到小的顺序排列；第二部分所有任务 $ST > 0$ ，且按照 IT 从小到大的顺序排列，记为 $\Gamma_n = \{\tau_1, \dots, \tau_k, \dots, \tau_n\}$ 。对于任务 τ_i 而言，其调度可行条件为

$$\sum_{j=1}^{i-1} lat_j + lat_i \leq dl_i \quad (6)$$

式(6)不满足时，任务将不能在其截止时间之前完成，违反了实时性要求。实际的嵌入式应用中，在不会导致整个应用非实时的前提下，一定数量的非实时总线任务是可以容忍的。因此，理论上无调度可行性的任务在本文模型中仍有调度的余地。此时有两种选择，将当前所有的任务推迟到下一任务完成后，或者优先进行当前调度，而推迟之后任务的启动时间。两种选择都会带来缓冲区增大及总时间延长的损耗，图 3 显示对于并发任务 e_0 与 e_1 分别采用以上两种选择后的调度结果。

总的调度开销 C 的计算体现了系统对于片上缓冲与完

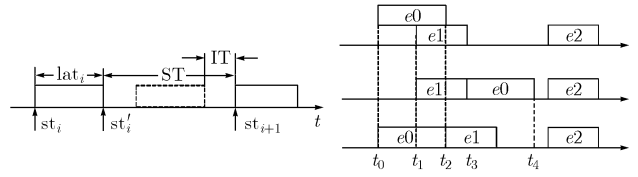


图 2 各变量定义

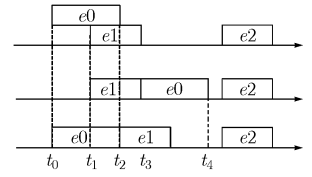


图 3 两种不同调度选择

成时间的关注度的高低。片上缓冲在 C 中占据的比例越大，总线调度越倾向于选择缓冲开销小的方法进行；反之，则倾向于选择更短的完成时间。为此在 C 的计算中引入了参数 α 来控制优化倾向， α 越大时，表明更倾向于优化缓冲区开销；反之，更强调对于总线完成时间的优化。式(8)为 C 的表达式，不难看出， $\alpha = 0$ 时，忽略片上缓冲，仅需要根据完成时间的大小来选择调度策略； $\alpha = 1$ 时，仅需要根据片上缓冲的开销进行选择。

$$C = \alpha \cdot \Delta M + (1 - \alpha) \cdot \Delta T \quad (7)$$

图 4 所示为分别使用 RR、SBA 与本文方法对任务 $e_0 \sim e_3$ 进行调度的情况。由图中可知，使用 RR 调度时任务 e_3 开始时间较其截止时间晚 Δt_1 ，不满足任务实时要求；而使用 SBA 调度，总线时间消耗较本文多 Δt_2 。

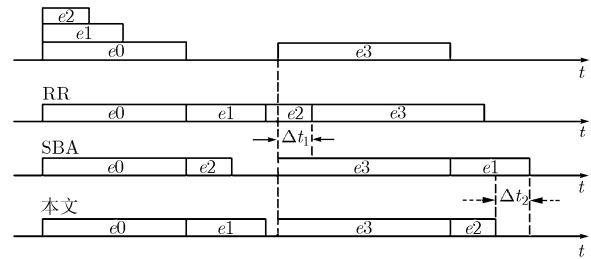


图 4 RR SBA 与本文调度对比

4 实验与结果比较

为了验证本文提出调度策略的有效性，以自主研发的两款嵌入式处理器 Spock 与 Schubert^[10,11]为基础搭建了 H.264/AVC 解码双核 SoC 平台。Spock 为单标量 RISC 处理器，其上运行并行度较低的比特流解析、熵解码与解码控制等应用；Schubert 是 SIMD 与 VLIW 混合架构的矢量处理器，运行反变换、运动补偿、帧内预测等计算并行度较高的解码算法核心部分；而环路滤波算法复杂度高，控制流程复杂，用专用硬件加速器实现^[12]。系统结构如图 5 所示。将 H.264/AVC 标准映射到此平台，完成通信建模后其结构框架如图 6。

