

TMS320VC5501 的自启动方式研究

卓 为, 管 庆

(电子科技大学 美国德州仪器 DSPs 技术中心 四川 成都 610054)

摘 要: 在介绍 TI DSP 芯片 TMS320VC5501 系列的自启动 (BootLoader) 模式的基础上, 重点分析了 EMIF 的并行启动模式。阐述了 BootLoader 的概念与基本原理, 并与 TMS320VC5402 的自启动进行比较, 最后给出 BootLoader 的具体工程实现, 从而指导用户完成 DSP 目标系统的产品化设计。

关键词: DSP; EMIF; TMS320VC5501; BootLoader

中图分类号: TP33

文献标识码: A

文章编号: 1004-373X (2005) 20-050-03

BootLoader Investigation of TMS320VC5501

ZHUO Wei, GUAN Qing

(Texas Instruments DSPs Technology Center, University of Electronic Science & Technology of China, Chengdu, 610054, China)

Abstract: The BootLoader modes of TMS320VC5501 of TI DSP have been introduced in this paper, following by analysis of parallel EMIF boot from 16 b external asynchronous memory, and based on the concepts and basic principles of the technology of BootLoader are introduced. The BootLoader of TMS320VC5501 and that of TMS320VC5402 have been compared in the article. And how to realize BootLoader is also discussed in detail to help DSP engineers accomplish the DSP design in product period.

Keywords: DSP; EMIF; TMS320VC5501; BootLoader

在 CCS 开发环境下, PC 机通过不同类型的 JTAG 电缆与用户目标系统中的 DSP 通信, 帮助用户完成调试工作。当用户在 CCS 环境下完成开发任务, 编写完成用户软件之后, 需要脱离依赖 PC 机的 CCS 环境, 并要求目标系统上电后可自行启动并执行用户软件代码, 这就需要用到 BootLoader 技术。DSP 系统的 BootLoader 是指在系统上电时, DSP 将一段存储在外部非易失性存储器中的代码搬到内部的高速存储单元中执行。这样既利用了外部存储单元扩展 DSP 本身有限的 ROM 资源, 又充分发挥了 DSP 内部资源效能。同时, BootLoader 也指由 TI 在生产芯片时预先烧制在 DSP 片上的 ROM 中, 完成该功能的一段代码名称。

1 BootLoader 的基本原理与模式

本文主要介绍 TMS320VC5501/C5502 中的 BootLoader 技术。因为 C5501/C5502 是 RAM 型器件, 掉电后不能保持任何用户信息, 所以需要用户把执行代码存放在外部的无挥发存储器内, 在系统上电时, 通过 BootLoader 将存储在外部媒介中的代码搬到高速的片内存储器或系统中的扩展存储器内, 搬移成功后自动去执行代码, 完成自启动。

BootLoader 技术提供很多种不同的启动模式:

(1) HPI 口启动

由外部处理器(即主机)将执行代码通过 C5501 的 HPI 口搬到片内 RAM, 当代码搬移完毕, 开始执行代码。这种方式下无需启动表。

(2) 外部 16 b 的异步存储器的并行 EMIF 启动

通过 EMIF 接口从外部的 16 b 的异步存储器读取启动表, 这种启动方式将 CE1 空间配置为 16 b 异步操作, 并且从字地址 200000H 开始读取启动表。

(3) 通过 MCBSP0 的标准 16 b 串行启动

通过工作在标准模式下的 MCBSP0 从处理器接收启动表, 并根据启动表的信息装载代码。

(4) 通过 MCBSP0 的 SPI E²PROM 串行启动

通过 MCBSP0 从 SPI 格式的串行 E²PROM 装载启动表。

在这种模式下, 将 MCBSP0 配置为工作在 SPI 方式下, 就可以从 SPI E²PROM 或兼容 SPI 的串口读取数据。这种自启动方式只支持基于 24 b 编址的 SPI 设备。

(5) 通过 I²C 口的 I²C E²PROM 启动

通过 I²C 口从 I²C E²PROM 或工作在 I²C 模式下的设备上读取启动表并将其存储到片上存储器中。这种模式只支持使用 2 B 为内部存储空间编址的 I²C 设备。

(6) 串口启动

通过串口从主机读取启动表, 根据启动表的信息装载代码。

(7) 无启动模式——直接从外部异步存储器执行

C5501 提供两种无启动模式(即直接从16 b 或32 b 外部异步存储器执行)。在这种方式下,DSP 上电时片上 ROM 被禁用(MPNMC=1),并且强制从外部存储空间的0xFFFF00 地址处取第一条指令。

2 外部16 b 的异步存储器的并行EMIF 启动模式分析

在上述各种启动模式中,外部16 b 的异步存储器的并行EMIF 启动模式实现简单,速度较快,在实际系统中应用也最为广泛,本文重点介绍这种启动模式。

启动方式是通过设置3 个启动方式选择引脚BOOTM [2: 0] 来设置的(如表1),这3 个引脚的值在复位时被采样。BOOTM 引脚与通用I/O 引脚(GPIO)复用。

