|  |
| --- |
| 浙江大学嵌入式系统研究中心 |
| TSDR节点  FPGA详细设计文档 |
| V1.0.0 |
|  |
| **王益民** |
| **2017-8-3** |

|  |
| --- |
| **Confidential** |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 编号 | 章节名称 | 修订内容简述 | 修订日期 | 修订前  版本号 | 修订后  版本号 | 修订人 | 批准人 |
| 1 | 所有 | 创建 | 2017-3-6 | 1.0.0 | 1.0.0 | **王益民** |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |

目录

[1、目的 4](#_Toc327014365)

[2、范围 4](#_Toc327014366)

[3、芯片综述 4](#_Toc327014367)

[4、主要特性 4](#_Toc327014368)

[5、芯片详述 5](#_Toc327014369)

[5.1、总体框图 6](#_Toc327014370)

[5.2、接口信号 7](#_Toc327014371)

[5.3、模块特性 8](#_Toc327014372)

[5.3.1 顶层控制 8](#_Toc327014373)

[5.3.2 SPI\_clock模块 8](#_Toc327014374)

[5.3.3 SPI\_interface模块 9](#_Toc327014375)

[6、器件说明 11](#_Toc327014376)

[6.1、器件型号 11](#_Toc327014377)

[6.2、封装型号 11](#_Toc327014378)

[6.3、芯片描述 11](#_Toc327014379)

[6.4、电气特性 12](#_Toc327014380)

[7、管脚说明 14](#_Toc327014381)

[7.1、管脚分布 14](#_Toc327014382)

[7.2、管脚说明 15](#_Toc327014383)

[8、地址映射 16](#_Toc327014384)

[8.1、逻辑内部寄存器空间分布 16](#_Toc327014385)

[8.2、逻辑外部RAM空间分布 16](#_Toc327014386)

[8.3、逻辑内部RAM空间分布 16](#_Toc327014387)

[9、寄存器描述 17](#_Toc327014388)

[9.1、寄存器映射 17](#_Toc327014389)

[9.2、寄存器描述 17](#_Toc327014390)

[9.3、寄存器块1 17](#_Toc327014391)

[9.4、寄存器块2 17](#_Toc327014392)

# 1、目的

本文档详细描述了TSDR节点的FPGA的逻辑功能，接口和操作说明等。

# 2、范围

本文档仅适用TSDR节点模块FPGA逻辑设计。

# 3、芯片综述

TSDR节点以Microsemi公司Smartfusion2高性能FPGA为核心设计,实现接收数模转换模块发送的数据，并将数据发送到桥接模块，以及节点之间的数据的转发。

**4、主要特性**

TSDR节点模块以Smartfusion2系列M2S060T 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。

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。

5.1、总体框图



TSDR节点的FPGA主要实现接收数模转换模块发送的数据，并将数据发送到桥接模块，TSDR节点间的时钟同步与通信，节点与桥接模块之间的通信功能。节点的FPGA模块可分为3个部分，LVDS收发模块，时钟控制模块和AD采集数据接收模块。

5.2、LVDS模块逻辑

**帧头监听**



**SAV检测**



**数据接收**

数据缓存开3个RAM，RAM1缓存帧头数据，对帧头进行CRC校验并获取帧头信息，RAM2缓存目的地址为本地的数据，将数据发向M3；RAM3缓存目的地址非本地的数据，转发给下一节点。

数据长度为0的包：延迟测量命令包、时钟切换命令包。

数据长度最大为1728Bytes（96个数模转换元）。

**帧头信息提取**

接收端检测到SAV后即开始缓存数据，同时利用状态机跳到获取帧头信息的状态，根据目的地址是否本地地址将数据分别缓存在2个ram中，根据数据长度的大小选择缓存数据的长度。发送给M3的数据按照1ms发一个中断通过AHB总线读取。转发的数据则存放在ram中按照优先级发出。

**数据帧格式**

接收与发送的10位并行数据通过8B/10B转换成为8位数据，参照标准的网络帧头定义帧头为24’b10101010\_10101010\_10101011，即8’aa\_8’aa \_8’ab。

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{8’AA,8’hAA，8’hAB}，大小为3\*8bits，当检测到SAV后，通过FIFO或者RAM缓存得到一个包的数据，再对包头目的地址进行判断，若目的地址是本地地址则将数据缓存到用以存放本地数据的ram中，若目的地址不是本地地址，则将数据缓存在存放转发数据的ram中。

数据包主体由头部和数据两部分组成。头部大小为8Bytes，包括2Bytes目的地址，2Bytes源地址，2Bytes长度信息， 1Byte的端口与功能选择信息，1Byte的保留字节以及1Byte的头部CRC校验码。数据长度为24\*16\*6=288Bytes（16个数模转换元，每包6个采样点），DATA CRC长度为4Bytes。