实验选择 4CIF(688×576 像素)格式的 News、Akiyo、Foreman 与 Mobile 序列做为输入，帧数为 298，帧率为 30 帧每秒(frame per second, fps)。总线调度策略以 SystemC 模块方式实现，使用统一的 arbitrate()函数接口，可直接嵌入原通信模型中进行仿真与性能评估。

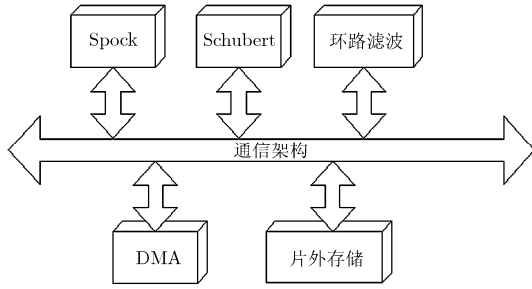


图 5 双核 SoC 系统结构

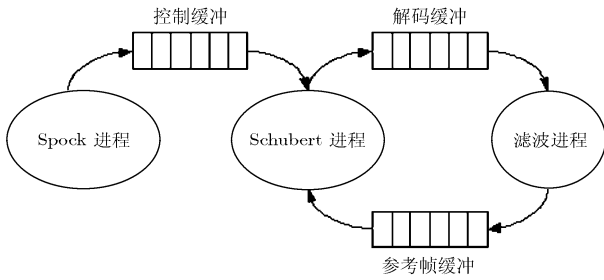


图 6 通信模型结构图

首先令 $\alpha = 0.5$ ，比较了 FP、RR、SBA 与本文方法完成解码应用时分别所消耗的总线时间及不满足实时要求的任务数量。表列出总线时间的对比情况(为简化讨论,假设系统所有设备,包括总线均工作于相同频率,表中数据单位为 Mcycle(百万时钟))。本文方法较 FP、SBA 与 RR 策略总线用时平均分别缩短了 16.6%、13.2%与 9.7%。同时,由于复杂度较低的视频序列存在较大的总线空闲时间,采用本文方法进行调度时,效率更高。如表 1 所示,News 序列与 Akiyo 序列平均可缩短 18.7%,而 Foreman 与 Mobile 序列为 15%。而违反系统实时要求任务的平均数量如图 7 所示。由于本文采用了实时性优先的两层调度,较其他三种策略可以更好的满足任务的实时要求,较性能最接近的 SBA 策略,平均可减少 59.4%的未能实时完成的任务数量,而相对于 FP 与 RR,这一数据分别为 86.8%与 82.9%。

其次,选择不同 α 值,研究了本文方法随 α 变化带来的片上缓冲与总线时间的变化关系,如图 8 与图 9 所示。随着 α 的增大,式中总线时间所占比重下降,向着有利减少额外缓冲区消耗的方向进行;反之,倾向于减少总线时间。 $\alpha = 0$ 时最大限度优化完成时间,而 $\alpha = 1$ 时最大限度优化片上缓

表 1 总线时间消耗对比

	FP	SBA	RR	本文
News	113.6	108.9	105.3	93.7
Akiyo	120.7	110.8	119.5	96.5
Foreman	173.2	169.0	157.0	147.5
Mobile	178.7	175.3	159.8	151.6
平均	146.6	141.0	135.4	122.3

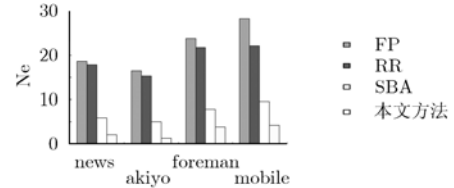


图 7 违反实时要求次数对比

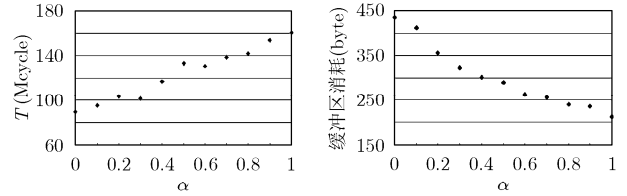


图 8 总线时间消耗随 α 变化情况

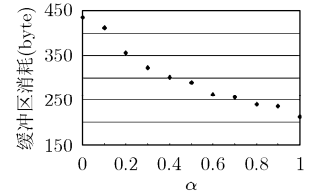


图 9 缓冲区消耗量随 α 变化情况

冲。由图中可知,最大的缓冲区开销为仅 435 字节,在可接受的范围内,而此时系统完成时间可优化至 89.2Mcycle;最小缓冲区开销为 212 字节,此时系统总线时间将增加至 160.8Mcycle。

