

基于 TMS320VC5402 DSP 的 HPI 方式自举的设计与实现^①

HPI Bootloader Based on TMS320VC5402 DSP

邹 翼 曾文海 陈续喜 (湖南大学 电气与信息工程学院 湖南 长沙 410082)

摘 要: 在 DSP 系统中, 程序加载是其关键技术之一。TMS320C54xx DSP 支持多种自举方式, 在深入分析各种加载方法之后, 选择了利用单片机来实现 DSP 的主机接口自举加载方式。结合单片机和 DSP 芯片的特点, 设计了单片机和 DSP 主机接口的连接电路, 编写了其运行程序, 实现了 DSP 的主机接口方式自举加载。实验结果表明, 该方法可以降低系统的复杂度和成本, 提高系统的通用性, 有很大的实用价值。

关键词: 主机接口 单片机 自举 数字信号处理器 引导模式

1 引言

近年来, 数字信号处理器(DSP)芯片以其强大的运算能力在通信、图像处理、医疗设备、网络设备、自动控制等各个领域得到了广泛的应用。由于 DSP 的控制功能不是十分强大, 在应用中 DSP 往往专门负责复杂的运算, 而另外使用一个主机对整个系统的运行进行控制。将 DSP 和单片机构成双 CPU 处理平台可以充分利用 DSP 对大容量数据和复杂算法的处理能力, 以及单片机接口的控制能力。DSP 和单片机功能互补, 已经成为很多 DSP 系统中的两个重要部分。

程序自举是开发 DSP 系统的关键技术之一, 自举从本质上说就是在 DSP 启动后通过某种方式从外部获取运行代码并装载到其程序区内全速运行。在深入分析了各种加载方法之后, 结合本 DSP 系统的特点, 采用了单片机来实现 DSP 的主机接口方式自举加载。

2 TMS320C54xx的引导过程

TMS320VC5402^[1]是 54xx 系列 16 位定点 DSP 芯片, 是德州仪器公司推出的具有较高性价比的定点数字信号处理器, 其内部总线结构为改进的哈佛结构、专用的硬件逻辑单元、片内 RAM、片内 ROM、片内外设和专业指令集。此外它还具有灵活多样的引导加载方法。TMS320C54xx 的自举加载共有并行

EPROM(Flash)、并行 I/O、串行口、HPI 口和热自举五种方式^[2]。上电或硬件复位后首先采样 MP/\overline{MC} 引脚, 如果为高电平, 则选择微处理器模式, DSP 从片外的 FF80H 开始执行。如果为低电平, 则选择微计算机模式, DSP 从片内的 FF80H 开始执行引导程序, 在片内 ROM 的 FF80H 地址处有一条跳转到 Bootloader 程序的语句, Boot 在运行搬移程序之前, 首先进行初始化, 初始化工作包括: 使中断无效 ($\overline{INTM}=1$), 内部 RAM 映射到程序/数据区 ($OVLY=1$), 对所有程序和数据区均设置 7 个等待状态 ($SWWSR=7FFFFH$), 当程序和数据空间切换时, 插入一个等待周期 ($BSCR=0FFFFH$) 等。然后检查是否有效, 以此决定是否从主机接口(HPI)加载。如果无效, 系统继续检测其他几种引导方式是否发生。如果其他模式引导都无效, 回头再从 HPI 开始检测, 如此反复直至检测到某种引导模式有效为止。选择过程简化为如图 1 所示:

当 DSP 检测到 HPI 自举方式有效后, DSP 片内 0x7f 地址的值被置为 0, Bootloader 不断检验 0x7f 地址处是否出现了可用的程序指针的跳转地址。当发现该地址内的值不为 0 时, 即判定为 DSP 已由外部主机进行了 HPI 自举程序加载, 并按照该值跳转 PC 指针, 开始运行, 从而完成 HPI 方式自举。