头部地址用来指示数据包的源地址以及目的地址。主要区分广播包、地址指定包。广播包主要用于桥接模块对节点模块进行广播。当节点模块收到广播包后，在判定数据传输没有出错后会立即将该数据包从另一端口loop出去。地址指定包主要用于节点向桥接模块发送数据包以及桥接模块向指定节点模块发送命令包。

地址信息表参照如下表所示：

|  |  |
| --- | --- |
| 具体意义 | addr(2Bytes) |
| 广播包 | 0xFF\_FF |
| 地址指定包 | 除0xFF\_FF以外的地址 |

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），若数模转换元数量为96个，则数据长度最大为24\*96\*6=1728Bytes，加上数据头部和CRC校验，则数据包长度为1741Bytes，故Length的长度取2Bytes。

Port & Func中的内容用于FPGA进行发包端口的判定以及数据包类型判定。

Port占1位，Func占5位，高2位保留，用于功能扩展。因此该字节的格式为



Port以及Func字段实现具体意义如下表所示

|  |  |
| --- | --- |
| 具体意义 | Port(1 bit) |
| 左端口 | 1’b0 |
| 右端口 | 1’b1 |

|  |  |
| --- | --- |
| 具体意义 | Func(5 bit) |
| 延时测量命令包 | 5’b00000 |
| 延时配置命令包 | 5’b00001 |
| 开始 | 5’b00010 |
| 停止 | 5’b00011 |
| 干线数据包 | 5’b00100 |
| 支线数据包 | 5’b00101 |
| 非声传感器数据 | 5’b00110 |

**数据包类型**

LVDS模块逻辑主要实现数据的接收与发送，数据包的类型分为四种（每个数据帧有一特定字节表示该数据包类型）。

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。收进来数据存放在本地数据ram的特定地址上，经M3 CRC校验后的数据存放在转发的ram的特定地址上，Port1 收到的数据由Port1发出去，Port2收到的数据由Port2发出去。



若数据包为延迟测量包，延迟测量包由桥接发出并统计，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，再判断该延迟测量包目的地址是否本地地址，若是则将源地址填充为本地地址，目的地址改为桥接地址，并立刻从同向端口回包；若不同，则loop至下一个节点。桥接统计每个节点的延迟值后给每个节点发延迟配置命令包，数据内容是节点的延迟值。节点同样判断该延迟配置包地址是否与本地地址匹配，若是，则获取该延迟值，并做延迟逻辑（延迟逻辑的实现通过节点的同步时钟计数实现，每个节点按照延迟值的大小先后给将同步时钟发给AD采样的mclk），否则转发给下一个节点。



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。若数据包为链路检测包，链路检测包由桥接发出，当节点出现丢包等故障时用来定位链路出问题的具体位置。节点收到链路检测包后从原端口回包，桥接统计节点的回包信息从而确定故障位置。



节点工作流程为：上电 -> 延迟测量 –> 延迟值配置 -> 数模转换元同步时钟分频输出 –> 接收数模转换元数据并上传桥接模块 。不同类型的数据（命令和采样数据）缓存放在ram里面不同的地址。

同步包：空闲状态节点发01数据同步包以确保链路维持。

**数据优先级**

节点发包优先级：本地延迟测量包>其他节点延迟测量包>本地命令包>其他节点命令包>本地数据包>其他节点数据包。

**8bits/10bits编码**

通过组合逻辑的方法实现，可参考源码资源。

5.3、时钟控制模块

**同步时钟传输**



每个节点将解串器解出来的时钟RCLK赋给该节点同一链路的发送端时钟TCLK，构成级联关系。根据延迟测量包对每个节点延迟值的计算，对节点的RCLK进行延迟校准并分频产生MCLK，作为AD采样的同步时钟。



**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。

**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：



在实验室条件下利用ZYNQ-7000平台，对LVDS传输同步时钟方案进行在板验证。测试时，FPGA发送模块工作时钟为20MHz，传输同步时钟频率为1Mhz。接收端能够从数据链路中恢复出同步时钟并进行内部备份。当LVDS链路发生传输错误时，FPGA能够检测到同步时钟传输出错并将AD控制时钟切换至内部备份时钟。当链路恢复时，FPGA能够将AD控制时钟切换为外部同步时钟。

