ISSN1672-4305
CN12-1352/N
实 验 室 科 学
LABORATORY SCIENCE
第13卷 第1期 2010年2月
Vol. 13 No. 1 Feb. 2010

EDMA3 在视频采集和显示系统中的运用

李 静，安 康
（杭州师范大学 钱江学院，浙江 杭州 310012）

摘 要：TMS320DM6446（简称 DM6446）是基于达芬奇技术的双核、高性能数字视频信号处理器。该处理器设计的改进型 EDMA3 极大地提升了处理器内部以及和外设之间的高速数据传输能力。该文详细讨论了 EDMA3 特点、性能和使用方法，并给出了其在视频采集和显示系统中的实际应用方法和技巧。

关键词：DM6446；EDMA3；达芬奇
中图分类号：TP368.1 **文献标识码**：A doi:10.3969/j.issn.1672-4305.2010.01.030

Application of EDMA3 in video capture and display system

LI Jing, AN Kang
(Qianjiang College, Hangzhou Normal University, Hangzhou 310012, China)

Abstract: TMS320DM6446 (DM6446 for short) is a high performance dual-core digital video processor based on Davinci technology. This processor' s 3rd generation of Enhanced Direct Memory Access (EDMA3) controller enhances its high speed data transfer capability both within processor and between peripherals. This paper discusses the feature and the performance of EDMA3 in detail and also presents its application skill in video capture and display system.

Key words: DM6446; EDMA3; Davinci

DM6446 是具有高集成度的视频芯片。该芯片包括一个 ARM 子系统、一个 DSP 子系统和一个视频处理子系统（VPSS），其中 VPSS 包括一个视频前端（VPFE）和一个视频后端（VPBE）[1]。ARM 采用 ARM926EJ-S™ 内核，主频达 297MHz；DSP 采用 C64x+™ 内核，主频达 594MHz，指令数可达 4752MIPS；视频处理子系统支持图像缩放、自动聚焦/曝光/白平衡、OSD 等一系列图像处理功能。

DMA 传输是指在没有 CPU 参与下自动完成不同存储器空间之间的数据搬移，主要应用于需要大批量数据传输的场合[2]。传统的 DMA 传输功能比较简单，只能完成两块连续存储空间之间的批量数据传输。在视频采集和显示系统中，经常会遇到一些更加复杂的批量数据传输，DM6446 的 EDMA3 控制器正好可以解决这些较复杂的数据传输，从而更好地减轻 CPU 负担，增强系统性能。

1 EDMA3 控制器

EDMA3 属第三代增强型 DMA，它基于 RAM 结构。DM6446 共提供 128 个 DMA 信道，其中 64 个对应 DMA 信道，8 个对应 QDMA（快速 DMA）信道，其余作为信道链接（Link）时的参数 RAM。每个信道对应一个参数 RAM，总共 128 个参数 RAM 组成一个参数 RAM 集。如图 1 所示，每个参数 RAM 由 32 字节控制参数组成。参数 RAM 中的信道选项参数（OPT）共四个字节，包含传输优先级、FIFO 深度、数据传输方式等诸多控制选项。传输源地址和目的地址指出了数据搬移中源和目的数据块的起始地址。传输阵列计数、帧计数和块计数用于设定本次传输的数据大小和组织格式。帧索引和块索引分别给出了信道相邻阵列间距和相邻帧间距。链接地址保存了链接参数存放地址，而重载选项用于更新信道参数 RAM[3]。

基金项目：国家自然科学基金（项目编号：60372058；项目编号：60772101）。

信道选项参数（OPT）	
信道传输源地址	
帧计数	阵列计数
信道传输目的地址	
目的帧索引	源帧索引
重载选项	链接地址
目的块索引	源块索引
保留	块计数

图1 参数 RAM 结构

EDMA3 数据分成三维[3]。一定字节的数据可以组成一个阵列，称为第一维；多个阵列数据可以组成一个帧，称为第二维；多个帧的数据又可以组成一个块，称为第三维。对应的 EDMA3 数据传输方式也可分成三种。一次传输一个阵列称为 A 同步传输；一次传输一个数据帧称为 AB 同步传输；一次传输一个数据块则称为 ABC 同步传输[4]。

如图 2 所示，EDMA3 包括两个信道控制器和两个传输控制器。图中虚线左面为信道控制器，虚线右面为传输控制器。其中 DMA/QDMA 信道逻辑用于捕获外部触发或手动触发的 EDMA 事件，根据寄存器设定的各事件的优先级对多个事件进行排序，并送往事件队列。事件队列有两个，每个队列各能接受 16 个事件排队。队列头部的事件根据当前事件对应的参数 RAM 控制参数分别向传输控制器 TC0/1 提交传输请求。

