

一种支持多级存储架构的嵌入式多核操作系统模型^①

郭海林, 陈香兰

(中国科学技术大学 计算机科学与技术学院, 合肥 230027)

摘 要: 多核处理器已经成为主流, 并且被广泛应用于嵌入式设备中. 在操作系统如何有效支持多核处理器方面的研究中, 目前国内外大多基于常见的紧耦合共享存储架构的多核处理器, 而对一些特殊存储架构的多核处理器研究并不多. 本文针对内存受限的多级存储架构的多核处理器, 提出一种单代码多数据的嵌入式多核操作系统模型. 实验表明, 该模型应用在具有多级存储架构的八核 DSP 上, 比 AMP 模型减少约 80% 的代码空间开销; 与 SMP 模型相比, 与实时性紧密相关的时间开销减少约 10 倍.

关键词: 多核处理器; 多核操作系统; 操作系统模型; 多级存储

Embedded Multicore Operating Systems Model for Multi-Level Memory Architecture

GUO Hai-Lin, CHEN Xiang-Lan

(School of Computer Science and Technology, University of Science and Technology of China, Hefei 230027, China)

Abstract: Multicore processors have already become mainstream and are being widely applied in embedded devices. Current research on how operating systems support multicore processors has focus on tightly coupled and shared memory architecture, which is the most common architecture for multicore processors, but ignores other special memory architectures. In this paper, a new embedded multicore operating system model based on a single code and multiple data is proposed for memory-constrained and multi-level memory architectures. Experiments show that, when applied to an eight-core DSP based on a multi-level memory architecture, this model can efficiently reduce the code space overhead by 80% compared to AMP; and the time cost related to real-time is ten times lower than that on SMP.

Key words: multicore processors; multicore operating systems; operating systems model; multi-level memory

微处理器自诞生以来, 处理器的主频不断提升, 但通过提高主频来提升处理器性能, 随之带来的功耗问题和散热问题也日益凸显, 这些问题在嵌入式系统中表现尤为明显. 在这样的背景下, 多核嵌入式处理器逐渐引起了人们的关注, 设计多核处理器成为提升处理器性能、满足不断提升的应用计算需求的主要途径.

多核处理器^{[1][2]}是将两个或多个独立的处理器核封装在一个芯片内部, 从而能以较低的主频获得较高的处理器性能. 根据多个处理器核是否相同, 可以将多核处理器分为同构多核处理器(Homogeneous Multicore Processor)和异构多核处理器(Heterogeneous

Multicore Processor)^[3], 前者单个芯片内多个核完全相同、地位也对等, 后者则是将不同处理特性的核集成在一个芯片上. 在硬件方面, 多核技术已成为芯片设计的主流技术, 并已被广泛用于嵌入式 DSP 芯片设计^[4]. 但在软件方面, 多核软件的发展仍远远落后于多核处理器的发展^[5,6].

作为用户应用与硬件的中间接口, 操作系统对多核处理器的有效支撑是合理利用多核处理器性能优势的关键. 鉴于此, 近年来支持多核架构的操作系统的研究已成为热点. 目前, 多核操作系统在资源管理、核间通信与同步、代码迁移等问题上面临的诸多挑战^[7-10].

① 基金项目: 国家“核高基”重大专项基金(2012ZX01034001-001); 国家自然科学基金(61379040, 61272131)

收稿时间: 2015-03-31; 收到修改稿时间: 2015-05-07

解决上述问题的复杂度取决于操作系统的多处理模型,即操作系统是如何支持多核处理器的。常见的操作系统模型有三种:非对称多处理(Asymmetric Multiprocessing, AMP)、对称多处理(Symmetric Multiprocessing, SMP)以及限定多处理(Bound Multiprocessing, BMP)^[5,6,11]。然而这三种模型都是基于紧耦合、共享存储的多核处理器提出的,并不适用于一些特殊存储架构的处理器。本文研究的出发点来自于一款嵌入式多核 DSP,这是一个四簇、每簇双核、共八核的处理器。该多核 DSP 没有采用纯粹的共享存储架构,而是每个核均拥有自己的私有数据空间、簇内拥有共享数据空间,同时八核共有全局代码空间和全局数据空间。对于这种拥有复杂存储架构的多核处理器,在选择操作系统的多处理模型时除了考虑上述热点问题外,还需考虑各级内存的空间限制、大小与访问速度差异。