① 收稿时间:2008-09-09

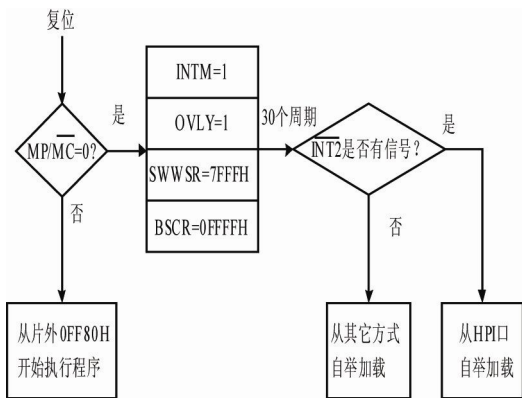


图 1 自举加载方式的选择过程

3 单片机和DSP的HPI口连接的电路设计

3.1 HPI-8 接口概述

主机接口(HPI)是主设备或主机处理器与 DSP 的接口,而增强型 8 位主机接口,通常也表示为 HPI-8^[3],是标准 8 位主机接口(HPI)的增强版本。HPI-8 使用 8 位的外部数据接口,但可以为 DSP 提供 16 位的数据传输。其传输过程是: HPI-8 寄存器先将 16 位的数据分成两个 8 位数据进行传输,并由 HBIL 信号决定当前传输的数据是高字节还是低字节;当接收方收到全部数据后,将按约定的顺序将两个 8 位数据合并成 16 位数据。

HPI 主要由五个部分组成,如图 2 所示:

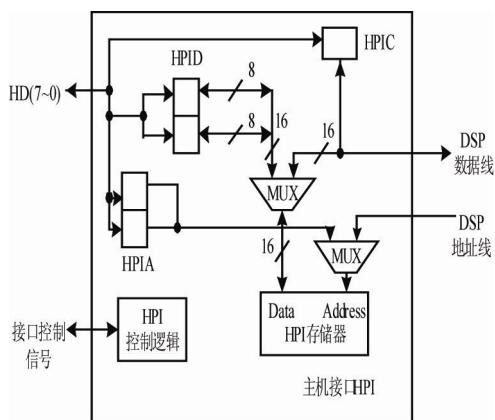


图 2 主机接口的组成框图

① HPI 存储器(DARAM): 用于 C54xx DSP 与主机之间传送数据。

② HPI 地址寄存器(HPIA): 只能由主机对其直接访问。该寄存器中存放着当前寻址 HPI 存储单元的地址。

③ HPI 数据锁存器(HPID): 只能由主机可以对其直接访问。如果当前进行读操作,保存着从 DSP 片内存储器读出的数据;如果当前进行写操作,包含了将要写到 DSP 片内存储器的数据。

④ HPI 控制寄存器(HPIC): DSP 和主机都能对它直接访问,包含控制和状态位。

⑤ HPI 控制逻辑: 用于处理 HPI 与主机之间的接口信号。

HPI-8 的使用是通过 HPIA、HPIC 和 HPID 三个寄存器赋值实现的。主机通过外部引脚 HCNTL0 和 HCNTL1 选中不同的寄存器。由于 HPIC 是 16 位寄存器,而 HPI-8 是 8 位的数据宽度,所以在主机向 HPIC 写数据时,需要发送两个一样的 8 位数据,首先发送第一个字节,此时 HBIL 为 0,然后发送第二个字节,此时 HBIL 为 1。而地址寄存器 HPIA 选择后,直接向它写数据就可以了,但是要注意高位和低位的顺序。另外, HPIA 具有自动增长的功能,在每写入一个数据前和每读入一个数据后, HPIA 会自动加 1。这样只需设定一次 HPIA,即可实现连续数据块的写入和读出^[4],由于主机不需要更新 HPIA 寄存器的值,所以极大的提高了系统的性能。

3.2 单片机和 DSP 的接口电路

本系统中单片机采用 STC89LE 系列的 51 单片机, TI 54xx 系列的 DSP 采用的是 TMS320VC5402, 由于工作电压匹配,本系统不需要电平转换。HPI-8 的 8 位数据总线(HD0~HD7)和单片机的 P00~P07 连接,负责从主机接收数据^[5]。HBIL 和 P20 连接,用于识别传输的是第 1 个或第 2 个字节,HCNTL0/1 分别和单片机的 P21、P22 连接,用于选择内部寄存器,根据其电平高低,可以分别选择 HPI 控制寄存器, HPI 地址寄存器,数据锁存寄存器(地址不变方式和地址自加方式)。HR/ \bar{w} 读写控制信号接 P23 用于读写控制。 \bar{HCS} 、 $\bar{HDS1}$ 、 $\bar{HDS2}$ 分别接 P24、 \bar{RD} 、 \bar{WR} ,用于片选、读选通和写选通或数据选通。 \bar{HAS} 、 \bar{HRDY} 、 \bar{HPIENA} 接高电平。当 \bar{HAS} 接高电平时,后面的 \bar{HCS} 、 $\bar{HDS1}$ 和 $\bar{HDS2}$ 实际上控制着 HCNTL0/1、HR/ \bar{w} 和 HBIL 的采样。在上电复位时 \bar{HINT} 会产生一个低压脉冲,如果与直接相连系统就判断为 HPI 自举模式,所以把它们短接。其具体电路图如图 3 所示:

