

基于 $\mu\text{C}/\text{OS} - \text{II}$ 和 DSP 的微机保护核心平台

曾成山¹,何海军²,谢培元³,曾次玲⁴

- (1. 贵州乌江水电开发有限公司, 贵州 贵阳 550002;
- 2. 南京南自电力控制系统工程有限公司, 江苏 南京 210003;
- 3. 湖南省电力调度通信中心, 湖南 长沙 410007;
- 4. 华中科技大学 电气工程学院, 湖北 武汉 430074)

摘要: 介绍了基于 TMS320VC5402 的新型微机保护核心平台的硬件系统, 及 $\mu\text{C}/\text{OS} - \text{II}$ 在 TMS320VC5402 上的移植。将实时多任务系统和传统的前后台系统进行了比较, 最后阐述了其基于 $\mu\text{C}/\text{OS} - \text{II}$ 多任务实时操作系统的软件设计思想, 尽量使软件和硬件脱离, 把整个系统分成几个任务, 然后赋予每个任务唯一的 ID 号, 并对各个任务指定其优先级。进入运行状态后, 在主任务中启动其他任务, 在这些任务中又可以启动另外的任务。

关键词: TMS320VC5402; $\mu\text{C}/\text{OS} - \text{II}$; 多任务实时操作系统; 核心平台

中图分类号: TM 77; TP 316 **文献标识码:** A **文章编号:** 1006-6047(2005)07-0057-04

微机保护在我国已经经历了近 20 年的发展历史, 形成了一整套相当成熟的理论体系。随着计算机技术的迅猛发展, 微机保护在硬件和软件两大方面仍然在不断完善和发展。从早期的 8 位机, 到现在 16/32 位机, 性能得到了提高, 运算能力也大大加强了。相对于硬件的快速发展, 软件方面的发展相对滞后, 大多数继电保护软件的规模较小, 功能单一, 智能化程度低, 而且多是采用汇编语言编写的线性程序^[1-4]。因此软件的编写和硬件的选择是一项关键性的工作。区别于传统的前台/后台程序开发机制, 同时在采用了高速 DSP 的基础上, 本文提出了基于 $\mu\text{C}/\text{OS} - \text{II}$ 和 DSP 的微机保护核心平台。

1 硬件平台的构建

1.1 硬件系统的整体设计思路

硬件系统的主控原理框图如图 1 所示, 整个硬件

系统以 TI 公司的 TMS320VC5402 为核心, 由于 DSP 芯片的速度较高, 为提高 DSP 的运行效率和达到最大处理能力, 一般对其外围芯片的速度也要求较高, 同时又要考虑系统在线编程调试的要求。所以, DSP + CPLD + FlashROM 的配置是 DSP 系统设计的一般要求, 这个结构的最大特点是结构灵活, 有较强的通用性, 适于模块化设计, 从而能够提高算法效率; 同时其开发周期较短, 系统易于维护和扩展, 适合于实时信号处理。对于片外数据空间一般选用高速的 SRAM, 尽量减少 DSP 的等待周期。用户程序一般在上电时从外部 ROM 加载到片内 RAM 区运行。该系统提供丰富的通信接口, 使装置具有较强的组网功能以及通信功能, 形成的硬件平台应该具有扩充升级的余地。