本文提出一种单代码多数据的嵌入式多核操作系统模型,该模型适用于内存受限的多级存储架构的多核处理器。本文首先介绍了三种传统的多核操作系统模型,指出每种模型的优缺点以及适用场景;接着描述多级存储架构的多核 DSP 硬件特征,分析传统的操作系统模型并不能很好地利用在该 DSP 上;针对该 DSP 体系结构,提出一种单代码多数据的多核操作系统模型,通过实验比较分析了三种多核操作系统模型在占用内存以及实时性上的表现,验证了本文提出的新模型的正确性。

1 嵌入式多核操作系统模型

开发者通常根据硬件平台的特征以及应用的需求来选择合适的操作系统模型,这将决定将单核代码移植到多核平台上以达到最大的并行性的难度。本节将介绍三种常见的多核操作系统模型,并总结各自的优缺点。

1.1 非对称多处理 AMP

AMP 模型的主要特点是每个核都运行一个独立的操作系统,应用被分配到特定核上运行,如图 1 所示为非对称多处理模型的架构图。

AMP 模型的主要优点为:

1)方便操作系统和应用的开发。在 AMP 模型下操作系统和应用都被指定到特定的核上运行,执行环境与单核类似。不仅将原有的单核应用程序移植到多核

平台非常简便快捷,同时将已存在的单核操作系统改造成该种模型的多核操作系统也相对容易。

2)支持异构多核的环境。嵌入式设备具有专用性,通常将通用的处理器和专用处理器集成到一块芯片上,形成异构多核架构,AMP 模型为每个处理器部署适合的操作系统并在其上运行特定的应用,能充分发挥通用处理器和专用处理器的优势,因此十分适合这种场景。

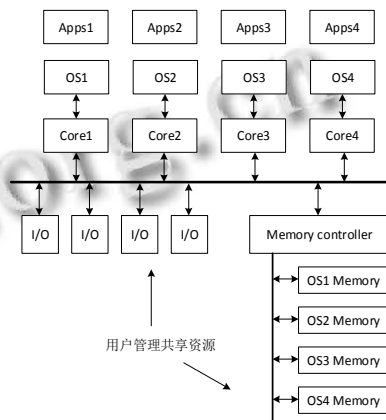


图 1 非对称多处理模型

AMP 模型的主要缺点是:

1)资源分配对用户不透明。开发者在 AMP 模型下需要为每个核手动分配硬件资源,这对开发者要求较高,需要开发者能掌握每个任务消耗的资源,合理地为其进行分配处理器核和其它硬件资源。对于紧耦合的应用来说,当该应用需要不超过两个核的情况下,AMP 模型还可以很好地适用,一旦需要更多的核时,其设计复杂度就呈指数级别增加。

2)不利于充分利用处理器资源。如果某个核负载过大,它上面的应用并不能轻易地转移到其它空闲的核上运行。

1.2 对称多处理 SMP

SMP 模型的主要特点是所有核运行同一个操作系统实例,如图 2 所示,应用程序在操作系统的统一调度下可以在任何一个核上运行。

SMP 模型的主要优点为:

1)系统资源利用率高。SMP 模型下所有核运行同一个操作系统映像,所有进程、资源的调度和管理以及核间通信均由该操作系统负责,操作系统可以动态地为每个应用分配资源,可以以较少的代价实现负载均衡,可以大大提高系统资源的利用率。

2)SMP 模型具有较好的可扩展性。增加或减少处

理器核的个数,对操作系统的部署及用户编程影响并不大。

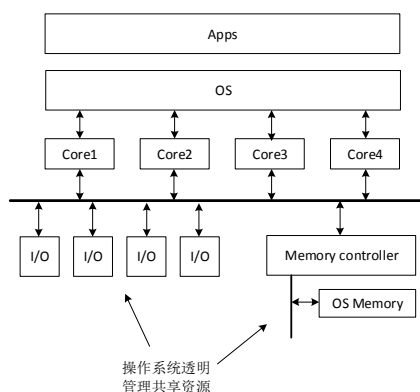


图 2 对称多处理模型

相比 AMP 模型来说, SMP 模型的缺点主要有:

