

即时加速非对齐数据传输的 dma 设计方法

张彤^{1,2} 周芝梅^{1,2} 赵东艳^{1,2} 张海峰^{1,2}

(1. 北京智芯微电子科技有限公司 国家电网公司重点实验室 电力芯片设计分析实验室 北京 100192;

2. 北京智芯微电子科技有限公司 北京市电力高可靠性集成电路设计工程技术研究中心 北京 100192)

摘要: 针对软件处理字节非对齐数据传输时效率低的问题,提出了一种资源占用少、低功耗的即时拼接直接存储器存取(DMA)设计方案。用户按照字节传输方式配置 DMA 进行非对齐数据传输,DMA 控制器自动进行适当的字节拼合,从而以字传输方式完成数据传输。该方案不仅能够应对 32 位系统所有字节不对齐和递增递减方向的组合,而且完全由硬件电路处理、软件编程没有额外的开销,仿真验证表明,该设计方法能够显著加快 32 位低功耗 SOC 系统字节非对齐数据的传输速度。目前已经应用于一种用电采集 MCU 芯片,用于协助软件高速处理加解密数据、通信协议帧等,结果表明,该芯片的 DMA 模块传输速率高、操作简便,可以满足设计要求。

关键词: 直接存储器存取设计;字节非对齐;即时加速;低功耗 SOC

中图分类号: TN402 文献标识码: A 国家标准学科分类代码: 510.4030

Method and implementation for DMA on the fly expediting transfer of unaligned data

Zhang Tong^{1,2} Zhou Zhimei^{1,2} Zhao Dongyan^{1,2} Zhang Haifeng^{1,2}

(1. State Grid Key Laboratory of Power Industrial Chip Design and Analysis Technology, Beijing Smart-Chip Microelectronics Technology Co. Ltd., Beijing 100192, China; 2. Beijing Engineering Research Center of High-reliability IC with Power Industrial Grade, Beijing Smart-Chip Microelectronics Technology Co. Ltd., Beijing 100192, China)

Abstract: We propose the design of DMA controller for 32-bits low power SOC system, which expedites transfer of unaligned data through on-the-fly reassembling bytes by hardware circuit, requiring fewer resources, power and no extra software overhead. Although user configures the transfer of unaligned data in byte format, the DMA controller can automatically reorganize bytes in proper way, so as to transfer data in word format correctly and faster. Verification shows the design can evidently improve transfer efficiency of unaligned data. The design method has been applied to a MCU chip for electricity information acquisition system, and assists software to deal with encryption and decryption, to form communication frames, etc. The results show that the DMA controller works with high efficiency while easy to use, so the design meets the requirements.

Keywords: DMA controller; unaligned data; on-the-fly expedite; low power SOC

0 引言

SOC 系统需要接直接存储器存取(direct memory access, DMA)来协助 CPU 频繁地搬移加解密、通信协议、多媒体等数据,但是实际工作中经常面临字节地址非对齐数据传输效率低的难题。本文试图从 DMA 控制器电路设计上找到突破口,达到功耗、传输效率均较佳的平衡点。

DMA 是通过硬件电路实现存储器与存储器之间、存储器与外部设备之间直接高速传送数据,传输效率远远高于 CPU,并且可以将 CPU 从繁重而重复的数据搬运工作

中解放出来,在某些应用场景下,甚至大部分时间允许 CPU 进入休眠状态,大幅度降低动态功耗。

非对齐数据普遍存在于 SOC 系统的各种应用场景,比如安全、通信、图像、音视频、多媒体扩展等。文献[1]指出,加密算法只对帧数据部分进行加/解密,而不对帧头、帧尾及空白字符进行处理。不同协议的帧头长度不同,加密数据的字节起始地址不一定与字地址对齐,搬移前后也不一定对齐,如图 1 所示。文献[2]指出越来越多的处理器集成了 SIMD 扩展,但是非对齐访问对系统的多媒体处理性能造成损失。这类研究提出了数据重组、向量化技术^[3]、编译