由上面硬件电路图,可以得出 HPI 主机接口的各个寄存器分配的地址和操作的类型。例如: #define

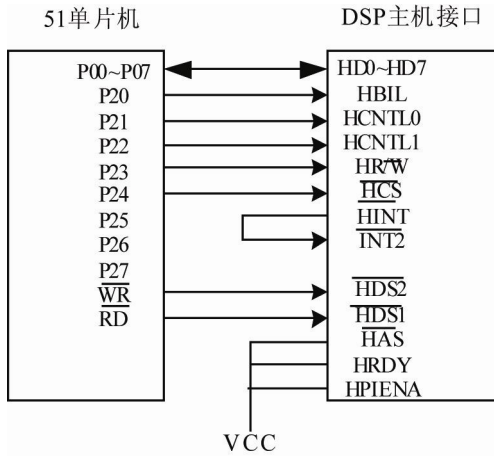


图 3 硬件电路设计的连接框图

WR_HPILO_DXBYTE[0xE200],单片机引脚 P27-P20 为(1110 0010),片选信号有效($\overline{HCS}=0$),写操作($HR/\overline{W}=0$),选中数据锁存器($HCNTL1/0=01$),传输的是第一字节(HBIL=0),就可以表示选择了 HPI 数据锁存器,并且其操作类型是以自加的方式写第一个字节。具体见表 1

表 1 HPI 寄存器端口的地址和操作方式

名称	地址	操作方式
HPICL	0xE000	分别写 HPIC 的第 1, 2 个字节
HPICH	0xE100	字节
WR_HPILO_D	0xE200	以地址自加方式分别写
WR_HPIHO_D	0xE300	HPID 的第 1, 2 个字节
WR_HPIL_A	0xE400	分别写 HPIA 的第 1, 2 个字节
WR_HPIH_A	0xE500	字节
WR_HPIL1_D	0xE600	以地址不变方式分别写
WR_HPIH1_D	0xE700	HPID 的第 1, 2 个字节
RD_HPILO_D	0xEA00	以地址自加方式分别读
RD_HPIHO_D	0xEB00	HPID 的第 1, 2 个字节
RD_HPIL1_D	0xEE00	以地址不变方式分别读
		HPID 的第 1, 2 个字节

4 HPI方式自举的软件设计

在 DSP 集成开发环境 CCS^[6] (Code Composer Studio)中,用 C 语言或汇编语言编写好源程序,进行编译、汇编、链接,最后生成的是.out 文件,该文件的格式不利用被单片机识别,通过一个批处理文件把.out 文件转化为易于被单片机识别的.h 头文件,这样 DSP 运行时所需的代码就包含在这个头文件中,把它作为单片机程序的一部分下载到单片机中,这样

DSP 运行的程序就加载到单片机的存储区了。

转化后的头文件具有这样的格式:

① 由许多行构成,每行有很多字节。

② 每行第一个字节的数值表示此行中需传到 DSP 中的字节个数。

③ 第二,三个字节分别作为一个字的高 8 位和低 8 位,一起构成此行数据在 DSP 中存储的首地址,余下的为要传输到 DSP 中的数据。

④ 最后一行只有一个 0,作为结束标志。

根据头文件的格式和 HPI 主机接口的各个寄存器分配的地址和操作的类型,就可以编写单片机程序了。单片机程序采用流行的 Keil C51 编写,Keil uVISION2 是众多单片机应用开发软件中优秀的软件之一,它支持众多不同公司的 MCS51 架构的芯片,它集编辑,编译,仿真等与一体,支持汇编和 C 语言的程序设计,功能十分强大^[7]。在编写的单片机程序中,先判断每行的第一字节是否为 0,以判断其是否结束,如果不为 0,根据 2, 3 字节计算其存放首地址,接着传送数据,此行数据传完后,判断是否出错,如果错误,重新传输此行,如此反复,直到遇到结束标志。最后,把程序的入口地址写到 0x007f,自举加载完成。其流程如图 4 所示:

