◎产品、研发、测试◎

针对嵌入式系统的存储器管理单元设计

朱贺飞,陆 超,周晓方,闵 昊,周 电
ZHU He-fei,LU Chao,ZHOU Xiao-fang,MIN Hao,ZHOU Dian

复旦大学 专用集成电路与系统国家重点实验室,上海 201203
ASIC & Systems State Key Laboratory,Fudan University,Shanghai 201203,China
E-mail:hfzhu@fudan.edu.cn

ZHU He-fei,LU Chao,ZHOU Xiao-fang,et al.Memory management unit design for embedded system.Computer Engineering and Applications,2007,43(1):96-99.

Abstract: To support Linux operating system,a high efficiency memory management unit for a 32-bit embedded processor is proposed in this paper.The added pre-comparing circuits can greatly improve the efficiency of address translation when the processor needs to access the same virtual address page.A TLB auto refilling mechanism by hardware is also adopted to improve the speed of physical address fetching when TLB miss.Experiments show that RSIC embedded processor with MMU can run Linux perfectly.

Key words: embedded system;memory management unit;Linux TLB

摘 要:针对 Linux 操作系统,实现了面向 32 位 RSIC 嵌入式处理器的存储器管理单元。通过在指令快表中增加预比较电路,提高了处理器连续访问同一虚拟页面时的地址转换效率。快表失效时,设计了专门的硬件来实现页表查询及快表填充,处理速度明显优于软件。论文设计的 MMU 能够很好地和 Linux 配合,完成地址映射及存储权限管理。

关键词:嵌入式系统;存储器管理单元;快表
文章编号:1002-8331(2007)01-0096-04 **文献标识码**:A **中图分类号**:TP33

1 引言

随着移动通信、移动计算的普及,嵌入式系统已经越来越深入人们的生活。许多现实应用要求在嵌入式处理器上运行的操作系统能够支持多进程处理。例如 Windows Mobile 以及嵌入式 Linux,这些操作系统可以使处理器同时运行多个进程,并在不同进程之间切换。

运行支持多进程的操作系统需要为处理器在硬件上增加一个存储器管理单元(MMU,Memory Management Unit)来配合操作系统实现虚拟内存和物理内存的转换[1]。由于虚拟内存和物理内存的转换工作是软硬件协同完成的,因而在设计 MMU 时就要充分考虑操作系统的内存管理机制,使 MMU 能很好地支持操作系统。本文针对 Linux 的内存管理机制,为家庭网络核心 SOC 平台中使用的 32 位嵌入式 RISC 处理器[2]设计了支持虚、实地址转换和存储权限控制的 MMU。为了提高处理器访问指令和数据的效率,MMU 中包括了彼此独立的指令快表(ITLB)和数据快表(DTLB),用来实现地址映射关系的存储和比较。在实现 MMU 基本功能的基础上,通过在指令快表中增加指令预比较电路来减小对指令快表的访问频率,提高地址转换的效率;快表失效时,设计了专用硬件来处理对两级页表的访问,不需要操作系统的干预。

2 家庭网络SOC 平台及 Linux 的内存管理体系

家庭网络 SOC 平台中使用的嵌入式系统架构如图 1 所示,包括自行设计的 32 位 RSIC 处理器核、MMU 、Cache 以及片外主存。处理器给出的虚拟地址经过 MMU 进行地址映射后得到物理地址,用来访问 Cache 或主存。处理器中运行的 Linux 操作系统采用页式内存机制,内存空间分为内核(Kernel)空间和用户(User)空间,处在内核空间的地址使用线形映射的方法,处在用户空间的地址使用页表映射的方法。

本文设计的 MMU 支持两级页表,页表存放在片外主存中,页面大小为 4 Kbyte。每个页表项中有一个 Present 标志位,表示该页面是否在内存中;一个 Writable 标志位,表示该页面是否可写[3]。

