# 基于 DM642 的 DSP 中的数据拷贝

袁 辉，达 力，周剑杨
（厦门大学信息科学与技术学院 福建 厦门 361005）

【摘 要】：本文就基于 DM642 的 DSP 中数据拷贝进行分析，首先从理论上论述了在 DSP 上如何进行数据拷贝，然后通过实验，对各个 DSP 中数据拷贝方式进行比较。从而可以使得我们在开发 DSP 过程中，选择合适的数据拷贝方式。

【关键词】：拷贝，DSP。

0、绪论：

DSP 即数字信号处理器(Digital Signal Processing，简称 DSP)是一门涉及许多学科而又广泛应用于许多领域的学科。数字信号处理是一种通过使用数学技巧执行转换或提取信息，来处理现实信号的方法，这些信号由数字序列表示。在过去的二十多年时间里，数字信号处理已经在通信，图像处理等领域得到极为广泛的应用。

DSP 是一种独特的微处理器，是以数字信号来处理大量信息的器件。其工作原理是接收模拟信号，转换为 0 或 1 的数字信号，再对数字信号进行修改、删除、强化，并在其他系统芯片中把数字数据解译回模拟数据或实际环境格式。它不仅具有可编程性，而且其实时运行速度可达每秒数以千万条复杂指令程序，远远超过通用微处理器，是数字化电子世界中日益重要的电脑芯片。它的强大数据处理能力和高运行速度，是最值得称道的两大特色。

1、DSP 中的数据拷贝方式：

DSP 中需要处理大量的数据，数据的搬运，移动，拷贝需要占用大量的系统资源。如何正确的使用 DSP 系统中提供的拷贝方式，将会对 DSP 的运行提供很大的便利。在 DSP 中，提供的数据拷贝方式主要有 DSP 提供的数据拷贝函数，直接存储器访问(Direct Memory Access 简称 DMA)方式，加强型直接存储器访问(Enhanced Direct Memory Access 简称 EDMA)方式，快速直接存储器访问(Quick Direct Memory Access 简称 QDMA)方式。本文就基于 TI 公司提供的 DM642 中的数据拷贝方式进行研究，而在 DM642 中 EDMA 方式是 DMA 的加强版，所以只是介绍 EDMA 方式和 QDMA 方式。

1.1 DSP 提供的数据拷贝函数：

在 DSP 函数库中，提供的数据拷贝函数主要有两种，分别是字符串的拷贝和内存的拷贝。其中，字符串的拷贝函数主要有：strcpy、strncpy。他们的定义分别如下：

原型声明：extern char *strcpy(char *dest,char *src);

功能：把 src 所指由"\0"结束的字符串拷贝到 dest 所指的数组中，返回指向 dest 的指针。

说明：src 和 dest 所指内存区域不可以重叠且 dest 必须有足够的空间来容纳 src 的字符串。

原型声明：extern char * strncpy(char *s1,char *s2,size_t n);

功能：将字符串 s2 中最多 n 个字符拷贝到字符数组 s1 中，返回指向 s1 的指针。

说明：如果源串长度大于 n，则 strncpy 不拷贝最后的"\0"结束符，所以是不安全的，拷贝完后需要手动添加字符串的结束符才行。如果拷贝到"\0"就结束拷贝。

内存的拷贝函数主要有：memcpy，memmove。他们的定义如下：

原型声明：extern void *memcpy (void *dest, void *src, unsigned int count);

功能：由 src 所指内存区域拷贝 count 个字节到 dest 所指内存区域。

说明：src 和 dest 所指内存区域不能重叠，函数返回指向 dest 的指针。

原型声明：extern void *memmove(void *dest, const void *src, unsigned int count);