等软件方面的解决方法,但是较少从电路设计和实现角度思考解决方案。

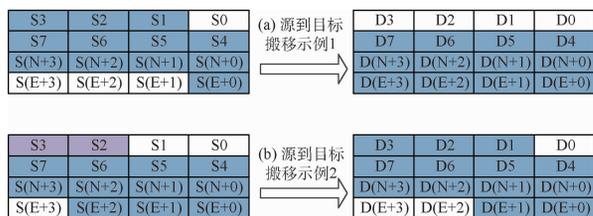


图 1 非对齐数据传输示例

1 SOC 系统传输非对齐数据的瓶颈

图 1 中,每个格代表 1 Byte 的数据,需要传输的数据以灰色方格表示。 $S(N)$ 表示源字节地址, $D(N)$ 表示目标字节地址, $N=(\text{Word Row}-1)\times 4$,即字地址行数减 1; $S(E)$ 表示结尾行。图 1(a)要求将从 $S1$ 至 $S(E+0)$ 的若干个字节数据搬移到 $D4$ 至 $D(E+3)$ 位置。

一般情况下,DMA 采用字节传输方式处理此类问题,由于 32 位系统通常按照字格式(32 bit)访问存储器和外设,实际每次只能传输 1 Byte(8 bit)的有效数据,由此将浪费 3/4 的带宽和功耗。

传输数据不对齐是指源起始字节地址、目标起始字节地址互相不对齐。其特点是字节地址的交叉组合多,源字节地址(字地址内包含 0、1、2、3 4 种字节地址)到目标字节地址(4 种)的组合可达 16 种。

如果同时考虑传输时地址向上递增和向下递减的处理方法不同,则有 32 种组合。

如图 2 所示,如果源区域和目标区域有重叠,图 2(a)使用递增方式,会覆盖源数据;图 2(b)使用递减方式,不会覆盖源数据。

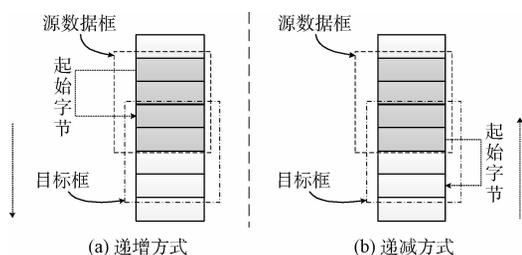


图 2 传输方向递增和递减的区别

基于此,本文对比并提出新的 DMA 设计方案,同时具体说明关键模块的设计实现,并验证和分析相关结果。

2 解决思路和实现方案

以往视频、图像处理等多媒体芯片的 DMA 设计中,出现过几种数据不对齐的处理方法。文献[4]提出了一种二

维 DMA 方法,自动补齐不在图像边界内的子图像块数据。但是这种方法不能满足低功耗 SOC 系统应用:仅能够处理数据地址不对齐转变成对齐的情况,不能处理从一种形式的字节地址不对齐(或者对齐)转变成另一种形式的字节地址不对齐(或者对齐)的情况;额外需要一定深度的 FIFO 暂存对齐数据。

文献[5]提出了将对齐专用 DMA 模块、加密模块、去对齐专用 DMA 模块组成一套电路的方案。这种方案固化了 DMA 的用途,而且对齐和去对齐占用两路 DMA,对于功能多样化的 SOC 系统来说难以接受。

文献[6]针对 SIMD 单元主要适用于寄存器长度对齐的矢量数据的特点,在 DMA 电路内部设计专门的硬件单元,即时将地址不对齐的代码先转成地址对齐或者其它形式的不对齐。但这种方法展开和移位逻辑复杂,需要指定几个专门的存储空间(全局存储器、本地存储器、Cache),用于暂存对齐数据。

