

DSP 空间相机控制器的在轨程序重注

武星星,刘金国,周怀得,徐东,孔德柱

(中国科学院 长春光学精密机械与物理研究所,长春 130033)

摘要: 提出了一种基于 DSP 的空间相机控制器在轨程序重注方法,给出了重注程序接收、存储、校验和执行的流程和部分代码。并给出了 DSP 重注程序的编写和生成方法。在验证实验中以重注死循环程序为例,验证了在轨程序重注方法和重注程序编写方法的正确性。

关键词: 数字信号处理器;空间相机;程序重注

中图分类号: V445.8

文献标识码: A

Program Reloading of DSP Space Camera Controller on Orbit

Wu Xingxing, Liu Jinguo, Zhou Huaide, Xu Dong, Kong Dezhu

(Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Sciences, Changchun 130033, China)

Abstract: An on-orbit program reloading method of DSP space camera controller is advanced. Flows of reloading program reception, storage, verification and execution as well as partial source codes are provided. How to write and produce reloading program are discussed and solved. In validation experiment, an infinite loop reloading program is used and proves the validity of the on-orbit program reloading method of DSP space camera controller.

Key words: digital signal processor; space camera; program reloading

引言

以卫星或宇宙飞船为平台的空间相机工作在距地球 100 km 以上的外层空间,一旦发生故障无法修复,或修复的成本极高,如美国 2009 年 5 月对哈勃望远镜的第 5 次修复就耗费约 9 亿美元^[1]。如果空间相机控制器具备在轨程序重注功能,即当空间相机在轨发生故障或发现软件缺陷时,能在地面通过测控天线将重注程序的二进制码流发送给在轨飞行的空间相机,空间相机在校验成功后执行,即可对故障和缺陷进行修复,延长空间相机寿命,节约昂贵的维修费用。

数字信号处理器(Digital Signal Processor, DSP)采用了哈佛结构、超标量流水线、多总线和专用硬件运算部件等技术,可以实现低功耗下的高速实时信号处理,是理想的空间相机控制器^[2-3]。本文提出了一种基于 DSP 的空间相机控制器在轨程序重注方法,给出了重注程序接收、存储、校验和执行的流程和代码,并给出了 DSP 重注程序的编写和生成方法。并以重注死循环程序为例,对在轨程序重注方法进行验证。

1 在轨程序重注方法的实现

图 1 为 DSP 空间相机控制器的组成原理^[4],DSP 程

序目标代码和常量表固化在只读存储器 PROM 中,数据存储器 SRAM 中保存变量和堆栈。Flash 中保存用于像移计算的电子地图。相机控制器从上位机接收指令和数据,并转发给成像单元和调焦单元。本文所用的 DSP 型号为 TMS320VC33。DSP 空间相机控制器上电或复位后从 PROM 运行固化的程序。为了实现在轨程序重注功能,必须在固化程序中留有处理上注程序的接口。重注程序从地面通过测控天线传送给上位机,上位机再转发给 DSP。

DSP 除了同上位机通信外,还要完成像移计算、图像对时信息采集等任务,因此当重注程序较大时,需要分多包多次上注。DSP 处理重注程序的流程如下:

① 重注程序包计数器加 1。

② 当收到第一包时,从重注程序接收缓冲区中读取重注程序包数,同时将重注程序过往包加载成功标志置

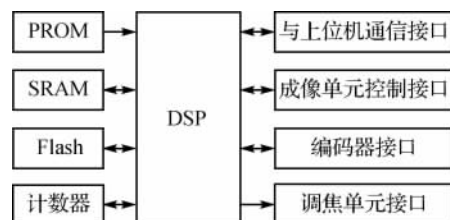


图 1 DSP 空间相机控制器组成原理图



位,并清零重注程序总长度、重注程序总累加和与重注程序可用性标识。

③ 当从重注程序接收缓冲区中读取的当前包序与重注程序包计数器值一致,即包序正确时,依次将重注程序从重注程序接收缓冲区加载到 RAM 或 Flash 中指定位置,然后再从该位置读出,将 32 位数据拆分成 8 位数据,计算累加和。如果包序错误,重注程序包计数器减 1,跳至模块出口。

