# 2相关技术介绍

本论文研发的基于LS1024A的海洋油气勘探数据采集传输系统，使用嵌入式技术开发系统软件框架，使用PCIe接口技术和DMA技术实现处理器和FPGA之间的数据传输。本章将介绍本系统在整个研发过程当中所设计到的关键技术，包括嵌入式处理器，高速串行通信总线接口（PCIe）、内存直接访问技术（DMA）和循环冗余校验（CRC）。

## 2.1嵌入式处理器

嵌入式处理器是嵌入式系统的核心部件，掘不完全统计，目前全世界的嵌入式处理器品种已经有成千上万种之多。随着人工智能、物联网等行业的快速发展，人们对于嵌入式处理器的需求也不断增高，同时也对嵌入式处理器的运算能力、可扩充能力、系统稳定性、功耗和集成度等各方面提出了更高的要求，为了不断适应各方面需求，嵌入式处理器的体系结构也经历了指令集从复杂指令集运算（Complex Instruction Set Computing，CISC）、精简指令集运算（Reduced Instruction Set Computing，RISC）、显式并行指令集运算（Explicitly Parallel Instruction Computing，EPIC）、到超长指令字指令集运算（Very Long Instruction Word，VLIW）,位宽从4位、8位、16位、32位到64位；寻址空间从64kB到256MB甚至更大；运算速度从0.l MIPS到5000 MIPS甚至更快；常用封装从8个引脚到725个引脚甚至更多。另外，处理器的集成度进一步提高，并且功耗也有了明显的降低；。

### 2.1.1 嵌入式处理器概述

嵌入式处理器可以分为以下几类：嵌入式微处理器（Embedded Microprocessor Unit，EMPU），嵌入式微控制器（Microcontroller Unit，MCU），DSP处理器（Embedded Digital Signal Processor，EDSP）以及嵌入式片上系统（System on Chip）。

嵌入式处理器的基础是通用计算机中的CPU，专门设计在指定装配的电路板上，只保留与嵌入式应用有关的母板功能，但是电路板上必须包括总线接口、ROM、RAM、以及各种外接设备等。目前主要的嵌入式微处理器有X86、DSP、PowerPC、MIPS、ARM系列等。

一般情况嵌入式微处理器具备以下4个特点：

1、实时的多任务处理能力。能够完成实时多任务，并且具有较短的中断响应时间，从而使内部的代码和实时内核的执行时间减少到最低限度；

2、强大的存储区保护能力。由于嵌入式系统的软件结构已模块化，为了避免各软件模块之间出现非预期的交叉作用，必须设计功能强大的存储区保护机制，同时也有利于软件诊断；

3、可扩展的处理器架构。能够最迅速地开发出满足应用的最高性能的嵌入式微处理器；

4、较低的功耗。嵌入式微处理器大都用在便携式场景，比如无线设备，移动通信设备等需要靠电池供电的嵌入式系统，其需要的功耗只有毫瓦甚至微瓦级别。

### 2.1.2 Layerscape系列处理器

Layerscape架构是NXP公司基于ARM架构推出的新一代片上系统（System on Chip）的底层软件架构，旨在充分提取、开发现实条件，让程序员能够以非常轻松的方式高效地“释放”每一块芯片的性能。Layerscape架构延伸了目前多核芯片设计发展的趋势，以便让芯片性能最大化，同时，该架构也可提取足够复杂的硬件，使得软件开发变得更加灵巧、高效、可维护并且相对简洁。简而言之，Layerscape架构实现了手动编写汇编代码的高性能、高效率，高级语言的简易性和代码易维护性三者之间的平衡。

Layerscape内部架构框图如下所示：

图2.1：Layerscape架构内部框图

如图2.1所示，在Layerscape架构中，按照逻辑方式划分可将处理器划分为以下三层：最底层快递包输入/输出层（EPIL），中间层加速包处理层（APPL）和最顶层通用处理层（GPPL），分别粗略地代表了标准OSI模型的低、中和高层。