这类设计方案对于主要处理加解密电路、通讯协议等的低功耗 SOC 系统而言,占用资源过多。相比之下,本文提出了一种避免使用 FIFO、SRAM 等消耗大量电路资源的 DMA 设计方法,在传输过程中即时进行字拼接。利用该方法,源字节地址和目标字节地址不对齐时,软件仍然按照字节形式对 dma 进行配置,如果满足一定的条件,专门的硬件电路可利用少量移位寄存器和拼接逻辑电路,以字为单位加速完成传输,将传输效率提高约 4 倍,且传输过程不需要软件干预。

2.1 基本设计思路

将非对齐传输过程分成 3 个区域,分别是开头区域、中间区域和结尾区域,在每个区域按一定的规律进行预读、拼接,如图 3 所示。

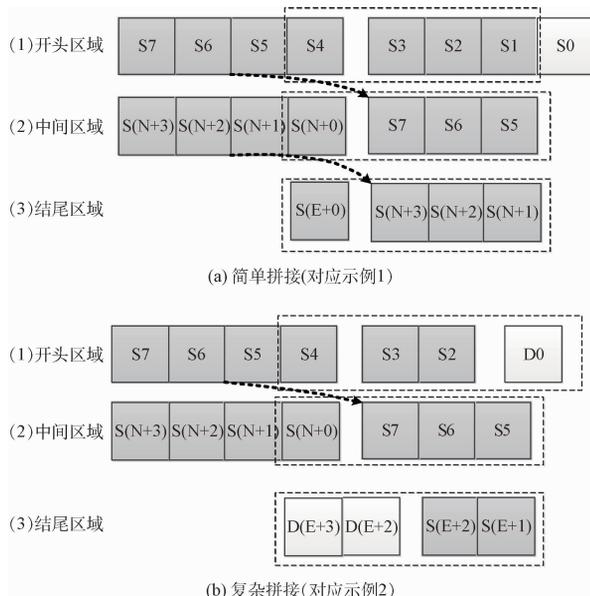


图 3 分区拼接

这种处理方式开头区域和结尾区域的动作较多,如果传输字节数较少,则未必能够提高传输效率。

2.1.1 简单拼接示例

针对图 1(a)的传输要求,开头区域进行两次预读取,并选取 S4、S3、S2、S1 四个字节的数据合并为第 1 个字,写入目标地址,如图 3(a)所示。

中间区域和结尾区域将上一次遗留的数据和本次的部分数据合并为字,写入相应目标地址。

2.1.2 复杂拼接示例

在更复杂的情况下,比如图 1(b),开头区域需要做 3 次预读取和 1 次写入,结尾区域会做两次预读取和 1 次写入。

第 1 次,当开始字节不是边界位置时,需要从目标起始地址的边界位置读取 1 个或者多个字节,以便作为 1 个字整体写回。如图 3(b)中的 D0,第 2 次读取 S3、S2,第 3 次读取 S4。同理,当结束字节不是边界位置时,第 1 次读取 S(E+2)、S(E+1),第 2 次要 从目标结束地址的边界位置读取 1 个或者多个字节,比如图 3(b)中的 D(E+2)和 D(E+3)。

2.2 DMA 控制器结构和实现

本文提出的 DMA 控制器设计借鉴了文献[7]和[8],针对拼接的特点修改了有限状态机,并增加了多功能移位寄存器,如图 4(a)所示。本设计使用 Verilog-2001 HDL 进行电路描述。

