

介绍 **dsp** 知识, 为大家提供最新的 **dsp** 资讯, 更多内容可以去南京研旭电气科技有限公司的官网 www.njyxdq.com www.f28335.com 或者官方论坛, 嵌嵌 **dsp** 论坛 www.armdsp.net 进行交流学习

欢迎大家收听嵌嵌 **dsp** 论坛的官方微博

<http://t.qq.com/qianqiandsp>

DSP 笔记

第一章 绪论

1、哈佛结构:

采用双存储空间, 程序存储器和数据存储器分开, 有各自独立的程序总线 and 数据总线, 可独立编址和独立访问, 可对程序进行独立传输, 使取指令操作、指令执行操作、数据吞吐并行完成, 大大地提高了数据处理能力和指令的执行速度, 非常适合于实时的数字信号处理。

2、冯。诺依曼结构:

采用单存储空间, 即程序和数据公用一个存储空间, 使用单一的地址和数据总线, 取指令和取操作数都是通过一条总线分时完成的。

当进行高速运算时, 不但不能同时进行取指令和取操作数, 而且还会造成数据传输通道的瓶颈现象。

3、流水线技术

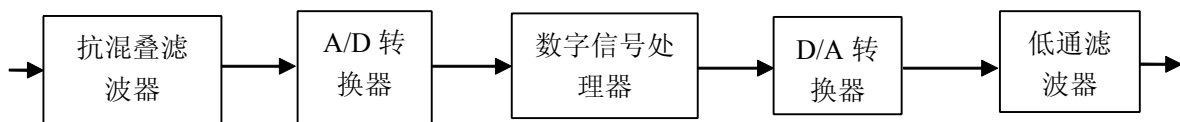
每条指令可通过片内多功能单元完成取址、译码、取操作数和执行等多个步骤, 实现多条指令的并行执行, 从而在不提高系统时钟频率的条件下减少每条指令的执行时间。

4、DSP 的分类

- 1)、按基础特性分类: 静态 DSP 芯片和一致性 DSP 芯片
- 2)、按用途分: 通用型芯片和专用型芯片
- 3)、按数据格式分 (即按精度或动态范围): 定点 DSP 和浮点 DSP (大多数定点 DSP 芯片都采用 16bit 定点运算, 只有少数 DSP 芯片为 24bit 定点运算)

5、DSP 系统的构成

抗混叠滤波器、数据采集 A/D 转换器、数字信号处理器 DSP、D/A 转换器和低通滤波器等



6、DSP 系统的处理过程

- 1) 将输入信号 $x(t)$ 进行抗混叠滤波, 滤掉高于折叠频率的分量, 以防止信号频谱的混叠。
- 2) 经采样和 A/D 转换器, 将滤波后的信号转换为数字信号 $x(n)$
- 3) 数字信号处理器对 $x(n)$ 进行处理, 得到数字信号 $y(n)$
- 4) 经 D/A 转换器, 将 $y(n)$ 转换成模拟信号
- 5) 经低通滤波器, 滤除高频分量, 得到平滑的模拟信号 $y(t)$

第二章 TMS320C54x 的硬件结构

1、TMS320C54x 的结构是以 8 组 16 位总线为核心, 8 组总线分为 1 组程序总线、3 组数据

总线 和 4 组地址总线

- 2、程序总线 PB：主要用来传送取自程序存储器的指令代码和立即操作数。
- 3、数据总线：3 条数据总线（CB、DB、EB）CB、DB 总线用来传送从数据存储器读出的数据，而 EB 用来传送写入存储器的数据。
- 4、地址总线：4 组。分别为 PAB、CAB、DAB 和 EAB
- 5、CPU 的基本组成：
 - 1) 40 位算术逻辑运算单元（ALU）
 - 2) 2 个 40 位累加器（ACCA、ACCB）
 - 3) 1 个支持—16~31 位移位的桶形移位寄存器
 - 4) 乘法器—加法器单元（MAC）
 - 5) 比较、选择和存储单元（CSSU）
 - 6) 指数编码器
 - 7) CPU 状态和控制寄存器