### 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

5.4、RS485模块逻辑

RS485 底层传输接收端已写好并验证

RS485模块根据自定义RS485协议判断接收到的数据包并进行校验，将数据缓存进FIFO中等待传输；RS485非声数据存储到FIFO中

RS485模块发送端根据 lvds模块接收到的命令发送数据

5.5、代码更新

Smartfusion2处理器代码分为FPGA与M3两个部分，FPGA部分调试更新可通过JTAG烧写，M3部分调试更新通过处理器扩出来的MII接口网络更新。另外节点还支持在线更新功能，更新数据包由桥接下发通过FPGA判断内容后完成FPGA或者M3的更新。

# 6、器件说明

## 6.1、器件型号

FPGA芯片选用的是Microsemi公司的Smartfusion2系列处理器M2S060T FCS325I。

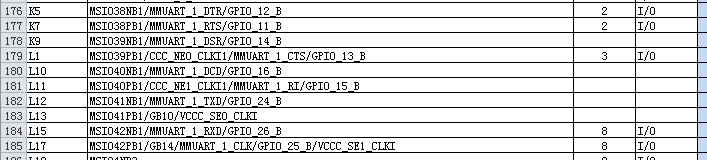
## 6.2、封装型号

芯片的封装是FCS325I。

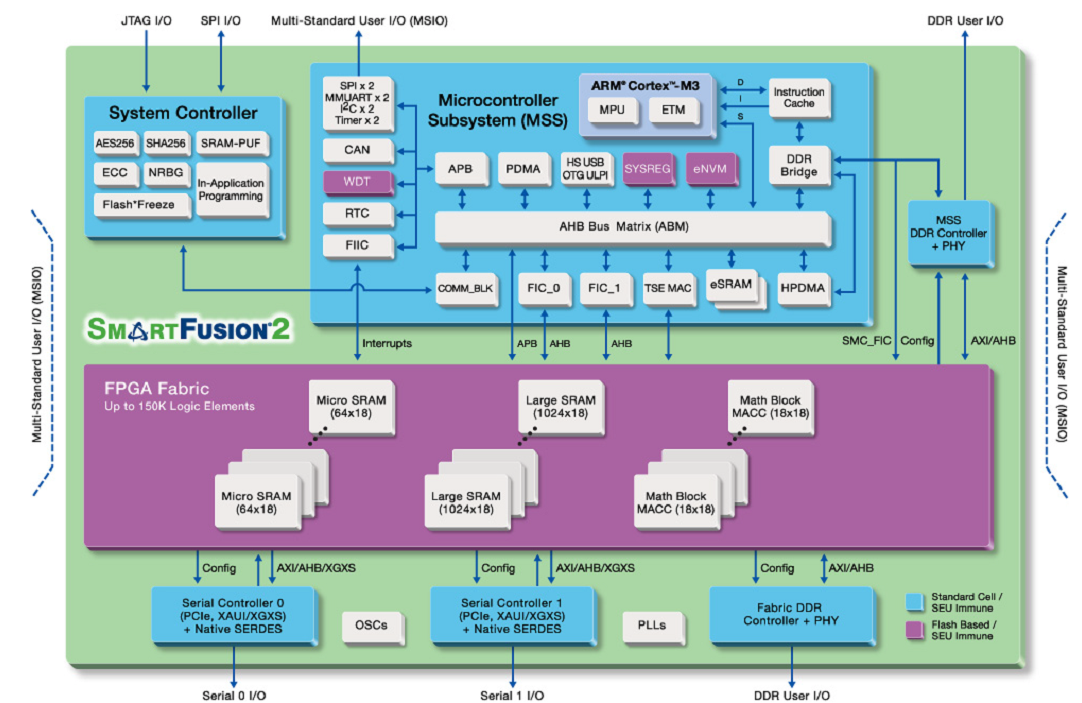
## 6.3、芯片描述

SmartFusion2系列是Microsemi公司的高性能SoC处理器，是一款专为满足当今嵌入式应用、低功耗、高可靠性要求的处理器。M2S050-FCS325的主要外围接口包括：

* 2个I2C接口；
* 1个DDR2/3(LPDDR兼容)控制器；
* 1个调试JTAG口。
* 2个串行外围设备接口（一个SPI Flash，一个SPI外设），
* 1个通用异步收发口（UART）。(FCS\_325封装只找到一个)