3 MMU 硬件电路结构及工作原理

如图 1 所示,MMU 由控制逻辑、指令快表和数据快表三部分组成。控制逻辑采用 Verilog 硬件描述语言实现,主要负责对地址映射的时序、访问权限进行控制,同时负责向处理器给出异常请求。指令快表和数据快表采用 32 路全相联形式[4],分别可以存储 32 个虚拟地址和物理地址的映射关系。虚拟页号(虚拟地址高 20 位)和物理页号(物理地址高 20 位)分别存贮在快

基金项目:国家 863 高技术研究发展计划资助项目(2003AA1Z1120);上海市科委 SDC 资助项目(037062020)。
作者简介:朱贺飞(1982-),男,硕士研究生,主要研究方向为专用集成电路设计;周晓方,男,高级工程师,主要研究方向为微处理器设计;周电,男,博士生导师,主要研究方向为高速集成电路设计、计算机辅助设计等。

表(TLB)的 CAM[5](Content Addressable Memory)阵列和 SRAM 阵列中。当 Linux 运行至用户空间,访问的虚拟地址需要进行页表映射时,处理器给出的虚拟页号将同 CAM 阵列中存储的虚拟页号比较,如果有一路命中就将相应一路 SRAM 单元中的物理页号读出,与虚拟地址的低位拼接在一起形成物理地址[6],虚拟地址的页面映射过程如图 2 所示;如果快表中的虚拟地址没有命中(称之为快表失效),MMU 就会查询主存中的两级页表,找出映射关系,填入 TLB 中,再用取得的物理地址访问 Cache。

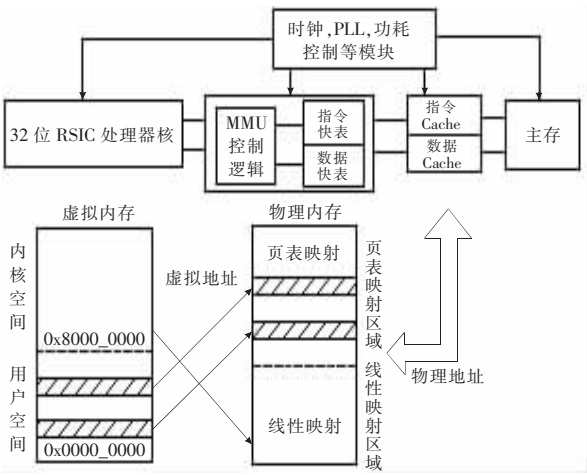

图 1　MMU 的功能及其在系统中的位置

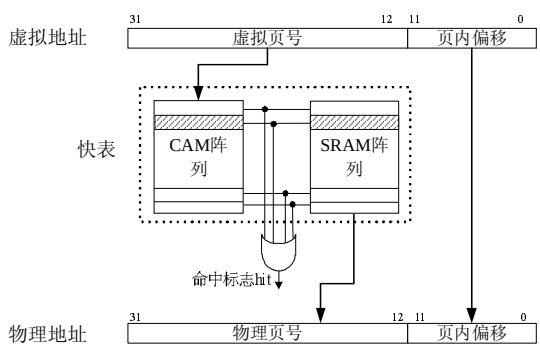

图 2　页面映射原理图

4　高效率的指令快表设计

传统结构的指令快表需要处理器在每次访问指令空间时都进行虚拟页号的比较,判断 TLB 是否命中,而存储虚拟页号的 32 路 CAM 阵列进行全并行比较不仅效率低下,还将带来大量的功耗损失[7]。根据程序的临近性原理,60%-80%的指令都是顺序执行的[8]。当指令顺序执行时,处理器给出的指令地址顺序增加,并且很有可能处于同一个虚拟页面的范围内(虚拟地址高 20 位相同)。这时完全可以省略指令快表的查找,直接利用上一次访问的结果取得物理页号,这将在很大程度上提高 MMU 进行地址转换的效率,减少进行 32 路 CAM 并行比较的次数。文献[9]采用了一种称为 Filter TLB 的结构,实际是一种两级 TLB 的结构。Filter TLB 命中时可以省略对 main TLB 的查找,当 filter TLB 具有较高命中率时就可以在很大程度上提高地址映射的效率。但这种方法将两级 TLB 工作在串行模式下,Filter TLB 失效时还需访问第二级 TLB,会给整个处理器的关键路径增加延时。

