

# 应用于语音信号处理的FPGA并行访问设计

朱永金, 成友才

(四川职业技术学院电子电气系 四川 遂宁 629000)

**【摘要】**为了实现实时语音处理,利用FPGA并行访问的特性,采用低地址物理存储空间由FPGA可配置逻辑模中的存储资源实现;高地址的物理存储空间由存储阵列实现,设计了动态置换算法来控制逻辑地址空间到物理空间的映射,将频繁访问且冲突概率高的数据块映射到低地址的物理存储空间上。实验数据表明,采用并行访问控制的动态置换算法能够实现稳定的访存性能。

**关键词** 动态置换; FPGA; 并行访问; 语音信号处理

中图分类号 TP393.03

文献标识码 A

doi:10.3969/j.issn.1001-0548.2012.01.030

## FPGA-Based Parallel Access Design for Applications of Voice Signal Processing

ZHU Yong-jin and CHENG You-cai

(Electric and Electronic Department, Sichuan Vocational and Technical College Suining Sichuan 629000)

**Abstract** In order to realize real-time voice processing, a parallel access design based on FPGA is proposed. Low memory space is implemented by distributed CLBs while the high memory space is constructed by the embedded arrays on FPGA. The mapping from logical memory to physical memory is introduced. With the help of dynamical replacement algorithm, the frequently accessed memory block is mapped into low memory space. Our experiment results show that, the dynamic replacement algorithm keeps memory access performance in a suitable range.

**Key words** dynamic replacement; field programmable gate array (FPGA); parallel access; voice signal processing

随着逻辑门数量的迅速增加,FPGA被广泛用于语音信号处理等领域<sup>[1-2]</sup>。实时语音传输(VOIP)是互联网的重要应用,由于互联网存在延迟和丢包等现象,直接播放从互联网上接收到的语音数据包并不能够获得令人满意的语音效果。引入回放机制控制数据包的接收、回放和解码是获得满意语音效果的一种有效的途径。对于VOIP网关之间大量的语音数据处理,软件实现的回放机制并不能够很好地满足实时性需要<sup>[3]</sup>。鉴于此,如何使用FPGA有限的可编程资源设计合适的语音信号处理模型及访问控制算法,是实现基于硬件的实时语音信号的方法和途径。

基于硬件的语音信号处理模型包括:语音数据包接收模块、回放算法模块、音频解码模块、播放模块。基于硬件的语音信号处理模型的工作流程为:语音数据包接收模块接收到语音数据后存放到共享存储器中,并且通知回放算法模块进行处理;回放

算法接收到回放算法模块的消息后进行数据处理,将处理后的数据存放到共享存储器中,并且通知音频解码模块进行解码工作;音频解码模块完成解码任务后,同样将数据存放到共享存储器中,通知播放模块读取数据并转化为声音信号。在整个语音信号处理过程中,每个模块都需要访问共享存储器<sup>[4]</sup>。如果共享存储器只提供一个访问端口,那么每个模块只能采用分时复用的办法以共享这个访问端口,这样语音信号的处理只能串行进行;如果能够提高各个模块对共享存储器并行访问的速度,整个语音信号处理就可能实现流水作业,极大地提高吞吐率。

## 1 基于FPGA的并行访问设计

### 1.1 存储器框架

FPGA能够提供的存储资源包括两类:一类是存储阵列;另一类是可配置逻辑模块。存储阵列是

FPGA的主要存储资源,其宽度、深度以及端口数目可以按照用户需求进行配置,如Xilinx芯片上的存储阵列可以配置为2K×9、1K×18、512×36 bit单端口存储块,或者是16K×1、8K×2、4K×4 bit的双端口存储块<sup>[5]</sup>。

对存储器的访问,可使用地址交叉技术<sup>[6]</sup>实现并行访问控制电路设计;也可将该技术应用于FPGA,把可配置逻辑块的存储资源映射成为单端口电路<sup>[7-8]</sup>,并且采用地址交叉技术将多个单端口电路组成小容量的并行访问控制电路;另外,还可将存储阵列映射为双端口电路<sup>[9]</sup>,并且同样使用地址交叉技术将多个双端口电路组成大容量的并行访问控制电路。对比几种方法,文献[9]的设计几乎不消耗可配置逻辑块资源,但是无法将冲突概率保持在一个较低的水平<sup>[10]</sup>。因此本文采用了图1所示的混合模式,以64 KB的小容量存储器为例,0~4K存储电路由可配置逻辑模块中的存储资源实现,用于实现处理单元之间的高并行度数据共享;4K~64K存储电路由存储阵列实现,用于实现处理单元之间低并行度的数据共享。