1.2 TMS320VC5402 简介

TMS 320VC5402(简称 VC5402)是美国 TI 公司

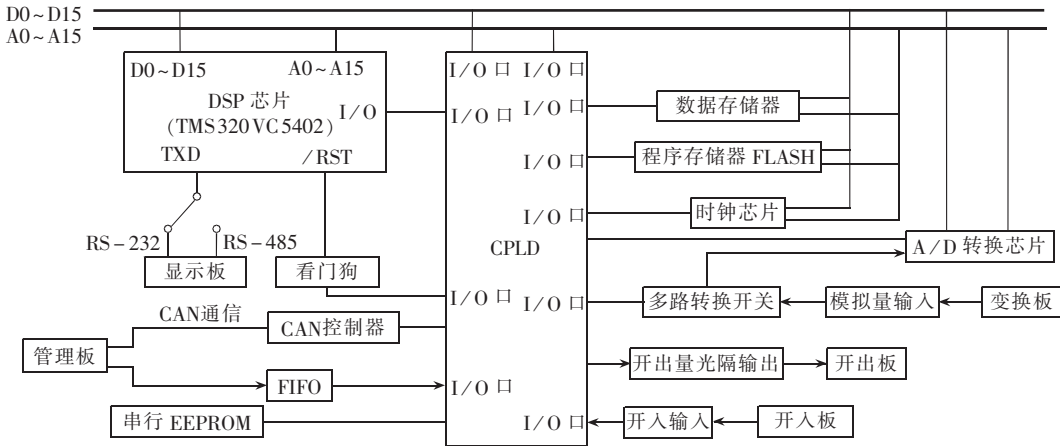


图 1 主控板原理框图

Fig.1 The block diagram of main control board

1999年10月推出的一款性价比极高的定点DSP芯片,该芯片的主要特点是^[5]:运算速度快;强大的寻址能力;优化的CPU结构;智能外设;低功耗方式;体积小;开发方便。

TMS320VC5402以其独有的高性能、低功耗和低价格优势,是选择VC5402作为该装置的数字信号处理器的主要原因。

1.3 存储器

64 kWRAM 全局数据存储器(实际可用32 kW)和64 kWRAM 局部数据存储器;64 kW Flash RAM 外部程序存储器;2 MW (2 M × 16 位)掉电不失NVRAM作为海量存储器,可存储故障报告,记录掉电前的数据;8 kB (8 k × 8 位)串行EEPROM存储器,存储定值安全可靠。

1.4 通信接口

主控板有一组RS-485和RS-232复用异步串行通信,一组CAN异步串行通信与显示板、采集板通信,其原理框图如图1所示。

1.4.1 RS-485和RS-232复用异步串行通信

RS-485通信和RS-232通信复用,由跳线来选择,使用DSP的串行口进行传输,用于和显示板的通信;RS-485通信使用差分传输芯片MAX1487控制传输数据,输入输出通道经光电隔离,并在输出口加一个双向瞬态电压抑制TVS (Transient Voltage Suppression)管,用来抑制浪涌电流;RS-232通信使用芯片MAX202E进行TTL-EIA双向电平转换。

1.4.2 CAN异步串行通信

CAN通信用来与管理板通信;CAN控制器采用SJA1000控制器,将并行数据转为串行数据传输,它有64 Byte的接收缓冲区,能通过验收屏蔽码寄存器和验收码寄存器进行单/双向接收滤波,有很强的通信能力;CAN控制器接口采用82C250,它提供对总线的差动发送能力和对CAN控制器的差动接收能力;同样的,输入输出通道经光电隔离,并在输出口加一个双向瞬态电压抑制TVS管,用来抑制浪涌电流。

1.5 A/D转换

32路模拟量经4片DG409多路转换开关,进入一片A/D转换芯片MAX125。DG409为8选2多路转换开关,每次同时选通4片,8路模拟量进入A/D转换芯片MAX125,MAX125为8通道(A、B各4通道),14位精度。每次进行4通道A/D转换,转换时间为12 μs。为了做到各个通道的采样同步,可考虑用4片MAX125,这样一次可接收来自变换板32路的模拟量,每次可进行16通道的连续转换。

1.6 开入回路和开出回路

开入回路和开出回路均由快速光电隔离芯片和逻辑编码电路组成,增加了电路的抗干扰性能。

2 软件系统设计

2.1 传统的前后台系统和实时多任务系统的比较

2.1.1 前后台系统

在前后台系统中,应用程序由后台运行的无限

循环和前台的中断服务子程序组成。后台主要对数据进行读取、处理和显示等操作,它甚至可以是一个空循环。而运行在前台的中断服务子程序主要负责处理时间要求相对严格的操作(如响应异步事件)。一般,中断服务子程序只对外部事件完成一个简单的处理,就把处理信息交给后台来执行接下来的操作。

由于后台程序必须运行到处理该信息的代码时,才能真正对信息进行处理,因此,系统对信息处理的及时性表现较差。最坏的情况时需要一个循环的执行时间,这样的等待时间通常称为任务级响应时间。当后循环程序修改后,运行一个循环的时间和时序都改变了,这时任务响应时间也不确定。

由于在系统中实效性强的处理都是通过中断服务子程序处理的,所以它又被称为中断驱动程序。系统中中断的现场保护和恢复,中断是否运行嵌套,中断处理流程和中断与后台主程序的共享资源问题都是在设计系统时要认真考虑的。