本文设计的 ITLB 充分考虑了连续访问同一虚拟页面对于提高地址转换效率的积极作用,它的结构如图 3。ITLB 包括用于存储虚拟页号的 CAM 阵列,用于存储物理页号的 SRAM 阵列,用于保存输入数据、记录每次访问的虚拟页号和物理页号的锁存器,以及一个比较器和一个多路选择器。ITLB 首先对每一次访问的虚拟页号和读出的物理页号都进行锁存,当处理器给出新的虚拟地址时,通过比较器来判断前后连续两次访问的虚拟页号是否一致。如果比较器比较命中,Hit_previous 信号置高,使 CAM 比较使能信号 CAM_EN 无效,不再进行 ITLB 查找,避免了 32 路 CAM 阵列的并行比较以及对 SRAM 阵列的读取,直接从锁存的结果中取得物理页号。由图 3 可以看出,只有当指令预比较电路与快表的输入信号锁存及时序控制部分并行工作时,才能避免对系统关键路径的负面影响,这就对指令预比较电路中使用的比较器提出了苛刻的设计要求。本文采用中芯国际 0.18 μm 集成电路工艺对 20 位的指令预比较电路进行全定制设计,仿真结果表明,所采用的动态比较电路结构[10]完成一次比较的最大延时(输入的虚拟页号和锁存的虚拟页号只有一位不同)为 0.43 ns,完全能够将指令预比较的工作与

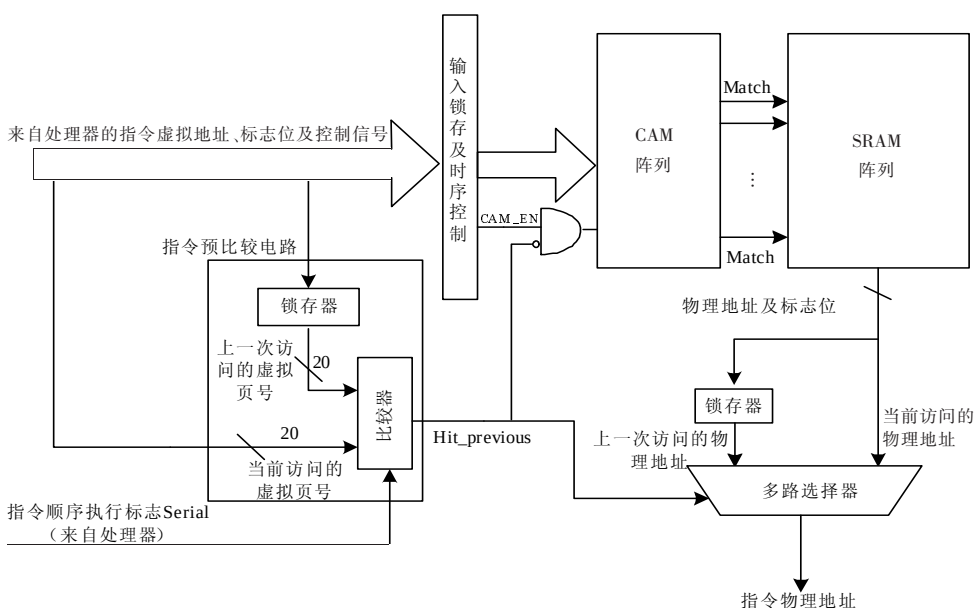

图 3　指令快表电路结构

输入地址锁存并行进行,不再引入额外延时。从图 3 中还可以看出,若处理器执行的指令并非顺序执行,Serial 信号为低,使指令预比较电路停止工作,不会引入任何额外功耗。