5、当 16 位数据存储器操作数通过数据总线 DB 或 CB 输入时，ALU 将采用以下两种方式对操作数进行**预处理**

- 1) 若数据存储器操作数在低 16 位时，则
 - 当 SXM=0 时，高 24 位（39~16 位）用 0 填充
 - 当 SXM=1 时，高 24 位（39~16 位）扩展为符号
 - 2) 若数据存储器操作数在高 16 位时。则
 - 当 SXM=0 时，39~32 位和 15~0 位用 0 填充
 - 当 SXM=1 时，39~32 位扩展为符号，15~0 位用 0 填充
- SXM 位符号位扩展方式控制位，位于状态寄存器 ST1 的 8 位

6、溢出处理

当运算结果发生溢出时

- 1) 若 OVM=0，则对 ALU 的运算结果不进行任何调整，直接送入累加器
- 2) 若 OVM=1，则需对 ALU 的运行结果进行调整
 - 当正向溢出时，将 32 位最大正数 007FFFFFFFH 装入累加器
 - 当负向溢出时，将 32 位最小负数 FF80000000H 装入累加器
- 3) 状态寄存器 ST0 中与目标累加器相关的溢出标志 OVA/OVB 被置 1

7、累加器 A 和 B

C54x 芯片有两个独立的 40 位累加器 A 和 B

累加器 A 和 B 之间的唯一区别是累加器 A 的 16~31 位能被用做乘法—累加单元中的乘法器输入，而累加器 B 则不能

	39~32	31~16	15~0
累加器 A	AG	AH	AL

	保护位 39~32	高阶位 31~16	低阶位 15~0
累加器 B	AG	AH	AL

保护位 高阶位 低阶位

例题：假设累加器 A=FF 0123 4567H，分别执行带移位的 STH 和 STL 指令后，数据存储单元 T 中的结果如下：P31

STH A, 8, T ; A 内容左移 8 位后, AH 存入 T 中, T=2345, A 的内容不变
STH A, -8, T ; A 内容右移 8 位后, AH 存入 T 中, T=FF01, A 的内容不变
STL A, 8, T ; A 内容左移 8 位后, AL 存入 T 中, T=6700, A 的内容不变
STL A, -8, T ; A 内容右移 8 位后, AL 存入 T 中, T=2345, A 的内容不变

8、桶形移位寄存器

40 位桶形移位寄存器主要用于累加器或数据区操作数的定标。它能将输入数据进行 0~31 位的左移和 0~16 位的右移。

9、乘法—累加单元 (MAC)

乘法—累加单元 MAC 是由乘法器、加法器、符号控制、小数控制、零检测器、舍入器、饱和和逻辑和暂存器几部分组成。见 P33 图

MAC 单元具有强大的乘法累加运算功能,可在一个流水线周期内完成一次乘法运算和一次加法运算。

MAC 单元包含一个 17x17 位硬件乘法器,可完成有符号和无符号的乘法运算。

乘法器的 XM 输入数据来自 T 寄存器、累加器 A 的 32~16 位以及由 DB 总线提供的数据存储器操作数;而 YM 输入的数据可以取自累加器 A 的 32~16 位、由 DB 总线和 CB 总线提供的数据存储器操作数以及由 PB 总线传送过来的程序存储器操作数。

10、MAC 单元的乘法器能进行有符号数、无符号数以及有符号数与无符号数的乘法运算,根据操作数的不同情况需进行一下处理

1) 若是两个有符号数相乘,则在乘法运算之前,先对两个 16 位乘数进行符号位扩展,形成 17 位有符号数后再进行相乘。扩展方法是:在每个乘数的最高位前增加一个符号位,其值由乘数的最高位决定。

2) 若是两个无符号数相乘,则在两个 16 位乘数的最高位前添加“0”,扩展为 17 位乘数后再进行乘运算。

3) 若是有符号数与无符号数相乘,则有符号数在最高位前添加 1 个符号位,其值由最高位决定,而无符号数在最高位前添加“0”,然后两个操作数相乘。

乘累加? P33

11、C54x 提供三个 16 位寄存器来作为 CPU 状态和控制器,它们分别为:

状态寄存器 0 (ST0)、状态寄存器 1 (ST1) 和处理器工作方式状态寄存器 (PMST)

12、TMS320C54x 共有 192 千字的寻址存储空间。这 192 千字的存数空间分成 3 个独立的可选择空间:

64 千字的程序存储空间;

64 千字的数据存储空间;