④ 如果累加和校验成功,置位当前重注程序包加载成功标识,否则清零当前重注程序包加载成功标识。

⑤ 将当前重注程序包加载成功标识和重注程序过往包加载成功标识进行与运算,结果赋给重注程序过往包加载成功标识。

⑥ 从重注程序接收缓冲区中读取当前包重注程序累加和,将之累加到重注程序总累加和缓存值。

⑦ 从重注程序接收缓冲区中读取当前包重注程序长度,将之累加到重注程序总长度缓存值。

⑧ 当重注程序包计数器值等于重注程序包数,即收到最后 1 包时,清零重注程序包计数器,然后判断重注程序过往包加载成功标识。如果重注程序过往包加载成功标识被置位,说明所有包都加载成功,置位重注程序可用性标识;否则清零重注程序可用性标识。

在重注程序加载成功后,接收到执行重注程序指令后 DSP 开始执行重注程序。执行重注程序标识被置位时,若重注程序可用性标识被置位,依次读出全部重注程序,将 32 位数据拆分成 8 位数据后计算字节累加和,如果计算得到的累加和和重注程序总累加和缓存值相等,在对相关寄存器压入堆栈后,跳转到重注程序加载的首地址开始执行,返回时将相关寄存器弹出堆栈;如果计算得到的累加和与重注程序总累加和缓存值不相等,清零重注程序可用性标识。部分源代码如下:

.....

```
if (sumtemp == ReProgramSum) { /* 如果校验成功 */
asm(" PUSH R7"); /* 将 R7 压入堆栈 */
asm(" LDI @_ProgramStartAddr,R7 ");
/* 将重注程序加载的首地址送入 R7 */
asm(" CALLU R7");
asm(" POP R7"); /* 将 R7 弹出堆栈 */
}
else{ /* 如果校验失败 */
.....
}
```

从源代码中可以看出,由于要进行堆栈操作和跳转操作,必须使用汇编指令才能实现,因此采用了 asm 语句内嵌汇编的方法。这里需要注意的是,使用 CALLU 调用指令,而非 BR 跳转指令,由于 CALLU 指令自动将当前 PC

指针的内容压入堆栈,然后将 R7 的内容赋给 PC 指针。而 BR 指令只是将 R7 的内容赋给 PC 指针,没有对当前 PC 指针的内容进行保存,从而会出现一开始执行重注程序就跑飞的现象。

2 重注程序的编写

重注程序可以分为 3 类:补丁型重注程序、替代型重注程序和死循环重注程序。补丁型重注程序在执行完毕后返回主循环,继续执行固化程序,并作为主循环的一部分,主循环每循环一次就执行一次重注程序。替代型重注程序重新构造一个主循环,而不返回执行固化程序,为了避免看门狗复位,需要在重注程序的主循环中喂狗,为了能够替代原固化程序完成与上位机的通信功能,并影响其他中断,需要重新对中断向量表的位置进行设置。在空间相机控制器的地面测试中使用最多的是死循环重注程序。当上注死循环程序并执行后,DSP 进入一个无限循环,但没有喂狗操作,导致看门狗电路产生复位脉冲,使 DSP 相机控制器复位,这实质上是一个热启动过程。通常要求 DSP 相机控制器具有热启动计数功能,即当 DSP 相机控制器由于受到干扰或供电电压波动而热启动时,对热启动计数器加 1,并将热启动计数作为遥测返回。因此通过上注死循环重注程序可以同时验证 DSP 的程序重注功能、看门狗电路功能和热启动计数功能。

表 1 为重注程序结构。由于上注的重注程序可以各种各样,每包重注程序的长度也变化不一,因此需要通过参数长度来通知 DSP,和上位机通信的过程中需要连续收多少个字节的数据。重注程序包数和重注程序包序用于 DSP 在收到多包重注程序时对上注顺序的正确性进行检查。重注程序加载地址为本包重注程序需要加载到的绝对地址,重注程序代码长度给出了本包重注程序加载结束的位置。重注程序帧校验和用于对本包重注程序进行和校验,只有校验正确,才会进行加载。

表 1 重注程序结构

序号	数据含义	字节数	说明
1	参数长度	1	重注程序数据帧参数码的字节数
2	重注程序包数	1	标识重注程序的总包数
3	重注程序包序	1	标识当前包为第几包,从 00 开始
4	重注程序代码长度	1	本包重注程序的有效代码长度
5	重注程序加载地址	3	标识本包重注程序需要加载到的绝对地址
6	重注程序代码	-	重注程序有效代码
7	重注程序帧校验和	1	重注程序包数到重注程序代码的字节累加和的低字节