本文对 Linux 操作系统运行过程中的指令进行统计,图 4 比较了 MMU 对传统结构的指令快表和本文设计的高性能指令快表各自所需的访问次数。图中同时给出了文献[9]中介绍的 selective filter-TLB 运行若干种测试程序时 filter-TLB 的命中情况(若 filter-TLB 命中,则可以省略对 main TLB 的查找,其平均命中概率为 80%-90%)。由于处理器连续访问同一个虚拟页面的概率非常大,而本文设计的指令快表和文献[9]中的快表都只有在虚拟页号变化时才进行 TLB 的访问,因而具有相似的访问频率,都能在较低的访问次数下高效地完成地址映射的任务。表 1 为三种结构的快表平均所需访问时间的比较,本文指令快表中的指令预比较电路与输入信号锁存及时序控制的过程并行进行,不会为地址映射增加延时,故平均访问时间与传统结构相同。文献[9]的 selective filter-TLB 在 filter TLB 失效时还必须访问 main TLB,故其平均访问时间会比传统结构稍大。

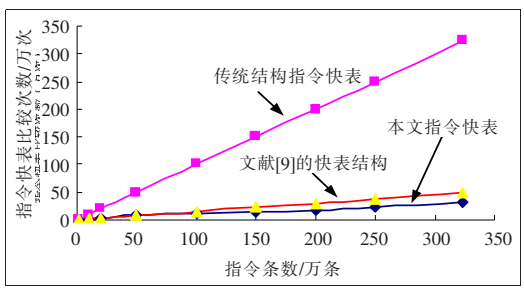

图 4 指令快表访问频率比较

表 1 三种结构快表平均访问时间比较(时钟周期)

	传统结构	本文结构	文献[9]
访问时间	1.0	1.0	1.2

5 快表失效时的地址转换

MMU 对处理器给出的虚拟地址进行映射时,若发现所访问的虚拟页号还未在快表中存储,就发生了快表失效。针对快表失效通常有两种处理方法:一种是 MMU 发出异常请求后,由操作系统来查询页表并填充 TLB;另一种是在 MMU 中设立专门的硬件,直接完成页表查询和 TLB 填充。两种方法各有优劣:由操作系统查询页表的方案硬件结构简单、关键路径短、灵活性好,但是效率低下,很多 RISC CPU 使用这种方案(如 MIPS 4Ke 系列 CPU[11]);而硬件查询页表的方案中硬件结构较复杂、需要和操作系统有很好的配合,但效率很高,CISC CPU 大多使用这种方案,其中最著名的就是 x86 体系的 CPU[1],另外一些高端的 RISC CPU 也会使用这种方案如 ARM11 MPCore Processor[12]。为了在快表失效时以较高的效率从主存中查询两级页表取得物理地址,本文采用了硬件直接查询页表的方案。

图 5 介绍了快表失效时由硬件实现两级页表查询的过程。首先将处理器给出的 32 位虚拟地址分为第一级偏移(31 至 22 位)、第二级偏移(21 至 12 位)和页内偏移(低 12 位)3 部分。MMU 从处理器中的专用寄存器 PGD_base 取得第一级页表基址,与第一级偏移拼接,访问主存后得到所需的第二级页表基址;再用第二级页表基址和第二级偏移拼接,再次访问主存,取得物理页号;最后用物理页号与页内偏移拼接得到最终的物理地址,用来访问处理器需要的指令或数据。如图 5 所示,MMU 每次进行主存访问时,都会对所访问地址的读写权限进行检测,若发生越权访问的情况,将向处理器发出异常请求。查询两级页表的过程结束以后,MMU 还会将取得的地址映射关系填入快表中,以备下次访问同一虚拟页面时使用。快表失效时的整个地址映射过程完全由 MMU 中的硬件电路实现,不需要操作系统的干预。本文采用 Mentor Modelsim v5.8 对处理器运行 Linux 操作系统的过程进行仿真,图 6 为一次数据快表失效时采用硬件查询两级页表的仿真波形。此时处理器的程序指针 PC 为 0x0042ad64,当前指令 0xac430000 是一条对外存数据空间的 store 操作,所访问的数据空间虚拟地址为 0x10004634,在 DTLB 未命中的情况下,经过两级页表查询后该虚拟地址被映射为物理地址 0x00445634。表 2 比较了分别采用硬件和软件来完成页表查询所需的时钟周期数。可见,无论是指令快表失效还是数据快表失效,采用硬件完成上述操作所花费的时钟周期都明显少于软件,能够显著改善页表查询的效率。

表 2 硬件与软件完成页表查询操作效率比较(时钟周期)

