**基于LVDS的双环路传输**

**系统概要设计**

**王益忠**

2017.4

# 概要说明

## 背景介绍

TSDR节点模块以Smartfusion2系列M2S050 FCS325为核心，将同步时钟转发给数模转换元进行同步采样，数模转换元和非声传感器采样数据通过RS485方式发送给数据TSDR节点，TSDR节点进行整理后通过高速串行LVDS总线将数据上传，上传链路采用双环路备份。

## 功能描述

TSDR节点具有3种数据接口：“支线数据总线”接口，“传感器总线”接口和“干线数据总线”接口，且各类型接口均为2个，分布于节点两端。水声信号的数据由2个“支线数据总线”接口进入TSDR节点处理后再由左右两端的“干线数据总线”均衡输出上传。传感器信号的数据由“传感器总线”接口进入TSDR节点处理后再从“干线数据总线”均衡输出上传。TSDR节点正常工作时通过2个“干线数据总线”接口均衡输出本地采集的数据，当链路出现故障时，各TSDR节点可自动选择正常链路将本地数据完整送入干端桥接模块。TSDR节点的优选采用多芯双绞线和总线拓扑结构的数据总线，每种接口均需具备双向通信功能。

TSDR节点具有同步授时功能，且时钟频率上位机可控制。同步机制所需信号通过“干线数据总线”传输完成。时钟信号输出有两种形式：其一、同步时钟线路与“支线数据总线”复用的授时方式给模数转换元提高同步时钟，其二、输出差分时钟信号供检测和系统扩展。两种形式时钟属同频同相关系。

TSDR节点需充分考虑系统会出现的故障问题和应对策略，当系统出现故障时需最大程度的保证系统的功能, 也能够提供充足的手段排查问题并准确定位故障所在的位置。比如链路断裂后的数据单向传输处理，时钟模块失效后的时钟源及同步处理，数据包丢失的重发机制，软件升级的稳定性和可靠性等等。

## 需求性能指标

* “干线数据总线”接口采用多芯UTP5E双绞线传输，数据线≤4根，数据传输带宽≥300Mbps，传输距离≥90米。
* “支线数据总线”接口数据采用多芯双绞线传输，数据线≤2根，数据传输带宽≥2.5Mbps，传输距离≥50米。
* “支线数据总线”接口采用主、从架构设计，主接口位于TSDR节点，其接口要求外围电路和软件协议简洁。主接口对从接口数量的增减具有自动识别功能，且传输带宽内无需软件干涉可自由增减从接口数量。挂载从接口数量≥8个。
* “传感器总线”采用RS485通讯接口并兼容RS422通讯接口，在RS485接口支持多点接入，数量≥64个，传输带宽≥1Mbps，传输距离≥100米。
* 各TSDR节点输出的时钟信号同频同相，误差应控制在ns级别，时钟传输距离≥50米且支持多点挂载。同步时钟频率可以程控，其范围为 256fs（1k≤fs≤12k取整数）。
* 进行微功耗设计，电流波动小,工作电压：6～15V，最大功耗≤3.5W。
* 模块尺寸：长度≤80mm，宽度≤16mm，高度≤9mm，
* 水下电子模块满足-10℃～50℃工作温度和-40℃～70℃的存储温度，适应轻蜡油浸泡，且可承受4.5Mpa最大静压力，以及舰船用电子模块的振动、颠震等要求。
* 电子模块具有抗静电、抗雷击能力。