2.1.2 实时多任务系统

对于一个复杂的嵌入式实时系统而言,当采用中断处理程序和一个后台主程序这种软件结构难以实时、准确、可靠地完成时,或存在一些互不相关的过程需要在一个计算机中同时处理时,就需要采用实时多任务系统,随着应用的复杂化,一个嵌入式控制系统可能要同时控制/监视很多外设,要求有实时响应,有很多处理任务,各个任务之间有多种信息传递。如果仍采用原来的程序设计方法将存在两个问题:一是中断可能得不到及时响应,处理时间过长,这对于一些控制场合是不允许的,对于网络通信方面则会降低系统整体的信息流量;二是系统任务多,要考虑各种可能出现的情况(尤其在任务使用共享资源时,如果任务调度不当就可能导致系统死锁,从而降低软件可靠性),导致程序编写任务量成倍增加。为降低系统的复杂性,保证系统的实时性,可维护性是必不可少的。

实时多任务系统的实现必须有实时多任务操作系统的支持。操作系统主要完成任务切换,任务调度,任务间的通信、同步、互斥,实时时钟管理以及中断管理。实时多任务系统实际上是由多个任务、多个中断处理过程和实时操作系统组成的有机整体。每个任务是顺序执行的,并以并行性的方式通过操作系统完成,任务间的相互通信和同步需要操作系统的支持。

继电保护对实时性要求较高、任务较多,故本文提出基于μC/OS-II和DSP的微机保护核心平台。

2.2 μC/OS-II嵌入式实时多任务操作系统介绍

μC/OS-II是一个完整的,可移植、固化、裁剪的占先式实时多任务内核。μC/OS-II是一种专门为微控制器设计的抢占式实时多任务操作系统,它以源代码的形式给出。其内核主要提供进程管理、时间管理、内存管理等服务。系统最多支持56个任务,每个任务均有一个独有的优先级。由于其内核

为抢先式,所以总是处于运行态最高优先级的任务占用 CPU。系统提供了丰富的应用程序接口 API (Application Programming Interface)函数,实现进程之间的通信以及进程状态的转化。 $\mu\text{C}/\text{OS}-\text{II}$ 的使用使得应用程序的设计和扩展变得容易^[6],不需要大的改动就可以增加新的功能。复杂的应用程序可以分解为若干独立的任务,每个任务的调试、修改几乎不影响其他模块, $\mu\text{C}/\text{OS}-\text{II}$ 使得应用程序的设计过程大为简化。基于 $\mu\text{C}/\text{OS}-\text{II}$ 上的 C 语言程序具有极大的可移植性。在 $\mu\text{C}/\text{OS}-\text{II}$ 基础上可以编写出各种硬件驱动程序、库函数,和通用性的应用程序,都可以进行移植,大大减轻了软件工程师的劳动强度,缩短了产品投入市场的时间。

2.3 $\mu\text{C}/\text{OS}-\text{II}$ 系统结构分析与在 VC5402 的移植

图 2 说明了 $\mu\text{C}/\text{OS}-\text{II}$ 的软件体系结构。

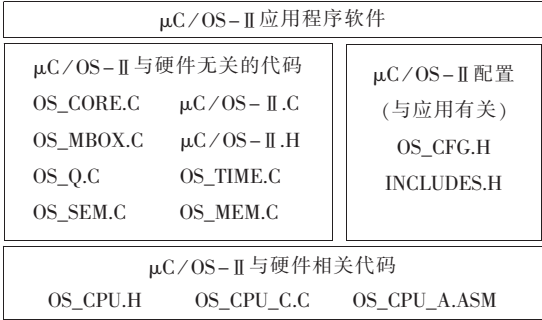


图 2 $\mu\text{C}/\text{OS}-\text{II}$ 系统的软件体系结构

Fig.2 The SW structure of $\mu\text{C}/\text{OS}-\text{II}$ system

从图 2 中可以看到,若使用 $\mu\text{C}/\text{OS}-\text{II}$,必须为其编写 OS_CPU.H,OS_CPU_C.C,OS_CPU_A.ASM 三个文件。这三个文件是与芯片硬件特性有关的,它们主要提供任务切换与系统时钟的功能。其他文件用 C 写成,它们为系统提供任务管理、任务之间通信、时间管理及内存管理等功能。 $\mu\text{C}/\text{OS}-\text{II}$ 的移植需要满足以下要求(VC5402 完全具备这些条件):

- a. 处理器的 C 编译器可以产生可重入代码;
- b. 可以使用 C 调用进入和退出 Critical Code(临界区代码);
- c. 处理器必须支持硬件中断,并且需要一个定时中断源;
- d. 处理器需要能够容纳一定数据的硬件堆栈;
- e. 处理器需要有能够在 CPU 寄存器与内存和堆栈交换数据的指令。