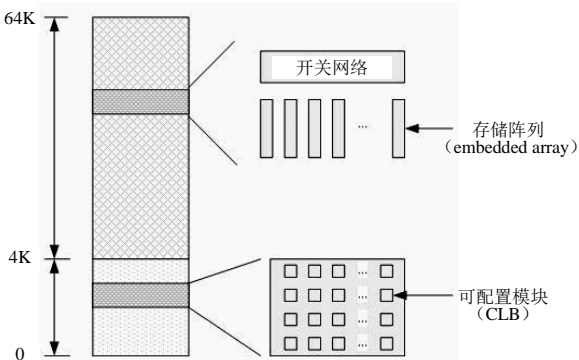


图1 混合型的并行访问控制

### 1.2 存储空间管理

如图1所示,0~4K存储电路对应于可配置逻辑块构成的低冲突概率存储器;4K~64K存储电路对应于由存储阵列构成的高冲突概率存储器。前者更适合于并行访问要求更高的领域,即适合某个应用中并行访问要求更高的部分。在实际应用中,用户很难准确地将低冲突概率的存储资源分配给有需求的处理模块(实际上,各个模块在不同的时间段上访问共享存储的情况也是不同的)。因此,采用动态置换方法,把低地址物理空间留做备用,优先分配高地址物理空间,以大小为N的数据块作为粒度记录冲突概率,动态地将频繁访问的数据块映射到低地址空间上。图2给出了逻辑地址空间到物理地址空间的映射示意图。

假设音频解码模块申请16 KB的空间用于存放解码产生的数据。存储器首先在高地址物理空间中分配16 KB,然后将逻辑地址映射到分配的物理空间。在解码过程中,当发现某个4 KB数据块的访问频率很高且冲突概率较高(假设粒度大小为4 KB),存储器会将该数据块复制到低地址物理空间,并将逻辑地址重新定向到对应的物理空间。

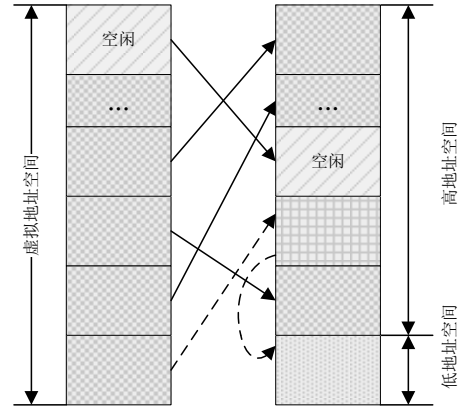


图2 逻辑地址空间到物理地址空间的映射

需要强调两点:首先,动态置换设计存在额外开销,检测和记录粒度的冲突次数需要额外的硬件电路来实现。发生动态置换时,数据的复制和重定向操作会影响存储器的正常访问。其次,粒度的大小和具体应用相关,粒度越大,开销越小,不利于降低冲突概率;粒度越小,开销越大,能够降低冲突概率。用 $f(N)$ 表示粒度大小对冲突概率的影响,用 $g(N)$ 表示粒度大小对硬件开销的影响,静态置换开销为:

$$Cost_{static} = g(N)f(N)$$

另外,为了描述和运算方便,引入局域性因子 $\alpha$ 用于标识一段时间内访问地址集中在存储段的概率。如前所述,动态置换本身会有一些的硬件开销,频繁发生的置换操作会大大降低存储器的访问效率。理论上,局域性因子 $\alpha$ 越高,置换操作的次数越少,动态置换的效率也就越高。但是,实际情况的处理却没有这么简单。如依次执行如下操作:

- 1) 访问0x001000地址100次;
  - 2) 访问0x000001地址100次;
  - 3) 访问0x001000地址100次;
  - 4) 访问0x001000地址100次。
- 如果以每400次访问为采样值计算局域性因子 $\alpha$ ,那么这4组访存操作的 $\alpha=75%$ ,需要进行一次动态置换;但是,如果以每100次访问为动态采样范围计算局域性因子 $\alpha$ ,那么这4组访存操作的 $\alpha=100%$ ,需要进行3次置换操作。显然,以每400次访问为采样值能够取得更好的效果。因此,

真正影响动态置换的效率的应该是局域性因子 $\alpha$ 的计算方式,该方式必须使得结果不能过于灵敏,也不能过于迟钝。鉴于此,引入动态采样范围设计:当置换次数上升时,减弱局域性因子 $\alpha$ 的灵敏度;当置换次数下降时,增强局域性因子 $\alpha$ 的灵敏度。结合静态开销公式,动态置换开销为:

$$\text{Cost}=\alpha g(N)f(N)$$

如果进一步考虑历史动态置换开销统计数据对 $\alpha$ 的反馈作用,当前动态置换开销可进一步细化为:

$$\text{Cost}=k(\text{Cost}_1, \text{Cost}_2, \dots, \text{Cost}_n) g(N)f(N)$$

其中,  $\text{Cost}_x$ 是最近几次动态置换开销统计数据; $k(x,y,z,\dots)$ 表示历史数据对 $\alpha$ 的反馈作用。

### 1.3 置换操作的优先级

多个模块对存储器的访问是并行的,而动态置换操作是串行的,当两个或者两个以上的数据块满足动态置换条件时,优先处理哪个数据块是需要考虑的问题。本文采用先到先服务策略,从低地址到高地址扫描各个数据块,依次完成满足触发条件的各个数据块。实际上,先到先服务策略的灵活性较差,对性能存在一定影响。如解码模块对内存的读写访问明显要多于其他模块,所以更应该优先把冲突概率低的存储空间分配给该模块,这样可以避免冲突带来的性能下降。考虑到实现该功能需要更加复杂的硬件电路,根据语音处理的实际,仍然采用先来先服务策略来保证整个设计运行的合理性。