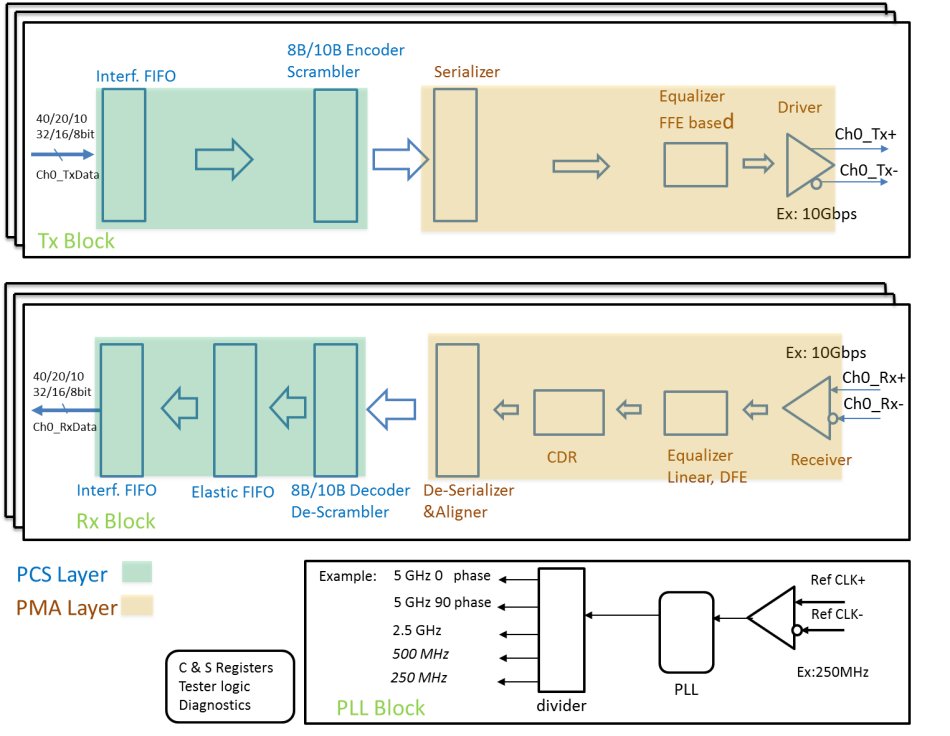
## 功耗统计

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  | 数字电源（mA） | | | | |
| 芯片型号 | 数量 | +3.3V | +1.8V | +1.5V | +1.2V | +1.0V |
| M2S050 | 1 | 29.34 | 41.09 |  | 416.38 |  |
| DS92LV1021A | 2 | 80 |  |  |  |  |
| DS92LV1212A | 2 | 115.76 |  |  |  |  |
| CLC001 | 2 | 169.70 |  |  |  |  |
| LMH0074 | 2 | 125.45 |  |  |  |  |
| MAX3362 | 2 | 4 |  |  |  |  |
| N25Q128A13ESE40F | 1 | 20 |  |  |  |  |
| MT46H32M16LFBF | 1 |  | 70 |  |  |  |
| 总计 |  | 537.44 | 111.09 |  | 416.38 |  |

# 系统硬件详细说明

## 原理介绍

节点模块之间采用LVDS进行数据传输。硬件结构如下图所示：



发送方向(Tx)信号的流向: FPGA软逻辑(fabric)送过来的并行信号，通过接口FIFO(Interface FIFO), 送给8B/10B编码器(8B/10B encoder)或扰码器(scambler)，以避免数据含有过长连0或者连1。之后送给串行器(Serializer)进行 并->串 转换。串行数据经过均衡器(equalizer)调理，由驱动器(driver)发送出去。

接收方向(Rx)信号的流向, 外部串行信号由线性均衡器(Linear Equalizer)或DFE (Decision Feedback Equalizer)结构均衡器调理，去除一部分确定性抖动(Deterministic jitter)。CDR从数据中恢复出采样时钟，经解串器变为对齐的并行信号。8B/10B解码器(8B/10B decoder)或解扰器(de-scambler)完成解码或者解扰。如果是异步时钟系统(plesio-synchronous system)，在用户FIFO之前还应该有弹性FIFO来补偿频差。

上图中的PCS层为标准的CMOS数字逻辑，可以使用硬逻辑（相关asic）实现，也可以使用FPGA软逻辑（比如Xilinx提供了相关的软核IP）。PMA层为数模混合（CML/CMOS）电路，主要实现串并转换、驱动均衡、解串等功能，一般采用ASIC实现。

桥接模块和节点模块之间使用光纤作为传输介质。硬件上利用smartFusion上的mgt模块以及光电转换模块实现了差分电信号和光信号之间的转换。

系统的大致结构框图如下所示：

系统结构框图

## 模块功能单元划分和业务描述

### 桥接模块

桥接模块功能框图

TSDR桥接模块以Freescale公司PowerQUICC III™系列高性能网络处理器MPC8569E为核心设计，CPU主频为1066MHz，支持4路千兆以太网通信（最多两路SGMII），支持最多8路百兆以太网通信。型号为XC7A35T的FPGA与处理器通信，负责整理节点上传的数据以及下发处理器命令。桥接模块通过QE接口外扩4个1000M以太网PHY，2个100M以太网PHY，实现与主控PC机网络通信与调试功能；桥接模块通过16bit总线接口外扩16MB Flash程序存储器，通过专用UART口扩展两个RS-232串行通信接口；桥接模块扩展1GB DDR SDRAM存储器。