表1 不同启动方式下的BOOTM 引脚电平值

BOOTM2 (GPIO2)	BOOTM1 (GPIO12)	BOOTM0 (GPIO0)	
0	0	0	无启动模式(直接从16 b 外部异步存储器执行)
0	0	1	通过MCBSP0 的SPI E ² PROM 串行启动
0	1	0	通过MCBSP0 的标准16 b 串行启动
0	1	1	外部16 b 的异步存储器的并行EMIF 启动
1	0	0	无启动模式(直接从32 b 外部异步存储器执行)
1	0	1	HPI 口启动
1	1	0	通过I ² C 口的I ² C E ² PROM 启动
1	1	1	串口启动

当复位时BOOTM [2: 0] = 011b 时,就选择了外部16 b 的异步存储器的并行EMIF 启动方式。这种自启动方式从16 b 的外部异步存储器读取启动表,数据宽度是在自启动程序复位时配置的,在自启动过程中不能修改。

并行EMIF 启动的方式,从字地址200000H 开始读取启动表,这段地址在CE1 空间。包含启动表的外部存储空间必须从这个地址开始,代码执行的入口地址也包括在启动表中,并且是可编程的。

当初初始化这种方式时,EMIF 的可编程时间参数是如下设置的:

- (1) READ SETUP 为 15 cycles (1111b)。
- (2) READ STROBE 为 63 cycles (111111b)。
- (3) READ HOLD 为 7 cycles (111b)。

READ SETUP, READ STROBE 和 READ HOLD 被设置为最保守的值来确保适应大范围的存储器速度。虽然如此,但是如果认为这个默认设置太慢(每次访问需85 个时钟周期)可以使用端口地址(port-address) 寄存器来重新设置。这些时间参数是由 EMIF CE1 空间控制寄存器1 (CE1_1) 来设置的。

必须注意的是在启动过程中改变 EMIF 的这些时间参数可能会导致自启动失败。外部的CE1 空间必须一直保持设置为异步存储,并且数据宽度必须和原来选择的启动

模式一致。在重新配置 CE1_1 EMIF 寄存器时,写到 MTYPE 中去的值必须和原来的启动模式匹配。对 EMIF 控制寄存器的修改在变为有效时有一些延迟。在配置生效前不能对 EMIF 进行读请求。所以在重新配置 EMIF 后,启动表的入口必须跟不少于10 个 EMIF 时钟周期,使 EMIF 的配置完成。

同时还要注意,由于时钟生成频率是基于周期的,使用寄存器配置属性来改变时钟生成频率会改变 EMIF 生成的存储时间信息。仔细确认时钟和编程配置的 EMIF 信息与使用的外部存储器的存储时间信息一致。

在这种启动模式下,启动开始时 GPIO4 变为低电平。接着,在执行启动表中的可编程延迟时,GPIO4 变为高电平。当延迟结束,GPIO4 又变为低电平。在启动过程的装载结束时,GPIO4 又变为高电平,而且 DSP 将从程序的入口地址处开始执行。GPIO4 对于存储器来说不是必须的,但是如果有其他源为 EMIF 生成数据时可以用来作为握手信号。要使用并行 EMIF 启动模式,复位时 GPIO6 引脚必须拉高。否则,EMIF 会被禁用,BootLoader 程序就不能从外部存储器读取启动表。

EMIF 并行启动有如表2 所示的两种方式,通过将相应引脚上拉或下拉即可实现需要的启动方式。

表2 EMIF 并行启动GPIO 引脚电平值

BOOTM [2: 0]	GPIO2	GPIO6	GPIO7	
011	0	1	0	BootLoader 将 CE1 空间配置为16 b 异步存储内部晶体提供时钟源 EMIF 和 HPI 可用 (HPI 工作在复用模式)
011	1	1	0	BootLoader 将 CE1 空间配置为16 b 异步存储外部时钟提供时钟源 EMIF 和 HPI 可用 (HPI 工作在复用模式)

3 与 TMS320VC5402 自启动方式的比较

C55x 是 TI C5000 系列 DSP 中最新的子系列。C55x 是从 C54x 发展起来的,并与之源代码兼容,下面主要比较两种系列的并行启动模式。

TMS320VC5402 并行引导 是从数据空间并行启动。启动表可以是 4000H~0FFFFH 的任意有效地址。该地址应在系统上电前,存放在 I/O 空间的 0FFFFH 或数据空间的 0FFFFH 中。启动程序根据这个地址读入启动表的第一个字,如果是 08AAH,说明是 8 位并行启动;如果是 10AAH,说明是 16 位并行启动(C54x 中的地址为字地址)。