最底层快递包I/O层提供了具有决定性线速性能的支持L2+转换功能的所有网络接口，同时还包含了诸如Ethernet、Interlaken和RapidIO等网络数据包接口。虽然PCI Express并不属于网络接口，但由于其通常在堆栈中刀片之间的接口，，故将其概括在此处。至于其它一些重要但与此不相关的接口，比如USB接口或者SATA接口，不属于该接口层的组成部分，其被分至芯片系统接口（System Interface）模块，如图左侧所示。

中间层加速包处理层包含芯片的包处理元素，这些元素可通过定义规范明确的接口与传统处理器通信。它们有的是可编程引擎，有的是硬接线加速器，或者是二者的组合。通过传统同步的、顺序的、完整运行的模型，加速包处理层可提供客户自定义的附加值功能，并且可通过嵌入式C语言的结构化编程实现。

最顶层通用式处理器层属于通用性质层，面向用户/开发人员免费提供，可用于操作系统、高级语言代码、应用和其它附加值功能。与Layerscape架构提倡的效率和硬件独立性的价值观念保持一致。这一层可以同时支持ARM内核与Power Architecture。正如我们所知道的，ARM架构技术通常属于小端字节序（Little-Endian），而Power Architecture技术通常采用大端字节序（Big-Endian），但是Layerscape架构很完美地支持了这两种技术。

Layerscape架构结合了全世界普遍采用的模块化、高水平编程模型以及目前性能最强的通信处理器，使得无需进行高级别的硬件工程设计，便可轻松获得高级通信引擎。除此之外，更难能可贵的是它可以直接作为一代QorIQ LS系列处理器直接被其后继产品所取代而不用重新去学习每一个芯片的详细细节。具有清晰定义和界限的编程模型可以在芯片之间、代代之间保留下来，这并非是视硬件工程实现更新任务弃之不用，而是基于开发人员的工作构建产生。简而言之，Layerscape架构保留了团队在开发过程中最重要和最有价值的东西保留了下来：即它独具特色的软件。

### 2.1.3 LS1024A处理器

本论文研究的数据采集传输系统以LS1024A处理器为核心进行设计，LS1024A处理器以LS102MA为基础进行改进，具有非常高的处理能力，其I/O接口与飞思卡尔创新的多层总线架构相结合，可跨所有数据接口实现无阻塞的并发事务处理，从而最大限度地减少片上数据包处理延迟，优化了数据包的处理速度。另外，LS1024A利用ARM的高能效核心技术和飞思卡尔的低功耗设计流程，使其实现同类产品中最低的功耗。符合本论文研究系统的低功耗，小型化指标。

LS1024A处理器的架构图如图2.x所示。在LS1024A内部集成两个ARM Cortex A9内核器件，其主要性能特征如下：

* 精简指令集（Reduced Instruction Set Computing，RISC）体系结构；
* 单内核主频达到1.2GHz，共可提供高达6000 DMIPS的处理性能;
* 每个内核都有自己的高速缓存单元，包括32KB的高速指令缓存，32KB的数据缓存以及总共256KB的L2缓存;；
* 一个带宽高达533MHz的16/32位DDR3 SDRAM控制器；
* 一个高速串行外设接口（HS-SPI），带宽高达50MHz，2个从机选择；
* 一个低速串行外设接口（LS-SPI），带宽为16MHz，3个从机选择；
* 两个支持Gen1（2.5Gbps）和Gen2（5Gbps的）的PCIe接口；
* 三个可编程SerDES接口，最高带宽可达5Gbps；
* 64个通用输入输出接口（GPIO）；