1) 不支持异构平台。SMP 模型所有核运行同一操作系统映像, 因此只能支持同构平台。

2) 并发问题格外突出。SMP 模型下多核共享系统资源, 无论是操作系统还是用户程序, 在编程时都需要考虑多核并发访问有可能导致的同步问题。鉴于此, 将原有的单核代码移植到 SMP 环境下时, 通常需要对软件进行重新设计, 否则可能造成程序运行出错。同样, 如果将已存在的单核操作系统改造为适合 SMP 模型的多核操作系统, 难度也比较大。

3) 除此之外, SMP 模型还有例如 Cache 抖动、非确定性、潜在的竞态条件等等缺点。

1.3 限定多处理 BMP

BMP 模型在 QNX 软件系统^[12]中首先被提出, 这种模型有着 SMP 模型的透明资源管理的优势, 但又提供给开发者将某个应用绑定到特定核上的能力, 这对于将原有的单核代码移植到多核环境中很有帮助。

相比 SMP 模型来说, BMP 模型具有以下优势:

1) 允许为单核环境而编写的代码无需经过修改就可以移植到多核并发环境下正确运行;

2) 可以将共享同一数据集的应用绑定到同一个核上, 从而消除 SMP 中降低性能的 cache 抖动现象;

3) 由于允许一个应用的所有线程在同一个核上运行, 所以 BMP 提供了比 SMP 更简单的调试。

2 面临的问题

本文所涉及的多核 DSP 处理器是某所研制的具有

自主知识产权的高端 DSP 芯片, 可广泛应用于各种高性能计算领域, 如雷达、精度制导、通信保障、图像处理等信号处理领域。该多核 DSP 在单个芯片上集成了四个 DSP 簇, 而每个簇又由两个独立的 DSP 核构成。

多核处理器的内存架构可分为集中共享式存储架构和分布式存储架构, 它们在扩展性、访存带宽、多核通信开销等各有优缺点, 近年来基于分布式存储器架构的簇状内存架构也获得学术界广泛关注^[13]。针对数字信号处理应用的特殊需求, 该款多核 DSP 结合了多种多核内存体系架构的特点, 如图 3 所示(为了简化起见, 此处只画了两个簇)。

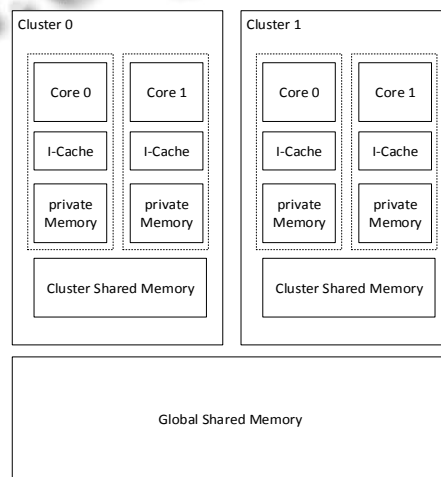


图 3 多核 DSP 内存体系结构简图

从图 3 中可以看出, 与紧耦合共享存储的体系架构处理器相比, 该款多核 DSP 除了拥有全局共享存储器之外, 对于每个核还拥有自己的私有主存以及簇内共享存储, 构成了三级存储的内存体系架构, 而且越靠近处理器, 内存体积越小, 访存速度越快。

DSP 需要较大的运算量和较高的运算速度, 为提高数据吞吐量, DSP 内存通常采用哈佛结构, 即数据与代码存储分开, 该款多核 DSP 也采用了哈佛结构。

从地址空间来看, 每个 DSP 核均拥有 4G 的访存能力, 但分为三个部分:

1) 核私有地址空间

每个核都有自己的私有地址空间, 如图 4 中浅灰色部分表示, 私有空间大小为 18M, 从 0x00200000 到 0x013FFFFFF, 主要用于映射核寄存器、中断及 DMA 寄存器以及数据存储器等。每个核的私有地址空间相