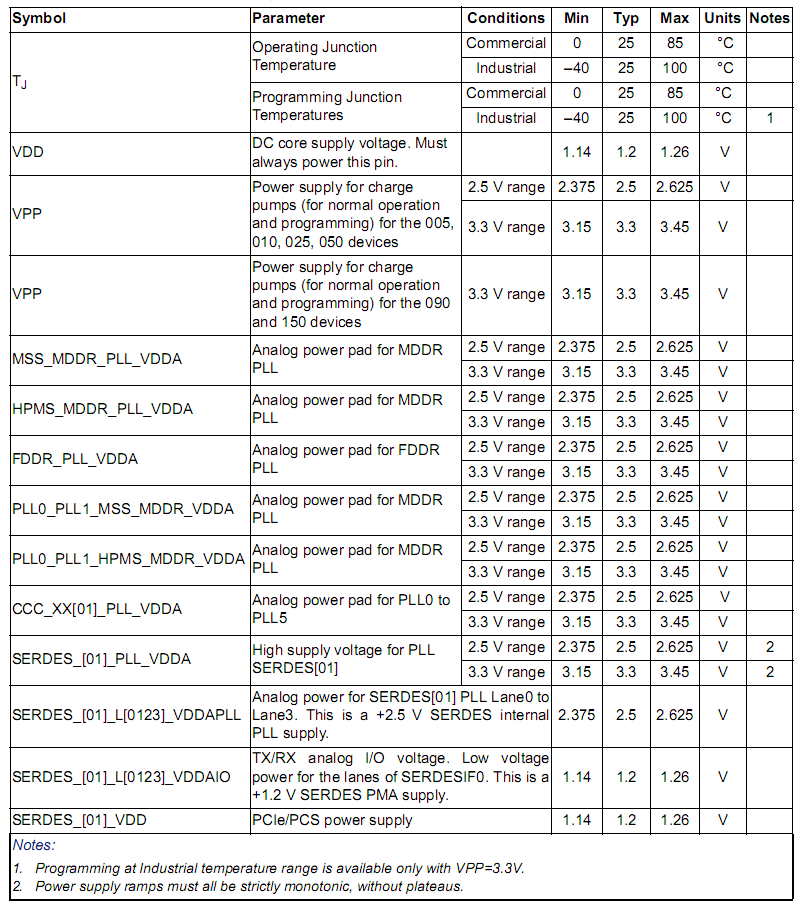
SmartFusion2系列的结构图如图表4所示：

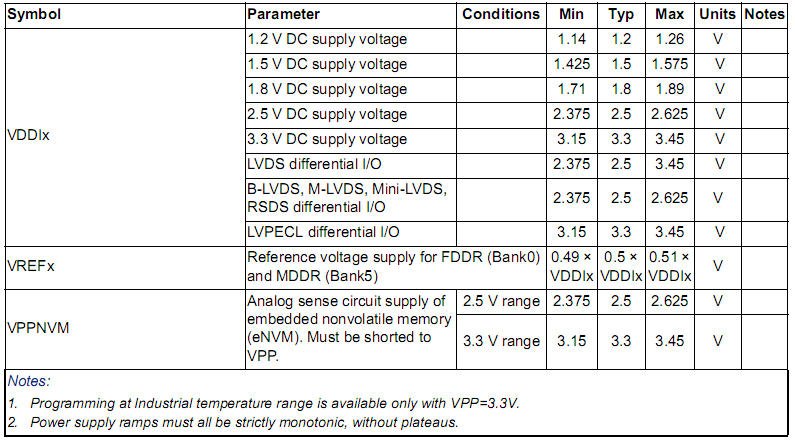


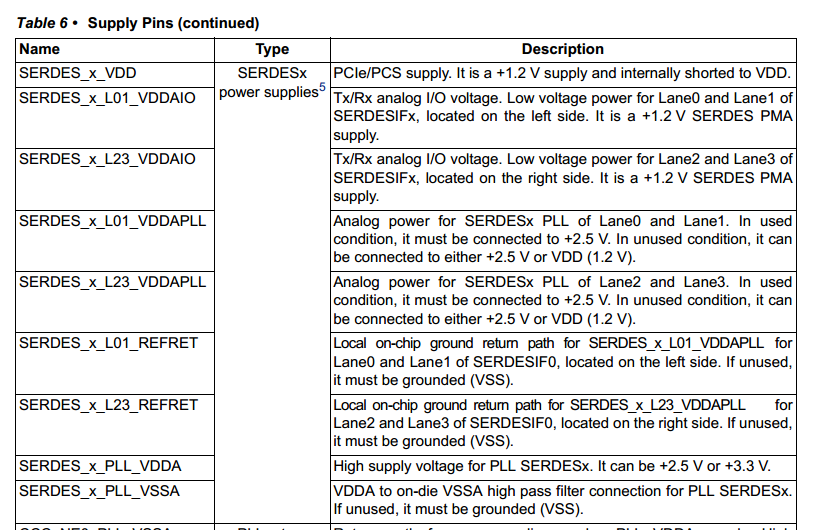
图表4 SmartFusion2结构图

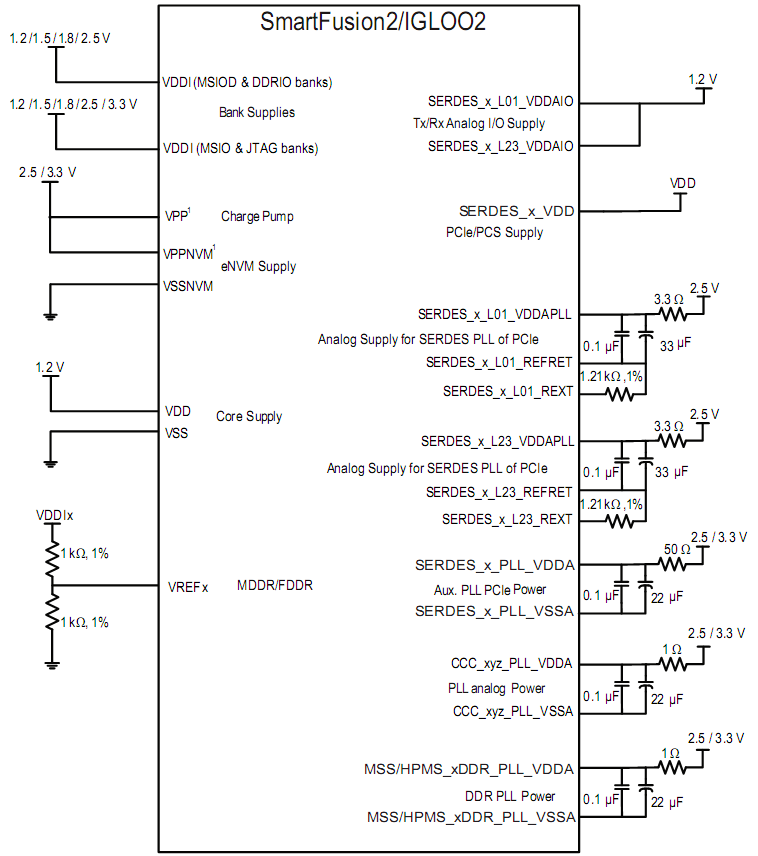
* High Temperature Operation to 125°C
* IEEE 1149.1 (JTAG) Compliant and At-Speed BIST Test Mode
* Clock Recovery from PLL Lock to Random Data Patterns
* Ensured Transition Every Data Transfer Cycle
* Chipset (Tx + Rx) Power Consumption < 60 mW (Typ) @ 80 MHz
* 800 Mbps Serial Bus LVDS Data Rate (at 80 MHz Clock)
* **10-bit Parallel Interface for 1 Byte Data Plus 2 Control Bits**
* **Synchronization Mode and LOCK Indicator**
* **Programmable Edge Trigger on Clock**
* **High Impedance on Receiver Inputs When Power is Off**
* **Bus LVDS Serial Output Rated for 27**Ω **Load**
* **Small 49-Lead NFBGA Package**
* **DDR**