TSDR桥接模块主要任务为接收节点数据和整理、发送并口数据，发送网口数据、响应串口命令等。桥接模块根据信号处理机RS-232命令，转发控制数据，下传给节点，实现节点命令控制与状态切换功能。

### 节点模块

模块的功能框图和信号流向如下所示：



图表 7 节点模块功能框图

TSDR节点包括1块Smartfusion 处理器 M2S050 FCS325以及串行器DS92LV1021A、解串器DS92LV1212A、驱动芯片CLC001，均衡器LMH0074和外围设备等。主要实现接收前端模块数据并对其进行整理和上传。模块通过485接收来自前端模块数据。TSDR节点之间的数据发送通过10位总线并行数据发出，通过两种方式进行传输：一种方式是串行器解析为差分信号，并通过驱动芯片提升长线传输的驱动能力；接受模块由均衡器接收传输过来的差分信号，发送至解串器恢复成并行数据，输入到FPGA中进行处理。另一种方式是通过FPGA的SERDES模块将差分信号直接通过驱动器发送，均衡接收的方式进行传输，中间通信的介质为光信号，带宽可达1Gbps以上。

节点主要由Smartfusion2系列处理器及外围设备组成，外围设备包括，SPI FLASH，DDR，晶振，SP809，JTAG，RS485时钟接收芯片和电源模块等。

## 对外接口

白头宽度小于16mm，高度小于9mm。  
TSDR节点接口说明，左侧插座共12根：数字电源4根，LVDS传输线8根。

|  |  |  |
| --- | --- | --- |
| 信号线序号 | 信号名称 | 说明 |
| 1，2 | V+IN (数字正电源输入) |  |
| 3，4 | LVDS传输线1发正发负 | 双绞 |
| 5，6 | LVDS传输线2发正发负 | 双绞 |
| 7，8 | GND (数字地输入) | 双绞 |
| 9，10 | LVDS传输线1收正收负 | 双绞 |
| 11，12 | LVDS传输线2收正收负 | 双绞 |

图表 23 TSDR节点左侧接口说明

TSDR节点接口说明，右侧插座共12根：

|  |  |  |
| --- | --- | --- |
| 信号线序号 | 信号名称 | 说明 |
| 1，2 | 模数转换元时钟信号正负 | 双绞 |
| 3，4 | 水声数据正负 | 双绞 |
| 5，6 | 非声传感器通道数据 | 双绞 |
| 7，8 | NC |  |
| 9，10 | NC |  |
| 11，12 | 非声传感器通道数据 | 双绞 |

图表 24 TSDR节点右侧接口说明

# 时钟传输

本系统需要传输同步时钟，用于AD采样控制。时钟包含于数据传输通路中，如下图所示：



发送端每个参考时钟沿发送一个Byte数据（8bits），其中高2位代表AD的同步时钟，低6位为链路传输的数据。对于时钟位Byte[7:6]，当接收端接收到Byte[7:6]=2’b00时，认为是MCLK（AD同步时钟）低电平。当接收端收到Byte[7:6]=2’b11时，认为MCLK为高电平。而当接收端收到Byte[7:6]=2’b01或者2’b10时，则认为MCLK在传输中发生错误。

低6位代表数据位，在该6bits数据链路上进行协议封装，实现数据校验、帧检测、数据包转发等功能。

## 延迟校正

### 线路延迟测量方式

延迟校正功能主要实现桥接模块到各个节点模块的物理线路延迟值测量与收集，并且配置各个节点模块的延迟值，实现系统各个节点模块的同步采样。双环路传输系统简图如下：



系统桥接模块以及各个节点模块都可以双向收发数据包，整个系统构成了双环路。因此对于任意节点模块，从桥接模块发出的数据包都有2条往返路径，如下图所示：



对于一次线路延迟测量，首先桥接模块发送一个延时测量数据包，同时开启计数器（FPGA逻辑实现），数据包中包含目标节点模块地址，用于目标节点进行包识别；目标节点模块收到延时测量数据包后立刻反送一个延时测量回应数据包，其中也包含目标节点模块地址，用于桥接模块进行识别。桥接模块收到相应延时测量回应数据包后，关闭相应计数器。若此时桥接计数clk为K Hz，计数值为n，在假设线路延时是对称的情况下，可以得到目标节点的线路延时T=n/2K；

