

DSP 应用的结构和发展方向

许 伟

摘 要: 简要概述了 DSP 与单片机的关系, 然后详细介绍了 DSP 应用软件结构, 并对 DSP 应用的未来趋势——DSP 和 RTOS 的结合进行了初步分析。

关键词: DSP 实时操作系统 循环调度

1 DSP、MCU、MPU 的关系

微控制器 MCU 通俗的称呼是单片机, 它与微处理器 MPU 是微机技术的两大分支。MPU 的发展动力是人类对无止境的海量数值运算的需求, 速度越来越快。MCU 的发展是为了满足被控制对象的要求, 向高可靠性、低功耗、低成本发展。一般 MCU 的引脚数在 60 以下, MCU 以 8 位机为主、32 位机为辅。有趋势提高 MCU 的运算功能, 将 DSP 集成到 MCU 中, 比如 32 位的 MC68356 集成了 Motorola 的 DSP56002。

微控制器 MCU 一直存在两种基本结构: 哈佛(Harvard)结构和冯诺伊曼(vonNeumann)结构, 还可进一步讲是对应成复杂指令集计算机 CISC 和精简指令计算机 RISC。冯诺伊曼结构具有单一总线 PRAM 或 DRAM 都映射到同一地址空间, 总线宽度与 CPU 类型匹配。哈佛结构具有独立的程序总线 and 数据总线, CISC 的指令一般是微码 miccode, 每条指令由 CPU 解码为许多基本指令, 基于 CISC 的微控制器一般很复杂, 都采用冯诺伊曼结构, 所需要的程序存储器比 RISC 产品少。微码在 CPU 产生而限制了 CISC 器件的带宽, 其指令集也比 RISC 器件大。

68000 的 MPU 是准 32 位的 MPU, 内部 32 位, 外部总线是 16 位。苹果机就是用 68000 系列, 它的运行分成系统态和用户态, 其设计是面向分时多任务或实时操作系统的, 68000 的总线后来变成 VME 总线标准。到 68020 就是全 32 位了。

1991 年 IEEE1149.1 即 JTAG 的公布满足了 IC 制造商的措施需求, 也给 ASIC、MCU、MPU、DSP、PLD、FPGA 等的用户带来方便。一般十万门以上的 IC 都有 JTAG 接口, 1993 年 IEEE1149.5 对 JTAG 作了修正(5 线接口)。IC 的测试分成晶片级、IC 封装级、电路板与系统级, JTAG 完成了前两者的测试。适于 68000 系列的 32 位机的开发工具 ICD32 是一段扁平电缆, 一端接 IC 的 JTAG 的 5 线接口, 一端通过 25 芯头(里面有 GAL)接 PC 机并口。传统上, 微控制器 MCU 与微处理器 MPU 是两大分支, 而 DSP 是 MCU 的一种特殊变形。但是从实质讲, MPU 多半是 CISC, 除了 DSP 之外的 MCU 也是 CISC。而 DSP 是 RISC。所以比较时更适合 DSP 与 MPU 相比, MPU 适宜于相同管理这样的应用中, 以条件判断为主的应用, 以软件管理的操作系统为核心的产品, MPU 的设计侧重于不妨碍程序的流程, 以保证操作系统支持功能及转移预测功能等。而 DSP 侧重于保证数据的顺利通行, 结构尽量简单。

2 DSP 应用的结构

DSP 应用不同于主流的嵌入式系统应用, 在那些应用中, 你可以依赖一个通用的多优先级核心(Kernel)的服务。而在 DSP 领域, 该核心与 DSP 模块可能是两回事, 尽管核心都应该提供 CPU 资源、中断处理、通信机制等。所以, 一个有丰富细节的核心和操作系统通常是不加以考虑的, 因为它加在紧凑的 DSP 上实在勉强。开发者常常自己设计系统软件框架, 作为目标代码的一部分一起运行。开发者甚至没有意识到自己编写了一个小型操作系统。

可见, 这样的核心/操作系统随着应用的不同是多种多样的。也需要有各种核心, 支持从简单到复杂的应用。至于是自己写核心或者从别的专业公司获得帮助, 那是需要在效率、紧凑、灵活、扩展性、安全等因素进行折衷考虑的。

2.1 核心的种类

一般来说,框架从简单到复杂分成:单任务和一个中断服务程序(ISR)、多个互相通信的任务、多个分时循环的线程、多优先级的结构。

· 单任务和中断服务流程 ISR(基本核心结构)