TMS320VC5501 并行引导 通过 EMIF 接口从外部的 16 b 的异步存储器读取启动表,这种启动方式将 CE1 空间配置为 16 b 异步操作,并且从字地址 200000H 开始读取启动表。

通过 EMIF 接口从外部的 16 b 的异步存储器读取启

动表。该启动模式只有 16 位并行启动一种方式。

C5402 的启动表可以由 4000H~0FFFFH (字地址) 的任意有效地址, 而 C5501 的启动表只能是 200000H (字节地址) 的有效地址, 因而 C5402 比 C5501 能更为灵活地选择启动表的起始地址; 但这是以操作的复杂性为代价的。C5402 启动表的起始地址存放在 I/O 空间的 0FFFFH 或数据空间的 0FFFFH 中, 而这个地址需要在烧制 EPROM 时, 在烧制的文件的地址 0FFFFH 处手工修改。

C5501 在启动时不需要判断 16 位还是 8 位启动, 启动更加简单。

4 如何建立启动表

如上所述, 为完成系统自启动, 需要为 BootLoader 提供启动表, 在里面写入 BootLoader 需要的所有数据。该启动表可以使用 TMS320VC55x 汇编语言提供工具包提供的十六进制转换工具来生成, 该工具文件名为 hex55.exe。下面就介绍该工具 hex55 调用格式及常用选项。十六进制转换工具 hex55 的调用格式为:

hex55 [-options] filename

-options: 提供附加信息用来控制十六进制转换处理过程。可在命令行里或一个命令文件里使用多个选项, 并且除了选项 -q (quiet) 必须用在其他选项前外, 不区分顺序与大小写;

filename: COFF 文件名或命令文件名;

其常用选项如下:

(1) 通用选项: 用来控制转换工具的全部操作

-map filename: 该选项能生成转换报告;

-o filename: 指定转换输出的十六进制文件名;

(2) 映像选项: 建立一段目标存储器的连续映像

-fill value: 指定填充在段与段之间的值, 只有使用选项 -image 时才有效;

-image: 指定使用映像模式;

-zero: 在使用映像模式时将起始地址复位至 0;

(3) 存储器选项

-memwidth value: 定义 DSP 系统存储器宽度 (默认值为 16 b);

-romwidth value: 定义用户 ROM 存储器宽度 (默认值由使用格式确定);

(4) 输出选项, 指定输出十六进制文件的格式, 如下所示:

输出选项	输出格式	可寻址宽度	默认用户存储器宽度
-a	ASCII-Hex	16	8
-i	Intel	32	8
-m1	Motorola-S1	16	8
-m2/-m	Motorola-S2	24	8
-t	TI-Tagged	16	16
-x	Tektronix	32	8

其中, 可寻址宽度决定了输出格式所支持的地址信息

比特数, 所以 16 b 寻址宽度最高仅支持 64 k 地址。

(5) 启动选项

-boot: 将所有已初始化的段转换到启动表里;

-bootorg: 定义启动表的源地址;

-e: 指定代码搬移完成后开始执行的入口地址, value 可为数值地址 (C5501 中必须为字节地址) 或全局符号;

-parallel16: 用于 16 b 并行启动 (-memwidth 16 and -romwidth 16);

-parallel32: 用于 16 b 并行启动 (-memwidth 16 and -romwidth 32);

-serial8: 用于 8 b 串行启动 (-memwidth 8 and -romwidth 8);

-serial16: 用于 8 b 串行启动 (-memwidth 16 and -romwidth 16);

-vdevice: revision: 说明器件版本。

除了以上使用命令行的形式进行文件格式转换外, 还可以使用命令文件的形式。

5 举例说明

最后以实际开发的项目为例来说明 C5501 的 boot loader。在项目开发中, 使用的是银杏科技发展有限公司与德州仪器培训中心联合开发的 des320pp-u 教学实验箱。在该教学实验箱上使用 C5501 子卡。该教学实验箱提供两块 8 b EPROM 映射到 C5501 的 CE1 空间 (字节地址 200000H 开始), 通过设置 BOOTM [2: 0] 为 011, 选择 EMIF 并行启动方式。

接下来本文以 C5501 为例说明启动表的生成。我们的项目是用 CCS 开发完成的, 使用十六进制转换工具将编译后的 COFF 文件转换为 .hex 文件, 最后将 .hex 文件烧制到用户 EPROM 中。

用上面介绍的 2 种方法来转换生成的 COFF 文件:

方法一: 因为使用的外部存储器是 8 位的 EPROM, 且使用的是 EMIF 并行启动模式, 所以就要使用两片 8 b EPROM 来进行启动。

生成的文件数 = memwidth/romwidth;

做一个批处理文件, 例如下例批处理文件 boot.bat:

```
rem boot.bat
hex55 %1 -boot -v5510:2 -memwidth 16 -romwidth 8 -e
c:\int00 -a
```