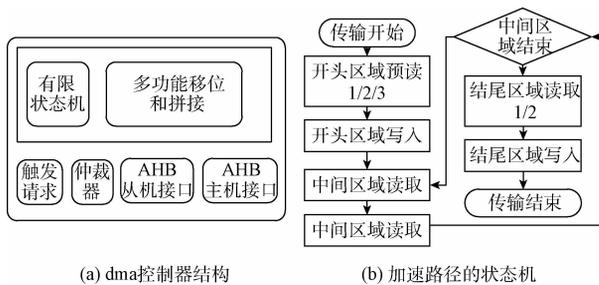


图 4 结构与加速路径

图 4(a)中灰色长方形代表单个通道,一个 DMA 可以包含多个通道^[9]。各通道共享从机接口、主机接口、仲裁器等子模块。AHB 总线从机接口用于接收来自 CPU 或者上位机的传输配置信息。AHB 总线主机接口用于发起存储器、外设访问。当多于一个通道同时工作时,仲裁器模块负责仲裁那个通道工作,采用 round-robin 等指定算法。单个通道内的触发请求模块负责收集来自外设的传输请求,通知有限状态机进入传输开始状态。

2.2.1 设计要点 1: 移位寄存器

单个通道内的 8 Byte 多功能移位寄存器 (shifter) 负责存放和拼接数据。在开头区域,根据配置判断需要几次预读、读取哪些位置、存放在移位寄存器的哪些位置,其中 4 Byte 存放当前需要写入目标位置的数据,另外 4 Byte 存放下一次可能会用到的数据 (saved)。

完成写入后,进入中间区域。在读取新字的同时,移位寄存器将适当的字节移位。根据配置的传输数量、记录的已传输数量等条件,判断中间区域是否已经结束。

结尾区域也需要判断是否需要从目标位置预读用于写回的数据,拼接成字,写入目标位置。结尾区域结束后,一次传输完成。

移位和拼接逻辑的编程示例如下:

```
always@ (posedge clk or negedge rstn)
```

```
.....//省略复位、其他分支
```

```
4'h1: begin // 递增、源起始字节为 1、目标起始字节为 0 的分支
```

```
For (pos=0; pos<3; pos=pos+1) begin//pos 遍历多个字节,处理 shifter[2:0]
```

```
if (headread_phs1) //开头阶段第一次读
```

```
shifter[pos] <= HRDATA[pos+1];
```

```
else if (read_phs1 || endread_phs1) //中间、结尾阶段第一次读
```

```
shifter[pos] <= saved[pos]; //使用存的字节
```

```
else if (endread_phs2 && destkeep(pos, dest_addr [1:0], trans_number [1:0], increment)) //结尾阶段第二次读,判断是否需要读取并保留目标位置的某些字节
```

```
shifter[pos] <= HRDATA[pos];
```

```
end //第一个 for 结束
```

```
for (pos=0; pos<4; pos=pos+1) begin//pos 遍历多个字节,处理 shifter[6:3]
```

```
if (headread_phs1) //
```

```
shifter[pos+3] <= HRDATA[pos];
```

```
else if (endread_phs2 && destkeep(pos+3, dest_addr [1:0], trans_number [1:0], increment) && pos == 0) //此组合下仅在 pos==0 位置判断是否保留
```

```
shifter[3] <= HRDATA[3];
```

```
end //第二个 for 结束
```

```
end //分支结束
```

```
.....//省略其它分支
```

2.2.2 设计要点 2: 有限状态机

单个通道内的有限状态机包括加速和非加速两条路径。非加速路径相对简单,与文献[10]对传输过程的描述类似,主要包括等待状态、传输开始、读取状态、写入状态(根据传输数量在读取和写入之间循环)、结束状态等。设计创新主要体现在加速路径,如图 4(b)所示。在未开启拼接使能、或者拼接加速条件不满足时,状态机按照普通传输状态跳转;如果开启拼接使能而且加速条件成立,则按照加速路径下的状态跳转。

本设计可以保证加速路径的状态机与多功能移位寄存器共同完成各种非对齐组合、递增递减组合情况下数据的即时拼接、暂存、读写操作。这种设计方式兼容了加速和非加速两种模式,保留了两种模式的优点,通过软件配置可以