首先在 Code Composer Studio(简称 CCS)中使用 DSP 汇编语言编写死循环程序如下:

```
.text
_main:   AND 0Dfffh,ST
lps:     NOP
         BR lps
         RETSU
```

可以看出,在程序中首先关闭所有中断,然后进入一个无限循环中,循环中不作任何操作。关闭中断的目的是为了停止所有中断服务程序的运行,避免由于中断服务程序中的喂狗操作导致系统不复位的情况。对应的链接命令文件(*.cmd)编写如下^[5]:

```
MEMORY{
.....
SRAM: org=0091A000H len=00005000H
}
SECTIONS{
.....
.text: > SRAM
}
```

这里需要注意的是,DSP 程序中的标号在编译时由 CCS 根据 CMD 文件中 SECTION(段)指定的地址自动转换为真实的地址,因此必须将 SRAM 的 org 设定为重注程序的加载地址。由于本文要把死循环重注程序加载至地址 91A000H 处,因此将 SRAM 的 org 设定为 91A000H。编译后形成的十六进制目标代码如下:

```
02 F5 DF FF 0C 80 00 00 60 91 A0 01 78 80 00 00
```

其中 60 91 A0 01 对应 BR lps 指令,可以看出标号 lps 在编译时被自动替换为地址 91 A0 01。则按表 1 构造的完整重注程序如下:

```
18 01 00 10 91 a0 00 02 F5 DF FF 0C 80 00 00 60 91 A0 01
78 80 00 00 2D
```

由于死循环重注程序很短,一共只有 1 包,包序号为 00。

3 实验验证

为了验证本文提出的 DSP 空间相机控制器在轨程序重注方法和重注程序编写方法的正确性,将前面生成的死循环重注程序通过上位机发送给 DSP 空间相机控制器,用 CCS 通过 DSP 仿真器观察重注程序的加载情况。加载结果如图 2 所示,从图中可以看出,目标代码正确加载到了地址 91A000H~91A003H 处。

图 3 为开始执行重注程序后的结果。从图中可以看出 PC 指针变为 91A000H,全速运行后,PC 指针在 91A001H 和 91A001H 之间反复切换,说明重注程序得以正确执行。当不用仿真器,而直接从 PROM 中运行 DSP 程序时,在上注死循环重注程序并执行后,从遥测结果上

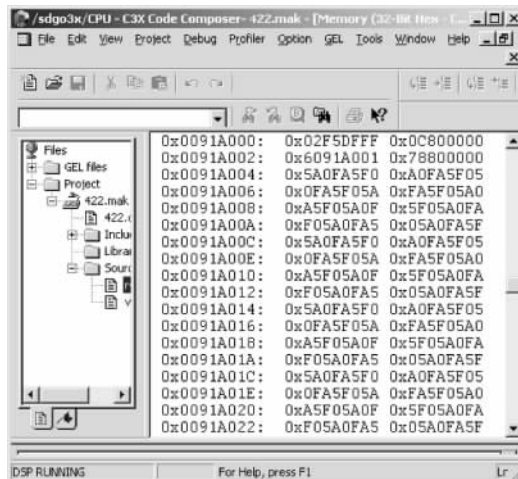


图 2 死循环重注程序加载结果

看热启动计数加 1,同时验证了 DSP 的程序重注功能、看门狗电路功能和热启动计数功能。说明本文提出的 DSP 空间相机控制器在轨程序重注方法和重注程序编写方法正确有效。