功能：由 src 所指内存区域拷贝 count 个字节到 dest 所指内存区域。

说明：src 和 dest 所指内存区域可以重叠，但拷贝后 src 内容会被更改。函数返回指向 dest 的指针。

其中，strcpy 和 strncpy 的区别很明显，一个是将一个有以 NULL 为结尾的字符串传递给另外一个，一个是将最多 n 个字符拷贝到另外一个字符串。而 memcpy 和 memmove 之间的区别在于一个指定 src 和 dest 所指的内存区域不能重叠，一个可以重叠。这两种函数都可以用于 dsp 中的数据移动。但是对于字符串的拷贝函数来说，只能作用于字符串。如果要拷贝的数据中存在"\0"并且不是结尾，则会出现错误。而内存的拷贝是将制定的内存内容拷贝到另外的内存区域中，不论数据中存在不存在"\0"。当然，在拷贝字符串的时候，如果参数 dest 所指的内存空间不够大，可能会造成缓冲溢出(buffer Overflow)的错误情况，如果使用 strcpy 效果会好一点。在速度上，memcpy 快于 strcpy。memcpy 被编译优化为 MOVSB，MOVSW 等的机器码内存块搬动。strcpy 不容易用 MOVSB。因为必须有一个求长度(块大小)的过程，即使用 MOVSB 速度也被这个过程搞慢了。但是，无论是字符串的拷贝函数还是内存的拷贝函数，在整个拷贝的过程中，都需要 CPU 的支持。如果 CPU 存在大量的空闲资源，使用这些函数对整体的速度影响不大。但如果将数据的搬运交给 DMA 控制器，则会节省大量的 CPU 时间。

1.2 EDMA 的数据拷贝方式

EDMA 是在 DMA 的基础上发展起来的一种传输方式，它拥有 DMA 传输的全部特性，除了具有 DMA 后台操作(即不占用 CPU 时钟)，高吞吐率等特点外，EDMA 比 DMA 通道的功能更加强大，而且控制器和结构都比 DMA 有了很大改进。期增强之处包括：

· 提供了 64 个通道；

所谓通道是可以配置的 EDMA 传输的方式，每个通道是和特定的系统事件绑定的。每一个通道对应一个 PaRAM 的配置，当启动一个通道的时候，数据的传输就按照 PaRAM 中的配置来传输。

· 通道间的优先级可以设置；

即当有多个通道传输同时被请求时，根据优先级的设定来判断先执行哪一个通道的传输。在 PRI 字段中配置。

· 支持不同结构数据传输的链接；

EDMA 有两种类型的数据传输：1D 和 2D 的(即有 4 种组合方式)，数据的维数表明了数据的组成方式。

· 支持对 8BIT、16BIT、32BIT 数据的存取；

源/目的地址是在元素大小的边界对齐的，因此要注意指向源/目的地址的指针的类型需要和 ESIZE 匹配

· 基于 RAM 的配置结构(PaRAM)。

Parameter RAM 位于 EDMA 控制器内部，只有设备总线可以对之进行访问。PRAM 表大小为 2KB，其组成为：

① 64 个 24 字节(即每项 6 个 32bits 的字 WORD)的表项，用于保存 64 个通道的参数，也可作为保存 Linking 时候需要进行重载的通道参数；

② 21 个 24 字节(即每项 6 个 32bits 的字 WORD)的表项，用于保存 Linking 时候需要进行重载的通道参数；

③ 剩余 8 个字节.

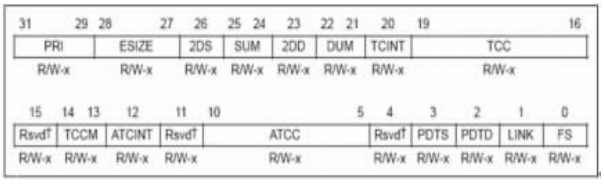
下图就是 EDMA 中每个通道对应的 PaRAM 结构示意图。

| EDMA Channel Options Parameter (OPT) | |
|---|---|
| EDMA Channel Source Address (SRC) | |
| Array/frame count (FRMCNT) | Element count (ELECNT) |
| EDMA Channel Destination Address (DST) | |
| Array/frame index (FRMIDX) | Element index (ELEIDX) |
| Element count reload (ELERLD) | Link address (LINK) |