互独立,即使地址相同,实际上指向不同的物理存储。

2) DSP 簇内共享地址空间

DSP 簇内共享地址空间如图 4 中深灰色部分表示,地址范围从 0x01400000 到 0x093FFFFF,共 128M,主要用做簇内两个核的共享数据存储器,簇间独立,即其它簇上的核不能跨簇访问。

3) DSP 全局共享地址空间

DSP 全局共享地址空间分为两段,如图 4 中白色表示的部分,第一段从 0x00000000 到 0x01FFFFFF,共 2M,是八个 DSP 核的共享程序存储空间;第二段从 0x09400000 到 0xFFFFFFFF,是八核共享的片外 SRAM 存储空间和 DDR3 存储空间等。

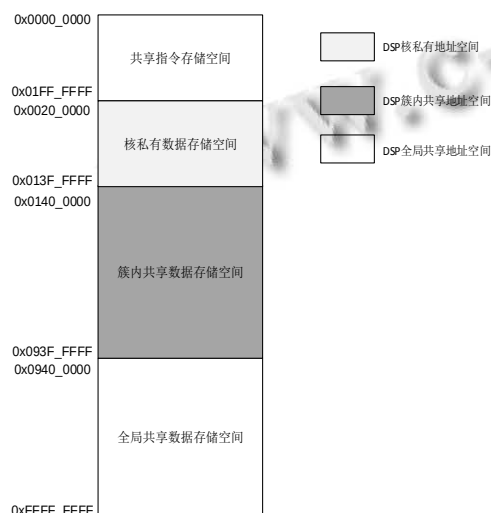


图 4 多核 DSP 地址空间示意图

上述常见的三种操作系统模型均不太适用于此多核 DSP 的存储架构,原因如下:

1)若采用 SMP 模型,为共用一个操作系统映像,必须将操作系统、应用程序及数据等全部部署到访问速度较慢的全局共享空间上,无法发挥核私有空间和簇内共享空间的访问性能优势,无疑会大大影响系统的实时性及整体性能。

2)若采用 AMP 模型,则对于四簇八核的 DSP 来说就需要八个操作系统映像,对于只有 2M 字的全局共享的代码空间来说,每个核只能平均分配到 256K 字代码,对于复杂的用户应用或使用 libc 库的应用,代码空间将会严重不足。

3)BMP 模型属于一种特殊的 SMP 模型,因此也不适用该多核 DSP 的体系结构。

综上,急需设计一种适用于这种复杂的多级内存架构的、新的多核操作系统模型,使其既能充分利用近核端的访存速度优势,又能兼顾全局、大小受限的代码空间限制。

3 设计与实现

从操作系统代码和数据的耦合度来看,SMP 模型只有一份操作系统代码和数据;AMP 模型则有多份操作系统代码和数据。前者易形成核间的竞争,会引起同步与互斥等问题;而后者对于内存受限的平台来说可能一方面存在代码空间不足、另一方面存在代码空间浪费的问题。针对同构多核处理器平台,为了适应多级内存体系结构,本文提出一种单代码多数据(Single-Code and Multiple-Data, SCMD)的操作系统模型。该模型通过将代码和数据分开,提供单一代码映像和各核私有数据映像,在对 AMP 和 SMP 模型各有继承的基础上相互融合,既能将各核代码统一存储到大小有限的全局共享代码空间上,又能将各核数据分别存储到核私有空间上以充分利用其性能优势。

3.1 基于单代码多数据的多核操作系统模型

经分析,在为本文所涉及的多核 DSP 选择多核操作系统模型时,遇到的问题是:

1)程序存储空间受限,无法像 AMP 模型那样为每个核提供一个操作系统映像;

2)全局共享地址空间访存速度最慢,若像 SMP 模型那样部署,则无法利用近核端的数据访存性能优势。

为兼顾上述两个方面的问题,本文考虑在 AMP 模型的基础上,将每个核的操作系统代码和数据分离开来,并将所有核的操作系统代码合并为一个操作系统映像,从而得到一个统一的操作系统代码映像和每核独立的数据映像,然后将它们分别部署到全局共享代码存储空间和核的私有数据存储空间上。从操作系统代码的角度看,新模型类似 SMP 模型,代码的合并减少代码空间的开销;而从数据的角度看,新模型又与 AMP 相似,可以利用核私有数据存储空间的独立性和访存性能优势。