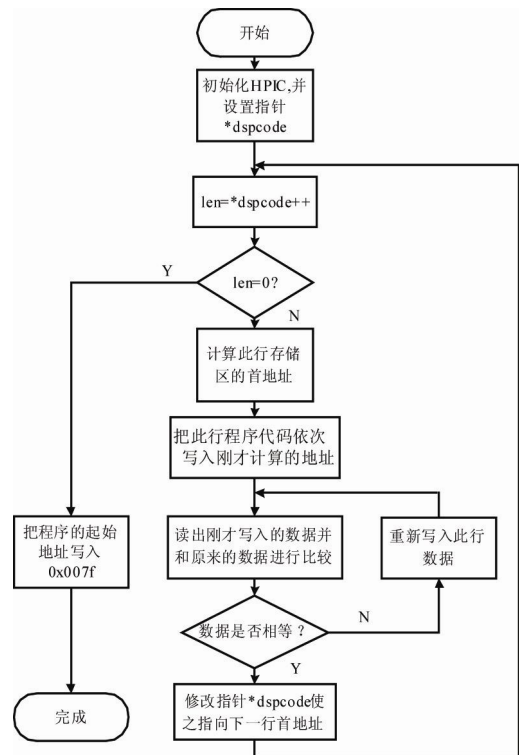


图 4 单片机和 DSP HPI 数据传输程序流程图

其核心代码及说明如下:

```

HPICL=0x08; //初始化 HPI 控制寄存器, 写入第 1
字节
HPICH=0x08; //初始化 HPI 控制寄存器, 写入第 2
字节
len=*dspcode++; //一行中需传输的数据个数
if(len==0) break; //如果长度为 0, 跳出, 传输完
成
address=((*dspcode++)*256)+(*dspcode  ++);
//数据在 DSP 中存储的地址
address --; //写操作前加 1, 所以先自减
WR_HPIL_A= address /256;
//向 HPI 地址寄存器写地址高 8 位
WR_HPIH_A= address %256;
//向 HPI 地址寄存器写地址低 8 位
for(i=0;i<len/2;i++) //循环次数, 因为每次传输
2 字节, 所以数据长度除以 2
{
WR_HPIL0_D=*dspcode ++;
//以地址自加的方式向 HPI 数据锁存器写数据的高 8
位
WR_HPIH0_D=*dspcode++;
//以地址自加的方式向 HPI 数据锁存器写数据的低 8
位
}

```

经过上述软硬件设计后, 把在 Keil 中生成的 .hex 文件下载到单片机中。DSP 上电或硬件复位后, 自动检测 MP/ 引脚信号, 由于在硬件设计时已经保证了 DSP 从 HPI 口进行自举加载, 所以单片机便按照程序把 DSP 的运行代码加载到 DSP 的 RAM 中, 最后, 把 DSP 程序的入口地址写到 0x007f, 自举加载完成,

DSP 就能按照程序正常的运行了。掉电后程序保存在单片机的 ROM 中, 下次上电或复位时, 单片机重新对其进行加载, 这样 DSP 系统就可以独立的运行了。

5 结语

本文分析了 54xx 系列 DSP 的引导过程, 给出了一种利用单片机控制 TMS320VC5402 DSP 的 HPI 口进行程序自举的硬件连接电路图和软件设计方案, 并给出了部分核心代码。这种方法省掉了 DSP 的 EPROM、使 DSP 只使用片内 RAM、实现简便、附加硬件少、成本低, 可为设计 DSP 系统的程序自举提供一定的参考。

参考文献

- 1 TI, TMS320VC5402 Fixed-Point Digital Signal Processor Datasheet. [2000203].
- 2 戴明桢, 周建江, TMS320C54x DSP 结构、原理及应用. 北京: 北京航空航天大学出版社, 2001.
- 3 TI, TMS320C54x DSP Reference Set Volume5: Enhance Peripherals. Texas Instruments Incorporated (SPRU302).
- 4 张雄伟, 曹铁勇. DSP 芯片的原理与开发应用(第 2 版). 北京: 电子工业出版社, 2000.
- 5 邓思豪, 曾春年. C54X DSP 的 HPI 口与 PC 机并口的接口技术. 系统工程与电子技术, 2004, 26(11): 1749 - 1752.
- 6 TI, TMS320C54x DSP Reference Set, Volume 4: Applications Guide, 2001.
- 7 徐爱均, 彭秀华. Keil Cx51 V7.0 单片机高级语言编程与 uVision2 应用实践. 北京: 北京电子工业出版社, 2004.