图2.2：ARM LS1024A处理器结构框图

## 2.2 PCIe接口技术

PCI-Express（Peripheral Component Interconnect Express），官方简写为PCIe，是一个高速串行计算机总线标准，其原名为“3GIO”，是由Intel公司在2001年提出的，旨在替代老的PCI，PCI-X和AGP标准[]。PCIe是高速串行点对点双通道高带宽传输，连接PCIe两端接口设备分别独享通道带宽，不占用总线带宽，可支持端对端的可靠性传输。在2002年，PCIe交由PCI-SIG（PCI特殊兴趣组织）并由其拟定并发布了PCI-Express1.0标准，才正式改名为“PCI-Express”。在2003年，PCI-SIG发布PCI-Express 1.1标准，进一步对一些规范重新进行声明和定义。在2007年，PC1-Express 2.0标准发布，其中，X1模式的数据传输速率达到了500MB/s，比PCI-Express1.1标准数据传输熟虑提升了一倍。另外，X4模式的数据传输速率达到2GB/s，而最高的X32模式的数据传輪速率甚至能够达到16GB/s。目前最新的PCIe标准PC1-Express 3.0，其比特率为8Gbps，约为上一代标准带宽的两倍。并且增加了数据发射接收均衡、锁相环改善以及时钟数据恢复等一系列十分重要的功能，极大的改善了数据传输和保护性能。

既然PCIe总线标准支持如此高的传输带宽，如果直接使用CPU来负责整个PCIe的数据传输过程，CPU需要从来源把数据的每一个片段拷贝到暂存器，再写到新的地方，这必然导致整个系统的业务处理能力下降。再加上一般的嵌入式CPU主频普遍较低，这时如果再进行大数据传输，CPU的绝大部分资源将会浪费在数据传输上，以至于CPU没有空闲去执行系统的其它业务。通常在这种情况下，我们会选择PCIe专用的内存直接访问机制（Direct Memory Access）来进行实际的数据传输。

## 2.3 DMA技术

DMA（Direct Memory Access）是一种允许外接设备在不使用系统CPU的情况下直接访问系统内存的硬件机制。它允许不同速度的外设进行沟通，不需要依赖CPU的中断负载。CPU只需要对DMA控制器进行相应的配置，后续的传输动作本身由DMA控制器来执行和完成，这样的操作不仅没有增加CPU的工作拖延，反而可以将CPU腾出去执行其它任务，大大提高了系统的处理能力。

DMA的使用在带来高效数据传输的同时也导致了缓存一致性的问題。由于当前的CPU都带有缓存（cache），CPU直接读写cache的数据，而DMA访问的是外部内存中的数据。因此就会出现以下两种情况：

1．CPU读取超前：即当DMA更新完外部内存中的数据后，cache没有进行该数据的更新，导致CPU访问到cache中的旧数据；

2．DMA读取超前：即当CPU更新完cache的数据后，没有将该数据更新至外部内存，导致DMA访问到外部内存中的旧数据。

对于这个问題，一般有两种解决方法：

1．使用一致性内存（Coherent Memory）：由硬件来确保cache数据和外部内存数据的一致性。当DMA向外部内存写数据后，通知缓存控制器更新cache数据，防止CPU读取超前；当DMA向外部内存读数据前，通知缓存控制器将cache数据全部清空到外部内存中，防止DMA读取超前。

2．对于使用非一致性内存（non-coherent memory）的系统，通常使用软件控制的方式来解决。当CPU向cache读数据时，由软件确保cache中的数据是最新的，防止CPU读取超前；当DMA读外部内存中的数据时，由软件来确保外部内存中的数据是最新的，防止DMA读取超前。

## 2.4 CRC校验

嵌入式系统处理器与其外接设备之间、与上位机之间通过各种传输接口传输数据时，不可避免的会被各种噪声影响，导致传输数据出错的情况。为了提高系统的可靠性，需要对传输的数据进行差错控制。循环冗余校验（Cyclic Redundancy Check，CRC）以其简单的编码解码方法、高效的检错能力被广泛运用。

CRC循环冗余校验时一种线性分组码[3]，在控制一定冗余的前提下就具备较强的检错能力。含有CRC校验码的传输数据信息由信息码元和检验码元两部分构成，其中信息码元为当前数据包的原始数据，检验码元为软件通过CRC添加的拓展位，其结构如下图所示：

图2.3：具有CRC校验码的传输数据信息结构图

假设传输的n位信息码为，将这n位的信息码M的每一位看成多项式的系数，在这个多项式的后面添加k个0，其多项式展开为：，将此多项式当作被除式，选择一个k次的多项式当作除式，相除后可得商式和最高次为k-1的余式，即：