将操作系统数据映像存放在核私有数据存储空间主要基于以下两点考虑:

1)核私有数据存储空间的访存速度最快,减少了上下文切换、中断响应等时间,提升系统的整体性能,增强操作系统的实时性。

2)核私有数据存储空间的独立性,即地址区间相同,但物理存储不同,为数据独立提供了可行性.各核代码统一到唯一的操作系统代码映像中后,受到链接的影响,各核在运行内核代码时对全局变量,例如就绪队列的访问地址也是一样的,但由于空间独立,各核访问相同地址得到的内容却是不一样的,从而保证了各核数据的独立性.这种空间独立性只有核私有数据存储空间具有,簇内共享地址空间和全局空想地址空间都不具有.

在用户应用上,也要考虑将代码统一链接到共享代码存储空间并将数据与之相分离.只不过不同核的应用代码并不合并.用户应用数据存放可以比较灵活,在数据存储空间还足够的情况下,将访问频率较高的数据优先存放在核私有数据空间上,其次是簇内共享数据空间,最后才是全局共享数据空间.但需要注意的是,除了共享内存之外,簇内不同核的应用的数据在簇内共享地址空间中存放地址不能重叠,不同核的应用数据在全局共享地址空间中存放的地址也不能重叠.对用于核间任务通信的共享内存,如果是簇内核间任务通信,优先在簇内共享内存上分配内存;如果是跨簇核间任务通信,则只能在全局共享内存上分配内存.

3.2 多核链接技术

为了实现上述代码和数据的分离和统一,需要正确的安排代码空间和数据空间,这需要解决多核链接技术.这里,多核链接是指将各核的操作系统代码映像和数据映像正确链接到全局共享程序空间和核私有空间中.该多核 DSP 平台现有的 C 语言编译器是基于开源编译器 GCC 开发的,已成功应用在单核平台上.但要支持上文所设计的单代码多数据的操作系统模型,还需要修改.

由于每个核上部署的应用不同,它们对操作系统的功能配置需求也不一样,因此每个核单独编译出来的操作系统映像是不一致的.具体而言体现在如下两个方面:①功能集合不一样.例如有的核需要操作系统提供消息通信机制,而有的核需要操作系统提供共享内存机制;②由于代码集合不同导致数据映像也不同.多核链接需要将不同核上操作系统的代码段合并去重并将数据段符号表进行归一化,以保证操作系统全局变量的地址保持一致.

下面以操作系统全局变量 `Clock_ticks` 地址为例来

说明多核链接的必要性. `Clock_ticks` 用于记录核上时钟中断产生的个数,每次时钟中断触发,在中断服务程序中将该变量加 1,对应的 C 语言代码以及编译之后的 `core0` 的汇编代码如下:

C 语言:

```
Clock_ticks += 1;
```

汇编代码:

```
[00001181]: U6 = __Clock_ticks: 0x0040002b
[00001183]: XR10 = [U6+0, 0]
[00001185]: XR11 = 0x00000001
[00001187]: XR10 = XR10 + XR11 (U)
[00001188]: [U6+0, 0] = XR10
```

上面汇编代码中加粗代码为取得 `Clock_ticks` 全局变量的地址,而该地址是在链接过程中进行符号解析时确定的.如果多个核的操作系统映像对该变量的链接地址不一致并且不加处理,则多个核分别对访问该变量时就会出现错误.本例中在编译 `Core0` 和 `Core1` 的操作系统映像时, `Clock_ticks` 变量地址分别是 `Core0: 0x0040_002b` 和 `Core1: 0x0040_0032`,那么 `Core1` 按照上述 `Core0` 代码片段中加粗的代码运行时就会错误引用到 `Core1: 0x0040_002b` 上而非 `Core1: 0x0040_0032`.

为此,有必要进行多核链接以合并并统一各核操作系统数据段的符号地址,使不同的核在运行相同的操作系统代码时,对全局变量的引用是同一个地址.

图 5 展示了各核编译和多核链接的过程,简化起见,只画了两核链接.