### 线路延迟值配置

当桥接模块得到所有节点的延时值后，将发送延迟值配置命令包，其包含每个节点的地址以及对应的延迟值。每个节点收到该命令包后根据地址从中解析出其对应所需的2个延迟值，从而实现桥接模块对所有节点的线路延迟值配置。

### 延迟实现方式

节点实现延迟采样的原理主要是对MCLK进行延迟输出。实现方式如下图所示：



当节点收到同步时钟时，对同步时钟的上升沿进行计数，当计数值和延迟值相等时，将MCLK连接到同步时钟上输出至AD，控制采样，即控制MCLK延迟若干个时钟。关于延迟值m，对于任意一个节点，若其延时值为t，MCLK=a Hz，则可得延迟值m=round(t\*a)。

### 延迟值切换

当线路发生故障导致环路断路时，将导致同步时钟无法通过单环路无法传达到某些节点，此时需进行时钟切换，同时由于某些节点的同步时钟传输线路发生变化，需要对其延迟值进行切换。一般可以将断路情形分为2类，如下图所示为其中一种情况：



如上图所示，若开始时，系统采用环路2的同步时钟作为同步时钟源，则每个节点都采用相应的环路2的延迟值对环路2同步时钟进行延迟控制AD同步采样。当环路2中4,5节点发生断路时，桥接模块只能收到节点1,2,3,4的数据包。通过对所有数据包的源信息进行提取，可以判断环路2发生断路且4,5节点之间必定存在传输问题。此时环路1仍正常工作。因此需要将同步时钟源切换至环路1同步时钟，同时通知各个节点采用环路1对应的延迟值对环路1同步时钟进行延迟控制AD同步采样。

另一种特殊情形如下图所示：



如上图所示，当4,5节点之间的链路都发生断路时，环路2只能收到1,2,3,4节点的数据包，而环路1只能收到5,6……N-2,N-1,N节点的数据包。桥接模块通过对所有数据包的源地址进行提取对比可以得知此时环路1,2都发生断路，且故障位置在4,5节点之间。对于这种情形，节点1,2,3,4需采用环路1的同步时钟作为同步时钟源，并置其相应延迟值，而节点5,6……N-2,N-1,N需采用环路2的同步时钟作为同步时钟源，并置其相应延迟值。环路1,2将收到的数据包进行汇总可以得到每个节点的数据包。

## 同步时钟传输

### 时钟备份

时钟备份是为了保证在信道通讯质量下降时仍能够使系统维持一定的正常工作时间。

如下图所示：



若系统一开始采用环路2的同步时钟源作为同步时钟，那么当环路在节点4,5发传输错误时，节点1,2,3,4将无法获取同步时钟，此时按照之前的讨论，需进行同步时钟源以及对应延迟值切换。但由于1,2,3,4节点没有参考的同步时钟，如果节点内部无法产生一个同频的时钟去控制AD采样，AD芯片将无法扇出数据，这样桥接模块也无法得到1,2,3,4的数据包。

因此，对于每个节点，都应该具备时钟备份的功能。当同步时钟正常接收时，每个节点按照各个系统的主时钟维持一个计数值。如图所示：



若MCLK为M Hz，节点FPGA工作时钟为 K Hz，备份时钟计数值n取round(K/M),该计数值可通过FPGA进行测量获取。每次系统进行时钟频率切换时，每个节点都需要重新对备份时钟计数值进行估计。

### 时钟传输错误判定

当链路信道质量下降时，节点接收到的同步时钟可能出现错误。为了等待链路恢复，保证系统持续正常工作， 需要利用备份的时钟计数值进行同步时钟的本地恢复。

为了检测环路同步时钟是否正常，需要实时对环路时钟进行监测，对于节点FPGA来说就是维持一个同步时钟计数值n’，将该值和备份时钟计数值的一半(n/2)比较可以判断同步时钟传输是否正常。

当时钟传输错误时，可能有多种情形，如下图所示是其中一种情况：



同步时钟可能长时间维持0或者1，或者其上升沿或下降沿提早到来。假设同步时钟传输的最大误差为a，那么正常的沿应该落在同步时钟计数值n’=(1-a)n/2~(1+a)n/2的范围内。当n’<(1-a)n/2时，若同步时钟沿到来，判定同步时钟传输错误；当n’>(1+a)n/2时，若同步时钟沿仍未到来，判定同步时钟传输错误。