在二者之间自由切换;且部分资源可以共享,因此新增资源占用较少。

3 DMA 设计方案应用

低压远程用电采集系统服务于用电信息管理,包含系统主站、集中器、电能表、采集器和通信信道等设备^[11]。用电采集系统目前支持的通信方式有电力线载波通信、无线公网通信、电力 485 通信等。采集终端、集中器等设备中安装安全加密模块,用于采集设备与主站、智能电能表之间进行的身份识别、安全认证、关键信息和敏感信息安全传输^[12]。

本文提出一种用电采集系统专用 MCU 芯片设计,能够满足集中器、采集器等设备的性能需求。首先选择合适的 CPU、存储器、通信外设等资源,其次将支持非对齐数据加速传输的 DMA 控制器集成到 MCU 芯片内。芯片架构如图 5 所示。

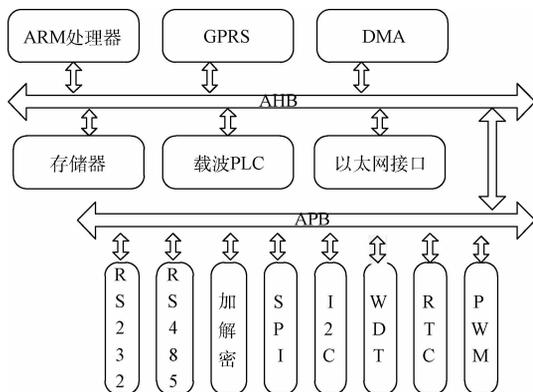


图 5 用电采集专用 MCU 架构

ARM 处理器选用 Cortex-M 系列,采用 AHB、APB 总线架构。

通信模块支持 GPRS 通信、载波通信、485 通信等用电采集常见的通信手段,满足多种组网条件^[13]。

安全模块集成有国家密码管理局认可的对称密钥加密算法和非对称密钥加密算法。

在此 MCU 系统中,非对齐数据主要来自于通信模块和解密模块。1)接收到的通信帧中,数据起始帧可能位于字中的任一位置,需要将每 2^N Byte 为 1 组送往解密电路进行解密,解密之后,搬移到内存区域。在某些特殊情况下,最后一组的数量少于 2^N Byte。如此循环输出密文^[14]。

2)发送时,从内存区域将待发送数据,每 2^N Byte 为 1 组送往加密电路进行加密。加密后的数据填充到通信帧内,起始位置也可能位于字内的任一位置。加密与解密流程类似^[15]。

为了开启非对齐加速功能,软件对 DMA 进行如下配置:1)字节传输模式;2)使能加速选项;3)指定源地址、目标地址;4)开启传输。

此外,由于数据帧的长度不定,且经常分成若干组,为了减少搬移过程中 CPU 的干预,DMA 电路可以使能触发模式。当加密模块完成了对上一组数据的运算之后,在 encry_done 触发信号上产生单个高脉冲,DMA 电路检测到此脉冲信号之后,开启下一组数据的传输。这样,CPU 在全部组传输完毕之后,处理一至两次中断即可。

4 EDA 仿真和 FPGA 验证结果

在用电采集系统专用 MCU 芯片及其验证平台上,对 DMA 控制器的非对齐传输模式进行了仿真。图 6 所示为通过 VCS-MX/I-2014.03 仿真器记录的单通道功能仿真波形。



图 6 单通道加速模式下字节拼接波形

在 xilinx 的 XC7K325T 上实现频率可达 50 MHz 以上,不开启加速功能时,单通道非对齐数据字节传输速率为 24.8 MB/s,开启加速功能时单通道速率可达 102 MB/s。

当系统频率为 32 MHz 时,若不开启即时加速传输功能,单通道非对齐数据字节传输 1 KB 数据耗时 75 μ s,开启及时加速功能后,单通道字节传输 1 KB 数据耗时 18.9 μ s。采用 SM2 算法对 1 522 Byte 长度报文进行加密实验,直接通过 CPU 和算法电路处理耗时 4.17 ms,使用 DMA 加速功能配合算法电路处理需要 0.25 ms,加密运算速度提升 16 倍。

