FSL91030(M)芯片

SoC 使用说明书

手册版本：V1.0

武汉飞思灵微电子技术有限公司
2022 年 12 月

目 录

1 功能概述

FSL91030M 芯片的 SoC 功能概述如下：

- 使用 Nuclei 内核，基于 RISC-V 架构的 64 位精简指令集。
- 配备片上 SRAM 作为 LM：
  - 64KB 紧耦合指令存储器
  - 64KB 紧耦合数据存储器
- 配备自定义的 SoC 总线。
- 配备多种外设 IP
  - 2 路异步串行收发接口
  - 一路 I2C 接口
  - 通用 GPIO: 支持 24 路通用输入输出。
  - 专用复位:
    - ✓ 1 个 System 复位接口;
    - ✓ 1 个 POR 复位接口;
  - QSPI Flash 接口
    - ✓ 支持通过 XIP 方式进行 SPI FLASH 启动

1.1 功能框图

FSL91030M 芯片的 SoC 的框图如图 1-1 所示。图中除了 RSIC V 内核和总线之外，还包含相关外设 IP。有关各外设的详细介绍，请参见第 9 章。

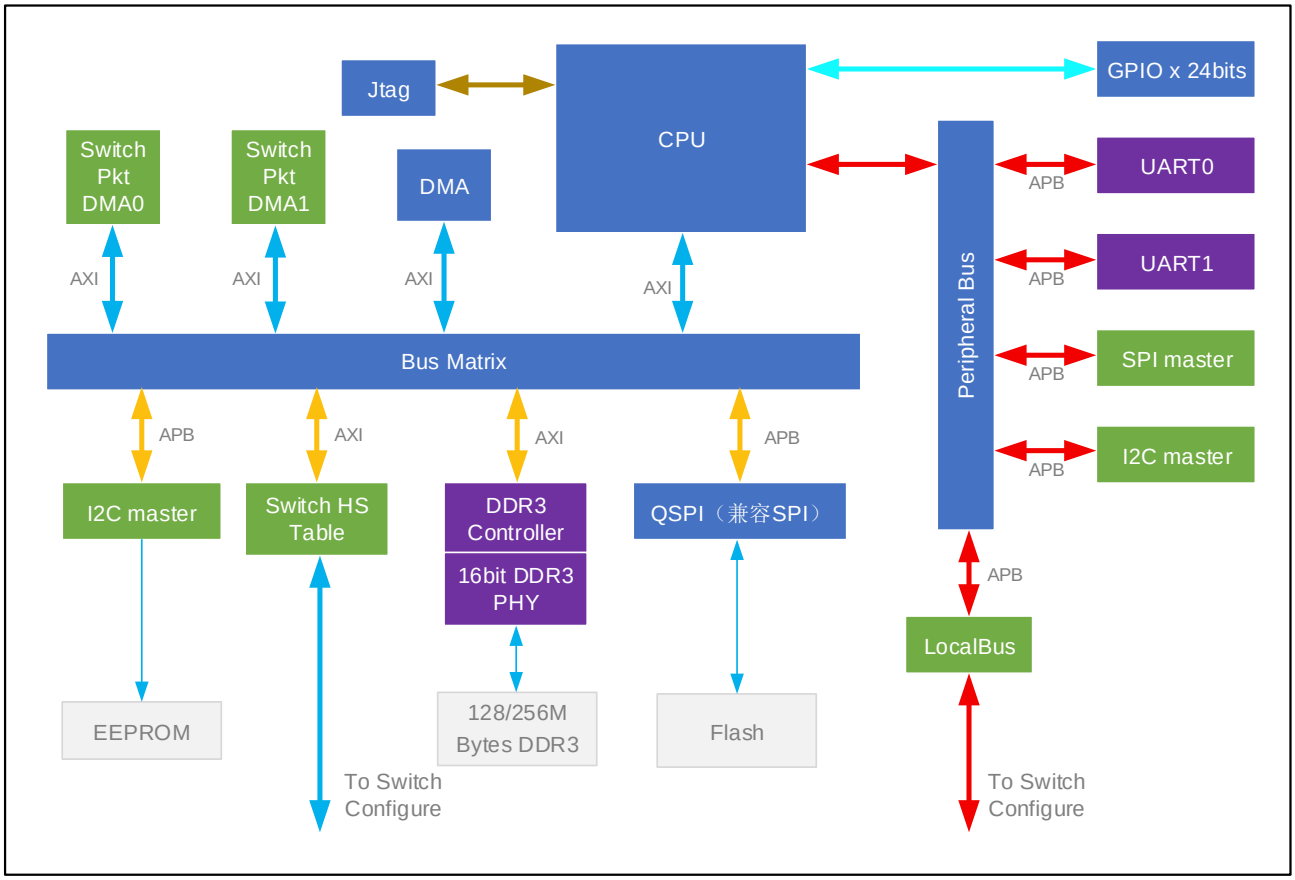

图 1-1 SoC 框图

SoC 的时钟如图 1-2 所示。

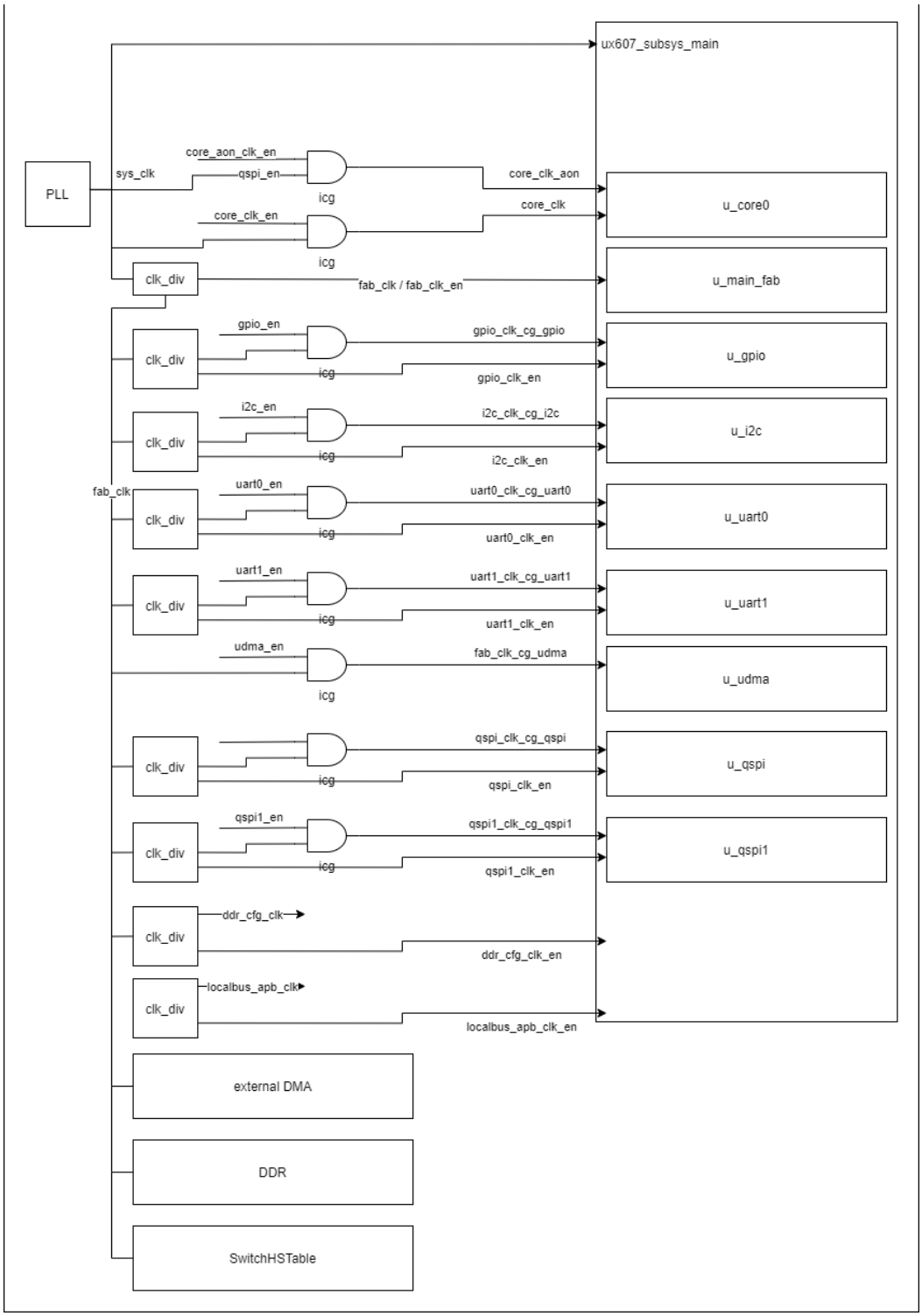

图 1-2 SoC 时钟

采用的分频单元如图 1-3 所示。

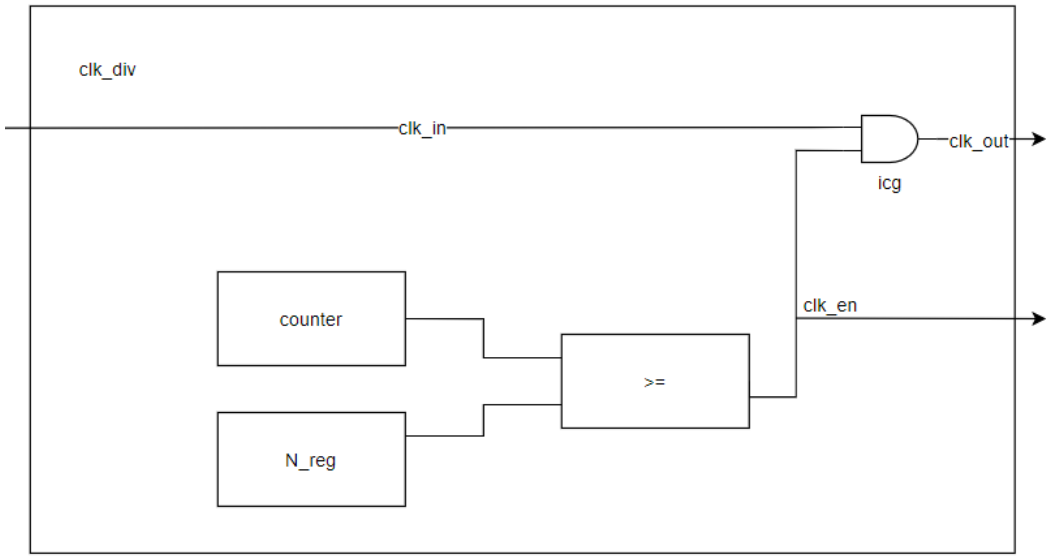

图 1-3 采用的分频单元

SoC 的复位如图 1-4 所示。

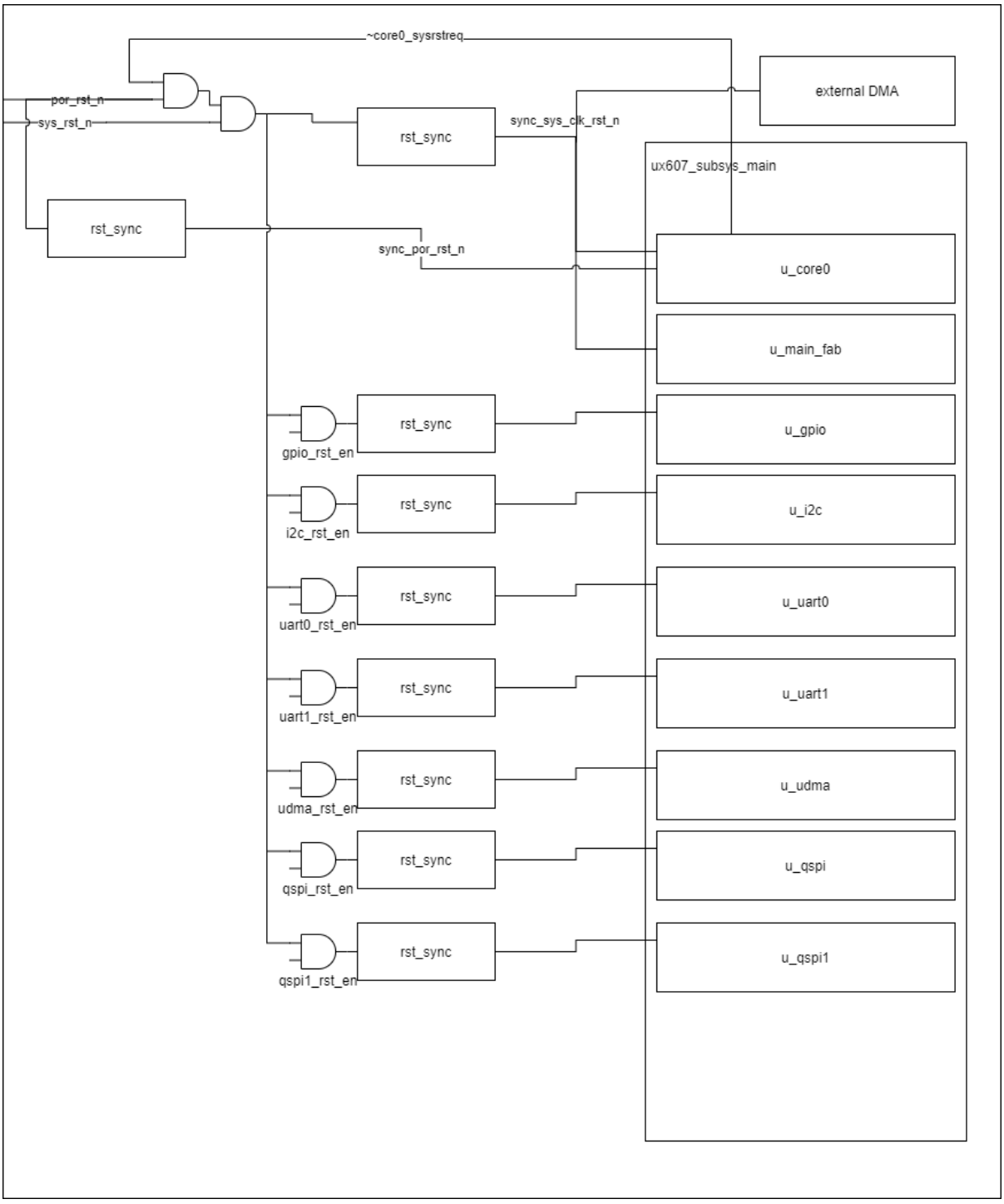

图 1-4 SoC 复位

1.2 总线特性

FSL91030M 芯片的 SoC 定义了一种自定义总线协议 ICB（Internal Chip Bus）。ICB 总线的初衷是为了能够尽可能地结合 AXI 总线和 AHB 总线的优点，兼具高速性和易用性，它具有如下特性：

