

# 基于 uCOS-II 的嵌入式数控系统实时性分析

刘 淼, 王田苗, 魏洪兴, 陈友东

(北京航空航天大学机器人研究所, 北京 100083)

**摘要:** 提出了一种嵌入式数控系统的软硬件体系结构, 该系统采用实时操作系统 uCOS-II 和 ARM 嵌入式微处理器构建。结合实时多任务操作系统的性能指标, 分析了嵌入式实时操作系统 uCOS-II 的时间特性, 并给出了 uCOS-II 中最长时间特性的计算方法, 同时, 在数控系统的平台上, 测试了系统任务调度各个阶段所用时间, 并对此系统的实时性和稳定性进行了分析。结果表明, 基于 uCOS-II 和 ARM 所构建的嵌入式数控系统的实时性好, 且具有低成本、高性能的特点, 具有广泛的应用前景。

**关键词:** 实时操作系统; uCOS-II; ARM

## Real-time Analysis of Embedded CNC System Based on uCOS-II

LIU Miao, WANG Tianmiao, WEI Hongxing, CHEN Youdong

(Robotics Institute, Beijing University of Aeronautics and Astronautics, Beijing 100083)

**【Abstract】** This paper discusses a hardware and software architecture of embedded numerical control system. This system is built on uCOS-II and ARM. The timing specification of uCOS-II is analyzed based on the performance of RTOS. The method for getting the longest timing specification of uCOS-II is described. And the time of system schedule is measured. The paper also analyzes the system real-time performance and stability. The result indicates that this embedded numerical control system has perfect real-time character, low cost and high-performance. It has extensive application.

**【Key words】** Real time operation system(RTOS); uCOS-II; Advanced RISC machine(ARM)

### 1 概述

嵌入式系统是面向应用的专用计算机系统, 目前已经广泛应用于军事国防、网络通信、工业控制等各种技术领域。和应用于桌面PC的操作系统相比, 嵌入式系统对实时性的要求逐渐提高。根据GMPTG(General Motors Powertrain Group)近10年的调查研究, 超过95%的应用对系统的实时性有着极高的要求<sup>[1]</sup>。

数控系统是一个典型的强实时性系统。也就是说, 该系统具有可确定性。可确定性主要是确保条件出现到由此引起的动作开始(或者结束)的时间在一个准确的时段内。在数控系统中, 条件是由操作员的指令(如: 紧急停止, 移动x轴等)或是机床的状态(如刀具破损等)引起的。实际上, 需要满足时间约束的情况主要是和系统安全(如对突发事件的反应等)以及切削精度有关, 因此数控系统具有硬实时任务。硬实时任务是指必须满足最后期限的限制, 否则会给系统带来不希望的破坏或者致命的错误<sup>[2]</sup>。

为满足嵌入式系统对实时性的需求, 在软件方面, 普遍采用的方法是对现有的分时系统进行实时的改造。即, 在非实时性的操作系统之上, 加一个实时任务调度器, 如RTLinux、RTAI、VenturCom的Windows NT实时扩展(Real-Time Extension, RTX)等。然而, 这种方式导致系统结构过于庞大和复杂, 实时任务的性能得不到很好的发挥。对于构建在Windows或者Linux平台上的数控系统, 系统的实时性很难保证。即使使用实时的扩展, 也很难估算系统的极端情况<sup>[3]</sup>——最长的中断等待时间、最长的任务切换时间等。

随着微电子技术的发展, 嵌入式微处理器的运行速度越来越快, 对实时操作系统提供了全面的运行支持。采用高性能

的嵌入式微处理器和实时操作系统构建嵌入式工业控制系统平台已经成为一种发展趋势。uCOS-II是专门为嵌入式系统设计的硬实时(Hard Real Time)多任务内核。它的内核精简, 多任务管理功能相对完善, 实时性能好, 可裁剪, 可固化, 源码开放, 可移植性强。uCOS-II应用广泛, 适应性强, 从8位到64位处理器都可以移植。uCOS-II已经成功地移植到大多数微处理器和微控制器上。新版本的uCOS-II通过了FAA(Federal Aviation Administration)认证, 很适合应用于对实时性有很高要求的嵌入式工业控制领域。

### 2 uCOS-II 构建的高精度加工数控系统

在高精度加工的数控系统中, 系统的伺服周期非常短(0.5ms)。通过实时扩展的Windows或者Linux平台上的数控系统很难满足要求。如图1所示, 本文提出了基于ARM和uCOS-II的嵌入式数控系统平台。该系统的硬件平台由ARM和FPGA组成, 硬件资源包括S3c44B0x处理器、32MB SDRAM、2MB Nor Flash、16MB Nand Flash。