	硬件	软件
数据快表失效	20	1 189
指令快表失效	17	1 014

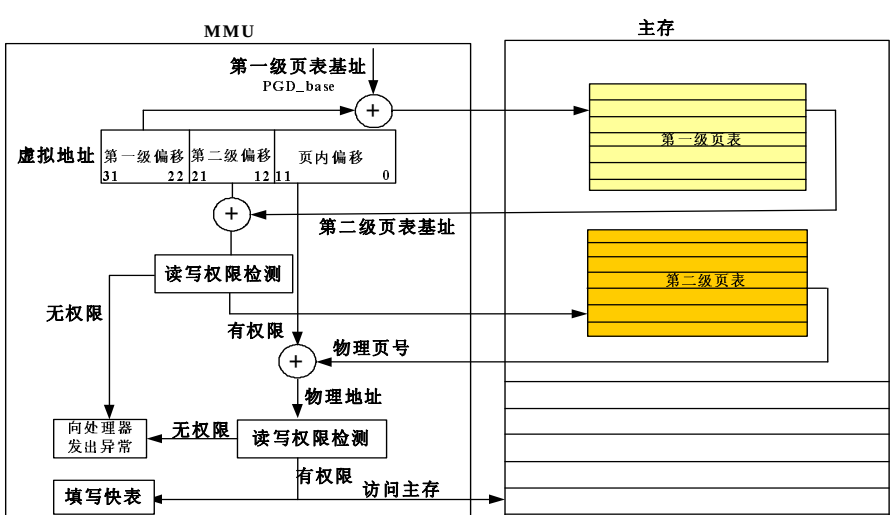

图 5 硬件实现两级页表访问

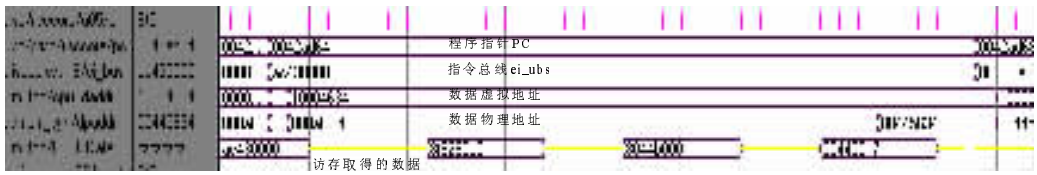

图 6　硬件实现页表查询仿真波形

6　Linux 的移植与验证

本文采用的 Linux 内核版本为 MIPS 公司提供的 Linux-2.4.18。虽然 MMU 是针对 Linux 的内存管理机制而设计的,但由于所采用的处理器是一款自行设计的 RISC Core,因而在一些细节上需要对 Linux 进行移植:

(1)在启动或激活一个进程时,要将该进程的页表目录基址填入 CPU 提供的专用寄存器(PGD_BASE Register)。

(2)切换进程时要调用 TLB flush 指令将 TLB 中的数据全部置为无效。

(3)发生 MMU 异常时从 CPU 提供的专用寄存器(Bad_Address Register)中获取发生异常的地址。

本文采用如图 7 所示的 FPGA 验证平台来测试所设计的 MMU 对 Linux 的支持。将整个嵌入式处理器及 MMU 的 Verilog 代码下载到一块 Altear 公司的 EP1S80B956C7 型号 FPGA 中,MMU 中的全定制 TLB 电路以及处理器原有的 Cache 电路都采用 RTL 级模型代替。用一片 256 Mbit 的 SDRAM 来实现嵌入式系统主存,其中存放了 Linux 的内核代码及部分应用程序,Linux 的 boot 程序则放在 4 片 Flash 中, 上电后由处理器自动读取。使用并行口将 Linux 运行中的各种信息从处理器发送到 PC 机上的虚拟终端模型, 使 Linux 输出的信息以文本的形式显示,方便调试。

