

浅谈 TMS320C6000 中断程序的实现方法

杨 辉, 秦品健

(中国科学院等离子体物理研究所, 合肥 230031)

摘要: 主要讨论中断服务程序在 TMS320C6000 DSP 中的几种用 C 语言来实现的方法。

关键词: TMS320C6000; DSP; C 语言; 中断服务程序

中图分类号: TP274 **文献标识码:** B **文章编号:** 1009—2552(2002)09—0036—02

General Methods of TMS320C6000 Interrupt Program Realization

Yanghui, Qinpinjian

(Chinese Academy of Sciences(ASIPP), Hefei 230031, China)

Abstract: Several methods of setting up TMS320C6000 DSP interrupts in C language are discussed in this paper.

Keywords: TMS320C6000; DSP; C language; Interrupt service program

0 引言

数字信号处理(DSP)技术近年来取得了高速发展,目前 DSP 芯片已经广泛的应用于通信、图象处理、语音处理、雷达等领域。TI 公司是当今世界 DSP 芯片的主要供应商之一,其 TMS320C6000 是 TMS320 系列产品中的新一代高性能 DSP 芯片。由于中断处理在现代实时处理领域占有举足轻重的地位,本文着重讨论在 C6000 系列 DSP 中如何实现中断的几种方法。

1 中断的定义及在 TMS320C6000 DSP 中的结构

中断是为使 DSP 具有对外界异步事件的处理能力而设置的。通常 DSP 工作在包含多个异步事件环境中,当这些事件发生时,DSP 执行这些事件所要求的任务。中断就是要求 CPU 暂停当前的工作,转而去处理这些事情,处理完以后,再回到原来被中断的地方,继续原来的工作。

TMS320C6000 DSP 中有三种类型的中断,即复位(RESET)中断、不可屏蔽中断(NMI)和可屏蔽中断(INT4 ~ INT15)。

用高级语言写 DSP 的结构程序是现在编程的趋势和必然,这里用 C 语言来实现中断服务程序的编写。一般中断服务程序的编写有以下几个步骤:

- (1)选择中断源和写中断服务程序。
- (2)建立和初始化中断矢量表。
- (3)使能合适的寄存器。
- (4)编译、连接。

2 举例说明以几种不同的方法来实现中断的配置和使能

现在以 TMS320C6201 DSP 的 INT14 为例,举例说明中断程序的几种 C 语言的实现方法,INT14 对应的中断源为定时器 0,假设项目文件名为 myprojection.mak.

第一步,建立和初始化中断矢量表。

用两种方法来配置中断矢量表,一是用汇编语言来描写中断矢量表,程序(vectors.asm)如下:

```
.ref    _myISR    //定义中断服务程序的入口地址

.ref    _c_int00 //主程序的入口地址
.sect   "vectors" //定义中断矢量表的段名

RESET_RST:
MVKL.S2 _c_int00, B0
MVKH.S2 _c_int00, B0
B.S2 B0
NOP 5
```

收稿日期:2002—01—14

作者简介:杨辉,男,1975 年出生,1999 年毕业于合肥工业大学精密仪器专业,现在主要从事 TMS320C6000 DSP 的软件编程和数据采集系统的软件编写工作。


```

/* 定时中断器,一旦赋值就执行控制 */
TrackTimeOut * pTrackTimeOut;
bool bTrackTimeOut_Alloced = 0;
/* 接收的主端数据 */
INPUT_STRUCT Input_Struct;
bool bInput_Alloced = 0;
/* 动态内存地址分配给规划数据 */
JOINTS_ADVALUE * adJoints_List;
/* 采样到的数据 */
SAMPLE_OUT Sample_Out;
/* 定时数目计数 */
WORD wTimeout_sum = 0;
/* 定时数目累积到该值后送规划数据的下一个点 */
/* /
WORD wTimeout_sum_upper_limit = 0;
/* 跟踪了多少个规划点 */
/* /
WORD wPassed_Plannum = 0;
TrackTimeOut:: TrackTimeOut ( DWORD msec ) :
VAsyncTimeOut(msec)
{}
/* 定时中断处理程序。功能:实时采样,PID 运算
及控制 */
VOID TrackTimeOut:: handler ( VMHANDLE hVM,
THREADHANDLE th, PCLIENT_STRUCT pRegs,
DWORD lag)
{.....}
/* 动态初始化 */
BOOL RobotvxdDevice:: OnSysDynamicDeviceInit()
{
dout << "OnSysDynamicDeviceInit()" << endl;
AD_Card_Init();
}

```

```

StopMachine();
VTD_Begin_Min_Int_Period(1);
return TRUE;
}
/* 动态退出 */
BOOL RobotvxdDevice:: OnSysDynamicDeviceExit()
{
VTD_End_Min_Int_Period(1);
StopMachine();
dout << "OnSysDynamicDeviceExit()" << endl;
return TRUE;
}
/* VxD 与应用程序的数据交互 */
DWORD
RobotvxdDevice:: OnW32DeviceIoControl ( PIOCTL-
PARAMS pDIOCPParams)
{.....}

```

编译后即可将 Robotvxd.h 和 Robotvxd.cpp 编译成 Robotvxd.vxd。

在 WIN32 应用程序中用 CreateFile 函数打开 Robotvxd.vxd,启动虚拟驱动程序。然后可调用 API 函数 DeviceIoControl 调用虚拟驱动程序完成实时位置闭环伺服控制。

参 考 文 献

- [1] 孙守迁. 微机操纵液压挖掘机规划级控制理论和方法的研究. 浙江大学博士学位论文, 1991. 9.
- [2] 袁南儿, 王万良, 苏宏业. 计算机新型控制策略及其应用. 清华大学出版社, 1998. 18 - 62.
- [3] 彭礼孝. 虚拟设备驱动程序开发起步与进阶. 人民邮电出版社.
- [4] 田志刚. Windows95 设备驱动程序体系结构的分析. 微计算机应用, 99. 9.

责任编辑:杨立民

~~~~~  
(上接第 37 页)/\* 全局中断使能 \*/

```

timer0_init(); /* 初始化 TIMER0,并启动,具体
程序这里省略了 */
while(1){} /* 等待中断 INT14 的产生 */
}

```

第四步 编译、连接。

把所有需要编译和连接的文件加入到 myprojec-  
tion.mak 文件中去,其中包括有 vectors.asm、main.  
c、.cmd 文件,如果用的是 DSP/BIOS 配置的,则还  
有 .cdb 文件,然后加入所需要用的 .lib 文件,程序就

可以编译、连接和执行了。

### 3 结束语

按照一定的流程,用 C 语言写中断程序是一个简单而行之有效的办法,上面讲述的无论包括中断矢量的初始化和中断使能的各个方法,可以在实践中选取自己喜欢的一个或多个组合来完成自己程序的主体框架,当然在编写 DSP 程序的过程中还需要注意很多其他地方的小节,这需要每位程序员在编程中慢慢琢磨!

责任编辑:杨立民