图1 嵌入式数控系统

**作者简介:** 刘 淼(1980 - ), 男, 博士生, 主研方向: 嵌入式机电控制技术; 王田苗, 教授、博导; 魏洪兴、陈友东, 博士

**收稿日期:** 2005-12-27 **E-mail:** threioaterl@163.com

该系统的操作系统使用 uCOS-II。把数控系统分为：(1)译码；(2)插补；(3)伺服；(4)逻辑控制；(5)显示等过程。其中，前3个过程，是数控加工过程中运行的周期性任务，实时性必须得到严格的保证。操作系统在保证强实时任务的同时，还必须有剩下的时间片处理界面显示、用户操作等任务。

数控系统的软硬件结构框图如图2所示。译码、插补和伺服过程之间通过先入先出(FIFO)队列通信。因此，系统的加工精度主要取决于伺服过程的稳定性。在系统中，译码和插补过程分别在 uCOS-II 的优先级较高的任务中执行。伺服过程则通过 FPGA 输入的外部中断信号得以执行。

该体系结构中，在满足译码和插补任务可调度的前提下，要提高数控系统的加工精度，就需要提高操作系统响应中断的稳定性。即系统的中断等待时间要尽可能的恒定。

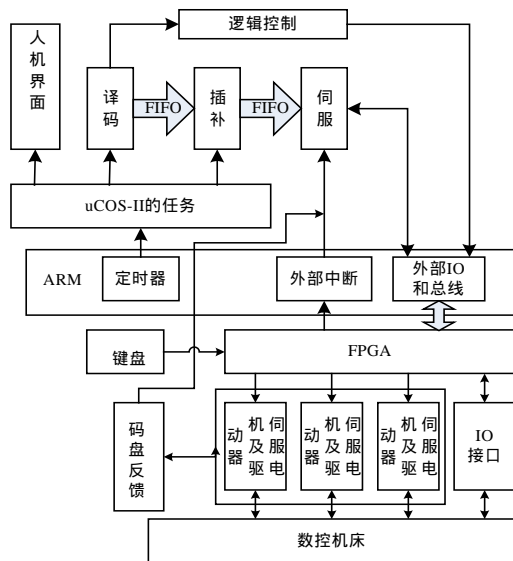


图2 嵌入式数控系统软件体系

### 3 uCOS-II 的实时性分析

#### 3.1 实时操作系统的性能指标

实时系统是指能够在确定的时间内执行计算或处理功能，并对外部的异步事件做出响应的计算机系统。多任务实时系统的性能主要体现在系统中每个任务的调度时间和执行时间的确定性。在系统硬件平台确定的情况下，系统的实时性完全取决于多任务调度算法。

与非实时系统相比，实时系统最大的特点在于，系统执行的正确性不仅仅在于计算的逻辑或者算术结果的正确性，还取决于系统产生结果的时间。如果一个实时系统的所有任务的执行时间都可以在一个规定的时间内完成，则认为此系统可调度。

按照Rhealstone方法<sup>[4]</sup>，定义了以下6项指标来衡量实时系统的性能：

(1)任务切换(Task Switch)时间，即系统在两个独立的、处在激活态并具有相同优先级的任务之间切换所需要的平均时间。切换时间取决于在存入和恢复上下文时处理数据结构的效率以及中央处理器(CPU)的结构和指令集。

(2)抢占(Preemption)时间，即系统将控制从低优先级的任务转移到高优先级的任务所花费的平均时间。

(3)中断等待时间(Latency)，即从CPU收到中断请求到执行中断服务程序的第一条指令所用的时间。

(4)信号灯(Semaphore)交替延迟时间，指从一个任务释放信号灯到另一个等待信号灯的任務被激活的时间延迟。

(5)死锁解脱时间，指系统解开死锁所需的平均时间。死锁解脱时间反映了操作系统解决死锁算法的效率。

(6)数据报(Data Gram)通过率是，当一个任务调用实时操作系统的原语把数据传送到另一个任务时，每秒传送的字节数。

#### 3.2 uCOS-II 操作系统任务调度分析

在 uCOS-II 中，最多支持 63 个任务，每个任务只能有独立的优先级，即 uCOS-II 中，不能有 2 个相同优先级的任务，系统也不支持时间片轮番调度。因此，按照 Rhealstone 方法的定义，uCOS-II 的实时性主要体现在：抢占(Preemption)时间，中断等待时间(Latency)。

uCOS-II 是一种基于优先级调度的占先式内核。在系统运行的过程中，可能发生任务调度的时机有：

(1)对任务操作的时候，包括：创建或者删除任务，挂起或者恢复任务，改变任务优先级。

(2)任务主动使用 OSTimeDly 函数延时的时候。

(3)发送(或者等待)信号量、邮箱、消息的时候。

(4)中断函数返回的时候。中断往往会使得一个更高优先级任务处于就绪状态，包括使用信号量等或者直接恢复某个任务。所以，在系统退出中断的时候，就需要判断是否有更高优先级的任务处于就绪状态，如果有，则需要进行任务调度。