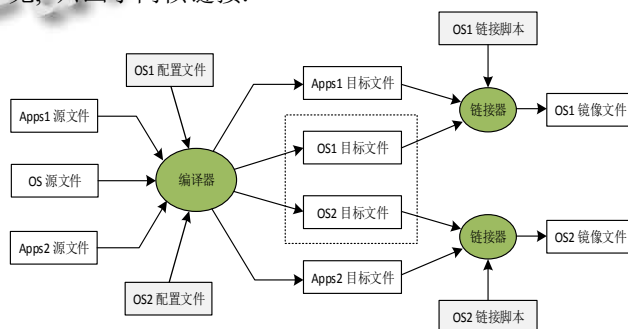


图 5 多核操作系统编译链接过程

上图中编译阶段与单核编译没有区别,结合用户对操作系统的配置文件,编译器分别将操作系统和用户应用程序的源文件编译成对应的目标文件.与单核链接不同的是,多核链接过程会更复杂一些,可概括

为下面的两步:

1) 多核操作系统的链接. 将多个操作系统的目标文件进行比较, 生成一个统一的操作系统目标文件, 即代码和数据的符号地址一致.

2) 生成各核操作系统映像. 每个核的应用目标文件与统一的操作系统目标文件进行分别链接, 最终生成每个核的操作系统映像文件.

图 6 是三种不同模型下操作系统映像加载之后代码和数据分布示意图, 其中 SCMD 模型中将应用的数据存放在核私有数据空间. 从图中可以看出, 多核链接完成的主要工作是将 AMP 模型中各个核操作系统代码集合 {OS0 Code, OS1 Code, ..., OS7 Code} 合并生成统一的操作系统代码 OS Code, 同时将各个核操作系统数据集合 {OS0 Code, OS1 Code, ..., OS7 Code} 生成统一的操作系统数据 OS Code, 并部署在各个核的私有地址空间, 使得操作系统数据具有统一的符号地址.

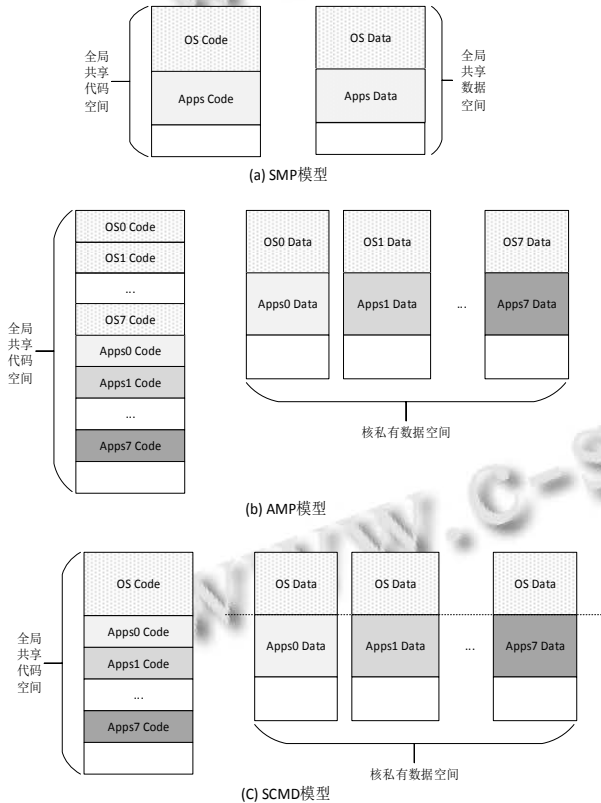


图 6 三种不同模型操作系统映像内存空间布局图

4 实验与性能分析

4.1 实验平台

RTEMS(Real-Time Executive for Multiprocessor

Systems)是一款开源实时嵌入式操作系统^[14], 最早用于美国国防系统, 现在由 OAR 公司负责版本的升级与维护. RTEMS 作为一个优秀的 RTOS, 具有实时性好、稳定性强、高度的可配置性等特点, 广泛应用于航空航天、医疗设备、网络等领域. 最新的 RTEMS 主代码分支中同时支持 SMP 和 AMP 两种多处理模式, 在编译时开发者通过宏定义 RTEMS_SMP 和 RTEMS_MULTIPROCESSING 来控制编译成何种模式的多核操作系统映像.

