

高效的多 DSP,RISC Core 软件开发工具

概述

RTOS (实时操作系统) 这个名词, 也许对您并不陌生, 九十年代初期即见诸一些学术性文献, 有关它的应用逐步成为系统工程师注目的焦点; 今天, 得益于微电子技术、软件技术的发展, 在高性能 DSP 应用领域, 它已经成为世界上 DSP 行业界最流行的实时应用开发工具。

应用领域:

实时系统(有严格的要求)

多任务应用系统

主从式应用系统/完全嵌入式系统

DSP 应用系统

ASIC Cores 应用系统

多 CPU(不仅限于 DSP) 系统



挑战-多 CPU 数字信号处理机软件编程面临的问题

随着科技的发展, 工程师所要完成的系统越来越复杂: 功能越来越多/性能日趋优化/. 无论是硬件软件均较以往有了较大的发展,

在硬件方面, 由于微电子技术日新月异, 芯片内部结构及资源日趋复杂/丰富, 且芯片品种大大增加, 芯片的更新换代周期越来越短, 硬件工程师熟悉新的器件的压力越来越大,

在软件方面, 系统实时性要求越来越强, 设计人员必须完成灵活的系统以尽快满足各种用户随时的特殊化要求, 以适应市场竞争, 对软件的可靠性要求越来越高, 且由于软件的生命期(乃至是整个系统的生命期)越来越短, 对软件上市前的开发的时间也是一再压缩, 软件开发成本也在提高(如人力资源等等)

另外, 竞争激烈的市场对产品提出了规范化, 标准化, 开放化的要求, 质量认证的要求 (ISO9000), 以便在系统集成, 系统升级, 二次开发等情况下更具竞争力

软件工程师面临的挑战还不只这些:

在构成多 CPU 数字信号处理机时, 编程可能会变得相当复杂, 因为每个 CPU 均需对其单独编程, 同时要考虑密切耦合的各 CPU 之间的通信, 这带来两个问题: 一个是要处理底层硬件操作, 而芯片结构日趋复杂, 另一个是要考虑各 CPU 之间的同步与协调, 前者不仅增加了编程的难

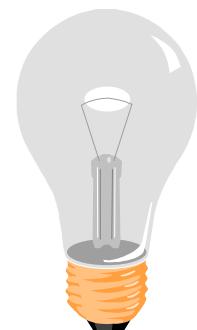


度，更带来了软件可靠性差的问题，后者则会由于同步不好而给整个系统带来性能上的瓶颈，直接影响系统实际工作的效率。

业界最优秀的解决方案：RTOS—Virtuoso

DSP 开发划时代的工具，让设计人员摆脱低水平的技术重复，极大的降低了软件人员的工作量，把软件工程师从传统繁冗的底层代码开发工作中解放出来，发挥更大的创造性：

- 以单 CPU 编程的方式为多 CPU 处理机编程
- 编程时不必考虑 CPU 之间的通信
- 把复杂的底层硬件操作“封装”起来
- 程序有极强的移植性，更换 CPU，改变 CPU 个数源代码无需更改
- 内核提供用于任务之间通信的系统调用
- 目标板可以通过申请主机的主控台服务和文件 I/O 服务
- 支持用户自己的目标板
- 实时性好、效率高、
- 极少占用 CPU 资源（1% 的计算开销，平均 2K 字的存贮开销）