在为 $\mu\text{C}/\text{OS}-\text{II}$ 编写任务切换代码时需要注意的是: $\mu\text{C}/\text{OS}-\text{II}$ 在每次发生中断后都会产生任务调度,但在中断结束后进行的任务切换,不能调用普通任务切换函数,这是因为在中断过程中往往伴随将 CPU 的状态寄存器压栈操作。在中断结束后进行的任务切换中必须对堆栈指针进行调整。

在系统移植过程中另一个较为重要的部分是系统时钟。 $\mu\text{C}/\text{OS}-\text{II}$ 要求系统能产生 10~100 Hz 的时钟节拍。该时钟节拍由硬件定时器产生。系统时钟中断服务子程序用汇编语言写成,由于其主要功

能在用 C 语言编写的子函数中实现,因此,编写该服务子程序的难度不大。移植 $\mu\text{C}/\text{OS}-\text{II}$ 的主要工作就是处理和编译器相关代码以及板级支持软件包 BSP(Board Support Packet)的编写。

2.4 $\mu\text{C}/\text{OS}-\text{II}$ 移植到 VC5402 上的代码编写

2.4.1 任务切换^[7]

$\mu\text{C}/\text{OS}-\text{II}$ 系统调用 OSCtxSw()实现任务切换,首先调用宏定义,在当前堆栈保存所有寄存器,然后恢复处理的寄存器。

2.4.2 $\mu\text{C}/\text{OS}-\text{II}$ 系统时钟

以 VC5402 的时基信号作为系统时钟,每经历一个时钟节拍的时间将产生一次中断,在中断服务子程序中会调用 OSTickISR()函数,主要是将全部寄存保存到当前任务的堆栈中,设置状态寄存器,最后从当前任务的堆栈中恢复全部寄存器。

2.5 软件结构设计

在软件设计时,尽量使软件和硬件脱离,改变传统的嵌入式软件过多依赖硬件的模式。针对保护装置的实际情况,把整个系统分成保护、A/D 采样、自检、显示、通信等几个任务,划分原则是在考虑系统实时性和软件效率的前提下尽量减少任务间的耦合,使功能清晰。任务确定以后,赋予每个任务唯一的 ID 号,并按照实时性要求对各个任务指定其优先级,本设计中把保护任务设置成较高优先级。进入运行状态后,在主任务中启动其他任务,例如显示、A/D 采样、通信等。在这些任务中又可启动另外的任务。下面给出了主程序的任务创建和设置邮箱部分代码:

```
main()
{ //此处略去系统初始化部分代码
  OSTaskCreate(TaskStart, (void *)0, (void *)
    & TaskStartStk[TASK_STK_SIZE-1], 0);
    //创建主任务
    OSStart();
}

void TaskStart(void *data)
{ //此处略去允许时间节拍中断代码
  OSTaskCreate(baohu, (void *)&0, (void *)&
    TaskStk[0][TASK_STK_SIZE-1], 4);
    //创建保护任务
    OSTaskCreate(xianshi, (void *)&0, (void *)&
    TaskStk[1][TASK_STK_SIZE-1], 5);
    //创建液晶屏显示等其他任务
    AckMbox = OSMboxCreate((void *)0);
    //设置两个邮箱
    TxMbox = OSMboxCreate((void *)0);
    .....//其他代码
}
```

设计中的任务调度机制采用优先级调度模式:CPU 被分配给最高优先级任务,如果几个任务优先级相同,CPU 就被分配给最先进入就绪队列的任务。

如果一个任务在执行过程中,另外一个优先级更高的任务进入就绪状态,则原来正在进行的任务进入就绪状态,新任务进入运行状态。

任务间的通信机制采用邮箱,邮箱主要用于在不同任务间传递数据,一个任务把信息发送到邮箱,以后另外的任务接收该信息。一个任务可以拥有多个邮箱。在本系统中多个任务间有信息的传递,所以采用了邮箱这个通信机制。如果有多个任务正在等待同一个邮箱里的邮件,一个到达邮箱的邮件被传送给等待队列中最靠前的任务。如果没有任务等待邮件,则该邮件被放到邮件队列中。如果一个邮箱中有多个邮件,任务取邮件时按照先到先取的顺序收邮件。如果没有邮件,任务被放进等待队列中。邮件可以是自己定义的结构。