PaRAM 示意图

其中，各个字段的含义如下：

1)、OPT：OPT 是 EDMA 中重要的一个参数。它决定着 EDMA 的优先级，单元数据大小等等。其结构图如下所示：




OPT 结构示意图

PRI：优先级设置，根据需要来具体设置，一般选择中优先级传输。

ESIZE：单元数据大小，ESIZE*ELECNT 是每一个帧的总的数据传输大小。

2DS：即一维或者二维的传输选择，如果是一维传输，则选择 0。

SUM：即源地址的更新模式，如果只是将固定源地址的数据传输，则 2DS=0;假如是从存储器传输数据，一般是采用源地址递增的模式。

DUM：即目的地址的更新模式，一般采用的是目的地址递增或者递减的数据传输，如是固定的地址的话，每一个传输的数据将覆盖上一次传输的数据。

TINCT：设置传输完成中断，即传输完成之后产生中断信号，执行相应的中断程序。TCC 和 TCCM：联合形成了传输完成代码，传输完成代码指定了相应的中断通道，即执行响应的中断程序。

LINK：即传输完成之后是否采用连锁传输，即加载指定通道的 PaRAM。

FS：帧同步，FS=0 时单元或者阵列同步，即每一次触发通道传输时传输一个单元（ESIZE 指定了大小）的数据，FS=1 帧同步，即每一次触发通道传输时传输一帧的数据，每一个帧的数据的大小由 ELECNT 指定；

2)、SRC：源地址，数据的输入地址。

3)、FRMCNT：总的帧计数，通常为 0；当只传输 1 帧时，也为 0。

4)、ELECNT：每一帧包含的单元数据的个数。

5)、DST：目的地址，即数据的输出地址。

6)、FRMIDX：每一个帧与下一个帧的间隔，如果是连续地址，第一个帧的大小是 32bit，第一个帧紧跟在第一个帧地址后面，则 FRMIDX 为 4；补充：帧索引+帧中最后的单元地址上=下一帧的起始地址。

7)、ELEIDX：每一个单元与下一个单元的间隔，如果是连续的单元存储，一个单元为 32bit，第二个单元紧跟在第一个单元后面，则 ELEIDX 为 4；单元索引+最后的单元地址=下一帧的起始地址；

8)、ELERLD：用于 1-D 传输中每帧最后一个数据单元传输结束后，重新加载传输计数值，一般设置为 0；

EDMA 数据传输有两种发起方式：

CPU 发起的 EMDA 数据传输(非同步方式)：需要传输时，CPU 设置 ESR 寄存器的相应位为 1，从而触发一个 EDMA 事件的产生，事件对应的通道参数被送往地址硬件并且完成相应的处理，这种非同步方式的实时数据传输无需设定 EER 寄存器；

事件触发方式 EDMA 数据传输（同步方式）：ER 寄存器保存外设发送过来的事件，一旦 CPU 设置 EER 寄存器的相应位为 1 后，ER 中的事件才会提交给事件编码器（Event Encoder），并且进一步引起相关的传输参数的发送给地址产生硬件；如果 EER 中对应于某事件的位没有置 1，则 ER 寄存器中的事件将保留，一旦置 1 则触发 EDMA 的传输，这种特性可以应用到 EDMA Chain 传输，需要 EER 和 CCER 结合使用；

1.3 QDMA 的数据拷贝方式

QDMA 与 EDMA 一样，能够支持几乎所有的 EDMA 传输模式，但是，QDMA 提交传输申请的速度要快很多，并可在 CPU 干预下控制搬移数据，因而可以大大增加系统的灵活性。QDMA 的操作由两组寄存器进行控制，其中第一组的五个寄存器寄存了 QDMA 传输所需的参数，而第二组的五个寄存器则是第一组寄存器的"伪映射"。QDMA 数据传输总是帧同步的，即对于 1D 数据传输而言每次同步事件传输一帧数据，对于 2D 数据传输而言每次同步事件传输一块数据。因此，QOPT.FS 对于 QDMA 是无意义的。另外，QDMA 是一次性快速传输的，因此也没有中间传输过程这个概念，即没有交替性传输完成中断。

QDMA 没有 Linking 方式的传输，但是有 Chaining 方式的传输。QDMA 有两组内存映射寄存器用于设定通道参数，如下图所示：

| QDMA Channe Options Parameter(OPT) | |
|---|---|
| QDMA Channel Source Address(SRC) | |
| Array/frame count(FRMCNT) | Element count(ELECNT) |
| QDMA Channel Destination Address(DST) | |
| Array/frame index(FRMIDX) | Element index(ELEIDX) |