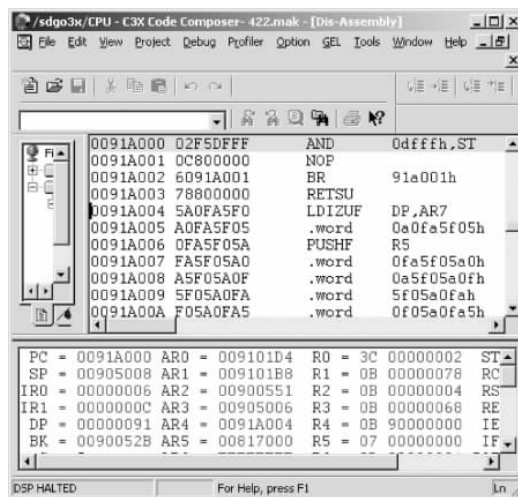


图 3 死循环重注程序执行结果

结 语

针对空间相机在轨发生故障或发现软件缺陷后难以修复这一难题,本文提出了一种基于 DSP 的空间相机控制器在轨程序重注方法,给出了重注程序接收、存储、校验和执行的流程和部分代码。并给出了 DSP 重注程序的编写和生成方法。利用该方法,当 DSP 的空间相机控制器在轨故障或发现软件缺陷时,可以将修复程序上注后执行。上注死循环重注程序的验证实验结果表明,本文提出的 DSP 空间相机控制器在轨程序重注方法和重注程序编写方法正确有效。



μC/OS-II 中软件定时器的研究与改进

王申良,梅静静,胡敏

(安徽理工大学,淮南 232001)

摘要: 研究了 μC/OS-II 实时操作系统中软件定时器功能,并分析了软件定时器的优缺点。给软件定时器加入优先级,使回调函数按优先级执行,提高系统的实时性。对改进后的系统在基于 Cortex-M3 内核的路虎 LPC1768 开发板上进行测试,经实验得出,改进后的系统实时性有所提高。

关键词: μC/OS-II; 软件定时器; 优先级

中图分类号: TP311.1 **文献标识码:** A

Research and Improvement of μC/OS-II Software Timer

Wang Shenliang, Mei Jingjing, Hu Min

(Anhui University of Science and Technology, Huainan 232001, China)

Abstract: This paper studies the function of software timer in μC/OS-II real-time operating system, and analyzes the advantages and disadvantages of the software timer. Adding the priority level to software timer, making the callback functions execute according to priority level, improves the real-time performance of the system. Testing the improved system on the LPC1768 development board based on Cortex-M3 kernel shows that the system performance is enhanced.

Key words: μC/OS-II; software timer; priority level

引言

μC/OS-II 具有小巧、性能稳定、开源等众多优点,并且 μC/OS-II 大部分用 ANSI C 语言编写,系统的移植非常容易^[1]。在 μC/OS-II 12.81 及以后的版本中^[2],加入了对软件定时器的支持,使得 μC/OS-II 操作系统更加完善。

μC/OS-II 是一种基于优先级的抢占式操作系统,实时性很强。而系统中软件定时器没有优先级,回调函数顺序执行,这样就降低了系统的实时性。因此,本文对软件定时器进行改进,定时器中加入优先级,回调函数按优先级执行,从而提高系统的实时性。

1 对软件定时器的介绍

μC/OS-II 系统中的时间管理功能包括任务延时与软件定时器,而软件定时器的主要作用是,对函数周期性或者一次性执行的定时,利用软件定时器控制块与“定时器轮”管理软件定时器。定时器控制块的结构如同任务控制块,创建一个定时器时,从空闲定时器控制块链表中得到一个空闲控制块,并对其赋值^[3]。

软件定时器也需要一个时钟节拍驱动,而这个驱动一般是硬件实现的,一般使用 μC/OS-II 操作系统中任务延时的时钟节拍来驱动软件定时器。每个时钟节拍 OSTmrCtr(全局变量,初始值为 0)增 1,当 OSTmrCtr 的

参考文献

- [1] 宗河. 美国航天飞机第 126 次飞行—第 5 次维修“哈勃”空间望远镜[J]. 国际太空,2009(8):31-36.
- [2] 孔德柱,刘金国,吕世良,等. 空间相机像移速度计算方法及 DSP 实现[J]. 光学精密工程,2009,17(8):1935-1941.
- [3] Wissam Rabadi, Raj Talluri, Klaus Illgner, et al. Programmable DSP Platform for Digital Still Cameras[J]. Texas Instruments Technical Journal,2000,17(1):1-11.

- [4] 武星星,刘金国,等. 基于混合编程的空间相机控制器自检方法[J]. 光学精密工程,2008,16(9):1635-1641.
- [5] Texas Instruments. TMS320C3x/C4x Optimizing C/C++ Compiler User's Guide,2001.

武星星(副研究员),主要研究领域为空间相机嵌入式系统设计、智能控制等。

(收稿日期:2010-12-14)