(5)给系统的任务调度器解锁的时候。

在大多数情况下，uCOS-II 中断级的任务调度是按照如图3所示的流程进行的。其中， $T_r$  为系统的中断等待时间。 $T_p$  为系统的抢占时间。 $T_s$  为系统的中断响应与查找的时间，主要包括中断入口程序的执行，注册在系统中的中断向量的查找等。 $T_d$  为执行中断处理所用的时间。 $T_w$  为系统判断并查找处于就绪状态下优先级最高的任务和中断级任务切换的时间。而  $T_c$  则可以认为满足： $0 < T_c < T_r$  系统的最长关中断时间。

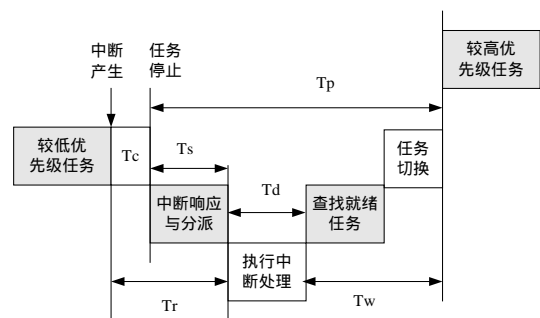


图3 uCOS-II 的任务调度

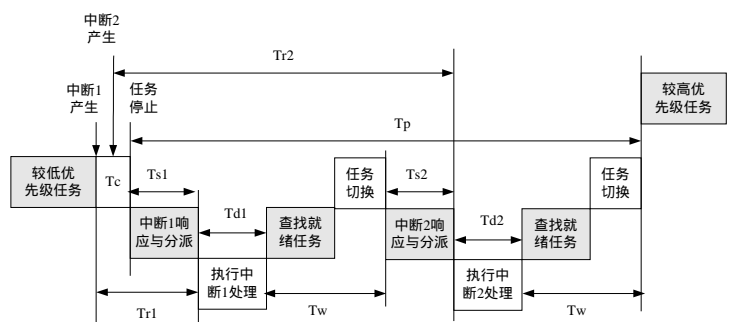


图4 多重中断的 uCOS-II 的任务调度

当有多个中断产生的时候, 如果因为中断的优先级低或者系统不允许中断嵌套, 则uCOS-II的中断级任务调度将按照如图4所示的流程进行。因为uCOS-II是依靠任务的查找表来确定处于就绪状态的最高优先级任务, 而任务切换的实质是对于处理器寄存器入栈和出栈的操作, 所以, 在同一系统中, uCOS-II中任务调度所花的时间 $T_w$ 是常数, 与应用程序中所建立的任务数无关<sup>[5]</sup>。

考虑系统的极端情况, 系统的最长中断等待时间是发生在多个中断同时产生的时候。假设系统中有  $1 - n$  个中断源, 中断的优先级按照由高到低的顺序排列, 且对于任意中断, 中断分派时间 $T_s$ 是固定的。则uCOS-II系统中, 当允许中断嵌套的时候, 第 $k(k=1,2,\dots,n)$ 个中断源的最长中断等待时间 $Tr_{k_{max}}$ 可以表示为

$$Tr_{k_{max}} = (k - 1) \times T_w + k \times T_s + \sum_{i=1}^{k-1} Td_i + T_c \quad (1)$$

当不允许中断嵌套的时候, 系统的最长中断等待的时间 $Tr_{max} = Tr_{n_{max}}$

系统最长的抢占时间 $Tpn_{max}$ 可以表示为

$$Tpn_{max} = n \times (T_w + T_s) + \sum_{i=1}^n Td_i \quad (2)$$

## 4 uCOS-II 系统实时性的测试

### 4.1 uCOS-II 实时性的测试方法

为测试 uCOS-II 操作系统中, 任务调度时各个时间片的精确时间, 如图5所示, 使用软件硬件相结合的方法, 通过处理器的 IO 口来标记各个时间点。

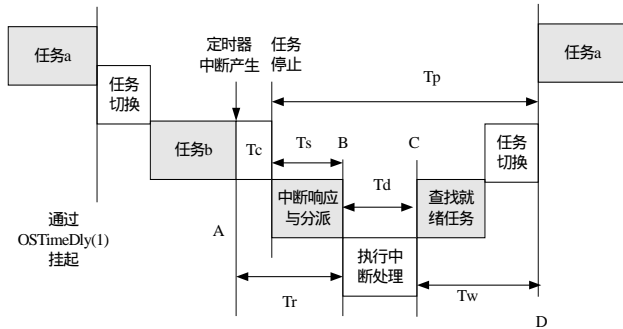


图5 uCOS-II 时间特性的测试点

信号1: 系统中只开启一个中断源, 即系统的定时器中断, 也就是uCOS-II的时钟。在S3C44B0处理器上, 定时器产生的中断源可以直接输出到IO上, 对应测试点A。