中断的优先级别高于任务的优先级别,不论是任何任务在执行,如果中断发生,一定是正在执行的任务被挂起;如果在低级别的中断服务进行期间,又发生更高级别的中断,则更高级别的中断得到响应,直到处理完毕,低级别中断才又得到响应,然后原先挂起的任务才获得 CPU 控制权。

$\mu\text{C}/\text{OS}-\text{II}$ 可支持 56 个任务。实现各种功能的保护程序按照功能划分成了一个任务,每一种任务实现一种功能,任务之间相互独立,只通过实时操作系统 RTOS(Real-Time Operating System)机制交换信息。这从根本上保证了软件的可靠性和实时性。

3 结语

本文介绍的基于嵌入式微处理器及多任务实时操作系统继电保护核心平台的设计方案采用了先进的硬件系统,并且摒弃了传统前后台式软件设计思路,依托嵌入式实时多任务操作系统,在嵌入式应用中引入了实时、多任务、并行运行的概念,采用现代软件工程方法开发软件,这些都预示着这种设计方案的潜力,代表着今后微机继电保护的发展方向。

参考文献:

- [1] 高 华. 新型继电保护发展现状综述[J]. 电力自动化设备, 2000, 20(5): 50-53.
GAO Hua. Survey on development state of new protective relay[J]. **Electric Power Automation Equipment**, 2000, 20(5): 50-53.
- [2] 陈德树. 计算机继电保护原理与技术[M]. 北京: 水利电力出版社, 1992.
- [3] 易永辉, 赵志华, 薛玉龙, 等. 一种新型的继电保护软硬件平台[J]. 继电器, 2002, 30(6): 26-28.
YI Yong-hui, ZHAO Zhi-hua, XUE Yu-long, *et al.* A new software and hardware frame of relay protection[J]. **Relay**, 2002, 30(6): 26-28.
- [4] 张振华, 许振宇, 张月品, 等. 第三代微机保护的设计思想[J]. 电力自动化设备, 1997, 17(3): 24-25.
ZHANG Zhen-hua, XU Zhen-yu, ZHANG Yue-pin, *et al.* Design consideration of the third generation microprocessor-based protection[J]. **Electric Power Automation Equipment**, 1997, 17(3): 24-25.
- [5] Texas Instruments Inc. TMS320VC5402 Fix-point digital signal processor(Literature number SPRS079)[EB/OL]. <http://www.ti.com>, 2003-09-21.
- [6] LABROSSE J J. 嵌入式实时操作系统 $\mu\text{C}/\text{OS}-\text{II}$ [M]. 邵贝贝, 龚 辉, 蒋俊峰译. 第 2 版. 北京: 北京航空航天大学出版社, 2003.
- [7] 戴明桢, 周建江. TMS320C54x 结构、原理及应用[M]. 北京: 北京航空航天大学出版社, 1999.

(责任编辑: 李 玲)

作者简介:

曾成山(1968-), 男, 贵州遵义人, 从事水电工程建设管理工作;

何海军(1976-), 男, 江苏江阴人, 研究方向为电力系统微机保护;

谢培元(1975-), 男, 四川南部人, 硕士, 研究方向为电力系统微机保护、电力市场;

曾次玲(1976-), 女, 湖南娄底人, 博士, 研究方向为电力系统自动化、电力市场。

Product platform for microcomputer protection system based on $\mu\text{C}/\text{OS}-\text{II}$ and DSP

ZENG Cheng-shan¹, HE Hai-jun², XIE Pei-yuan³, ZENG Ci-ling⁴

(1. Guizhou Wujiang Hydropower Limited Company, Guiyang 550002, China;

2. Nanjing Nanzi Electric Power Control System Limited Company, Nanjing 210003, China;

3. Hunan Electric Power Dispatch & Communication Center, Changsha 410007, China;

4. Huazhong University of Science and Technology, Wuhan 430074, China)

Abstract: The hardware system of a product platform for microcomputer protection system based on TMS320VC5402 is put forward. The transplantation of $\mu\text{C}/\text{OS}-\text{II}$ to TMS320VC5402 is introduced as well. The multi-task RTOS(Real-Time Operating System) is compared with the front-back system. The software design idea based on $\mu\text{C}/\text{OS}-\text{II}$ multi-task RTOS is explained, that is, separate software from hardware, divide the system into several tasks, afford each task an exclusive ID and appoint its priority. During running state, tasks may be initiated by main task and other tasks.

Key words: TMS320VC5402; $\mu\text{C}/\text{OS}-\text{II}$; multi-task real-time operating system; product platform