85 个 API
95 个 API

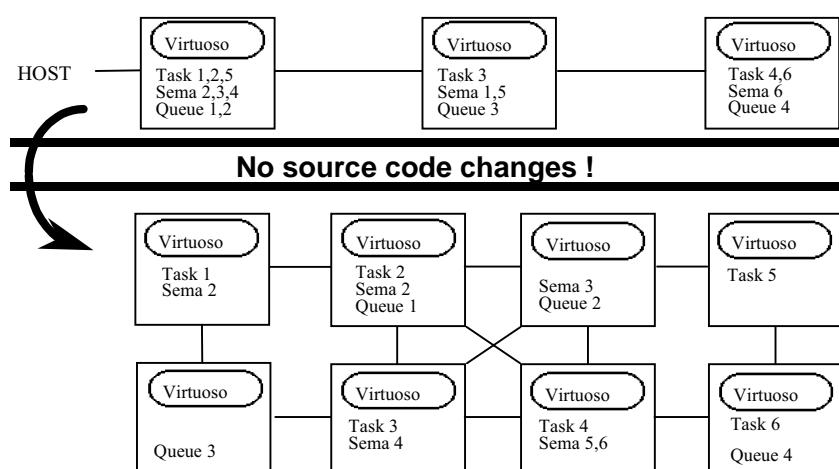
模块化设计方式
易学易用

技术描述

多 DSP 软件开发的支持-VSP 技术

Virtuoso 是一个专门用于简化实时应用中多 CPU 数字信号处理机编程的软件工具，它可以令程序员以单 CPU 编程的方式为多 CPU 处理机编程，即所谓的 VSP (Visual Single Processor) 技术。这种技术允许程序员将一个多 CPU 网络看作是一个“超级”CPU，而编程时不必考虑 CPU 之间的通信，这样，就大大降低了编程的难度，程序员可以在一个单 CPU 的平台上开发程序源代码，然后再移植到多 CPU 硬件平台上去（可多达上千个 CPU），完成在传统开发方式下无法解决的问题。

图 1 硬件由三个 CPU 增加到七个，源代码无需改变，只需重新编译即可



多任务并行处理的支持

在 Virtuoso 环境下，整个程序被分成多个任务，每个任务功能相对独立，任务之间呈“松耦合”状态，这些任务在 CPU 网络中并行运行。由于程序的复杂程度与代码量的关系并非线性，这种处理大大降低了程序的复杂程度。任务之间的通信，由 Virtuoso 环境处理解决，Virtuoso 环境将在目标系统 CPU 上生成一个实时多任务操作系统（RTOS: Real Time Operating System）内核，这个内核提供用于任务之间通信的系统调用，这些系统调用均为 C 语言函数形式，RTOS 可以提供用于通信的各种数据类型，象队列、邮箱、FIFO、信号、事件、存贮池等等，图 1 是一个多任务应用示例。