为了比较多核操作系统的 AMP、SMP 模型与本文提出的单代码多数据(SCMD)模型在多核 DSP 平台上性能差异, 通过移植 RTEMS 两种不同多核模型到多核 DSP 上, 生成 AMP 与 SMP 多核操作系统映像. 同时, 根据第三节提出的多核链接技术修改编译器, 将移植的 RTEMS 编译生成 SCMD 模型的操作系统映像. 三种不同多核操作系统模型在内存布局上也有所不同, SMP 模型操作系统映像必须要部署在全局共享空间, AMP 与 SCMD 模型代码部署在共享代码空间, 而数据则分散在核私有数据空间与簇内共享数据空间, 具体内存布局如表 1 所示.

表 1 三种不同模型内存布局表

模型	OS+应用代码	OS 数据	应用数据
AMP	共享代码空间	核私有数据空间	簇内共享数据空间
SMP	共享代码空间	全局共享数据空间	全局共享数据空间
SCMD	共享代码空间	核私有数据空间	簇内共享数据空间

实验设计三组测试用例, 主要用来测试操作系统映像所占用不同层次内存的大小, 以及与实时性能相关的几个关键性指标. 使用的测试用例如表 2 所示.

表 2 实验使用测试用例

用例功能	用例使用主要模块
test01 多任务测试	初始化、任务管理
test02 中断测试	初始化、任务管理、中断管理
test03 信号量测试	初始化、任务管理、信号量管理

实验是在为该四簇八核 DSP 开发的模拟器上进行, 该模拟器能模拟硬件的行为, 并且是时钟精确的, 因此可以用它比较容易地精确得到时间相关参数.

4.2 内存开销

RTEMS 具有高度的可配置性以及可裁剪性, 只有应用使用到的模块才可能被加入到最终生成的操作系统映像中. 本文分别测量和分析了三个测试用例生成的操作系统映像的代码大小, 以及操作系统内核、应

用数据所占内存空间。AMP 由于是多副本操作系统, 而 SMP 是单副本, 可以分析在八核处理器上 SMP 操作系统内核代码空间占用大约为 AMP 的八倍, 但 SMP 模式下有多核同步与互斥的额外开销。本文提出的 SCMD 模型, 采用的单代码多数据, 因此在代码空间上比 AMP 有很大的减少。测试和分析得出的结果如表 3 所示, 表中数据的单位为字。

表 3 三种不同模型所占内存对比

		AMP	SMP	SCMD
test01	OS+应用代码	1, 950, 208	395, 256	377, 280
	OS 数据	84, 864	11, 669	89, 107
	应用数据	11, 456	12, 602	11, 456
test02	OS+应用代码	1, 953, 280	398, 318	380, 352
	OS 数据	85, 925	11, 138	89, 237
	应用数据	11, 936	13, 130	11, 936
test03	OS+应用代码	1, 903, 104	384, 468	366, 912
	OS 数据	84, 672	11, 642	88, 906
	应用数据	11, 808	12, 989	11, 808

从表 3 中数据可以看出, 从代码空间开销来看, 由于 AMP 是每个核都需要一个操作系统副本, 因此占用内存较大; 而本文提出的单代码多数据模型将八个核的操作系统代码进行合并, 平均能减少约 80% 的代码空间开销, 有效解决了 DSP 代码存储空间不足的问题。从数据空间来看, 虽然 SMP 只保留一份操作系统数据, 但必须要存放在全局共享数据空间才能供所有核访问, 因此未能利用内存体系结构中高速访问的核私有内存和簇内共享内存; 而本文提出的单代码多数据模型, 将操作系统数据存放在核私有数据空间、应用的数据存放在簇内共享空间, 实验证明了该种方案的可行性, 下小节将从操作系统的实时性上分析这种模型也有独特的优势。

4.3 实时性分析

由于本文所使用的多核 DSP 就是为实时应用来设计的, 因此要求运行的操作系统也是实时操作系统, 操作系统的实时性是在选型时所需要考虑的一个重要的因素。Kar 在其论文^[15]中给出影响 RTOS 实时性的六个关键操作的时间量, 目前这些时间参数已成为普遍使用的操作系统实时性能指标。本文采用文献^[16]中提出的测量方法, 选取其中三个时间指标进行比较, 分别为上下文切换时间、中断延迟时间以及信号量混洗时间。该款多核 DSP 无数据 Cache, 因此代码执行