- 相比 AXI 和 AHB 而言，ICB 的协议控制更加简单，仅有两个独立的通道，如图 1-5 所示，读和写操作共用地址通道，共用结果返回通道。
- 与 AXI 总线一样采用分离的地址和数据阶段。
- 与 AXI 总线一样采用地址区间寻址，支持任意的主从数目，譬如一主一从，一主多从，多主一从，多主多从等拓扑结构。
- 与 AHB 总线一样每个读或者写操作都会在地址通道上产生地址，而非像 AXI 中只产生起始地址。
- 与 AXI 总线一样支持地址非对齐的数据访问，使用字节掩码（Write Mask）来控制部分写操作。
- 与 AXI 总线一样支持多个滞外交易（Multiple Oustanding Transaction）。
- 与 AHB 总线一样不支持乱序返回乱序完成。反馈通道必须按顺序返回结果。
- 与 AXI 总线一样非常容易添加流水线级数以获得高频的时序。
- 协议非常简单，易于桥接转换成其他总线类型，譬如 AXI，AHB，APB 或者 TileLink 等总线。

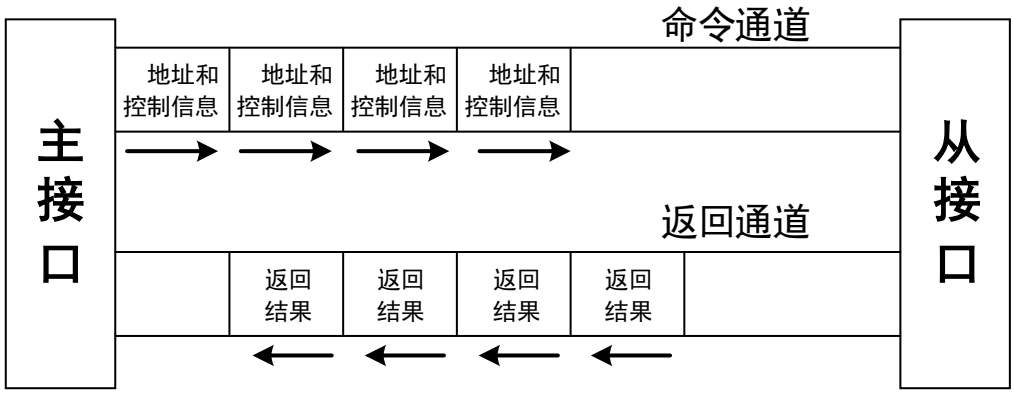

图 1-5 ICB 总线示意图

2 RISC V 处理器核心简介

RISC V 内核的特性列表如下：

- CPU 内核（CPU Core）
  - 6 级变长流水线架构，采用一流的处理器架构设计，实现业界一流的能效比与综合成本。
  - 支持动态分支预测。
  - 可配置的指令预取单元，能够按顺序预取两条指令，从而隐藏指令的访存延迟。
  - 支持机器模式（Machine Mode）、可配置的监督模式（Supervisor Mode）和可配置的用户模式（User Mode）。
  - 支持非对齐访问存储器。
  - 两级异常嵌套恢复。
  - 支持硬件单周期乘法器。
  - 支持硬件多周期除法器。
- 支持指令集架构（ISA，Instruction Set Architecture）

处理器核支持 64 位的 RISC-V 指令集架构，支持 RV64I/M/A/C

等指令子集的配置组合。其中 IMAC 为固定支持的指令子集组合。

- 总线接口
  - 支持 64 比特宽的标准 AXI 系统总线接口，用于访问外部指令和数据。
  - 支持 64 比特宽的指令局部存储器（Instruction Local Memory，ILM）总线接口（支持 SRAM 接口协议），用于连接私有的指令局部存储器。
  - 支持 64 比特宽的数据局部存储器（Data Local Memory，DLM）总线接口（支持 SRAM 接口协议），用于连接私有的数据局部存储器。
  - 支持 64 比特宽的私有设备总线（Private Peripheral Interface，PPI），支持标准的 APB 接口协议，用于连接私有的外设。
- 支持调试功能
  - 支持标准的四线 JTAG 接口。

- ✧ 支持 RISC-V 调试标准。
- ✧ 支持可配置数目的硬件断点（Hardware Breakpoints）。
- ✧ 支持可配置的调试器连接超时功能。
- ✧ 支持成熟的交互式调试工具。

➢ 支持低功耗管理

- ✧ 支持 WFI（Wait For Interrupt）与 WFE（Wait For Event）进入休眠模式。
- ✧ 支持两级休眠模式：浅度休眠与深度休眠。

➢ 支持增强的内核中断控制器（Enhanced Core Level Interrupt Controller，ECLIC）

- ✧ 支持 RISC-V 标准定义的的软件中断、计时器中断和外部中断。
- ✧ 支持可配置数目的外部中断。
- ✧ 支持可配置数目的中断级别和优先级，支持软件动态可编程修改中断级别和中断优先级的数值。
- ✧ 支持基于中断级别的中断嵌套。
- ✧ 支持快速向量中断处理机制。
- ✧ 支持快速中断咬尾机制。

➢ 支持 NMI（Non-Maskable Interrupt）。

3 存储资源

如图 1-1 中所示，FSL91030M 芯片的 SoC 中的存储器资源分为片上存储资源和片外 Flash 存储资源，本章节将分别予以介绍。

3.1 片上存储资源

FSL91030M 芯片的 SoC 的片上存储资源主要有 ILM，DLM 和系统 RAM。

- ILM 为处理器内核的指令存储器，其特性如下：
  - 大小 64KB。
  - 可配置的地址区间（默认起始地址为 0x8000_0000），见第 4 章中 SoC 的完整地址分配。
  - 在此配套 SoC 中，ILM SRAM 虽然主要用于存放指令，但是其地址区间也可以被用来存放数据。处理器核可以通过 SoC 的系统存储总线访问到 ILM。
- DLM 为处理器内核的数据存储器，其特性如下：
  - 大小 64KB。
  - 可配置的地址区间（默认起始地址为 0x8001_0000），见第 4 章中 SoC 的完整地址分配。
  - 在此配套 SoC 中，DLM SRAM 虽然主要用于存放数据，但是其地址区间也可以被用来存放指令。处理器核可以通过 SoC 的系统存储总线访问到 DLM。
- 系统 DDR RAM 为 SOC 总线空间的数据存储器，其特性如下：
  - 大小 256MB。
  - 可配置的地址区间（默认起始地址为 0x4000_0000），见第 4 章中 SoC 的完整地址分配。
  - 在此配套 SoC 中，DLM SRAM 虽然主要用于存放数据，但是其地址区间也可以被用来存放指令。处理器核可以通过 SoC 的系统存储总线访问到 DLM。

3.2 片外 Flash 存储资源

FSL91030M 芯片的 SoC 片外存储资源主要为 Flash 存储器，其要点如下：

- 外部 Flash 可以利用其 XiP（Execution in Place）模式，通过 QSPI0 被映射为一片只读的地址区间，地址区间为 0x2000_0000 ~ 0x2FFF_FFFF。
- 用户可以通过调试器将开发的程序烧写在 Flash 中，然后利用 Flash 的 XiP 模式，程序可以直接从 Flash 中被执行。