## 6.4、电气特性





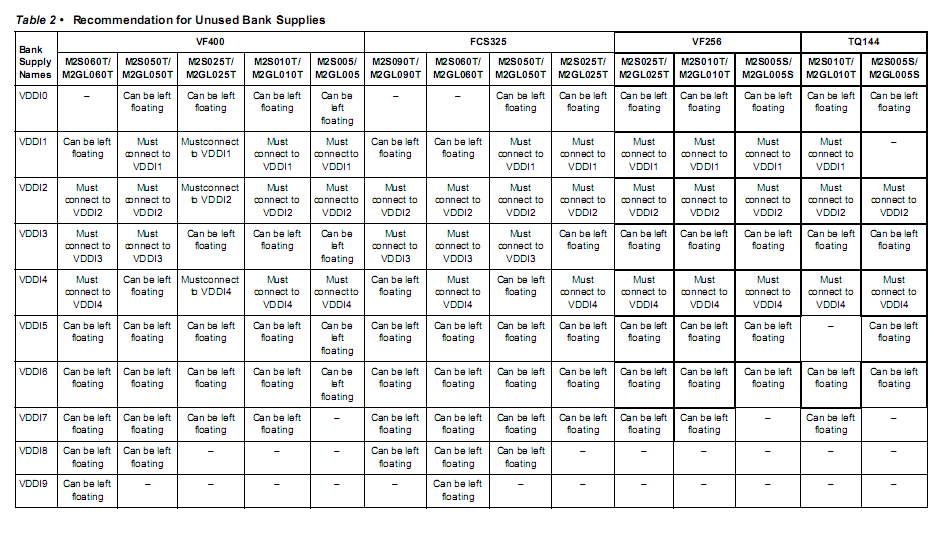


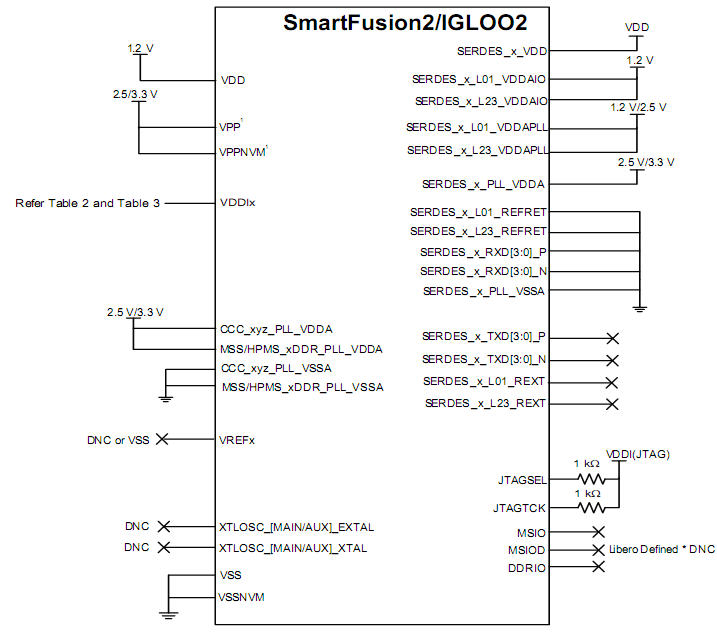


图表 11 处理器电压分配

电源上电没有严格的顺序要求，保证同时上电（预留PG控制Enable电阻）。当电源上电时，系统Power\_on reset有四种选择：50us，1ms，10ms，100ms。每种选择代表VDD和VPP上电的最大上升时间。

* Unused Bank Supplies





* 处理器时钟设计

SmartFusion2外部时钟可以用3.3V LVTTL电平，频率50MHz，精度要求为+/- 50ppm。

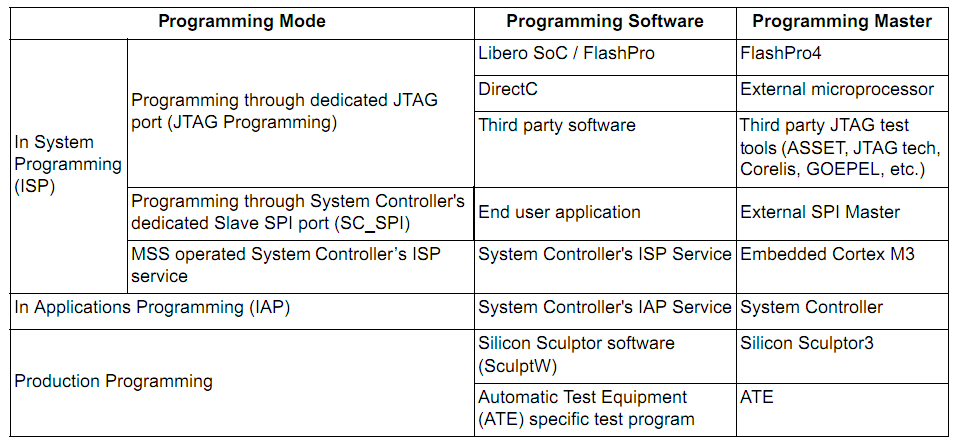
* UART接口设计

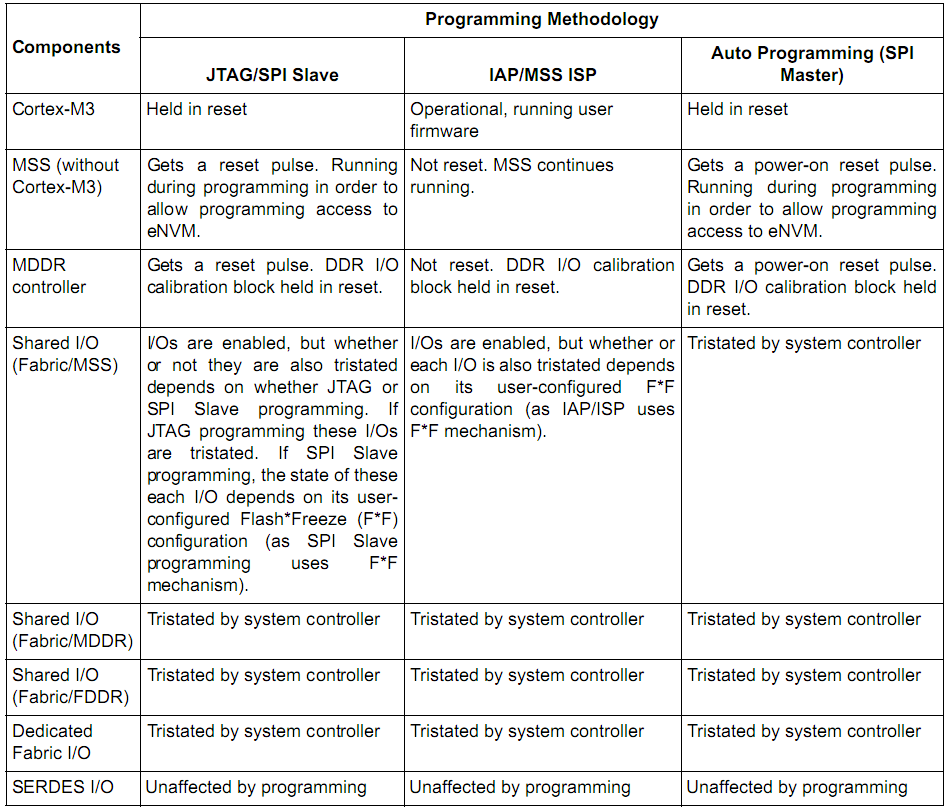
SmartFusion2有2个UART端口，扩展一个通过20Zif插件引出一个RS232串口，用于调试。

