

文章编号: 1671-1742(2007)02-0209-04

# DMA 控制器在 DSP 数据采集系统中的应用

马跃, 卿粼波, 滕奇志

(四川大学电子信息学院图像信息研究所, 四川 成都 610064)

**摘要:** DMA 控制器可以无需 CPU 介入而在内部存储器、外部存储器和芯片外设之间传送数据, 其在 DSP 系统中有广泛的应用价值。基于以 DSP 芯片 TMS320VC5509 为核心的数据采集处理系统, 通过对 FIFO、SDRAM 和 TMS320VC5509 DMA 控制器的介绍, 分别给出了具体的接口电路硬件设计和实际软件的代码示例。

**关键词:** TMS320VC5509; DMA 控制器; EMIF; 数据采集。

**中图分类号:** TN914.3

**文献标识码:** A

## 1 引言

随着信息技术的飞速发展, 数字信号处理器(Digital Signal Processor)技术应运而生, 并得到迅速发展。其开发手段不断改进, 功能日益强大, 性能价格比不断上升, 因而 DSP 系统被广泛应用于当今技术革命的各个领域。基于 TI 公司的 TMS320VC5509 为核心的数据采集系统, 着重介绍了如何利用 DSP5509 的 DMA 控制器, 实现系统中图像数据的传输功能。

## 2 基于 DSP5509 的数据采集系统

系统采用了 DSP + FPGA 的双核结构, 由 CMOS 感光芯片 OV7620 采集图像, 得到数据源, FPGA 作为辅助处理器, 控制部分外围器件并协助采集数据, DSP 芯片 TMS320VC5509 作为核心芯片, 主要负责数据处理。系统结构如图 1 所示:

DSP 芯片除了连接电、时钟和 JTAG 等其工作所必须电路外, 还外接了一块 FLASH 作为程序存储器, 一块 SDRAM 作为数据存储器。FIFO 作为 CMOS 所采集的图像数据的暂存器, 待 DSP 发出特定信号, 便可把其内部存储的数据传至 DSP。

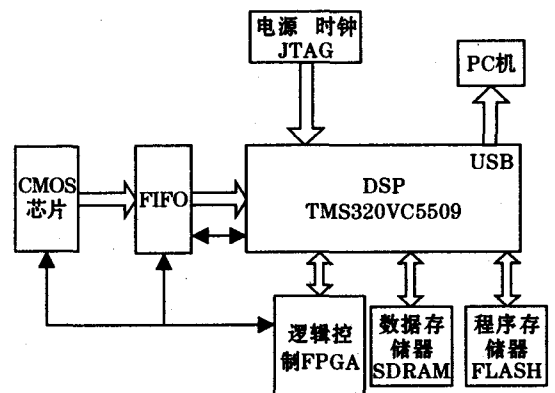


图 1 数据采集系统结构图

## 3 用 DMA 控制器实现数据传输的思想

### 3.1 硬件电路接口

#### 3.1.1 DSP 与 FIFO 的接口

系统中, FIFO 采用的是 AL422B, 这是一款由 AverLogic 公司专门推出的视频帧存储器, 它的容量大小为 384K \* 8bit, 其电路连接如图 2 所示。

此款 FIFO 的容量比较大, 可以完整存放由 OV7620 所采集的一帧图像。FIFO 的具体工作方式, 当系统发出采集信号, DSP 的 /AWE 和 /CE2

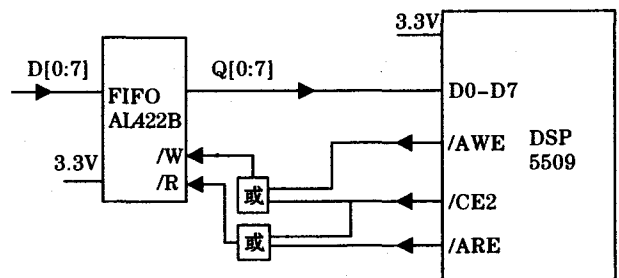


图 2 DSP 与 FIFO 的接口

管脚都输出低电平,通过FPGA所实现的或门,使得AL422B的/W管脚为低,此时FIFO可写,图像数据由OV7620进入FIFO;当采集完一帧图像后,DSP的/ARE和/CE2管脚都输出低电平,通过FPGA所实现的或门,使得AL422B的/R管脚为低,此时FIFO可读,图像数据由FIFO进入DSP。