64 千字的 I/O 空间。

13、所有的 C54x 芯片都包括内部随机存储器 (RAM) 和只读存储器 (ROM)。内部 RAM 又分为单寻址 RAM (SARAM) 和双寻址 RAM (DARAM) 两种类型。

14、与外部存储器相比内部存储器具有以下几个优点:

不需要插入等待状态;

与外部存储器相比,成本低;

比外部存储器功耗小

15、C54x 的存储结构与处理器工作方式状态寄存器 (PMST) 的设置有关,用户可以通过 PMST 中的 3 个控制位 (MP/MC、OVLY 和 DROM) 来配置存储器空间。

16、程序存储空间的配置

程序存储空间可通过 PMST 寄存器的 MP/\overline{MC} 和 OVLY 控制位来设置内部只读存储器的映

射地址。

1) $\overline{MP}/\overline{MC}$ 控制位用来决定程序存储空间是否使用内部存储器

当 $\overline{MP}/\overline{MC}=0$ 时, 4000H~EFFFH 程序存储空间定义为外部存储器, 而 F000H~FEFFH 程序存储空间定义内部 ROM。FF00H~FFFFH 程序存储空间定义为内部存储器。其工作方式微型计算机模式。

2) OVLY 控制位用来决定程序存储空间是否使用 RAM。

当 OVLY=0 时, 000H~3FFFH 全部定义为外部程序存储空间, 程序存储空间不使用 RAM, 此时内部 RAM 只作为数据存储器使用;

当 OVLY=1 时, 0000H~007FH 保留, 程序无法占用。0080H~3FFFH 定义为内部 DARAM。即内部 RAM 同时被映射到程序存储空间和数据空间。

17、数据存储空间的配置

C54x 的数据存储空间共有 64 千字, 采用片内和片外存储器统一编址。除了含有 SARAM 和 DARAM 外, 还可以通过软件将内部 ROM 映射到数据存储空间。用户可以通过设置 PMST 中的 DROM 位, 将部分内部 ROM 映射到存储空间。

若 DROM=0, 内部 ROM 不映射到数据空间;

若 DROM=1, 内部 ROM 映射到数据存储空间, 并且当 $\overline{MP}/\overline{MC}=0$ 时, 内部 ROM 同时映射到数据空间和程序空间。每次复位时, 处理器将对 DROM 位清零。

18、C54x 的片内外设主要包括:

通用 I/O 引脚;

定时器;

时钟发生器

主机接口 HPI;

串行通信接口;

软件可编程等待状态发生器;

可编程分区转换逻辑。

19、每一种 C54x 芯片都为用户提供了两个通用的 I/O 引脚: \overline{BIO} 和 XF

\overline{BIO} 为分支转移控制输入引脚, 用来监控外部设备的运行状态。

XF 位外部标志输出引脚, 主要用于程序向外设传输标识信息。

20、串行通信接口

同步串行口 SP

缓冲同步串行口 BSP

时分多路同步串行口 TDM

多通道缓冲串行口 McBSP

21、软件可编程等待状态发生器

软件可编程等待状态发生器能把外部总线周期扩展到最多 14 个机器周期, 这样可以方便地与慢速的外部存储器和 I/O 设备接口。

第三章 TMS320C54x 的指令系统 (重点)

1、TMS320C54x 有 7 种基本的寻址方式: 立即寻址、绝对寻址、累加器寻址、直接寻址、

间接寻址、存储器映像寻址和堆栈寻址。

2、直接寻址时利用数据指针和堆栈指针寻址，其特点是数据存储器地址由基地址（数据页指针 DP 或堆栈指针 SP）和偏移地址共同构成，共 16 位。

1) 当 CPL=0，以数据页指针寄存器 DP 中的 9 位为高位，以指令中的 7 位为低位，构成 16 数据存储单元的地址。

2) 当 CPL=1 时，将堆栈指针 SP=的 16 位与指令中的 7 位地址相加，形成 16 位的数据存储器地址。

3、两种特殊的间接寻址方式：循环寻址方式和位倒序寻址方式

4、循环寻址？

5、位倒序可以提高 FFT 等算法的效率。例如 16 点 FFT 的位倒序寻址如表 P79

？

6、C54x 的堆栈是从高地址向低地址方向填入的。处理器使用一个 16 位堆栈指针（sp）来对堆栈进行管理，SP 始终向存放在堆栈中的最后一个元素。

执行压入堆栈操作时，时先减小 SP 后再将数据压入堆栈的；而执行堆栈弹出操作时，则是先从堆栈弹出数据，然后再增加 SP 的值。

7、TMS320C54x 的指令系统共有 129 条基本指令，由于操作数的寻址方式不同，由他们可以派生 205 条指令。按指令的功能分类，可以分成数据传送指令、算术运算指令、逻辑运算指令、程序控制指令、并行操作指令和重复操作指令。