## 2 实验

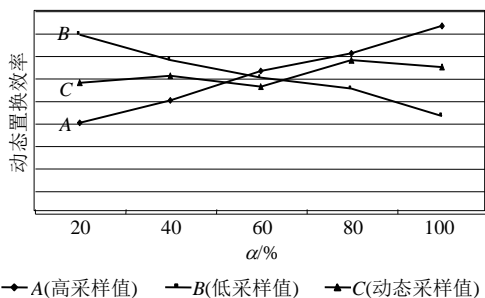


图3 动态采样值和固定采样值性能比较

实验使用Xilinx公司的Virtex-II系列的FPGA芯片<sup>[9]</sup>组成并行访问控制电路,并且评估动态置换算法的效率。图3比较了动态采样值和固定采样值的差异。图中,纵轴表示动态置换的效率,横轴表示平均局域性因子 $\alpha$ 的大小(平均指所有访问的统计值)。曲线A表示高采样值,曲线B表示低采样值,曲线C表示动态采样值。从图中可以看到,当局域性因子 $\alpha$ 很低时,低采样值能够取得较好的性能,高采样值则效率低下;当局域性因子 $\alpha$ 很高时,情况完全相反;

而曲线C无论在何种情况下都能够表现出稳定的访存性能。

## 3 结束语

基于FPGA实现的存储器并行访问控制设计,在应用于互连网络实时语音处理时,低地址的物理存储空间由可配置逻辑模块中的存储资源实现,用于处理单元之间的通信;高地址的物理存储空间由存储阵列实现,用于处理大规模数据时存储单元之间的共享。充分利用FPGA的并行性访问特性,设计了动态置换算法来控制逻辑地址空间到物理空间的映射,将频繁访问且冲突概率较高的数据块映射到低地址的物理存储空间。实验表明,动态置换算法能够实现稳定的并行访存性能。

### 参 考 文 献

- [1] TSAI W-T, WEI Xiao, PAUL R, et al. Service-oriented system engineering (SOSE) and its applications to embedded system development[J]. Service Oriented Computing and Applications, 2007, 1(1): 3-17.
- [2] PETROV P, ORAILOGLU A. Dynamic tag reduction for low-power caches in embedded systems with virtual memory[J]. International Journal of Parallel Programming, 2007, 35(2): 157-177.
- [3] 巢一波, 张曦煌. FPGA和NiosII软核的语音识别系统的研究[J]. 计算机工程与应用, 2010, 46(2): 60-62, 166. CHAO Yi-bo, ZHANG Xi-huang. Speech recognition system on FPGA and Nios II softcore[J]. Computer Engineering and Applications, 2010, 46(2): 60-62, 166.
- [4] 柳永胜, 黄祥伟, 赵武智. 多端口存储器在多机系统中的应用[J]. 单片机与嵌入式系统应用, 2001, 7: 8-12. LIU Yong-sheng, HUANG Xiang-wei, ZHAO Wu-zhi. Application of multi-port memory in multicomputer system [J]. Microcontrollers & Embedded Systems, 2001, 7: 8-12.
- [5] Xilinx Inc. Virtex-II platform FPGAs complete data sheet[G]. 2005.
- [6] MATTAUSCH H J. Hierarchical N-port memory architecture based on 1-port memory cells[C]//Solid-State Circuits Conference, 1997. ESSCIRC'97, Proceedings of the 23rd European. [S.l.]: IEEE, 1997: 348-351.
- [7] WANG Zuo, GU Yi-jun. A hierarchical architecture of N-port memory based on FPGA[C]//International Conference on Convergence Information Technology. Washington, DC, USA: IEEE Computer Society, 2007: 157-161.
- [8] Xilinx Inc. Virtex 2.5 V field programmable gate arrays[G]. 2000.
- [9] GU Yi-jun, WANG Zuo. Mapping N-port memory with dual-port array[C]//World on Computer Science and Information Engineering. Las Vegas: [s.n.], 2008: 444-447.
- [10] MATTAUSCH H J, YAMADA K. Application of port-access-rejection probability theory for integrated N-port memory architecture optimization[J]. Electronics Letters, 1998, 34(9): 861-862.

编辑 漆蓉