3.3 片上 SRAM 列表

3.3.1 ICache

Tag Single Port SRAM：8 块

宽度：54

深度：128

有效 Write-Mask: 1 bit

Data Single Port SRAM：8 块

宽度：64

深度：512

有效 Write-Mask: 18bit

Name	Direction	Width	Description
icache_tag0_cs	O	1	CS signal of Tag RAM0.
icache_tag0_we	O	1	Write enable of Tag RAM0
icache_tag0_addr	O	7	Address of Tag RAM0
icache_tag0_wdata	O	54	Write data of Tag RAM0
icache_tag0_rdata	I	54	Read data of Tag RAM0
clk_icache_tag0	O	1	Tag RAM0’s CLK signal
icache_data0_cs	O	1	CS signal of Data RAM0.
icache_data0_wem	O	8	Write enable of Data RAM0, support the byte enable
icache_data0_addr	O	9	Address of Data RAM0
icache_data0_wdata	O	64	Write data of Data RAM0.
icache_data0_rdata	I	64	Read data of Data RAM0.
clk_icache_data0	O	1	Data RAM0’s CLK signal

3.3.2 Dcache

Tag Single Port SRAM：2 块

宽度：23

深度：128

有效 Write-Mask: 1bit

Data Single Port SRAM：4 块

宽度：32

深度：512

有效 Write-Mask: 4 bit

Name	Direction	Width	Description
dcache_w0_tram_cs	O	1	CS signal of Tag RAM0.
dcache_w0_tram_we	O	1	Write enable of Tag RAM0
dcache_w0_tram_addr	O	10	Address of Tag RAM0
dcache_w0_tram_din	O	20	Write data of Tag RAM0
dcache_w0_tram_dout	I	20	Read data of Tag RAM0
clk_dcache_w0_tram	O	1	Tag RAM0’s CLK signal
dcache_dram_b0_cs	O	1	CS signal of Data RAM0.
dcache_dram_b0_wem	O	4	Write enable of Data RAM0, support the byte enable
dcache_dram_b0_addr	O	12	Address of Data RAM0
dcache_dram_b0_din	O	32	Write data of Data RAM0.
dcache_dram_b0_dout	I	32	Read data of Data RAM0.
clk_dcache_dram_b0	O	1	Data RAM0’s CLK signal

3.3.3 ILM

ILM Single Port SRAM：1 块

宽度：64

深度：64KB

有效 Write-Mask: 8bit

Name	Direction	Width	Description
iram_ram_cs	O	1	CS signal of Data RAM.
iram_ram_wem	O	8	Write mask of Data RAM, support the byte enable
iram_ram_we	O	1	Write enable of Data RAM
iram_ram_addr	O	13	Address of Data RAM
iram_ram_din	O	64	Write data of Data RAM
iram_ram_dout	I	64	Read data of Data RAM
iram_ram_clk	O	1	Data RAM’s CLK signal

3.3.4 DLM

DLM Single Port SRAM：2 块

宽度：32

深度：共计 64KB

有效 Write-Mask: 4 bit

Name	Direction	Width	Description
dram_ram_cs	O	1	CS signal of Data RAM.
dram_ram_wem	O	4	Write mask of Data RAM, support the byte enable
dram_ram_we	O	1	Write enable of Data RAM
dram_ram_addr	O	13	Address of Data RAM
dram_ram_din	O	32	Write data of Data RAM
dram_ram_dout	I	32	Read data of Data RAM
dram_ram_clk	O	1	Data RAM’s CLK signal

4 地址分配

FSL91030M 芯片的 SoC 总线地址分配如表 4-1 所示。

表 4-1 SoC 地址分配表

总线分组	组件	地址区间	描述
Core 内部私有	DEBUG	0x0000_0000 ~ 0x0000_0FFF	注意：调试模块（Debug Module）主要供调试器使用，普通软件程序不应该使用此区间
	ECLIC	0x0C00_0000 ～ 0x0C00_FFFF	ECLIC 中断管理单元地址空间
	PLIC	0x0800_0000 ～ 0x0800_FFFF	PLIC 中断管理单元地址空间
系统存储（MEM）接口	ILM	0x8000_0000 ~ 0x8000_FFFF	ILM 地址区间，区间大小取决于 ILM 配置的大小。
	DLM	0x8001_0000 ~ 0x8001_FFFF	DLM 地址区间，区间大小取决于 ILM 配置的大小。
系统外设	GPIO	0x1001_1000 ~ 0x1001_1FFF	GPIO 模块寄存器地址区间
	UART 0	0x1001_3000 ~ 0x1001_3FFF	第一个 UART 模块寄存器地址区间
	UART 1	0x1001_2000 ~ 0x1001_2FFF	第二个 UART 模块寄存器地址区间
	I2C	0x1001_8000 ~ 0x1001_8FFF	I2C 模块寄存器地址区间
	QSPI	0x1001_4000 ~ 0x1001_4FFF	XPI QSPI 模块寄存器地址区间
	QSPI0	0x1001_6000 ~ 0x1001_6FFF	QSPI0 模块寄存器地址区间
	DMA	0x1001_7000 ~ 0x1001_7FFF	DMA 模块寄存器地址区间
	Off-Chip QSPI Flash Read	0x2000_0000 ~ 0x2FFF_FFFF	QSPI 处于 Flash XiP 模式时将外部 Flash 映射的只读地址区间。
外部 AXI 接口地址空间	SwitchHStable	0x5000_0000 ~ 0x5FFF_FFFF	table DMA 访问 switch 寄存器时的地址区间
	DDR CFG	0x1001_9000~ 0x1001_9FFF	Ddr 配置寄存器地址区间
	Local Bus	0x6000_0000 ~ 0x6FFF_FFFF	直接访问 switch 寄存器时的地址区间
	DDR RAM	0x4000_0000 ~ 0x4FFF_FFFF	DDR RAM 地址区间，区间大小取决于 DDR RAM 配置的大小。
其他地址区间	表中未使用到的地址区间，均为写忽略，读返回 0		

5 上电流程控制

FSL91030M 芯片的 SoC 中处理器核上电复位之后，可更具 reset_vector 输入选择不同的地址进行执行程序。通常可从可设置为如下的地址：

- 从外部 Flash 开始执行
  - 由于映射的外部 Flash（Off-Chip QSPI0 Flash Read）的地址区间位于 0x2000_0000 ~ 0x2FFF_FFFF，因此如果从外部 Flash 开始执行，则 RISC-V 处理器核的 PC 复位值为 0x20000000。
  - 用户可以通过调试器将开发的程序烧写在 Flash 中，利用 Flash 的 XiP 模式，程序可以直接从 Flash 中被执行。
- 从 DDR 地址开始执行 。
  - 用户可以通过调试器将开发的程序烧写在 DDR 中，程序可以直接从 DDR 中执行。