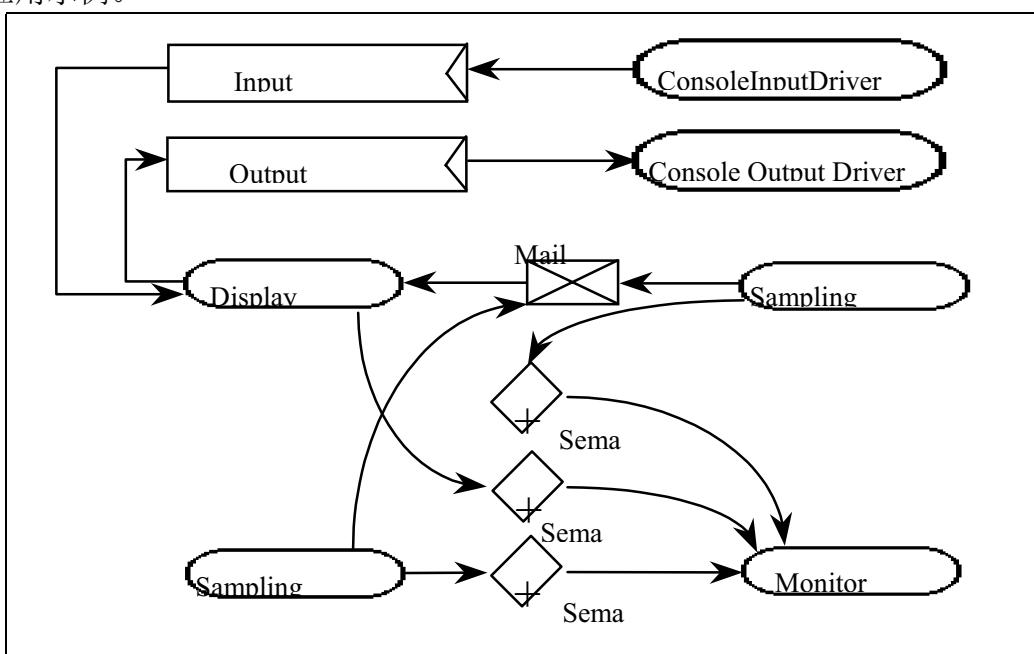


图 2：多任务应用示例一圆形表示任务，矩形表示数据对象：

Task: Console Input Driver, Console Output Driver, Display, Sampling, Monitor

FIFO: Input, Output,

Mailbox: Mail

Semaphore: Sema

任务之间的通信通过一些简单明了的系统调用完成，后者由内核提供支持，例如，图 2 所示的 DSP 系统中的 Sampling, Display 两个任务，Sampling 可以用如下方式向 Display 报告一些信息，在任务 Sampling 的代码中：

```

.....
K_MSG M; // 定义 M 为一个邮箱的数据头;
.....
M.size = 4 * sizeof (ACQUIRE_DATA); // 邮箱数据头赋值
M.info = id;
M.tx_data = &data; // 指向待传数据
  
```

```

M.rx_task = DISPLAY; //说明接收方
KS_SendW (Mail, 1, &M); //发送数据
.....

```

这里定义 Mail 为一个 Mailbox，M 为一个邮箱的数据头，上述代码中的 KS_SendW (Mail, 1, &M) 函数完成信息头向 Mail 的传输，

相应的，在任务 Display 的代码中，用函数 KS_ReceiveW (Mail, &M) 完成信息的接收；同时任务 Sampling 也可以是接收者，而由任务 Display 发送信息，只是这时需定义另外一个 MailBox:Mail2，在任务 Display 中用 KS_SendW (Mail2, 1, &M) 函数来发送信息，在任务 Sampling 中由 KS_ReceiveW (Mail2, &M)；函数来接收。

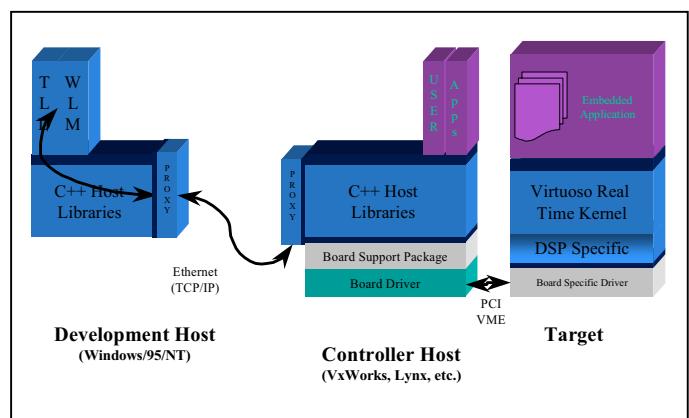
RTOS 会判断两个任务是否处于同一个 CPU 上，并判断所用到的 MailBox 是否为本地资源，并根据不同情况执行不同的操作，完成程序员所作调用的功能，如果需要进行 CPU 之间的通信，系统会利用 DSP 的通信口（例如 C40 的 Comport、SHARC 的 Liport）、DMA 等资源完成，而这些操作对程序员来说是透明的。这样，就把复杂的底层硬件操作“封装”了起来，大大降低了编程的难度，同时增强了代码的可读性，增强了代码的可维护性。

系统升级与源代码移植

这样做的另外一个好处是：由于程序中不涉及任何与硬件有关的描述，程序有极强的移植性，程序员可以在一块单 DSP 板子上开发程序，将来如果需要升级硬件，例如改用一块四 DSP 的板子，源代码无需作任何更改，只要在 Virtuoso 环境下的一个系统描述文件里作相应声明，重新生成目标文件即可。而传统方式下，如果硬件进行这种规模的升级，相应软件升级的工作将是很麻烦的。另外，这种极强的移植性还表现在，即便是更改了 CPU 的类型，如果 Virtuoso 支持这种 CPU，源代码仍无需修改即可适于于新的系统。

无需编写主机程序代码

通常，一个 DSP 系统会包含主机和目标板，所以，系统软件将由主机软件和 DSP 软件两部分组成，主机软件用于对目标板的控制，监测，并出借主机资源给目标板，例如目标板可以通过申请主机的主控台服务和文件 I/O 服务。通常，这部分代码由程序员自己完成，这要处理大量的计算机硬件及 Windows 底层操作，这部分工作常常给 DSP 专家带来一些不必要的麻烦，使得整个开发工作变得繁琐起来，相应的带来一些诸如系统可靠性差之类所问题。



Virtuoso 环境提供一个主机服务程序和一些实时运行库，即可完成上述主机软件所需完成的功能，如控制 DSP 目标板运行程序，并传送参数给目标板，目标板上的 DSP 可以调用图形函数（类似于 Borland C 的 BGI 库），由主机在屏幕上画出图形，例如，如下代码申请主机初始化图形状态：

.....

```

KS_LockW (GRAPHRES); /*等待其他任务释放主机资源，而后本任务得到该资源，并锁定它*/

```

```

initgraph (VGA, VGAHI);
KS_Unlock (GRAPHRES); /* 释放主机资源, 供其他任务申请*/
.....