### 3.1.2 DSP与SDRAM的接口

通过外部存储器接口(EMIF)DSP可与SDRAM实现无缝连接,系统SDRAM采用的是HYNIX公司的HY57V641620,其存储容量为4\*1M,连接电路如图3所示。

DSP的CEO为片选信号,BE0-BE1为字节使能信号,CLKMEM为同步时钟信号,行选通信号/SDRAS,列选通信号/SDCAS和写使能信号/SDWE分别于SDRAM的/RAS、/CAS、/WE管脚相连,DSP的地址线A和数据线D直接与SDRAM的相连。

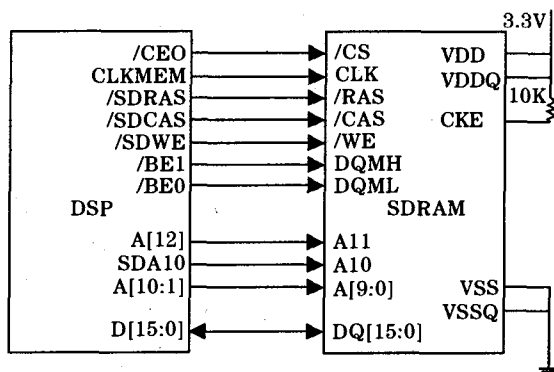


图3 DSP与SDRAM的接口

### 3.2 TMS320VC5509的DMA控制器的特点

DMA可独立于CPU工作;有4个标准端口与内部DARAM、SARAM、外部存储器和外设相连;有一个辅助端口用于HPI和存储器之间的数据传输;具有6个通道;可以设置每个通道的优先级;每个通道的传输可以由选定事件触发;当操作完成后,DMA控制器可向CPU发出中断。

### 3.3 DMA传输

TMS320VC5509的存储空间包括统一的数据/程序空间和I/O空间,数据空间地址用于访问存储器和内存影射寄存器。DSP5509的内部地址线为24位,但当访问数据空间时使用23位地址,并将23位地址左移一位,最低有效位置0,使得地址总线传输24位地址。故DSP在数据空间的寻址范围为8M,其MEMROY MAP如图4所示。

由图4可以看到,以CEO为片选信号的4M容量SDRAM占用了DSP数寻址空间的CE0-CE1两个片区,以CE2为读写控制信号的FIFO始终为CE2片区的起始地址0x800000。FIFO是作为数据源,且地址固定不变,SDRAM是作为数据传输的目的,其起始地址为0x040000,且每接收一个数据单元后,地址增加1。

DMA通道传输的目的端口和源端口由参数寄存器DMACSDP中的DST(SRC)字段来确定:

- 当DST(SRC) = xx00时,目的(源)端口为SARAM;
- 当DST(SRC) = xx01时,目的(源)端口为DARAM;
- 当DST(SRC) = xx10时,目的(源)端口为EMIF;
- 当DST(SRC) = xx11时,目的(源)端口为Peripheral。

DMA通道在数据传输过程中的地址修改方式由DMACCR寄存器中的DST(SRC)AMODE字段来确定:

- 当DST(SRC)AMODE = 00时,目的(源)地址为固定地址,用于单元传输;
- 当DST(SRC)AMODE = 01时,目的(源)地址再每个单元传输完后自动增加,根据数据的位数是8位、16位还是32位,地址分别增加1、2或4。
- 当DST(SRC)AMODE = 10时,目的(源)地址再每个单元传输完后自动增加一个索引值,索引值由单元索引寄存器DMACEI/DMACSEI确定。

当DST(SRC)AMODE = 11时,目的(源)地址再每个单元传输完后按单元索引和帧索引自动增加,索引值由单元索引寄存器DMACEI/DMACSEI和帧索引寄存器DMACFI/DMACSEI确定,又称为双索引。