6 复位处理

FSL91030M 芯片的 SoC 中有 2 个复位域，其中包括：

1. PoR 复位域：Debug 模块和 JTAG DMI 模块
2. 系统复位域：包括除了 PoR 复位域以外的系统其他模块

SoC 的复位有以下几个来源。

（1）来自 POR（Power-On-Reset）nPOR_RST 芯片引脚。

在芯片上电后，片外的 POR 在电压达到稳定阈值之前一直输出复位信号，保证芯片能够被正确地自动上电复位。此复位源可作用于 POR 和系统复位域。

（2）来自芯片引脚 nSYS_RST。

nSYS_RST 引脚可以用于外部复位。此复位源仅仅作用于系统复位域。

（3）来自 Core 的 Reset，其中包含 Debug Module 触发的 Reset。此复位源仅仅作用于系统复位域。

上述全局复位来源中的任何一种驱动都可以触发系统复位。

7 顶层引脚表

FSL91030M 芯片的 SoC 的顶层引脚如表 7-1 所示。

表 7-1 顶层引脚表

	名称	功能	辅助功能 1	辅助功能 2
时钟	syst_clk	系统时钟输入		
时钟	rtc_clk	RTC 时钟输入		
系统	nPOR_RST	系统上电复位信号输入		
系统	nSYS_RST	系统正常运行复位信号输入		
系统	JTAG_TCK	JTAG 调试口		
系统	JTAG_TDI			
系统	JTAG_TDO			
系统	JTAG_TMS			
系统	ext_nmi	NMI 输入		
系统	Test_mode	DFT test mode		
QSPI	qspi_d0			
	qspi_d1			
	qspi_d2			
	qspi_d3			
	qspi_sck			
	qspi_cs			
QSPI0	qspi0_d0			
	qspi0_d1			
	qspi0_d2			
	qspi0_d3			
	qspi0_sck			
	qspi0_cs			
I2C	sda			
	scl			
UART0	rx			
	tx			
UART1	rx			
	tx			
DDR	DDR interface			
	ddr_init_done			
GPIO	GPIO0	GPIO0		
GPIO	GPIO1	GPIO1		
GPIO	GPIO2	GPIO2		
GPIO	GPIO3	GPIO3		
GPIO	GPIO4	GPIO4		
GPIO	GPIO5	GPIO5		
GPIO	GPIO6	GPIO6		
GPIO	GPIO7	GPIO7		
GPIO	GPIO8	GPIO8		
GPIO	GPIO9	GPIO9		
GPIO	GPIO10	GPIO10		

GPIO	GPIO11	GPIO11		
GPIO	GPIO12	GPIO12		
GPIO	GPIO13	GPIO13		
GPIO	GPIO14	GPIO14		
GPIO	GPIO15	GPIO15		
GPIO	GPIO16	GPIO16		
GPIO	GPIO17	GPIO17		
GPIO	GPIO18	GPIO18		
GPIO	GPIO19	GPIO19		
GPIO	GPIO20	GPIO20		
GPIO	GPIO21	GPIO21		
GPIO	GPIO22	GPIO22		
GPIO	GPIO23	GPIO23		

8 中断处理

本章将对中断模块进行介绍。

8.1 IRQC 中断管理

IRQC 可以支持多个外部中断源，在具体的 SoC 中连接的中断源个数可能不一样。IRQC 在此 SoC 中连接了 GPIO 中断和其他外设中断作为外部中断源，其中断分配如表 8-1 所示。

表 8-1 IRQC 中断管理

IRQC 中断源编号	当前新版本
19	gpio_0~23
20	uart0
21	uart1
22	I2C
23	QSPI
24	QSPI0
25	udma
26	Watch dog
27	Switch
28	Pkt_dma_tx_end
29	Pkt_dma_rx_end
30	Pkt_dma_rx_req
31	ddr
32	Axi_cp_intr
33	Dp2reg_intr

8.2 NMI 中断管理

NMI 中断在本 SoC 的外部端口上。

9 外设介绍

本章将对挂载在 SoC 的私有设备总线上的外设模块进行介绍。

9.1 GPIO

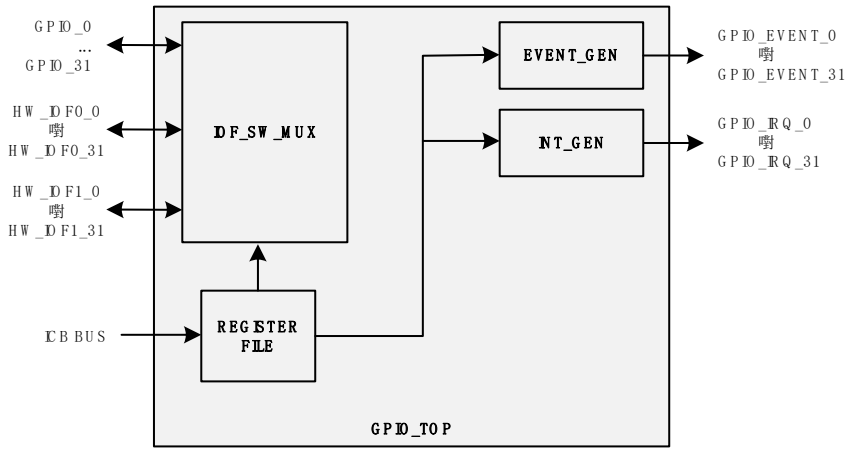

图 9-1 GPIO 数据流

9.1.1 GPIO 功能描述

GPIO 全称为 General Purpose I/O，其要点如下：

- GPIO 用于提供一组 I/O 的通用输入输出接口。每个 I/O 可用被软件配置为输入或者输出，如果是输出可以设置具体的输出值。
- 每个 GPIO 的 I/O 均作为一个中断源连接到 IRQC 的中断源接口上。
- 每个 I/O 还可以被配置为 IOF（Hardware I/O Functions），也就是将 I/O 供 SoC 内部的其他模块复用。
- 譬如 SPI，UART 等等。GPIO 被 SoC 内部模块的复用分配如图 9-1 所示，其中每个 I/O 均可以供两个内部模块复用，软件可以通过配置每个 I/O 使其选择 IOF0 或者 IOF1 来选择信号来源。

FSL91030M 芯片的 SoC GPIO 引脚是 SoC 与外界连接的主要通用接口。

9.1.2 端口操作描述

GPIO 都端口的每位可以有软件分别配置成多种模式：

- 输入上拉

- 输入下拉
- 开漏输出
- 推挽式输出
- 驱动强度输出

9.1.3 GPIO 中断请求

每个 GPIO 位可以产生一个中断位。中断可以由上升沿或下降沿驱动，也可以由电平值驱动，并且每个中断可以单独使能。

输入在被中断逻辑采样之前已同步，因此输入脉冲宽度必须足够长才能被同步逻辑检测到。

要启用中断，请将 rise_ie 和/或 fall_ie 中的相应位设置为 1。如果将 rise_ip 或 fall_ip 中的相应位设置为 1，则中断引脚将升高。

表 9-1 GPIO 中断请求

Interrupt Source	Interrupt Flag	Interrupt Control bit
rising edge from GPIO	RISE_IP	RISE_IE
falling edge from GPIO	FALL_IP	FALL_IE
high level from GPIO	HIGH_IP	HIGH_IE
low level from GPIO	LOW_IP	LOW_IE

9.1.4 硬件 I/O 功能（IOF）

每个 GPIO 引脚可通过 IOF_EN 寄存器实现最多 2 个 HW 驱动功能（IOF）使能。 通过 IOF_SEL0 和 IOF_SEL1 寄存器选择使用哪种 IOF。

当将引脚设置为执行 IOF 时，可能无法使用软件寄存器 IEN，OEN，OVAL，PUE，DS，PDE，OD，PUP 来直接控制引脚。而可以通过硬件驱动 IOF 来控制引脚。哪些功能由 IOF 控制，哪些功能由软件寄存器控制，在每个 IOF 的基础上固定在硬件中。如果使用 IOF_EN [x]配置的 pin [x]没有 iof_en，则该引脚将恢复为完全软件控制。