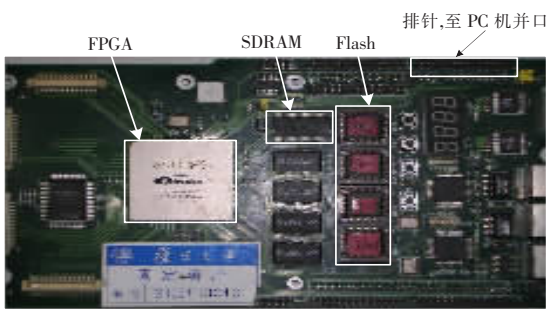

图 7　FPGA 验证平台

图 8 是家庭网络核心 SOC 平台运行 Linux 时,从并行口发送到 PC 机的一段输出信息,从中可以看到,Linux 内核已经正常启动并开始运行。从 Linux 的输出信息中可以看到其内核版本和 gcc 编译器版本, 本文采用的 RSIC 嵌入式处理器还包括了独立的指令 Cache 和数据 Cache。

7　结论

针对 Linux 操作系统的内存管理机制,本文为家庭网络核心 SOC 平台中使用的 32 位嵌入式处理器设计了一款能够支持虚、实地址转换及存储权限控制的存储器管理单元。通过在指令快表中增加指令预比较电路,以及采用硬件自动实现快表失效时的页表访问, 极大地提高了 MMU 进行地址转换的效率。本文硬件设计中的两项核心技术已经申请专利(专利申请号:200410067586.X 和 200510026403.4)。验证结果表明,本文设计的 MMU 能够高效地支持 Linux 的运行。

(收稿日期:2006 年 8 月)

```
CPU revision is:97906c
...
Linux version 2.4.18-MIPS-01.04 (hfzhu@hnet)(gcc version 2.96-sdelinuxmips-040127)#1 Wed Oct 5 20:06:50 CST 2005
homenet_setup()starts.
...
Memory:23520k/32768k available (564k kernel code,0k reserved,0k data,0k init,0k highmem)
...
Primary instruction cache 64 Kb,linesize 256 bits(4 ways)
 Primary data cache 64 Kb,linesize 256 bits(4 ways)
...
RAMDISK driver initialized:16 RAM disks of 4096K size 1024 blocksize
<5>RAMDISK:ext2 filesystem found at block 0
...
VFS:Mounted root(/sbin/init.e.2 filesystem)
...
```

图 8　Linux 运行时部分输出信息

参考文献:

[1] Bovet D P,Cesati M.Understanding the Linux kernel[M].2nd ed.O'Reilly Media,Inc,2002-12.

[2] 李侠,周晓方,张海清,等.可自适应变频嵌入式微处理器核的设计[J].小型微型计算机系统,2006,27(2):335-338.

[3] 毛德操,胡希明.Linux 内核源代码情景分析[M].杭州:浙江大学出版社,2001:38-39.

[4] Hennessy J L,Patterson D A.Computer architecture:a quantitative approach third edition[M].San Francisco:Morgan Kaufmann Publishers,2003.

[5] Rabaey J M,Chandrakasan A.Digital integrated circuits:A design perspective[M].2nd ed.Upper Saddle River,New Jersey:Pearson Education,2003:670-672.

[6] Juan T,Lang T,Navarro J J.Reducing TLB power requirements[C]//International Symposium on Low Power Electronics and Design 18-20 Aug 1997,c1997:196-201.

[7] Hsiao H Y L,Wang D H,Jen C W.Power modeling and low-power design of content addressable memories [J].Circuits and System,2001,4:926-929.

[8] Clark L T,Byungwoo C H.Reducing translation lookaside buffer active power [C]//Low Power Electronics and Design:ISLPED'03,2003:10-13.

[9] Lee Jung-Hoon,Park Gi-ho.A selective filter-bank TLB system [embedded processor MMU for low power][C]//ISLPED'03,2003:25-27.

[10] Edmondson J H.Impact of physical technology on architecture [M]//Chandrakasan A,Bowhil W,Fox F,eds.Design of High-Performance Microprocessor Circuits.New York:IEEE Press,2001:7-8.

[11] MIPS32™ architecture for programmers volume III:the MIPS32™ privileged resource architecture.Rev.0.95.MIPS Technology,2001-03-12.

[12] ARM11 MPCore™ Processor Revision:r0p2 Technical Reference Manual.ARM,2005.