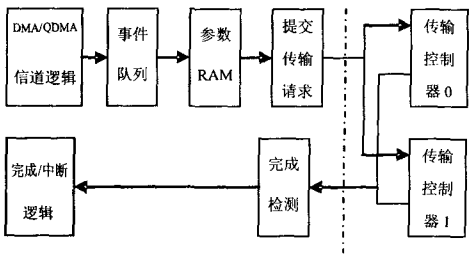

图2 EDMA3 框图

从上面的介绍可知，EDMA3 提供了丰富的控制选项，可以针对不同的应用场合实现不同的控制。其 3 维的数据控制格式，也可以实现各种复杂情况的数据搬移，比如在图像处理中，可以将图像数据从隔行排列的缓冲传输到逐行排列的缓冲，而实现方法也很简单，就是合理地配置其参数 RAM。此外，在复杂嵌入式系统中，系统通常需要传输音频、视频、网络等多种形式、多个器件之间的数据，如果没有很好的数据传输调度，系统性能就很难发挥。EDMA3 控制器队列可以同时接收 32 个 EDMA 事件请求，并根据每个事件的优先级确定其在队列中的位置，这样既保证了高优先级数据传输请求的优先响应，又同时兼顾了低优先级数据的传输需求，只要运用得当，就可发挥事半功倍的效果。

2 视频采集与显示系统中的应用

图 3 是基于 DM6446 的视频采集与显示系统框图，主要模块包括视频输入输出模块、音频输入输出模块、USB 接口和网络接口，以及用于音视频存储的硬盘。图中，前端摄像头采集的 CVBS 信号，经过视频解码芯片 TVP5146 后转变为 BT. 656 信号，最后输入到 DM6446 视频前端。视频前端处理好的数据可以通过 EMIF 接口存储到硬盘。在播放视频数据时，从硬盘读取数据，并经过视频后端输出。输出格式支持 CVBS 输出和 VGA 输出，其中 CVBS 输出由视频后端的编码模块编码后直接送监视器显示，而 VGA 视频信号需要先经过 PVX522 VGA 芯片处理再送给 VGA 显示器显示。

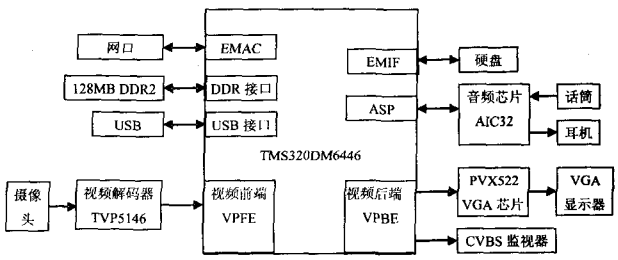

图3 视频采集与显示系统框图

在视频信号的输入端，为节约 MCU 的资源，通常使用 EDMA 来搬运视频数据[5]。视频输入端采集 BT. 656 信号是隔行信号，而 DSP 在处理图像时，通常都是按照逐行的格式处理，如果有一个办法在采集 BT. 656 数据的过程中，隔行数据自动按照逐行格式存放，就能提高系统效率。通过对 EDMA3 结构的研究我们发现，只要合理地配置 EDMA 参数就可以成功实现隔行信号到逐行信号的自动排列。下面的代码给出了奇场亮度（Y）信号接收的 EDMA 配置过程，而色度信号（U/V）以及偶场信号的接收配置方法类似：

```
Edma3_Init();        //初始化 EDMA 模块
Edma3_requestChannel (hEdma, &chId, &tcc);
//申请一个 EDMA 通道
paramSet. srcBIdx = 0;        //设置 srcBIdx 变量
paramSet. destBIdx = 720;        //设置 destBIdx
变量
paramSet. srcCIdx = 0;        //设置 srcCIdx 变量
```

```
paramSet.destCIdx = 1440;      //设置 descidx 变量
paramSet.aCnt = 720;      //设置 acnt
paramSet.bCnt = 1;      //设置 bcnt
paramSet.cCnt = 288;      //设置 ccnt
paramSet.bCntReload = 1;      //设置 Reload 变量
paramSet.linkAddr = &(even_EDMA);      //设置 linkAddr 变量
paramSet.srcAddr= &(FIFO);  //设置源地址
paramSet.destAddr=ViFrm[0].y;      //设置目的地址
Edma3_setPaRAM(hEdma, chId, &paramSet);  //将参数写入到 Param
Edma3_enableTransfer (hEdma, chId, MODE_MANUAL);      //启动 EDMA 传输
```