信号2: 在执行中断处理函数的开始对处理器的特定IO口置高(B点), 在中断处理函数的结束对IO口置低(C点)

信号3: 在系统中只创建两个系统任务: 任务a、任务b。任务a的优先级高于任务b。任务a的C代码如下:

```
for(;;){
    //测试 IO 口 GPG7, 置低
    rPDATG &= ~(1<<7);
    //任务切换
    OSTimeDly(1);
    //测试 IO 口 GPG7, 置高--对应测试点 D
    rPDATG |= (1<<7);
}
```

任务b的代码:

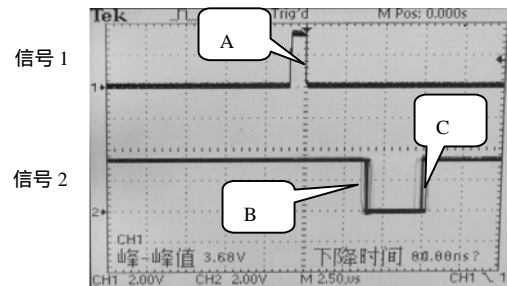
```
for(;;);
```

可见, 在任务a中, 不存在关闭中断函数, 所以, 可以认为  $T_c = 0$ , 即,  $Tr = T_s$ ,  $T_p = Tr + T_s + T_w$ 。在任务

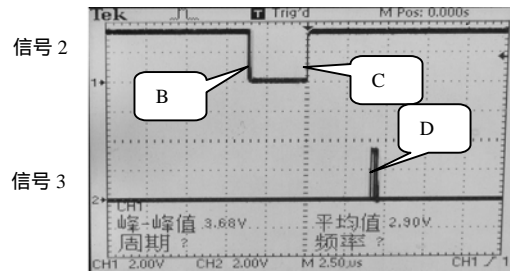
b中, 不允许调用更低优先级的任务, uCOS-II的最低优先级任务 idle 永远得不到调度。上述产生的 IO 口信号都是周期信号, 均可以用示波器测试。

### 3.2 测试结果

在所构建的嵌入式数控系统平台上, 采用高精度示波器对系统的实时性进行了测试, 其中系统的核心处理器——S3C44B0工作在67.5MHz的主频下, 开启8KB cache, uCOS-II操作系统在SDRAM中运行, 中断分配程序在flash中运行, 系统的定时器周期为0.5ms。测试得到的信号1、2、3波形如图6所示。



(a)中断等待与处理时间



(b)任务切换时间

图6 测试结果

因此, 可以准确地测试出系统的实时特性如表1所示。

表1 系统的实时特性

任务调度的各个时段	$Tr = Ts$	$T_d$	$T_w$	$T_p = Tr + T_d + T_w$
时间( $\mu s$ )	3.8	3.6	4.0	11.4

### 4.3 uCOS-II 下数控系统的实时性分析

在 uCOS-II 构建的数控系统中, 有两个中断源: (1)uCOS-II 系统的定时器; (2)由 FPGA 产生的外部中断源。在数控加工的过程中, FPGA 产生周期性中断, 插补和伺服的结果在中断中写入 FPGA 并得以执行。因此, 要想保证数控系统的加工精度, 必须保证 FPGA 产生的中断响应满足:

- (1)快速: 中断用尽量少的的时间得以执行;
- (2)稳定: 每次的中断等待时间尽可能的相同。

结合表1的测试结果和第2节中的假设与结论, 可以估算, 系统对于FPGA中断的最长等待时间为  $Tr_{f_{max}} = T_w + 2T_s + Td_1 = 15.2 (\mu s)$ , 最短的中断等待时间为  $Tr_{f_{min}} = T_s = 3.8 (\mu s)$ 。中断等待的误差为  $11.4 \mu s$ , 而系统的伺服和插补的周期为  $0.5ms$ 。即此 uCOS-II 组成的数控系统可以提供误差不超过  $2.28\%$  的插补周期。可见, 此平台上的 uCOS-II 操作系统完全可以满足数控系统的加工精度。

## 5 结束语

随着 SOC 技术的发展, 嵌入式微处理器的运行速度越来越快, 片上的外设资源越来越丰富, 采用高性能的嵌入式微

(下转第 226 页)