下图展示了另外一种时钟错误情形：



同步时钟采用数据通路Byte[7:6]来传输，当信道质量下降时，Byte[7:6]可能出现2’b01或者2’b10。如上图所示，当同步时钟计数值n’<(1+a)n/2时，只要出现这种情况，就判定同步时钟传输错误。

### 备份时钟启用

当节点判定时钟传输发生错误时，需要利用备份的时钟计数值进行同步时钟的本地恢复，并且应将时钟传输错误造成的延迟误差降到最低。

从逻辑上来说，时钟恢复可以分为2大类，如下图所示是其中一种情况：



若当判定同步时钟传输发生错误时，同步时钟计数值n’<(1-a)n/2，备份时钟源恢复逻辑开始工作，首先备份时钟源获取同步时钟发生错误时上一个状态值（图中为1），之后同步时钟计数值继续计数，当计数至n’=n/2时清零同步时钟计数值，同时启用备份时钟计数器n1’进行计数，当n1’=n/2时，清零n1’，同时对备份时钟源进行01翻转。（n为内部备份时钟计数值）  
 当同步时钟被判定发生传输错误时，需要将AD采样控制时钟在合适的时机切换至备份时钟源。

该逻辑采用状态机进行实现时需要考虑AD时钟在同步时钟和备份时钟源之间切换时产生的时序问题。因此需要对同步时钟MCLK延迟一个时钟，判错逻辑在同步时钟MCLK上进行，而AD采样控制时钟在切换同步时钟MCLK’和备份时钟源之间进行。

下图所示为另外一种同步时钟传输出错的情况：



当同步时钟计数值n’=(1+a)n/2时，同步时钟边沿仍旧没有出现，判定时钟传输出错。此时应立刻将备份时钟源切换至期望值（上图中为0），同时启用备份时钟计数逻辑，当备份时钟计数值n1’=n/2时备份时钟源进行01翻转。

为保持模块一致性，此时AD采样控制时钟仍旧在切换同步时钟MCLK’和备份时钟源之间进行切换。

### 同步时钟恢复

当同步时钟计数检测逻辑检测到稳定且正确的同步时钟（即连续多个稳定同步时钟）时，应将AD采用控制时钟从备份时钟源切换至同步时钟。



如上图所示，当同步时钟计数逻辑判断到连续P个正常同步时钟边沿时，判定同步时钟传输已经恢复正常。此时应该立刻将AD采样控制时钟从备份时钟源切换至系统同步时钟。由于采用备份时钟源时每个节点的晶振主频之间存在一定误差，导致同步时钟恢复时，每个节点的备份时钟源和系统同步时钟之间可能存在一定的相位差，最坏情况可能会导致一个采样点的丢失。下图展示了可能出现的最差情况：



## 链路检测

LVDS的数据接口为同步传输接口，且其无单独的时钟传输线。数据接收的时钟需要从串行链路中恢复出来。当链路发生断路时，对于接收端来说，串行时钟将无法恢复。这样将导致FPGA内部的模块无法正常工作（无工作时钟）。如下图所示：



上图所示的情况，若系统工作采用环路2，由于节点1,2,3,4都无法从链路中恢复时钟，则环路2桥接接收端将接收不到任何数据。对于任何两个节点之间，只要发生时钟锁死的情况，都会导致桥接模块收不到任何数据包。因此需要增加链路检测功能来进行故障定位。

故障检测模式的一般流程，首先桥接模块发送一个故障检测数据包，节点收到该数据包后进入故障检测模式，选择同端路的发送端进行回包（即若左端口收包，就用左端口的进行回包）。这样桥接模块可以通过接收到的包的数量判定出链路发生断路的位置，其工作示意图如下图所示：



上图中若第n-1个节点和第n个节点之间发生断路，则桥接模块将无法收到n-1节点之后所有节点的回包，桥接模块通过统计接收到的回包数量以及回包中的地址戳，可以定位链路发生故障的位置。

采用这种方式实现链路检测时需要考虑FPGA内部工作模块时钟切换的问题。

对于每个节点，接收以及发送逻辑的工作时钟如图所示：



当环路处于正常工作模式时，对于任何一个节点，可以从左接收端恢复出工作时钟并把该工作时钟用于右发送端，或者从右接收端恢复出工作时钟并把该工作时钟用于左发送端。

当系统工作于链路检测模式时，由于数据包需要反送，所以从左接收端恢复出的工作时钟将切换至左发送端用于回包发送；同理，从右接收端恢复出的工作时钟将切换至左发送端用于回包发送。切换的简图如上图所示。