9.1.5 输出反转

当配置为输出（由 SW 或 IOF 控制）时，可写 SW 的异或寄存器与输出组合以对其进行反相。

9.1.6 存储器映射

GPIO 寄存器的存储器映射如下表所示。GPIO 存储器映射为要求自然对其的 32 位存储器访问。

表 9-2 GPIO 存储器地址映射寄存器列表

Register Offset	Register Name	Description

0x00	IVAL	Pin 的值
0x04	IEN	Pin 的输入使能
0x08	OEN	Pin 的输出使能
0x0C	OVAL	输出端口值
0x10	PUE	上拉使能
0x14	DS	驱动强度
0x18	PDE	下拉使能
0x1C	OPEN DRAIN	开漏使能
0x20	PUP	推挽使能
0x24	RISE_IE	上升沿中断使能
0x28	RISE_IP	上升沿中断等待标志
0x2C	FALL_IE	下降沿中断使能
0x30	FALL_IP	下降沿中断等待标志
0x34	HIGH_IE	高电平中断使能
0x38	HIGH_IP	高电平中断等待标志
0x3C	LOW_IE	低电平中断使能
0x40	LOW_IP	低电平中断等待标志
0x44	IOF_EN	HW I/O function 使能
0x48	IOF_SEL0	HW I/O function 选择 0
0x4C	IOF_SEL1	HW I/O function 选择 1
0x50	EVENT_RISE_EN	上升沿事件使能
0x54	EVENT_FALL_EN	下降沿事件使能
0x58	OUT_XOR	输出异或
0x5C	SW_FILTER_EN	主动施密特触发器输入

9.1.7 IVAL

寄存器 offset：0x00

寄存器描述：Pin 的值

Bits	Name	R/W	Description	Default
[0:31]	ival	RO	输入值	0x0

9.1.8 IEN

寄存器 offset：0x04

寄存器描述：Pin 的输入使能

Bits	Name	R/W	Description	Default

0:31	ien	RW	输入使能 0: 输入禁止 1: 输入启用	0x0

9.1.9 OEN

寄存器 offset：0x08

寄存器描述：Pin 的输出使能

Bits	Name	R/W	Description	Default
0:31	oen	RW	输出使能 0: 输出禁止 1: 输出启用	0x0

9.1.10 OVAL

寄存器 offset：0x0C

寄存器描述：输出端口值

Bits	Name	R/W	Description	Default
0:31	oval	RW	输出值	0x0

9.1.11 PUE

寄存器 offset：0x10

寄存器描述：上拉使能

Bits	Name	R/W	Description	Default
0:31	pue	RW	上拉模式： 0: 禁止 1: 启用	0x0

9.1.12 DS

寄存器 offset：0x14

寄存器描述：驱动强度

Bits	Name	R/W	Description	Default
0:31	ds	RW	当配置成输出时，每个引脚都有一个 SW 可控制的驱动强度 0: 禁止 1: 启用	0x0

9.1.13 PDE

寄存器 offset：0x18

寄存器描述：下拉使能

Bits	Name	R/W	Description	Default
[0:31]	pde	RW	下拉模式： 0: 禁止 1: 启用	0x0

9.1.14 OPEN DRAIN

寄存器 offset：0x1C

寄存器描述：开漏使能

Bits	Name	R/W	Description	Default
0:31	open drain	RW	开漏模式： 0: 禁止 1: 启用	0x0

9.1.15 PUP

寄存器 offset：0x20

寄存器描述：推挽使能

Bits	Name	R/W	Description	Default
0:31	pull-up enable	RW	上拉模式: 0: 禁止 1: 启用	0x0

9.1.16 RISE_IE

寄存器 offset：0x24

寄存器描述：上升沿中断使能

Bits	Name	R/W	Description	Default
0:31	rise ie	RW	上升沿中断模式： 0: 禁止 1: 启用	0x0

9.1.17 RISE_IP

寄存器 offset：0x28

寄存器描述：上升沿中断等待标志

Bits	Name	R/W	Description	Default
0:31	rise ip	RW	上升沿中断挂起： 0：上升沿中断由 GPIO 上升沿驱动 1：上升沿中断由软件驱动	0x0

9.1.18 FALL_IE

寄存器 offset：0x2C

寄存器描述：下降沿中断使能

Bits	Name	R/W	Description	Default
0:31	fall ie	RW	下降沿中断使能： 0：禁止 1：启用	0x0

9.1.19 FALL_IP

寄存器 offset：0x30

寄存器描述：下降沿中断等待标志

Bits	Name	R/W	Description	Default
0:31	fall ip	RW	下降沿中断等待标志： 0： 下降沿中断由 GPIO 的下降沿驱动 1：下降沿中断由软件驱动	0x0

9.1.20 HIGH_IE

寄存器 offset：0x34

寄存器描述：高电平中断使能

Bits	Name	R/W	Description	Default
0:31	high ie	RW	高电平中断使能： 0：禁止 1：启用	0x0

9.1.21 HIGH_IP

寄存器 offset：0x38

寄存器描述：高电平中断等待标志

Bits	Name	R/W	Description	Default
0:31	high ip	RW	高电平中断等待标志： 0：下降沿中断由 GPIO 的下降沿驱动 1：下降沿中断由软件驱动	0x0

9.1.22 LOW_IP

寄存器 offset：0x40

寄存器描述：低电平中断等待标志

Bits	Name	R/W	Description	Default
0:31	low ip	RW	低电平中断等待标志： 0：低电平中断由 GPIO 的下降沿驱动 1：低电平中断由软件驱动	0x0

9.1.23 IOF_EN

寄存器 offset：0x44

寄存器描述： HW I/O 功能使能

Bits	Name	R/W	Description	Default
[0:31]	iof en	RW	HW I/O 功能使能	0x0

9.1.24 IOF_SEL0

寄存器 offset：0x48

寄存器描述： HW I/O 功能选择 0

Bits	Name	R/W	Description	Default
0:31	iof sel0	RW	每个 GPIO 引脚最多可以实现 4 个 HW 驱动功能（基于{iof_sel1,iof_sel0}） 00:选择 HW IOF0 01:选择 HW IOF1 10/11:反转	0x0

9.1.25 LOW_IE

寄存器 offset：0x3C

寄存器描述：低电平中断使能

Bits	Name	R/W	Description	Default
0:31	low ie	RW	低电平中断使能： 0：禁止 1：启用	0x0

9.1.26 IOF_SEL1

寄存器 offset：0x4C

寄存器描述： HW I/O 功能选择 1

Bits	Name	R/W	Description	Default
0:31	iof sel1	RW	每个 GPIO 引脚最多可以实现 4 个 HW 驱动功能(基于{iof_sel1,iof_sel0}) 00:选择 HW IOF0 01:选择 HW IOF1 10/11:反转	0x0

9.1.27 EVENT_RISE_EN

寄存器 offset：0x50

寄存器描述：上升沿事件使能

Bits	Name	R/W	Description	Default
0:31	event rise en	RW	上升沿事件使能： 0：禁止 1：启用	0x0

9.1.28 EVENT_FALL_EN

寄存器 offset：0x54

寄存器描述：下降沿事件使能

Bits	Name	R/W	Description	Default
0:31	event fall en	RW	下降沿事件使能： 0：禁止 1：启用	0x0

9.1.29 OUT_XOR

寄存器 offset：0x58

寄存器描述：输出反转使能

Bits	Name	R/W	Description	Default
0:31	out xor	RW	输出反转使能： 0: 禁止 1: 启用	0x0

9.1.30 SW_FILTER_EN

寄存器 offset：0x5C

寄存器描述：有效施密特触发器输入

Bits	Name	R/W	Description	Default
0:31	sw filter en	RW	在 PAD 接口中有效施密特触发器输入： 0：禁止 1：启用	0x0