第四章 汇编语言程序的开发工具

1、汇编器和链接器生成的目标文件，是一个可以由 C54x 器件执行的文件。这些目标文件的格式称为公共目标文件格式（COFF）

2、段是 COFF 文件中最重要的概念。所有的 COFF 目标文件都包含以下三种形式的段：

- 。Text 段（文本段），通常包含可执行代码
- 。Data 段（数据段），通常包含初始化数据
- 。Bss 段（保留空间段），通常为未初始化变量保留存储空间

3、COFF 目标文件有以下两种基本类型的段

1)、初始化段：包含有数据或程序代码

- 。Text 段是已初始化段
- 。Data 段是已初始化段
- 。Sect 汇编器伪指令建立的自定义段也是已初始化段

2)、未初始化段：在存储空间中，它为未初始化数据保留存储空间

- 。Bss 段是未初始化段
- 。Usect 汇编命令建立的自定义段也是未初始化段

4、汇编器对段的处理是通过段的伪指令来区别各个段，并将段名相同的语句汇编在一起。

汇编器有 5 条命令可识别汇编语言的程序各部分。

- 。Bss（未初始化段）
- 。Usect（未初始化段）
- 。Text（已初始化段）
- 。Data（已初始化段）
- 。Sect（已初始化段）

5、段命令举例 P105

6、Memory 伪指令。用来定义目标系统的存储器配置空间，包括对存储器各部分命名，以及规定它们的起始地址和长度。

Sections 伪指令。此命令告诉链接器如何将输入段组合成输出段，以及将输出段放在存储器中什么位置。

7、链接器命令文件举例 P127

8、使用 MEMORY 伪指令的链接器命令文件的例子 P128

9、多个文件的连接实例 P132

第五章 TMS320C54x 的汇编语言程序设计

1、如果程序要使用堆栈，必须先进行设置，方法如下： p148

```
size .set      120
stack .usect   "STACK",size
        STM    #stack+size,SP
```

2、程序控制中的分支操作包括：分支转移、子程序调用、子程序返回和条件操作。（能够判断即可）

3、P152---P161 例题

第七章 TMS320C54x 片内外设、接口及应用

1、定时器结构

图 p230

定时器主要由定时寄存器 TIM、定时周期寄存器 PRD、定时控制寄存器 TCR（包括预标定分频系数 TDDR、预标定计数器 PSC、控制位 TRB 和 TSS 等）及相应的逻辑控制电路组成。其中寄存器 TIM、PRD 和 TCR 都是存储映像寄存器，它们在数据寄存器中的地址分别为 0024H、0025H 和 0026H。TIM 是一个减一计数器。PRD 用来存放定时时间常数。TCR 中包含定时器的控制位和状态位。

2、定时周期=CLKOUT×(TDDR+1)×(PRD+1)

3、中断寄存器的作用

TMS320C54x 中断系统设置两个中断寄存器，分别为中断标志寄存器 IFR 和中断屏蔽寄存器 IMR。

中断标志寄存器：当一个中断出现时，IFR 中的相应的中断标志位置 1，知道 CPU 识别该中断为止。

中断屏蔽寄存器：如果状态寄存器 ST1 中的 INTM=0，IMR 寄存器中的某位置 1，就能开放相应的中断。

4、对于硬件可屏蔽中断，只有满足以下三个条件 CPU 才能响应中断：

- 1) 当前中断优先级最高
- 2) INTM 位清零
- 3) IMR 屏蔽位为 1

第八章 TMS320C54x 的硬件设计

1、介绍 DSP 硬件系统的基本设计，包括电源电路、复位电路、时钟电路等。

2、TMS320VC5402 芯片采用双电源供电方式，以获得

更好的电源性能，其工作电压分别为 3.3V 和 1.8V，其中 3.3V 位 I/O 电源 DV_{DD} ，主要供 I/O 接口使用，通常情况下可直接与外部低压器件记性接口，而不需要额外的电平变换电路。