QDMA registers

| QDMA Channe Options Parameter(OPT) | |
|---|---|
| QDMA Channel Source Address(SRC) | |
| Array/frame count(FRMCNT) | Element count(ELECNT) |
| QDMA Channel Destination Address(DST) | |
| Array/frame index(FRMIDX) | Element index(ELEIDX) |

*QDMA pseudo-registers*

因为这些参数和 EDMA 的参数功能类似，这里就不必详述。其中，QDMA 寄存器集只用于配置，QDMA psedudo 寄存器集可以用于提交 QDMA 请求。

2、DSP 中数据移动的实现：

为了保证数据的一致性和可比较性，我们主要是将 100 个每个含有 16 个字符的字符串拷贝到 DSP 的内存中，并将其打印出来。

2.1 使用字符串拷贝或者内存拷贝

字符串拷贝和内存拷贝是我们 C 语言中常见的数据拷贝方式，也是在 DSP 的应用中常用的数据拷贝方式之一。主要是应用简单，方便使用。

2.1.1 使用字符串的拷贝

字符串的拷贝应用的是 strcpy 和 strncpy 两个函数，两个函数的用途明显不同。一个用于拷贝一个字符串到另外一个，一个拷贝固定字数的字符串到另外一个。如果两个同时拷贝相同字数的字符串的话。则是 strcpy 占优势。比如在这次实验中，分别拷贝 100 个字符串占用的 CPU 时钟周期数是：strcpy 占用了 9602，而 strncpy 占用了 11200 个。总的程序运行(下转第 71 页)

模型库系统是森林经营决策支持系统的核心部分，其设计好坏直接影响到决策的结果，同时也影响到系统的质量和水平。事实上，一种数学模型在计算机中是以某种数据结构的形式进行存储，其建造过程是：首先从数据库中自动提取相关数据，同时通过屏幕向用户询问相关可控参数，再将这两方面的数据组成模型所要求的数据结构，同时将该数据结构存人模型库中，然后从方法库中选取适当的方法对该模型进行求解，求解结果存人数据库，可以将其显示到屏幕或打印输出，也可以做为为其它模型所调用；系统还应该设有良好的反馈功能，决策者能根据输出结果满意程序多次输入不同的可控参数，运行出多种方案供选择。建模过程见图 1

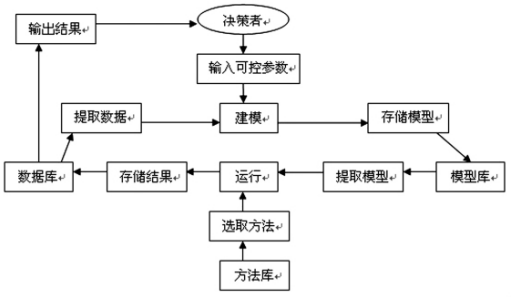



图 1

2.3 数据库系统

编制森林经营方案所涉及的数据是反映森林资源管理活动的一切数据，这些数据包括资源、经济、经营活动、人文等原始数据以及对这些数据的分析、预测和决策结果等数据。数据来源广，类型多，数量大。如何统一的组织管理这些数据，提高系统中数据流通效率，是建立数据库系统中最重要工作。

数据库通常被定义为计算机存储的数据集合，而数据库管理则是用于建立、维护、存储以及更新数据库的程序集合。建立数据库系统并对数据进行管理，其主要目的是实现数据的共享，保持数据的独立性和确保数据的安全。作为森林经营决策支持系统的数据库，很重要的一方面是支持对话和建模，将对话的参数和模型分离，使系统中参数的修改比较容易。由于参数可存入数据库，也为这些参数提供单独的子管理功能。数据库管理技术已经十分成熟，主要功能包括数据的管理、诊断、更新、统计以及数据析取。

1、数据管理包括输入、检索、增删、修改和维护等，这是其最主要的功能。

2、通过数据诊断实现自动查错功能，如通过逻辑检查，检查数据是否出错，以便及时修改。

3、数据更新指因林分的自然生长、更新造林和森林采伐等经营活动而引起资源数据的变化，需对数据进行更新，以保证决策信息的动态性。

4、数据统计是按照统计表格形式或图形形式，迅速、准确地对数据进行统计。

5、数据析取是指数据库与模型库之间的数据接口，提供模型所需数据，有时需要对数据进行预处理，才能为模型所利用。

3、结语

国内外决策支持系统的研究和应用多集中在商业和工业企业管理等领域，而在林业系统管理等领域的研制和开发应用比较少。但有理由相信，随着计算机技术、3S 技术、信息技术、林业科学技术的发展以及林业现代化管理水平的提高，森林经营决策支持系统的研究和应用将会不断得到发展并走向成熟。