9.2 QSPI

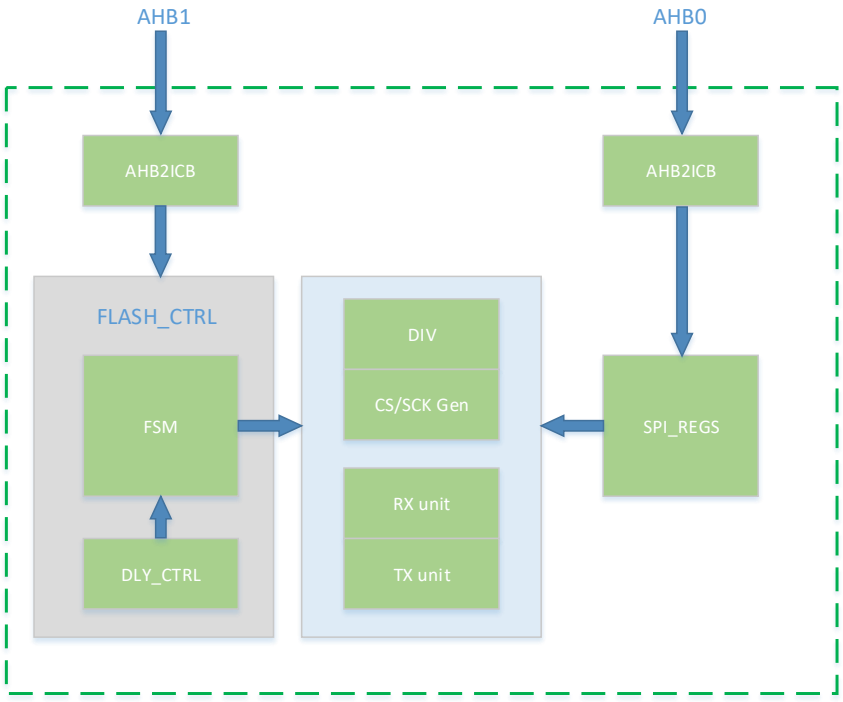

图 9-2 QSPI 数据流

在本 SoC 中，有一个 QSPI 主接口。作为 SPI 主机，支持发送和接收数据能力。虽然 Quad-SPI 有 4 根数据线，但是可以通过寄存器配置为单线（Single-SPI）、双线（Dual-SPI）和四线（Quad-SPI）模式。 支持发送和接受 FIFO 缓存，同时支持软件可编程的阈值（Watermark）以产生中断。支持通过寄存器配置 SPI 时钟信号 SCK 的极性和相位。

- 专用于连接外部 Flash 的 Quad-SPI（QSPI）接口，有专用的 SoC 顶层引脚。
- 并且该 QSPI 接口还可以被软件配置成为 eXecute-In-Place（简称 XIP）模式，在此模式下，Flash 可以被当作一段只读区间直接被当做存储器读取。在默认上电之后，QSPI 即处于该模式之下，由于 Flash 掉电不丢失的特性，因此可以将系统的启动程序存放于外部的 Flash 中，然后处理器核通过 eXecute-In-Place 模式的 QSPI 接口直接访问外部 Flash 加载启动程序启动。

9.2.1 寄存器描述

QSPI 的可配置寄存器为存储器地址映射寄存器（Memory Address Mapped）， SPI 作为从模块挂载在 SoC 的私有设备总线上，SPI 的可配置寄存器列表及其偏移地址如下表所示。

Register Offset	Register Name	Description
0x000	SPI_SCKDIV	SCK 时钟频率分频系数寄存器（xip/普通模式）
0x004	SPI_SCKMODE	SCK 模式配置寄存器（xip/普通模式）
0x00C	SPI_FORCE	SPI 未使用接口强制输出 1 使能
0x010	SPI_CSID	CS 选通标识（ID）寄存器（xip/普通模式）
0x014	SPI_CSDEF	CS 空闲值寄存器
0x018	SPI_CSMODE	CS 模式寄存器
0x01C	SPI_VISION	SPI 版本寄存器 1.0 版本
0x028	SPI_DELAY0	XIP 传输延迟控制寄存器 0（xip/普通模式）
0x02C	SPI_DELAY1	XIP 传输延迟控制寄存器 1（xip/普通模式）
0x040	SPI_FMT	传输参数配置寄存器（xip/普通模式）
0x07C	SPI_STATUS	传输状态寄存器（普通模式）
0x048	SPI_TXDATA	发送数据寄存器（普通模式）
0x04C	SPI_RXDATA	接收数据寄存器（普通模式）
0x060	SPI_FCTRL	XIP 模式控制寄存器（xip）
0x064	SPI_FFMT	XIP 传输参数控制寄存器（xip）
0x078	SPI_FFMT1	XIP 传输参数控制寄存器 1（xip）
0x080	SPI_RXEDGE	SPI 接收数据采样沿控制寄存器
0x050	SPI_TXMARK	SPI 发送 fifo 水位寄存器
0x054	SPI_RXMARK	SPI 接收 fifo 水位寄存器
0x070	SPI_IE	SPI 中断使能寄存器
0x074	SPI_IP	SPI 中断等待 pending 寄存器

9.2.2 SPI_SCKDIV

寄存器 offset：0x000

寄存器描述：用于设置 SPI 的 SCK 时钟频率

Bits	Name	R/W	Description	Default
0:11	div	RW	用于配置产生 SCK 信号的分频系数。	0x3

9.2.3 SPI_SCKMODE

寄存器 offset：0x004

寄存器描述：用于设置 SPI 的 SCK 时钟频率

Bits	Name	R/W	Description	Default
0	pha	RW	用于配置 CPHA。	0x0
1	pol	RW	用于配置 CPOL。	0x0

9.2.4 SPI_CSID

寄存器 offset：0x00C

寄存器描述：SPI 接口可以有多个使能信号。多个使能信号使用同一总线上连接多个 SPI 从设备成为可能，但是一次只能使能一个 SPI 从设备。SPI_CSID 寄存器用于选择设置 SPI 的使能信号。

Bits	Name	R/W	Description	Default
0:1	csid	RW	该域的值用于选择使能信号的索引。 00：不选通 01：选 SS0 10：选 SS1 11：选 SS2	0x0

9.2.5 SPI_CSDEF

寄存器 offset：0x014

寄存器描述：CS 空闲寄存器。

Bits	Name	R/W	Description	Default
0	cs0def	RW	该域的值表示 CS0 使能信号的空闲值	0x1
1	cs1def	RW	该域的值表示 CS1 使能信号的空闲值	0x1
2	cs2def	RW	该域的值表示 CS2 使能信号的空闲值	0x1
3	cs3def	RW	该域的值表示 CS3 使能信号的空闲值	0x1

9.2.6 SPI_CSMODE

寄存器 offset：0x018

寄存器描述：CS 模式寄存器。

Bits	Name	R/W	Description	Default

0:1	mode	RW	假设该域的值为 0，表示配置使能信号为 AUTO 模式。 假设该域的值为 2，表示配置使能信号为 HOLD 模式。 假设该域的值为 3，表示配置使能信号为 OFF 模式。	0x0