BT.656 每场信号包含 720x288 像素,所以设置 aCnt,bCnt 和 cCnt 分别为 720、1 和 288。由于视频接口使用 FIFO 结构用于缓冲接收数据,对于 FIFO 的访问使用相同的地址,所以设置 srcBIdx 和 srcCIdx 均为 0。为了实现隔行数据向逐行数据的转换,将 destBIdx 设置为 720,destCIdx 则设置为 1440。这样,在放完一行数据(720 像素)之后,下一行数据与前一行数据的开始点要间隔 1440 像素,中间空出了 720 个像素,正好预留了一行数据的空间。在偶场数据接收时也使用同样的策略,并且将 paramSet.destAddr 设置为 ViFrm[0].y1+720,也就是偶场数据存放的起始地址比奇场数据的起始地址偏移了 720 个像素。这样就顺利实现了一个完整数据帧的接收和存储。

为了实现奇场数据和偶场数据的连续接收,需要使用 EDMA 的链接功能。也就是在奇场数据接收完之后自动切换到对应偶场数据的接收,而当偶场数据接收完毕之后自动切换到奇场数据接收,如此往复。链接功能的实现主要是设置 linkAddr 参数。在前面的代码中可知,奇场数据 EDMA 配置中将 paramSet.linkAdd 设置为 &(even_EDMA),也就是在奇场数据结束后,调用偶场 EDMA 的配置参数,顺利实现两者的切换。同样在偶场数据 EDMA 配置中将 paramSet.linkAdd 设置为 &(odd_EDMA) 即可。

对于视频输出,配置过程正好与上述过程相反,此时 EDMA 将数据连续送到输出 FIFO。由于目的 FIFO 地址固定,所以将 destBIdx 和 destCIdx 设置为 0。其它配置过程类似。

前面介绍了输入和输出视频数据的 EDMA 详细配置过程,通过这些处理,视频数据的发送和接收就可以自动完成,CPU 只需对一些中断和错误做处理即可,大大减轻了 CPU 负担。不过仅有这些还不够,为了保证 DSP 处理数据过程的连贯性和实时性,需要构建发送/接收乒乓双缓冲,有时甚至需要构建三缓冲。

本系统中的乒乓双缓冲与多缓冲的实现方法类似,都放在中断处理函数中执行。视频信号在完成一场和一帧数据接收时都会产生一个中断,在中断处理函数中可以将视频帧号 index 计数,index 对 2 取余数就知道当前帧应该放入乒缓冲还是乓缓冲。代码实现也很简单,以视频输入为例,只要根据 index 不同判定 paramSet.destAddr 等于 ViFrm[0]的首地址还是 ViFrm[1]的首地址,即:

```
paramSet.destAddr=ViFrm[index++ % 2];
```

同理,如果要使用三缓冲,则代码配置为 paramSet.destAddr=ViFrm[index++ % 3]。

3 EDMA 的高级应用

在视频系统的调试过程中,我们发现了很多关于 EDMA 的使用技巧。通过这些技巧,可以减小 CPU 负担,同时可以减小总线带宽和内存占用,从而提高系统的整体性能,保证视频信号处理的实时性。在此给出其中的一个,其它应用原理相同。

在嵌入式视频系统中,为了节省带宽和 DDR 内存空间,所以视频信号进入 DSP 时经常会使用 4:2:0 的格式,所以在视频接收时需要将 4:2:2 格式的 BT.656 信号转变为 4:2:0 格式,而在视频后端输出,需要将 4:2:0 格式信号转换回 4:2:2 格式。如果能在视频输入/输出的过程中自动完成数据格式的变换,则可以很好地节省系统带宽。

在视频输入端数据接收的 EDMA 配置中,可以使用偶场 U/V 覆盖奇场 U/V 的方法来视频实现 4:2:2 到 4:2:0 的转换。而在视频输出时,则可以利用 U 信号每行取两遍,V 信号也是每行取两遍的方法。也就是说 U/V 信号在 EDMA 传输完一行信号后不直接传输下一行,而是将当前行信号重复传一遍。具体的 U/V 信号 EDMA 配置过程如下:

```
paramSet.srcBIdx = 0;   //设置 srcBIdx 变量
paramSet.destBIdx =0;   //设置 destBIdx 变量
```

```
paramSet. srcCIdx = 360;   //设置 srcCIdx 变量
paramSet. destCIdx = 0;    //设置 descidx 变量
paramSet. aCnt =360;   //设置 acnt
paramSet. bCnt = 2;    //设置 bcnt
paramSet. cCnt = 288;      //设置 ccnt
paramSet. bCntReload = 2;      //设置 Reload 变量
```