```

DSP 可以申请通过主机键盘得到一个按键，并且可以通过主机屏幕输出字符，存取磁盘文件，主机部分软件无需再编程。有时，用户可能需要主机具有一些特殊的功能，这可以通过修改主机服务程序来完成，Eonic 的主机扩展工具 HEK 中提供主机服务程序的 C++ 源代码，由于采用了面向对象的技术，使得修改主机服务程序的工作得以大大简化：覆盖基类的一些函数（有时可能要增加一些函数）即可。

支持用户自己制板

HEK 的另外一个重要用途是生成目标板的板级支持软件包 BSP，BSP 是用于主机和目标板通信的软件模块。如果用户选用一些商业现成板，BSP 通常可以直接得到，但有时用户出于一些特殊的考虑，需要自己制作目标板，用户需完成 BSP，由于 HEK 中有 C++ 源代码供继承，所以这一步工作也得以大大简化，用户需完成的代码很少。

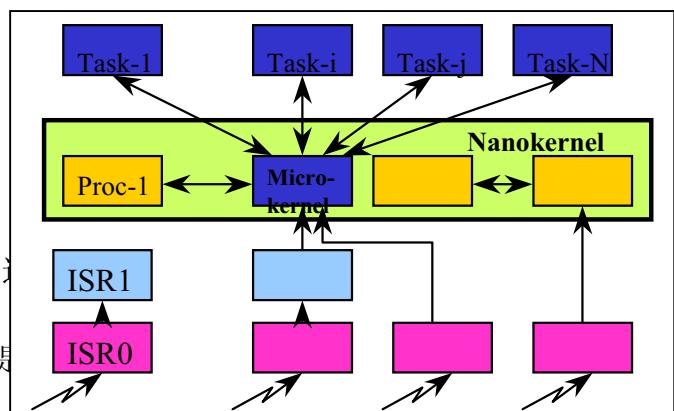
性能描述

V 的内核分成四个层次，ISR0 层，ISR1 层、
NanokernalMacro_kenal

效率

有一个问题可能是用户比较关心的，那就是在...
的损失. :

一般来说，效率的降低将来自两个方面，一方面是
有计算和存贮的开销。



在 C 语言编译器效率这方面，我们这样考虑这个问题，对于一个 DSP 系统软件来说，大部分执行时间将集中在少量的源代码……现成的基本 DSP 函数调用上，因为一般来说一个 DSP 软件算法的绝大部分往往由那些通用的基本部分组成，而 Virtuoso 环境及其第三方提供了大量的算法库，其中的 DSP 函数均为优化的手写汇编程序编译而成，其接口为 C 语言，这样 C 语言编译器的效率不会影响这个函数的执行速度。其他代码可能在行数上较多，但一般来说重复程度和计算量都远低于前者，即便 C 语言编译器的效率较低，对总体性能“加权”也轻。所以，总的来说，在 Virtuoso 环境下编程，C 语言编译器的效率对系统软件性能不会影响很大。

在 RTOS 本身计算和存贮的开销方面，可以分这么几点来考虑：

在存贮方面，一方面 Virtuoso 采用了 SoftStealth 技术，根据需要组成模块化 RTOS，删减未用到的系统功能可以使 RTOS 所占空间从 0.5KWord 开始至 10KWord，这在 DSP 芯片用 RAM 容量大大增加的今天，是微不足道的。

在计算方面的开销，可以分两点来考虑：

1. RTOS 要完成一些用于维护系统正常运行的功能，例如任务的调度，即同一个 CPU 中按时间片对多个任务进行切换，这需要一些计算开销，这在很大程度上取决于时间片的划分，一般来说，所用开销大约在 1%
2. RTOS 要完成 DSP 软件中的系统调用，系统调用要花费时间。在这一点上，RTOS 对于 DSP 芯片