其中 "%1" 是调用 boot.bat 时其后接的第一个参数, 为需要转换的 COFF 文件名。

当在 DOS 键入:

```
C:\boot.bat filename
```

boot.bat 将调用 hex55 将输入的 COFF 文件转换为 ASCII 格式的输出文件。

(下转第 56 页)

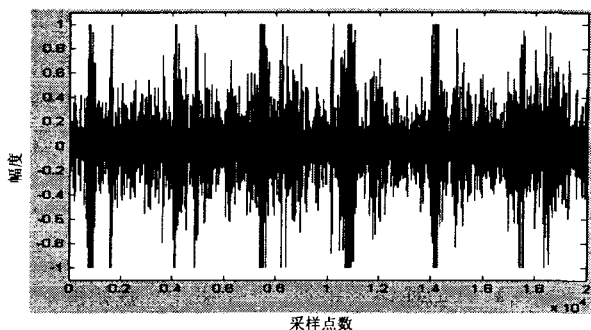


图6 胎儿心率信号

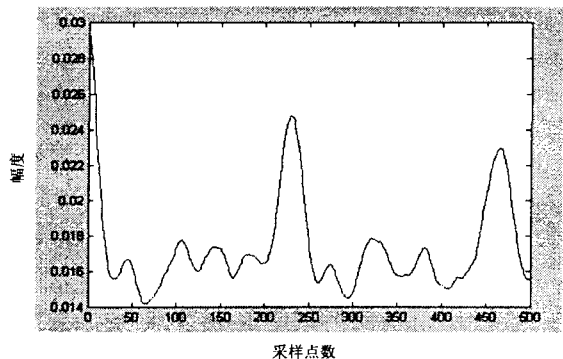


图7 通过Matlab计算的自相关函数

同样图6是采集自临床上的胎儿心音信号,通过本系统的计算心率是142,与Matlab计算出来的145基本接近。

作者简介 杨晓峰 1979年出生,西安交通大学生命学院,硕士。主要研究方向医学信号处理。

卞正中 1938年出生,西安交通大学生命学院,教授,博士生导师。

(上接第52页)

方法二:使用命令文件,如下命令文件my.cmd:

```
/* my.cmd */
myfile.out          /* 输入的COFF文件名 */
-boot              /* 将所有已初始化的段转换到启动表里 */
-v5510:2
-memwidth 16      /* 系统存储器宽度 16 b */
-romwidth 8       /* 用户ROM存储器宽度 8 b */
-e_c_int00        /* 入口地址 */
-a                /* 输出ASCII格式文件 */
```

当在DOS下键入:

C:\>hex55 my.cmd

boot.bat 将调用hex55将输入的COFF文件转换为ASC II格式的输出文件。

同样可以得到两个ASCII格式的十六进制的文件。

作者简介 卓为女,1981年出生,在读硕士研究生。主要研究数字信号处理领域。

管庆男,1968年出生,副教授,从事教学和科研工作。主要研究方向为数字信号处理(DSP),VXI总线,数据采集技术。先后参加“风洞试验动态参数实时处理系统”、国家九五预研“高速数据采集系统”等多个大型科研课题,担任主研、项目负责人。

经以上对比实验,验证了本系统的准确性。

6 结语

本文利用了TMS320VC5402实现了超声多普勒胎儿心率监护仪,在实验中,能稳定、可靠和快捷地计算并显示心率,同时也可以通过串口发送到远程医疗主控系统,实现远程监护。

参考文献

- [1] 熊庆,盖铭英.胎儿监护发展与现状[J].中华妇产科杂志,2000,35(1):7-10.
- [2] Drawes GS. Numerical Analysis of the Human Fetal Heart Rate; the Quality of Ultrasound Records [J]. Amer J Obstet Gynec, 1981, 141(1): 43.
- [3] TMS320C54xx DSP Enhanced Peripherals Ref_Set, SPRU-103G. 1999, 5.
- [4] TMS320C54x Optimizing C/C++ Compiler User's Guide. SPRU302. 2002.
- [5] 万明习,卞正中.医用超声学——原理与技术[M].西安:西安交通大学出版社,1992.
- [6] 李利. DSP原理及应用[M].北京:中国水利水电出版社,2004.

使用通用编程器将这2个文件分别烧到2片EPROM中,上电或复位时可自行加载用户代码,即实现了BootLoader。

参考文献

- [1] TMS320VC55x Assembly Language Tools User's Guide, SPRU280F. Texas Instruments, 2002.
- [2] Using the TMS320VC5501/C5502 BootLoader, SPRU911C Texas Instruments, 2004.
- [3] TMS320VC5501/5502 DSP External Memory Interface (EMIF) Reference Guide, SPRU621D. Texas Instruments, 2004.