参考文献：

[1] 陈文伟.决策支持系统及其开发[M].北京：清华大学出版社，2000

[2] 张怀清等.林业资源环境网络在线决策支持系统研究[J].林业科学研究，2002，15(6)：637~643

[3]W C Schou,B Richardson,M E Teske,H W Thistle.Spray Safe Manager 2-Integration of GIS With an Aerial Herbicide Application Decision Support System. 2001 ASAE Annual International Meeting, Sacramento, California, USA, July 30-August 1,2001

（上接第 91 页）

的 CPU 时钟周期数分别为：54809 和 55728

2.1.2 使用内存的拷贝

内存拷贝应用的是 memcpy 和 memmove 两个函数。这两个函数功能一样，只是对参数的要求不一样。不过这两个函数的性能上差距不大。对于此实验，memcpy 和 memmove 使用的 CPU 时钟周期分别为 399 和 396。其分别使用的 CPU 时钟周期数为：44774 和 44952。

2.2 使用 EDMA

EDMA 即加强型 DMA，其主要目的是使得在数据移动的时候，解放 CPU 的资源。使得系统可以在进行数据移动的同时，CPU 可以做其他的事情。本次实验采用的是传输 1D 的方式。采用 CPU 触发方式，方便管理与应用。实验中，EDMA 占用的 CPU 时钟周期是 45072

2.3 使用 QDMA

QDMA 称为快速 DMA。QDMA 与 EDMA 一样，能够支持几乎所有的 EDMA 传输模式。但是，QDMA 提交传输申请的速度要快很多，并可在 CPU 干预下控制搬移数据，因而可以大大增加系统的灵活性。全部程序占用的 CPU 时钟周期数为：44759。

3、实验结果分析：

通过实验，我们可以看到，在 DSP 中，效率最低的为字符串的拷贝。而且和内存拷贝的差距很大。由于存取的数据量想对于海量的数据来说，只是在 16*100 个字符的数据中，就比字符串的拷贝最少少占用了 10050 个时钟周期，相当于原来的 81.66%，可以节省大量的 CPU 时间。而 EDMA 从数据上看，总的时钟周期低于内存拷贝的时钟周期数。但是当数据为 32*100 个字符时，memcpy 占用的时钟周期数会超过 EDMA。而 QDMA 使用的时钟周期数已经少于内存拷贝。当然相对于内存的拷贝来说，改进的性能很小。但是，当数据量很大的时候，在此基础上一点微小的改进，意味着在海量基础上很大的改进。当然，对于少量的数据来说，使用内存的拷贝性能会优于使用 EDMA 和 QDMA。

所以总的来说，如果在对 DSP 系统资源要求不高的情况下，使用字符串的拷贝和内存拷贝基本上是没有问题。当然在此种情况下，如果是针对的字符串类型数据进行处理的话，使用字符串的复制，更能保证数据的完整性。在 DSP 系统资源比较紧张的情况下，如果说数据的拷贝量比较小的情况下，使用内存的拷贝性能也可以满足要求。如果需要大量的拷贝的情况下，我们就需要使用 EDMA 以及 QDMA 来进行数据的拷贝。

4、实验结论：

本文就 DSP 中的数据拷贝进行了分析，其中首先对 DSP 上的数据拷贝方式做了分析，其中重点说明了 EDMA 方式。然后针对这些分析，分别通过实验来观察它们的性能。并最后对每种拷贝分别的应用做了大致的总结。

参考文献：

[1]: 薛炜，李广军，郭志勇. 增强型 EDMA 的结构与典型应用[J]，单片机与嵌入式系统应用 2009 第一期：12~14。

[2]: 李俊，张朱明，李兵兵. 一种基于 QDMA 的 TMS320C6711D 与 FPGA 的接口设计[J]，电子元器件应用 2007.9 第九卷第九期。