单任务处理一个数据缓冲区,数据由一个ISR从特定的硬件取得(如AD器件)。系统首先需要初始化,使能中断,建立外设的正确状态。ISR的编程则应该保护现场,ISR的错误是调试很难检查的。通常一次ISR控制一个数据,而主任务需要一个数据缓冲区。解决办法是双缓冲结构。主程序于相同管理这样的应用中,以条件判断为主的应用,软件管理的常见流程是:等待ISR填满缓冲区;处理缓冲区数据,释放已经处理完的缓冲区。为了保证实时性,系统至少是双缓冲,因此主程序处理一个缓冲区时,ISR正在向另一个缓冲区送数。这种情况下,不需要特别的同步机制,缓冲区的满就是同步机制,整个系统除了主任务外就无所事事了,主任务则占用一切DSP资源。故此模式效率非常高。

· 通用的任务

系统有一个核心,它可以管理多任务,允许向系统列表加任务来扩展系统,占用如何扩展则与应用有关。首先以一个自动应答机来说明这种扩展,ISR是一个,而多个任务串行执行:音调检查、语音识别、语音压缩等。第一个任务控制CPU的缓冲区,执行完后主动彻底放弃控制,移交下一个任务。这种安排使得任务切换开销很小,因为你完全知道切换时,哪些状态需要保存而哪些不需要。这种方式的毛病就是任何一个任务失控,整个系统就崩溃了,因为核心没有办法取得对CPU的控制。

· 多个执行线程以循环(round-robin)模式切换

对前一种结构的改进是使用一种更安全的任务安排,使各个任务的依赖关系弱化。这需要一个基于时间段的任务切换器,在当前任务用完了规定时间片后将控制权移交下一个任务。在循环结构中,各个任务是平等的机会。这样CPU不会在为坏的任务而瘫痪,当然,安全的代价是你无法准确知道任务切换在什么时候发生,于是某个线程进入时的现场保护必须保护该线程要使用的所有寄存器,并且在退出线程时全部恢复。

· 循环(round-robin)线程与协作序列的组合

线程与协作序列的组合中,以条件判断为主的应用。

当处理一个较复杂但每个通路的处理是相同的时候,如语音信箱,无线基站,PXB等。可以将多个通路按照通用的多任务方式处理,而每个任务内以循环线程组成,每个任务是安全的,也就保证了整个系统的安全的。

· 通用的解决:多优先组结构

这是灵活性最强的方式,广泛应用于浮点和定点DSP系统。多优先级可以认为是循环方式的一种增强版,在资源可用时,高优先级的线程会被执行。多优先级核心应该小心使用,因为很难确定实时执行序列,特别是如果还允许动态改变优先级的话,调试更加困难。而且核心本身占用的CPU资源对定点DSP而言是相当大的开销。

除了这些缺点,本结构是某些应用的理想选择。比如,系统中的任务有的是时间苛刻的(高优先级),有些是可以后台运行的(低优先级),比如在蜂房电话中,DSP要迅速及时处理带内信号信令,对用户的按键则可以较慢地响应。随着DSP和MCU的功能上进一步集成,这种应用会越来越多。

2.2 其他性能综述

· 现场保护

任务切换中的现场保护(上下文保护)是影响性能的一个重要因素,它与中断响应时延是矛盾的,在任务列表被访问和管理期间,中断通常需要禁止。所以,任务机制越复杂,中断需要的时延越大。注意,在DSP领域,实时中断常处理几十kHz的信号,

对中断响应需要的时延是很敏感的。

- 实时与非实时
- 实时与非实时 (obin)线程与协作序列的组合中, 以条件判断为主的应用

对于基于核心的系统进行调试, 将系统作为一个整体来调试需要满足两个基本要求: 需要系统全速运行时可以观察和管理系统; 需要知道每次观察时影响的上下文。为达到此要求系统必须增加额外开销。

DSP 设计者并非处处需要实时调试工具, 模块的很多部分可以单步调试。但是, 大部分问题 (bug) 都是整个系统全速运行时暴露出来的。Go-DSP 的调试工具 Code Composer 的优越是在系统全速运行时仍然可以观察和修改状态。实时调试是需要代价的, 需要有 Debug Agent 伴随应用代码, 占用了 CPU 部分资源, 当然在最终产品上, Debug Agent 是完全去掉的。

3 实时操作系统 RTOS 与 DSP 应用的结合

3.1 用于嵌入式微处理器的传统的实时多任务操作系统

目前的趋势是一个微处理器 MCU 从单一任务结构变成多任务的结构, 初期软件设计是在应用中增加一个任务调用循环作为主程序, 随着软件规模上升和对实时性要求的提高, RTOS 作为一种软件开发平台, 成为嵌入式系统领域的主流。

RTOS 是一段 MCU 启动后首先执行的背景程序, 贯穿系统运行的始终。RTOS 的引入会增加系统的代码存储器占用和运行时间。RTOS 主要的性能指标是存储器占用、最小任务切换时间、最大中断延时。这三个指标与 RTOS 自身的设计、微处理器的设计、C 语言编译器的性能有关。