5 结 论

本文提出了一种新的即时加速非对齐数据传输的 DMA 控制器设计方法,并成功应用于用电采集系统专用 MCU 芯片设计中。一方面,整个方案仅需要 8 Byte 的移位寄存器用作拼接的空间,较使用 FIFO、RAM、CACHE 等方案可节省大量的存储器;另一方面,拼接逻辑电路高效地处理 32 种组合,整个过程完全由硬件电路自动处理,避免了复杂的软件编程,可大幅度地提升 32 bit 低功耗 SOC 系统的非对齐数据传输效率,对于 MCU 芯片性能的提升起到了非常重要的作用。

参考文献

- [1] 吴琼,刘建国,黄庆超,等. 基于以太网帧结构的 AES 加解密算法的实现[J]. 光通信技术,2016(11): 16-19.
- [2] 魏帅,赵荣彩,姚远,等. 面向 SIMD 的数组重组和对齐优化[C]. 全国高性能计算学术年会,2011.
- [3] 李玉祥,施慧,陈莉. 面向量化的局部数据重

- 组[J]. 小型微型计算机系统, 2009, 30(8): 1528-1534.
- [4] 方应龙, 汤加跃. 一种专为存取图象块优化的二维 dma 传输方法. CN 100552654 C[P]. 2009.
- [5] DOLGUNOV BORIS (IL), MINZ LEONID (IL); KROTMAN ROY (IL). Secure data processing for unaligned data [P]. 国外专利: WO2009057091, 2009-5-7.
- [6] VUJIC N, GONZALEZ M, CABARCAS F, et al. DMA++: On the fly data realignment for on-chip memories[C]. HPCA-16 2010 The Sixteenth International Symposium on High-Performance Computer Architecture, IEEE, 2010;1-12.
- [7] 时岩. 基于 AHB 的 DMA 控制器的设计[J]. 微计算机信息, 2010, 26(23):102-104.
- [8] 唐平, 郑建宏. 基于 AHB 总线的 DMA 控制器的实现与应用[J]. 电子测试, 2009(11):64-68.
- [9] 陈伍敏, 刘荣生, 罗阔阔, 等. 基于 Crossbar 的多通道 DMA 控制器设计与实现[J]. 中国集成电路, 2013(Z1): 52-57.
- [10] 乐湘云. DMA 传输技术与操作技术的探讨[J]. 计算机光盘软件与应用, 2010(13):50-51.
- [11] 章景平. 低压远程用电采集系统的集中器设计[J]. 电子技术与软件工程, 2015(16):105-106.
- [12] 胡江溢, 祝恩国, 杜新纲, 等. 用电信息采集系统应用现状及发展趋势[J]. 电力系统自动化, 2014(2): 131-135.
- [13] 徐文涛. 用电信息采集系统通信技术[J]. 中国科技信息, 2014(24):168-170.
- [14] 程海, 丁群, 杜辉, 等. 基于 FPGA 实现的 SMS4 算法研究 [J]. 仪器仪表学报, 2011, 32(12): 2845-2850.
- [15] 李湘锋, 赵有健, 全成斌. 对称密钥加密算法在 IPsec 协议中的应用[J]. 电子测量与仪器学报, 2014, 28(1):75-83.

作者简介

张彤, 1975 年出生, 工学硕士, 中级工程师, 研究方向为芯片设计。

周芝梅, 1977 年出生, 工学硕士, 中级工程师, 研究方向为集成电路设计。

赵东艳, 1970 年出生, 工学硕士, 教授级工程师, 研究方向为信号处理与专用集成电路设计。

张海峰, 1978 年出生, 工学硕士, 高级工程师, 研究方向为芯片设计。