* JTAG接口设计

SmartFusion2有1个JTAG端口，通过JTAGSEL来选择连接ARM还是FPGA,扩展一个通过20Zif插件引出一个JTAG口，用于下载程序和调试。

SmartFusion2的programming mode有三种：IAP，ISP，Production Programming





# 7、管脚说明

## 7.1、管脚分布

**LVDS收发模块**：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 信号名称 | 电平 | I/O | 引脚 | 备注 |
| Dout1[9:0] | LVCMOS33 | O | T5、R5、W1、V1、W2、Y2、U4、T1、A2、Y1 | 1路10位并行发送数据 |
| TCLK1\_R/FN | LVCMOS33 | O | N2 | 1路发送时钟边沿有效选择 |
| TCLK1 | LVCMOS33 | O | P1 | 1路发送时钟 |
| SYNC1\_1 | LVCMOS33 | O | L1 | 1路同步信号 |
| DIN1[9:0] | LVCMOS33 | I | Y10、W6、U11、V7、V8、U8、U9、AA6、AA7、AA11 | 1路10位并行接收数据 |
| RCLK1\_R/FN | LVCMOS33 | O | R8 | 1路接收时钟边沿有效选择 |
| LOCK1\_N | LVCMOS33 | I | Y7 | 1路失锁信号 |
| RCLK1 | LVCMOS33 | I | AA10 | 1路接收时钟 |
| REFCLK1 | LVCMOS33 | O | Y9 | 解串器参考时钟 |
| Dout2[9:0] | LVCMOS33 | O | N20、N21、K21、L21、K15、L15、J21、J20、J17、J18 | 2路10位并行发送数据 |
| TCLK2\_R/FN | LVCMOS33 | O | K17 | 2路发送时钟边沿有效选择 |
| TCLK2 | LVCMOS33 | O | L17 | 2路发送时钟 |
| SYNC2\_1 | LVCMOS33 | O | H17 | 2路同步信号 |
| DIN2[9:0] | LVCMOS33 | I | G18、F20、F19、E21、E20、E18、E17、B21、D21、H21 | 2路10位并行接收数据 |
| RCLK2\_R/FN | LVCMOS33 | O | C21 | 2路接收时钟边沿有效选择 |
| LOCK2\_N | LVCMOS33 | I | M18 | 2路失锁信号 |
| RCLK2 | LVCMOS33 | I | L20 | 2路接收时钟 |
| REFCLK2 | LVCMOS33 | O | L19 | 解串器参考时钟 |

**SERDES收发模块**：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 信号名称 | 电平 | I/O | 引脚 | 备注 |
| SERDES\_0\_TXD0\_P | LVCMOS25 | O | AA17 | 1路SERDES发正 |
| SERDES\_0\_TXD0\_N | LVCMOS25 | O | Y17 | 1路SERDES发负 |
| SERDES\_0\_RXD0\_P | LVCMOS25 | I | Y19 | 1路SERDES收正 |
| SERDES\_0\_RXD0\_N | LVCMOS25 | I | AA19 | 1路SERDES收负 |
| SERDES\_0\_TXD2\_P | LVCMOS25 | O | Y13 | 2路SERDES发正 |
| SERDES\_0\_TXD2\_N | LVCMOS25 | O | AA13 | 2路SERDES发负 |
| SERDES\_0\_RXD2\_P | LVCMOS25 | I | AA15 | 2路SERDES收正 |
| SERDES\_0\_RXD2\_N | LVCMOS25 | I | Y15 | 2路SERDES收负 |