的意义相当于 DOS、WIN 对于 80X86 的意义，在没有 RTOS 的情况下，这部分调用的功能本来是要由手写代码实现的，在时间要求严格的情况下，程序员仍然可以从较低的层次入手构筑软件，Virtuoso 环境允许程序员从两个不同的层次编程，结合使用，既能降低软件工作量，又能兼顾软件效率。

另外，在 DSP 硬件性能飞速发展，价格大幅度下降，而人力资源价格上涨的今天，以硬件性能的冗余换取编程工作的简化，是非常合理的。

可靠性

事实上，在一些项目的开发中，即便没有用到 RTOS，其软件的一部分功能必然是非常接近 RTOS 的，在一些规模较大项目的开发中，从这种意义上来说，程序员相当于自己完成了许多 OS 的代码，而在 RTOS 环境下，这部分代码是无需程序员编写的，相比之下，传统开发方式无形中浪费了大量的人力物力，且可靠性无法保障。Virtuoso 经过了众多用户的检验，广泛用于各种 DSP 系统中，用户中不乏航空、航天、航海、军方，科研、世界知名大公司等单位，在可靠性方面有较好的保障。

Virtuoso 相对于其他同类软件的特点

SPOX 的开发公司被 TI 买断，继而只支持 TI 芯片，3L 公司也被 SPECTRUM 买断，EONIC 公司是目前世界上唯一一家独立的 DSP RTOS 软件开发公司，支持多家 DSP 厂家的芯片，象

Analog Devices

21020 and 2106x SHARC (incl. ADA RT)

Texas Instruments

TMS320C3x, C4x, C51, [C54], C62/7

DSP Group 技术 Pine & Oak cores,

ARM6, ARM7, ARM7T

Hyperstone E1, E2-32

Older processors :

PC 80x86 (under MSDOS), Intel i960CA

Motorola 68K, 56K and 96K

SGS Thomson T4xx, T8xx

— 68HC11, 68HC16, R3052

易于开发：more than 85 services for:

Task management

Task synchronisation

Task communication

Memory management

Timer management

Interrupt handling

实时性好、效率高：比同类产品快几倍甚至是一个数量级，

— Minimum overhead

- Small code size (0.5 to 10 K)
- Minimum cycles used by RTOS (< 1 %)
- Low interrupt latency (0.25 - 4 us)
- Maximum speed
 - Fast task switching (100 ns -)
 - Fast inter-processor communication (up to 100 MB/s)

Virtuoso 有众多的第三方厂家，有者非常广泛的应用领域，隐然已是本行业的标准。

Virtuoso的用户及其应用领域

Aerospatiale	: 国防
Automotive Labs/Ford	: 防止碰撞装置
Alcatel (Belgium)	: 电话, ADSL Modem
Alex Computers	: OEM
Analogic	: 医疗设备
Australian Sonar Systems	: 声纳信号处理
Barco/LVD (Belgium)	: 机器人
BIRA (Belgium)	: 星载设备
CTBA (France)	: 高速视像监测
Dassault Electronique	: 国防
DASA (Dornier)	: 空间技术
ESA	: 空间技术
GEC-Thomson Marconi Sonar Systems	: 声纳(1600 DSPs)
Framatome	: 核电站车间移动机器人.
Ford Automotive	: 防止碰撞装置
GCHQ (UK)	: 监视系统
GEC Marconi	: 国防
Honeywell Space	: 空间技术
IBM (USA)	: 甚高速控制
Macdonell Douglas	: 国防
NSA	: 人工智能
NRL	: 激光探测器
Peugeot (PSA)	: 汽车数据分析
Saab	: 航天器处理单元
Schenck	: 便携仪器
Schlumberger	: 地震数据采集 (16000 processors)
Siemens/Alcatel/DB (RACE)	: 移动网络仿真

Siemens	: MPEG 解码
Sierra Nevada	: 海军项目
Spacebel	: 合成孔径雷达原型
SST	: 发动机控制
Thomson Defense	: 多项国防设施开发项目
TRW	: 国防
VDO (Germany)	: 航天器显示系统
Vatana (Canada) :	
Wavetek	: 仪器

旋极公司

DSP 系统事业部