5 结束语

本文提出了一种面向应用优化的总线调度策略,通过建模与仿真得到系统完整通信事件后,在优先保证任务实时性的基础上,最大限度利用总线空闲进行数据传输,并提供了可配置的优化权重参数用于调度性能与片上缓冲增大之间的取得设计折衷。通过最新视频编解码标准 H.264/AVC 解码器中的应用,证明本方法可以高效的进行复杂 SoC 系统片上总线的调度。下一步的工作是完成总线层数、传输协议等参数的优化设计。

参考文献

- [1] Kreutz M and Carro L, et al. Communication architectures for system-on-chip[C]. 14th symposium on integrated Circuits and Systems Design Pirenopolis, Brazil, Sep. 10-15, 2001: 14-19.
- [2] Meyerowitz T, Pinello C, and Vincentelli A. A tool for describing and evaluating hierarchical real-time bus scheduling policies[C]. Design Automation Conference, Anaheim, California, USA, Jun. 8-13, 2003: 256-261.
- [3] Pasricha S, Dutt N, and Ben-Romdhane M. Extending the transaction level modeling approach for fast communication architecture exploration[C]. International Conference on Hardware/Software Codesign and System Synthesis, Stockholm, Sweden, Sep. 8-10, 2004: 242-247.
- [4] 陈科明, 刘鹏, 王维东, 姚庆栋. 用于多处理器媒体SoC设计的实时总线调度优化策略[J]. 浙江大学学报(工学版), 2007, 41(9): 1546-1551.
Chen Ke-ming, Liu Peng, Wang Wei-dong, and Yao Qing-dong. Scheme to optimize real-time bus scheduling in multiprocessor SoC for media processing[J]. Journal of

- Zhejiang University (Engineering Science)*, 2007, 41(9): 1546-1551.
- [5] Jun M, Bang K, and Lee H, *et al.* Slack-based bus arbitration scheme for soft real-time constrained embedded systems[C]. Asia and South Pacific Design Automation Conference, Yokohame, Japan, Jan. 23-26, 2007: 159-164.
- [6] ISO/IEC 14496-10, ITU-T Rec, H.264, joint video specification [S]. 2002.
- [7] Cai L and Gajski D. Transaction level modeling: An overview[C]. International Conference on Hardware/Software Codesign and System Synthesis, Newport Beach, CA, USA, Oct.1-3, 2003: 19-24.
- [8] Salami E and Valero M. A Vector-uSIMD-VLIW Architecture for Multimedia Applications[C]. International Conference on Parallel Processing, Oslo, Norway, Jun.14-17, 2005: 69-77.
- [9] Pimentel A and Erbas C. An IDF-based trace transformation method for communication refinement[C]. Design Automation Conference, Anaheim, California, USA, Jun. 8-13, 2003: 402-407.
- [10] Yang Ye, Yang Jian, and Qin Xing, *et al.* GEM-SOC: A RISC/DSP dual-core platform for portable media applications[C]. International Conference on Solid-State and Integrated Circuit Technology, Shanghai, China. Oct.23-26, 2006, 1797-1799.
- [11] Liu Kunjie, Qin, Xing, and Yan Xiaolang, *et al.* A SIMD video signal processor with efficient data organization[C]. IEEE Asian Solid-State Circuits Conference, Hangzhou, Nov.13-15, 2006: 115-118.
- [12] Sima M, Zhou Yuanhua, and Zhang Wei. An efficient architecture for adaptive deblocking filter of H.264/AVC video coding[J]. *IEEE Trans. on Consumer Electronics*, 2004, 50(1): 292-296.
- 李德贤：男，1981年生，博士生，研究方向为专用 SoC 平台的设计空间搜索技术。
- 彭剑英：女，1982年生，博士生，研究方向为视频编解码标准的软硬件实现技术。
- 严晓浪：男，1947年生，教授，博士生导师，研究方向为集成电路 CAD 技术与集成电路设计。