1.8V 位内核电源 CV_{DD} ，主要为芯片的内部逻辑提供电压，包括 CPU、时钟电路和所有的外设逻辑。与 3.3V 相比，1.8V 可以大大降低芯片功耗。

3、TMS320VC5402 芯片加电次序

理想情况下，DSP 芯片上的两个电源同时加电，但在有些场合很那做到。若不能做到同时加电，应先对 DV_{DD} 加电，然后再对 CV_{DD} 加电，同时要求 DV_{DD} 电压不超过 CV_{DD} 电压 2V。

4、复位电路的设计

- 1) 上电复位电路
- 2) 手动复位电路
- 3) 自动复位电路

5、自动复位电路工作原理：

通过电路提供的的监视线来监视系统运行，当系统正常运行时，在规定的时间内给监视线提供一个变化的高低电平信号，若在规定的时间内这个信号不发生变化，自动复位电路就认为系统运行不正常，并对系统进行复位。

6、自动复位电路工作过程图 p285

当 DSP 处于不正常工作时，由程序所产生的周期脉冲 CLK 将会消失，自动复位电路将无法接收到监视信号，MAX706R 芯片将通过引脚 7 产生复位信号，使系统复位，程序重新开始运行，强迫系统恢复正常工作。

7、3.3V 与 5.5V 电平转换的形式

在一个系统同时存在 3.3V 和 5V 系列芯片时，必须考虑它们之间的电压兼容性问题。需注意：

- 1)、3.3V 的芯片是否能承受 5V 电压；
- 2)、驱动器件的输出逻辑电平与负载器件要求的输入逻辑电平是否匹配
- 3)、驱动电路允许的最大电流是否大于负载器件所要求的输入电流。

P290 图

8、根据不同的应用场合，3.3V 与 5.5V 电平转换有四种形式：

- 1)、5V TTL 器件驱动 3.3V TTL 器件 (LVC)
- 2)、3.3V TTL 器件 (LVC) 驱动 5V TTL 器件
- 3)、5V CMOS 器件驱动 3.3V TTL 器件 (LVC)
- 4)、3.3V TTL 器件 (LVC) 驱动 5V CMOS 器件

9、P291 图 DSP 与 5V EPROM 的接口

10、DSP 系统的组成 图 P317

- 1)、TMS320VC5402 芯片：作为整个系统的核心，主要用来完成语音压缩和解压缩在内的所有软件功能。
- 2)、模数转换电路：A/D 和 D/A 转换器完成语音信号的模数和数模转换。
- 3)、FLASH 存储器：用于存放系统程序和已初始化的数据。
- 4)、双口 RAM：用来与外部交换语音压缩数据。

11、DSP 系统的工作过程：

- 1)、系统加电 DSP 芯片后，由其内部存储器固化的自引导程序 (Boot) 将存于 FLASH 中的程序且数据移入内部 RAM。
- 2)、程序和数据移至内部 RAM 后，DSP 芯片开始运行程序，执行语音编码算法。每隔 10ms 运行一次编码算法，并与双口 RAM 交换一次数据；
- 3)、DSP 芯片将语音压缩后得到的数据写入双口 RAM，由外部系统读出送至信道；
- 4)、外部系统将对方的编码数据送至双口 RAM，由 DSP 芯片从双口 RAM 中读出，进行数据处理，还原为合成语音。

由于 TMS320VC5402 芯片的运算速度在 10ms 内足以完成语音的压缩和解压缩算法，

因此，系统可以采用全双工方式工作。

相信对你有帮助的：

[最实惠的 f28335 系列开发板](#)

[DSP 错误总结！](#)

[DSP 新手上路注意事项](#)

[DSP 汇编指令学习笔记](#)

[DSP 程序编写和调试的常见问题](#)

介绍 **dsp** 知识，为大家提供最新的 **dsp** 资讯，更多内容可以去南京研旭电气科技有限公司的官网 www.njyxdq.com www.f28335.com 或者官方论坛，嵌嵌 **dsp** 论坛 www.armdsp.net 进行交流学习

欢迎大家收听嵌嵌 **dsp** 论坛的官方微博

<http://t.qq.com/qianqiandsp>