的时间与访存数据的大小与访存时间紧密相关。表 4 为最终仿真的结果, 表中数据的时间单位为 cycles。

表 4 三种不同模型实时性能时间指标

模型	上下文切换时间	中断延迟时间	信号量混洗时间
AMP	946	514	1236
SMP	11508	4680	13272
SCMD	945	518	1189

从表 4 中结果可以看出, AMP 与本文提出的 SCMD 模型性能差别不大, 因为两者都将操作系统与应用的内存部署的位置相同; 但由于 SMP 将操作系统与应用数据部署在访存速度较慢的全局数据空间上, 因此会大大影响实时性能, 平均性能要比 SCMD 模型慢 10 倍左右。

5 结语

本文首先介绍了基于紧耦合共享内存的多核平台的嵌入式操作系统常用的三种模型, 同时分析了三种多处理模型在资源管理与分配、可扩展性以及操作系统和应用的移植性等方面的优缺点。嵌入式多核处理器由于其专用性, 因此设计时在其体系结构上会有所调整, 以适应特定应用的需求, 本文基于多级存储的内存体系结构的 DSP 提出了一种单代码多数据的嵌入式多核操作系统模型。实验表明通过对八个核共用同一份操作系统代码, 相比 AMP 模型可以减少约 80% 的代码空间额外开销; 同时将每个核的操作系统数据部署在每个核的私有内存上, 相比 SMP 模型, 与实时性紧密相关的时间开销减少约 10 倍。本文的研究对于特殊架构多核处理器下安装部署多核操作系统具有一定的指导意义。

参考文献

- 1 Conte, Thomas M, Levy M. Embedded multicore processors and systems. IEEE Micro, 2009, 29(3): 7-9.
- 2 Blake G, Dreslinski RG, Mudge T. A survey of multicore processors. Signal Processing Magazine, IEEE, 2009, 26(6): 26-37.
- 3 Kumar R, Tullsen DM, Jouppi NP, et al. Heterogeneous chip multiprocessors. Computer, 2005 (11): 32-38.
- 4 Karam LJ, AlKamal I, Gatherer A, et al. Trends in multicore DSP platforms. Signal Processing Magazine, IEEE, 2009, 26(6): 38-49.

- 5 Luecke K. What is the Current State of Software Development for Multi-core Processors? Infotech@ Aerospace 2011.
- 6 Vaidehi M, Nair TR. Multicore Applications in Real Time Systems. arXiv preprint arXiv: 1001.3539, 2010.
- 7 何翔,任晓瑞.支持多核的嵌入式操作系统关键技术研究.航空计算技术, 2013, 43(4): 86–90.
- 8 Trawick T. Multicore communication: today and the future. Embedded Computing Design, 2007: 29.
- 9 Betti E, Bovet DP, Cesati M, et al. Operating system's support for multicore systems: current and future trends . Techincal repport, System Programming Research Group Dept. of Computer Science, System and Industrial Engineering University of Rome Tor Vergata, 2007.
- 10 Nemati F, Behnam M, Nolte T. Efficiently migrating real-time systems to multi-cores. Emerging Technologies & Factory Automation, 2009. ETFA 2009. IEEE Conference on. IEEE, 2009: 1–8.
- 11 Freescale, Running AMP, SMP or BMP Mode for Multicore Embedded Systems, 92–97.
- 12 QNX Neutrino RTOS home page. <http://www.qnx.com/products/neutrino-rtos/neutrino-rtos.html>. 2014.
- 13 邓林.单芯片多核处理器存储优化技术研究[学位论文].长沙:国防科学技术大学,2011.
- 14 RTEMS home page.<https://www.rtems.org/>. 2014.
- 15 Kar RP, Porter K. Rhealstone-a real-time benchmarking proposal. Dr Dobbs Journal, 1989, 14(2): 14.
- 16 江建慧,唐智杰.测量嵌入式操作系统实时性能参数的新方法.同济大学学报(自然科学版),2008, 9: 21.