3.2 实时操作系统 RTOS 与 DSP 结合

DSP 软件开发越来越复杂, 开发者会发现自己已在两个矛盾的方向努力, 一方面, 设计者必须对底层代码优化以满足实时应用, 同时由于系统越来越复杂, 需要高层次的设计手段, 包括使用库和第三方软件包。

对 DSP 应用提供 RTOS 支持, 是 DSP 的性能和功能日益增加的必然结果。DSP 正在从高速数学引擎转变为包含主流控制器具有的特性的芯片, 因此需要 DSP 设计人员集中精力解决应用问题, 而不是重复实施系统级功能。

DSP 系统一般是两个极端, 简单的单片 DSP 设计, 用于 Modem 或峰房电话等。另一种是高性能的多处理器 DSP 系统, 用于大批输入流的实时处理。对于高档多处理器 DSP 系统设计, 有四个可能影响性能的主要因素: 通过系统的数据流 (流水线或星形); 主系统总线 (VME 或 PCI 总线); RTOS 的性能, 多 DSP 系统中, RTOS 在每个 DSP 上运行于嵌入式模式, RTOS 提供所需要的数据流和处理性能, 同时又允许主处理器继续在其固有模式 (Win 95/NT/Solaris) 中操纵整个系统; 处理来自 A/D 阵列的输入流的接口设计, 最好方法是使 A/D 转换子系统与系统其余部分有效隔离。

选择 RTOS 的关键考虑因素:

- 保证其可靠性足以应付 DSP 负载。
- 支持与 NT 或 Unix 主系统的互操作性。
- 操作核心要足够小; 应于有限的 DSP 存储空间。
- 有面向 DSP 的高级指令集, 便于迅速编程。
- 必要时可以对低级程序码手动优化。

嵌入式 RTOS 的主要功能是为 DSP 之间的实时协调与通信提供一个标准化的环境, 包括中断处理和存储区分配等, 以及和主机 OS 握手的所有功能。RTOS 运行于 DSP 之上, 所以必须很小, 还应该能进行分布操作和 DSP 任务的并行编程。RTOS 的结构应该使程序员很容易把单 DSP 任务变成分裂模式把单 DSP 任务变成分裂模式 足实时应

用,同时由于系统越来越复杂,需要高层次的设计手,即能运行于多个 DSP。RTOS 应该能支持顺畅地把任务分摊。面向 DSP 的嵌入式 RTOS 的主要功能是:多任务;动态进程;同步消息传递;信号机;时钟管理;等等。总之,目标是最小的运行开销和最大的硬件控制能力。

作为 Eonic 公司的 Virtuoso,可以用于浮点和定点 DSP。用于 TIC4x 和 ADSP1060 的版本的差别是有特殊的通信端口,便于多处理器系统的硬件开发。通过提供透明的多处理,使软件开发也容易。Virtuoso 对于单 DSP 的应用也提供了很多优越性,从一开始开发,你就有一个完全的多任务环境,这就意味着你可以将应用分成几个小的任务,更便于编程。有一套完整的工具用于任务间通信,任务同步,管理存储区和定时器,中断管理。提供了一个 ISR1 层支持嵌套中断。而 TI 缺省的 ISR0 层的中断服务中是不允许被中断的。如果系统中有突发性的中断发生,就有可能丢中断。Virtuoso 的 ISR1 层提供了可嵌套的中断机制,且响应速度与 ISR0 相当。多任务机制使设计人员能够充分发挥 DSP 的能力,因为可以使 DSP 是 100%的忙碌,否则 DSP 可能为了等待某个事件发生而处于空闲。特别是采用 TMS320C6201,不使用 RTOS 是无法充分发挥其性能的。Virtuoso 的另一个优点是可移植性,编写的 C 代码可以使用到 Virtuoso 支持的任何 DSP。

用于 DSP 的 RTOS 对加快开发进度、提供高级功能调用和标准的 I/O 库是非常有用的。许多 RTOS 支持多任务并包含 DSP 库,使用它有助于可移植性和可维护性。另一方面,RTOS 要占用处理器的开销,耗费本来可用于信号处理的 DSP 周期。如果是对于单处理器,支持多任务的开销可能会很可观。另外,成本也是要考虑的因素。

参考文献

- [1]Virtuoso--the RTOS for DSP.Eonic 公司,1998
- [2]Embed rprocessor programming.Electronic Design, 1997(10)
- [3]中兴新通讯.1998(3)

本文内容来自互联网,著作权归原作者所有。由电子零件城 (<http://www.epcity.com/>) 整理并制作成 PDF 文件,仅供个人学习之用,不得用于任何商业目的,否则后果自负。如果您认为本 PDF 文件侵犯了您的任何权利,请来信 epcity@epcity.com 通知,本站立即删除。

搜集整理:电子零件城-笨笨兔 (QQ: 154502842) 2004-04-10