在上面的配置中,通过将 aCnt 设置为 360,bCnt 设置为 2,bCntReload 设置为 2,而 srcBIdx 设置为 0,就相当于每传输完一行 U/V 信号(360 字节)之后,将该行信号从头部开始再传输一遍。srcCIdx 设置为 360 就相当于在重复传输一行 U/V 信号之后,下一次传输从下一行开始。

4 EDMA 调度策略

要想用好 EDMA,需要很好的理解 EDMA 的传输过程。EDMA 内部划分为 2 个优先级,分别是 Q0(Urgent)和 Q1(Normal),队列 0(Q0)的优先级比队列 1 高。每个队列可容纳 16 个传输请求,且都是 FIFO 结构,先提交的传输先执行传输操作,队列后面的操作必须等待前面的操作完成后才能够被执行,但是各个队列之间的传输可以按照优先级并行操作,当某个操作处于阻塞、等待状态时,其它队列中的任务可以得到执行[6]。

充分理解 EDMA 的结构对于 EDMA 资源的有效分配非常重要。在 DM6446 系统中,我们使用了如下的操作策略:高优先级的队列中执行传输数据少但紧迫的任务,比如 CACHE 访问和小批量的控制命令的发送等操作,因为这些操作对实时性要求很高,操作如果得不到快速的响应,会造成 CPU 的阻塞,从而严重影响系统整体性能[7]。低优先级队列中执行较低实时性要求的大容量数据传输。因为数据量的传输会很长时间的占用 DMA 操作和外部总线操作,如果将其放在高优先级队列中,会使那些数据量本来很少的操作因为优先级低而长时间得不到响应,影响系统性能。虽然视频、音频等大批量数据的优先级设置得比较低,但是由于 DM6446 EDMA3 带宽可以达到 2.4Gbps,所以其传输也能得到保证。

此外要注意不是所有的操作都要依赖 DMA,比如在 DMA 已经十分繁忙的情况下,如果传输的数据很少,也可以直接用 CPU 来访问,或者将两者配合使用来达到最好的效果[8]。

5 结束语

DM6446 的总线设计方式极大的增强了系统性能,提高了数据传输效率,在很大程度上减轻 CPU 的数据传输负担,非常适合于视频应用。中央交换资源的设计是对上一代 EDMA2 系统设计的重要改进。其对参数 RAM 的改进进一步提高了 DMA 传输控制的灵活性和多样性。EDMA 的 128 个信道可以为更多的外设提供服务。

EDMA3 的灵活使用是提高系统整体性能的关键,合理的参数配置和信道链接等功能的运用,可以很好地节省 CPU 资源和系统带宽。同时,在不同类型的 EDMA 类型传输时,需要做好优先级的分配和队列的选择。对于紧急的小批量数据可以选择优先级高的队列,而对于大批量的视频、音频等数据,则可以选择优先级较低一点的队列。这样系统性能可以得到最大的发挥[9]。

参考文献(References):

[1] Texas Instruments Incorporated. EDMA v3.0 (EDMA3) Migration Guide for TMS320DM644x DMSoC. Application Report SPRAAA6 [EB/OL], 2005.

[2] Texas Instruments Incorporated. TMS320DM6446 Digital Media System on-Chip. User data sheet SPRS283[EB/OL], 2005.

[3] Texas Instruments Incorporated. TMS320DM644x DMSoC Enhanced Direct Memory Access (EDMA) Controller User's Guide SPRUE23B[EB/OL], 2005.

[4] Texas Instruments Incorporated. TMS320DM644x DMSoC ARM Subsystem. Reference Guide SPRUE14, 2005.

[5] 李宝林,黄席樾. 实时图像处理系统中 EDMA 实现数据搬移. 信息技术[M],2005(9):35-37.

[6] 徐德文,史忠科. 利用 EDMA 实时传输数字视频图像. 计算机工程与应用[J],2004(19):91.

[7] 丁刚. 从 TMS320C6000 EDMA 的结构来理解和优化其应用. 无线电工程[J],2002(32):8.

[8] 滕小波,耿相铭,武丽帅. 基于 TMS320C6713B 的 EDMA 实时数据流传输[J]. 信息技术,2009(3):47-50.

[9] 林培杰,黄比得,黄春晖. 基于 TMS320C6713 的 McBSP 和 EDMA 实现串口通信[J]. 电子技术应用,2009(2):19-21,26.

收稿日期:2009-08-31

作者简介:李静(1982-),女,助教,主要研究领域为嵌入式系统。