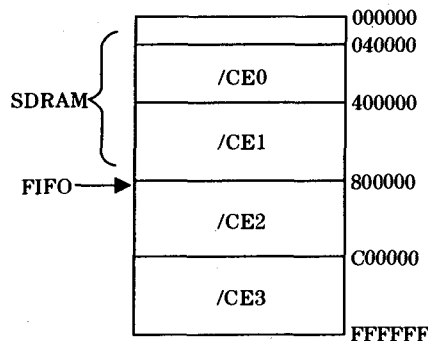


图4 DSP的MEMROY MAP

## 4 程序加载

调用 DMA 库函数首先要在头文件中包含 csl\_dma.h 文件,文件中定义了一名为 DMA\_Config 的结构体如:

```
typedef struct {
    Uint16 dmacsdp;           DMA 通道控制寄存器
    Uint16 dmaccr;           DMA 通道中断寄存器
    Uint16 dmaccir;          DMA 通道状态寄存器
    DMA_AdrPtr dmaccsal;     DMA 通道源起始地址(低字段)
    Uint16 dmaccsau;         DMA 通道源起始地址(高字段)
    DMA_AdrPtr dmacdsal;     DMA 通道目的地址(低字段)
    Uint16 dmacdsau;         DMA 通道目的地址(高字段)
    Uint16 dmacen;           DMA 通道数据单元数量寄存器
    Uint16 dmaefn;           DMA 通道帧数寄存器
    #if DMA_DST_AND_SRC_INDEX_SUPPORT
        对于 5509A,5510PG2

        Uint16 dmacsfi;
        Uint16 dmaceei;
        Uint16 dmacdfe;
        Uint16 dmaceei;
    #else
        对于 5509,5510PG1
        Uint16 dmaefi;       DMA 通道帧索引寄存器
        Uint16 dmaceei;     DMA 通道单元索引寄存器
    #endif
} DMA_Config;
```

在定义了结构体 DMA\_Config 后,便可以在主程序中声明所需的配置:

```
DMA_Config myconfig = {
    .....
}
```

声明配置结构后,需调用 DMA\_open 函数初始化 DMA 句柄:

```
DMA_Handle myhDma;
```

然后打开 DMA 通道 0:

```
myhDma = DMA_open(DMA_CHA0, 0);
```

调用 DMA\_config 函数对 DMA 进行配置:

```
myconfig.dmacssal =
(DMA_AdrPtr)((((Uint32)(myconfig.dmacssal)<<1)
&0xFFFF));
myconfig.dmacdsal =
(DMA_AdrPtr)((((Uint32)(myconfig.dmacdsal)<<1)
0xFFFF));
```

配置通道:

```
DMA_config(myhDma, &myconfig);
```

调用 DMA\_start 函数开始 DMA 传送:

```
DMA_start(myhDma);
```

等待 DMA 状态寄存器的帧状态位指示传输结束:

```
while (! DMA_FGETH(myhDma,DMACSR,FRAME)) {
```

```
}
```

关闭句柄:

```
DMA_close(myhDma);
```

## 5 结束语

着重介绍了如何利用 DSP 的 DMA 控制器在系统中从 FIFO 向 SDRAM 传输数据,经实践检验,此方法结构简单、可靠性高,并实现了 DSP 对并行数据的高速采集,不仅在系统中得到了很好的应用,在其它需要实时数据采集系统中也有很广泛的应用前景。

## 参考文献:

- [1] TMS320C55X Chip Support Library API User's Guide[Z]. TI Inc, 2004.
- [2] The TMS320VC5503/5507/5509/5510 Direct Memory Access (DMA) Controller Reference Guide[Z]. TI Inc, 2005.
- [3] 彭启琮. TMS320VC55x 系列 DSP 的 CPU 与外设[M]. 北京:清华大学出版社,2005.
- [4] 汪春梅,孙洪波,任治刚. TMS320C5000 系列 DSP 系统设计与开发实例[M]. 北京:电子工业出版社,2004.

## Application of DMA controller to DSP image collection system

MA Yue, QING Lin-bo, TENG Qi-zhi

(College of Electronics & Information Engineering, SCU, Chengdu 610064, China)

**Abstract:** The DMA controller can transmit data among the interior memorizer, exterior memorizer and chip peripheral without the CPU intervention. It has comprehensive application value in the DSP system. Based on the TMS320VC5509 image collection system the material demonstration for the interface circuit hardware design and the practical software code are provided with the introduction of the FIFO, SDRAM and TMS320VC5509 DMA.

**Key words:** TMS320VC5509; DMA controller; EMIF; data collection