多项式相除采用模2除法，加减采用没有进位借位的模2运算，从形式上讲等同于异或运算，因此，相同的多项式相加或者相减都为0。在式2.2两边都加上余式，则有：



从式2.3可知，式2.3左边表达式为除式的整数倍，它正好是多项式往左移k位后附加余式的结果，该多项式的系数为，其中高位的n位为原始的信息码，即有效数据，低位的k位则为CRC检验码，发送端最终发送出去的信息为，接收端在接收到信息时，将的多项式展开式除以协议定好的除式，判断余数是否为0，如果余数为0，则可判定信息在传输过程中并未出现差错，否则即表示传输过程出错。

图2.3演示了一个CRC循环校验码的生成过程，被除数“10101010”对应的多项展开式为：，除数“10011”对应的多项展开式为：，即CRC-4-ITU，所得的R6即为CRC校验码。

图2.4：CRC校验生成示例

通过上述例子可以看出：

* 每次迭代Rx的首位决定Gx的值，如果Rx首位为1，则Gx为“10011”，否则为“0”，示例中略去Rx首位为0的情况，所以看到的Gx都为“10011”；
* 每次迭代Rx的首位都会被移除，这样就可以舍去G的最高为来运算；
* 每次迭代只对信息码的前x位有效，考虑构建一个x位的寄存器Reg。

综合上述三点，计算步骤可改为读取信息码的前x位到寄存器Reg中，抛弃首位，向左移，同时将信息的后一位移入寄存器Reg，如果抛弃的首位是1，则与舍去最高为的G异或，如果抛弃的首位是0，则与0异或，循环至信息最后一位移入寄存器，计算可得CRC校验码。

传统的校验函数按位移动信息码，每处理1字节的信息码时，需要进行8次以为操作，校验速度很慢，这在RAM资源紧张的嵌入式操作系统中是非常不合理的。于是便有了直接查表法。同样以被除数“10101010”，除数“10011”为例，将被除数按每个块4位数据划分，算上填0的4位，共被分成了3个块。

表2.1 被除数分块表

|  |  |  |  |
| --- | --- | --- | --- |
| M | Block 1 | Block 2 | Block 3 |
| 1010 | 1010 | 0000 |

下图展示了4次迭代步骤，浅色背景表示在寄存器Reg中的数据。

|  |  |  |
| --- | --- | --- |
|  | Block 1 | Block 2 |
| Reg  Reg’  G0 | 1010  ~~1~~010  001 | 1010  1010  **1000** |
| Reg  Reg’  G1 | ~~1~~011  ~~10~~11  00 | 0010  0010  **0000** |
| Reg  Reg’  G2 | ~~10~~11  ~~101~~0  0 | 0010  1010  **0110** |
| Reg  Reg’  G3 | ~~101~~0  ~~1010~~ | 1100  1010  **0011** |

图2.5 CRC比特型迭代计算流程图

经过4次迭代计算，Block 1的数据被移出，不去关心，重点关心4次迭代对Block 2的影响，注意表中加粗数字的变化。明显G0的末四位（加粗数字，下同）取决于Block 1的首位，G1的末四位取决于Block 1迭代一次后的第二位（也就是Block 1的第一位和第二位决定），同理G2、G3的末四位也是由Block 1决定。由此可以考虑通过Block 1的值预先算出一张表，共有个数，其内存放着不同Block 1数字经过4次迭代后的G’，通过其值即可立马算出下一个块Block 2的校验值，省去了大量的移位操作和除法运算，极大地提高了运算效率。

|  |  |  |
| --- | --- | --- |
| Reg  Reg’  G | Block 1 Block 2  Block 2 Block 3  G’ | 使用Block 1查表得G’ |
| Reg  Reg’  G | Block 2’ Block 3  Block 3 Block 4  G’ | 使用Block 2’查表得G’ |

图2.6 CRC查表法迭代计算流程图

## 2.5本章小结

本章针对基于LS1024A处理器的海洋油气勘探采集传输系统在研发过程中所设计到的关键技术做了简要的介绍。包括嵌入式处理器平台、高速串行接口技术PCIe、内存直接访问技术DMA和循环冗余校验CRC，为本系统的后续研发工作提供了一定的理论基础。