**SPI模块**：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 信号名称 | 电平 | I/O | 引脚 | 备注 |
| SPI\_W\_N | LVCMOS33 | O | M1 |  |
| SPI\_HOLD\_N | LVCMOS33 | O | N1 |  |
| SPI\_MOSI | LVCMOS33 | O | P4 |  |
| SPI\_CS\_N | LVCMOS33 | O | P5 |  |
| SPI\_SCLK | LVCMOS33 | O | U2 |  |
| SPI\_MISO | LVCMOS33 | I | U1 |  |

**JTAG模块**：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 信号名称 | 电平 | I/O | 引脚 | 备注 |
| JTAG\_TRSTB | LVCMOS33 | O | Y3 |  |
| JTAG\_TMS | LVCMOS33 | O | AA3 |  |
| JTAG\_TDO | LVCMOS33 | O | R7 |  |
| JTAG\_TDI | LVCMOS33 | I | N9 |  |
| JTAG\_TCK | LVCMOS33 | O | L9 |  |
| JTAGSEL | LVCMOS33 | O | V4 |  |

**DDR模块**：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 信号名称 | 电平 | I/O | 引脚 | 备注 |
| MDDR\_A[13:0] | LVCMOS18 | I/O | A4、E7、E6、A5、B5、D5、D6、A2、F5、E5、B2、B1、E4、D4 |  |
| MDDR\_D[15:0] | LVCMOS18 | I/O | D13、E13、D14、D15、A13、A14、B14、G11、B12、B11、E11、E12、D9、E10、B9、A10 |  |
| MDDR\_UDM | LVCMOS18 | O | D11 |  |
| MDDR\_UDQS | LVCMOS18 | O | C11 |  |
| MDDR\_LDM | LVCMOS18 | O | A12 |  |
| MDDR\_LDQS | LVCMOS18 | O | A15 |  |
| MDDR\_BA1 | LVCMOS18 | O | B6 |  |
| MDDR\_BA0 | LVCMOS18 | O | C6 |  |
| MDDR\_CK\_N | LVCMOS18 | O | A7 |  |
| MDDR\_CK\_P | LVCMOS18 | O | B7 |  |
| MDDR\_CAS\_L | LVCMOS18 | O | D8 |  |
| MDDR\_CS\_L | LVCMOS18 | O | G10 |  |
| MDDR\_CKE | LVCMOS18 | O | G9 |  |
| MDDR\_WE\_L | LVCMOS18 | O | A9 |  |
| MDDR\_RAS\_L | LVCMOS18 | O | A8 |  |

**RS485模块**：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 信号名称 | 电平 | I/O | 引脚 | 备注 |
| UART\_RX | LVCMOS33 | I | F1 |  |
| UART\_TX | LVCMOS33 | O | G1 |  |
| UART\_CONTROL | LVCMOS33 | O | G5 |  |

## 7.2、管脚说明

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 信号名称 | 说明 | 接口名称 | 位宽 | 类型 |
| cpu\_reset | 8995复位 | CLK\_1M536 | 1 | 输入 |
| mclk | 1.536mhz时钟 | SYS\_RST\* | 1 | 输入 |
| PG8\_FPGA\_RDY | FPGA正常启动信号 | PG8 | 1 | 输出 |
| PH0\_L\_INT | 左中断 | PH0 | 1 | 输出 |
| PH1\_R\_INT | 右中断 | PH1 | 1 | 输出 |
| PH2\_INT\_GEN | 中断产生使能 | PH2 | 1 | 输入 |
| PH6\_DSP\_RDY | DSP正常启动信号 | PH6 | 1 | 输入 |
| ps0 | 配置8895的配置方式ps(1,0)=10 为spi方式 | ps0，ps1 | 1 | 输出 |
| ps1 | 1 | 输出 |
| pwd | 8995电源模式 | POWER | 1 | 输出 |
| sdi | 8995sdi数据输出线 | SDI | 1 | 输入 |
| sdo | 8995sdo 数据输入线 | SDO | 1 | 输出 |
| spic | 8995spi时钟线 | SPIC | 1 | 输出 |
| spis\_n | 8995spis 信号传输使能 | SPIS | 1 | 输出 |
| SYS\_NRST | 系统复位 | RST\_N | 1 | 输入 |
| PH3\_INT\_MOD | 进入中断模式 | PH3 | 1 | 输入 |

# 8、地址映射

## 8.1、逻辑内部寄存器空间分布

## 8.2、逻辑外部RAM空间分布

## 8.3、逻辑内部RAM空间分布

# 9、寄存器描述

## 9.1、寄存器映射

## 9.2、寄存器描述

## 9.3、寄存器块1