# 数据传输链路

### 帧格式

由于需要传输同步时钟，接收端经10b/8b转换后，损失了最高两位（Byte[7:6]代表同步时钟），因此参考接收端用户时钟，数据的宽度只有6bits，如下图所示：



若完全按照6bits作为基本Byte对数据进行校验传输，有非常大的缺点。一则，6bits的CRC校验没有一个固定的标准可循；二则，6bits的基本数据单元与一般上位机通用的byte不符，会给后续软件处理带来许多麻烦，因此需要对输入数据进行适当的6bits转8bits处理。

首先需要对SAV监听帧头进行定义，标准的网络监听帧头如下所示：



标准的网络帧头一般由4Bytes组成，对于网络底层来说，数据串行收发，即当接收端收到32’b10101010\_10101010\_10101010\_10101011时，认为一帧数据开始，网络数据链路层将开始数据包接收。但是对于32bits，无法转化为整数个6bits进行传送，因此需要对帧头进行改变。如上图所示，帧头采用24bits，共4个6bis，当收到24’b10101010\_101010\_101010\_101011时认为一帧数据包开始。

之后需要把输入的6bits位宽数据转换成8bit位宽数据进行后续帧头获取、帧头CRC校验、数据获取以及数据CRC检验。6bits转8btis过程可采用reg缓存的方式，示意图如下：



根据1.2以及1.3的讨论，协议封装在原先一般的协议基础上还需要增加3个特殊功能，即延迟测量、时钟选择以及相应的延迟配置、同步时钟频率选择。因此在原先长线传输的基础上还需要增加功能选择判断位。数据包完整的格式如下所示：



数据包的检测通过SAV来识别，SAV的固定构成为{6’h2A,6’h2A,6’h2A,6’h2B}，大小为4\*6bits，SAV不需要进行6bits转8bits操作，当检测到SAV后立刻启动6bits转8bits模块对输入6bits位宽数据进行转换，得到数据包主体。

数据包主体由头部和数据两部分组成。头部大小为12Bytes，包括2Bytes目的地址，2Bytes源地址，2Bytes长度信息，1Byte的方向信息，1Byte的端口信息，1Byte的功能选择信息，2Bytes的保留字节以及1Byte的头部CRC校验码。

IP内容主要区分广播包与IP指定包。广播包统一格式为255.255，IP指定包第1个字节固定为192，第二个字节确定节点地址。

IP信息表参照如下表所示：

|  |  |
| --- | --- |
| 具体意义 | IP(2Bytes) |
| 广播包 | 255.255 |
| IP指定包 | 192.XXX(0 - ?) |

Length中携带了后面Data部分的整体数据长度（不包括Data后面的CRC部分），由于目前已有最大数据包大小为中频节点1542Bytes，故Length的长度取2Bytes。

Direction内容确定硬件上双环路的数据包传输方向，内容解析如下表：

|  |  |
| --- | --- |
| 具体意义 | Direction(1Byte) |
| 左向 | 8’h01 |
| 右向 | 8’h02 |
| 双向 | 8’h03 |

Port中的内容用于软件内容的判断逻辑。

功能选择信息主要用来指示该包数据用于实现的功能，默认设置为数据包，其具体值与相应的功能对应如下表：

|  |  |
| --- | --- |
| 具体意义 | Function Select(1Byte) |
| 数据包 | 8’h00 |
| 延迟测量功能 | 8’h01 |
| 时钟切换 | 8’h03 |
| 时钟频率切换 | 8h04 |

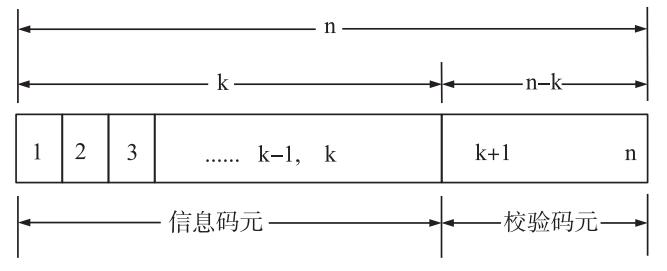
保留字节2Bytes，均默认置1。当功能选择为时钟频率切换时，该保留字节代表将要切换的频率。

### CRC校验

对于包头部分，由单独的CRC校验码对包头进行单独的CRC校验。包头CRC校验采用CRC-8校验方式，因此包头CRC长度为1Byte。

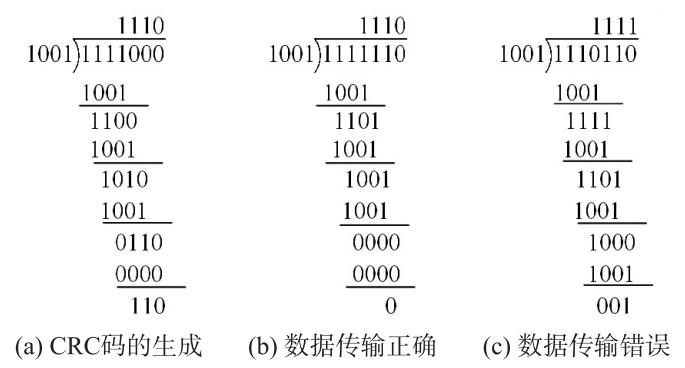
数据部分则是网络包的主要内容，分为控制命令和实际采集数据，其长度不确定，但长度信息可以通过包头的Length值得到，同样，数据部分也有单独的CRC校验码用来进行数据部分的CRC单独校验。数据部分的CRC校验采用CRC-32校验方式，因此数据部分的CRC长度为4Bytes。

关于CRC校验，CRC分组码码组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：



实现CRC校验拓展，需要生成一个二进制多项式作POLY，其与拓展位的关系为（拓展位的位数=生成多项式的位数 - 1）。而结合整体数据包的结构形式，其又符合另外一个公式：

CRC码的计算方式类似于模2除法，从高位部分开始，数据位与选择的多项式进行由高位到低位计算，每次计算所得到的余数过程通过按位异或的结果得到，之后作移位计算，计算结果可参考如下案例：



在FPGA完成该功能前，需要考虑以下两个公式：

最佳的考虑方式：保证带宽一定的使用率情况下，减少CRC部分所消耗的时间，从而确保传输不受到影响。

数据包数据部分的校验由软件实现，使用CRC-32校验方式，因此数据部分的CRC长度为4bytes。其实现原理和CRC-8类似。

### 数据包分类以及转发

数据包类型主要分为2种，一种是采样数据包，另一种是功能包。功能包主要实现延迟测量、时钟切换、时钟频率切换等功能。FPGA实现对数据包头部的解析以及CRC检验。根据解析结果，FPGA将数据包发送至相应模块进行后续处理：

* 若数据包为采样数据包，FPGA将该数据包通过AHB发送模块发送至M3，由软件去实现CRC32校验，校验成功后再由M3通过AHB总线发送至FPGA，loop至下一个节点。
* 若数据包为延迟测量包，对于延迟测量包，只有数据包头部，数据长度为0，FPGA若判定该数据包为延迟测量包，只需继续判定该数据包目标地址是否和本地地址相同。若相同，则FPGA立刻发送一个相同内容的回包；若不同，则loop至下一个节点。
* 若数据包为时钟频率切换数据包，对于时钟频率切换数据包其只有数据包头部，数据长度为0，FPGA若判定该数据包为时钟频率切换数据包，就可以直接获取相应频率切换值进行频率切换。同时将该数据包loop至下一个节点。
* 若数据包为时钟切换数据包，对于时钟切换数据包，其数据部分包含每个节点地址以及相应的延迟值。FPGA若判定该数据包为时钟切换数据包，应通过AHB发送模块将该数据包发送至M3，由软件去做数据部分的CRC32校验并获取相应的延迟值。CRC32校验成功之后再由M3通过AHB总线将数据发送至FPGA，loop至下一个节点。

根据以上分析，FPGA总体设计结构主要包括3部分，分别是LVDS数据收发模块，内部AHB总线传输模块以及相应的RAM缓存模块，其基本结构可参考下图：



上图中用不同颜色箭头标出了各种类型数据包的具体流向。其中时钟切换频率切换数据包以及延迟测量数据包不需要软件参与解析，整个解析以及loop由FPGA独立完成。采样数据包以及时钟切换数据包由于含有数据部分，因此需要由软件进行CRC32校验，同时对于时钟切换数据包还需要软件根据本地地址获取相应的延迟值以及时钟源。