9.2.7 SPI_DELAY0

寄存器 offset：0x028

寄存器描述：用于配置延迟周期参数。

Bits	Name	R/W	Description	Default
0:7	cssck	RW	该域的值指定在开始发送数据之前，在第一个 SCK 时钟前沿之前至少提前多少个周期会将使能信号（SS）置为有效值。	0x0
16:23	sckcs	RW	该域的值指定在结束发送数据之后，在最后一个 SCK 时钟后沿之后至少多少个周期内仍会将使能信号（SS）保持为有效值。	0x0

9.2.8 SPI_DELAY1

寄存器 offset：0x02C

寄存器描述：用于配置若干延迟周期参数。

Bits	Name	R/W	Description	Default
0:7	intercs	RW	该域的值指定使能信号从“有效值恢复为空闲值（de-assertion）后”到“重新置为有效值（assertion）”之间最少应该持续的空闲周期数（Mininum CS inactive time）。	0x0

9.2.9 SPI_FMT

寄存器 offset：0x040

寄存器描述：在 FIFO 发送接收模式下，可以通过 SPI_TXDATA 和 SPI_RXDATA 寄存器进行发送或者接收数据操作。当童工亵渎 SPI_TXDATA 和 SPI_RXDATA 发送和接受数据是，SPI_FMT 寄存器可以用于配置若干传输参数。

Bits	Name	R/W	Description	Default
0:1	proto	RW	如果该域的值为 2，则配置传输协议为 Quad-SPI。在此模式下，有四根数据线 DQ0、DQ1、DQ2、DQ3 工作。 如果该域的值为 1，则配置传输协议为 Dual-SPI。在此模式下，有两根数据线 DQ0 和 DQ1 工作。 如果该域的值为 0，则配置传输协议为 Single-SPI。在此模式下，有两根数据线 DQ0（作为 MOSI）和 DQ1（作为 MISO）工作。	0x0
2	endian	RW	如果该域的值为 1，则对数据先发送低位（LSB 优先）。	0x0

			如果该域的值为 0，则对数据先发送高位（MSB 优先）。	
3	dir	RW	如果该域的值为 1，则表示 TX，即发送。在此模式下，RX-FIFO 将不会接收数据。 如果该域的值为 0，则表示 RX，即接收。在此模式下，RX-FIFO 将会接收数据： 如果 proto 域配置的是 Dual 或者 Quad-SPI 协议，则所有的 DQ 数据线均处于接受数据的输入状态。 如果 proto 域配置的是 Single-SPI 协议，则根据普通 SPI 协议，DQ0（MOSI）仍然是会进行输出，DQ1（MISO）会作为输入接收数据。	0x0
16:19	len	RW	该域的值指定发送一帧数据的比特位数（长度值），有效的长度值范围为 0 到 8。	0x1

9.2.10 SPI_STATUS

寄存器 offset：0x07C

寄存器描述：用于指示当前的传输状态。

Bits	Name	R/W	Description	Default
0	tip	RO	该域的值用于指示当前的传输状态。	0x0

9.2.11 SPI_TXDATA

寄存器 offset：0x048

寄存器描述：在 FIFO 发送接收模式下，可以通过 SPI_TXDATA 寄存器发送数据。

Bits	Name	R/W	Description	Default
0:7	txdata	RO	/	0x0
31	full	RO	该位为只读域，用于表示 SPI TX-FIFO 的状态是否为满 如果 full 位为 1，则表示当前 SPI-TX-FIFO 已经状态为满，写入 txdata 域的数据将被忽略；反之，则为非满，写入 txdata 域的数据将被接收	0x0

9.2.12 SPI_RXDATA

寄存器 offset：0x04C

寄存器描述：在 FIFO 发送接收模式下，可以通过 SPI_RXDATA 寄存器接收数据。

Bits	Name	R/W	Description	Default
0:7	rxdata	RO	如果 empty 域为 0 时，软件读出 txdata 域的数据为有效数据 如果 empty 域为 1 时，软件读出 txdata 域的数据为无效数据	0x0
31	empty	RO	该位为只读域，用于表示 SPI RX-FIFO 的状态是否为满 如果 full 位为 1，则表示当前 SPI-TX-FIFO 已经状态为满，写入 rxdata 域的数据将被忽略；反之，则为非满，写入 rxdata 域的数据将被接收	0x0

9.2.13 SPI_FCTRL

寄存器 offset：0x060

寄存器描述：通过 SPI_FCTRL 寄存器使能 QSPI0 的 Flash Xip 模式。

Bits	Name	R/W	Description	Default
0	flash_en	RW	如果该域为 1，则表示使能 QSPI0 的 Flash XiP 模式。 如果该域为 0，则表示不使能 QSPI0 的 Flash XiP 模式，QSPI0 处于普通的 FIFO 发送接收模式。	0x0
1	flash_wen	RW	如果该域为 1，则表示使能 QSPI0 的 Flash XiP 写模式。 如果该域为 0，则表示不使能 QSPI0 的 Flash XiP 写模式，QSPI0 的 Flash XiP 模式只能使用读模式。	0x0
3	flash_burst_en	RW	如果该域为 1，则表示使能 QSPI0 的 Flash XiP 的 burst 模式。 如果该域为 0，则表示不使能 QSPI0 的 Flash XiP 的 burst 模式，只支持 Flash XiP 的 single 模式。	0x1

9.2.14 SPI_FFMT

寄存器 offset：0x064

寄存器描述：当 QSPI0 处于 Flash Xip 模式时，整个 QSPI0（外接 Flash）被映射为一篇只读的地址区间，从而被直接读取。全歼直接从此区间读数据或者取指令会自动触发 QSPI0 通过 SPI 协议读取外部的 Flash。QSPI0 通过 SPI 接口读取外部 Flash 的具体 SPI 协议行为受 SPI_FFMT 寄存器控制。

Bits	Name	R/W	Description	Default
0	cmd_en	RW	是否发送命令（Command）的使能。	0x1
1:3	addr_len	RW	地址位由多少个字节组成（0 至 4）。默认为 3 个字节（即 24 位）。	0x3
4:7	pad_cnt	RW	发送多少个 Dummy 读 Cycles。	0x0
8:9	cmd_proto	RW	发送命令（Command）阶段使用的 SPI 协议，参见 SPI_FMT 寄存器的 proto 域的定义。	0x0
10:11	addr_proto	RW	发送地址（Address）阶段使用的 SPI 协议，参见 SPI_FMT 寄存器的 proto 域的定义。	0x0
12:13	data_proto	RW	发送数据（Data）阶段使用的 SPI 协议，参见 SPI_FMT 寄存器的 proto 域的定义。	0x0
14	data_endian	RW	如果该域的值为 1，则对数据先发送低位（LSB 优先）。 如果该域的值为 0，则对数据先发送高位（MSB 优先）。	0x0
16:23	cmd_code	RW	具体的命令（Command）值。默认值 0x3。 是常用的 Winbond/Numonx Flash 串行 READ 命令（0x03）。	0x3
24:31	pad_code	RW	在 Dummay Cycles 中发送的头 8 个比特位。	0x0

9.2.15 SPI_FFMT1

寄存器 offset：0x078

寄存器描述：XIP 传输参数控制寄存器 1（xip）。

Bits	Name	R/W	Description	Default
0:7	wcmd_code	RW	Flash XiP 模式具体的写命令（Command）值。	0x2
8:11	wpad_cnt	RW	发送多少个 Dummy 写 Cycles	0x0

9.2.16 SPI_RXEDGE

寄存器 offset：0x080

寄存器描述：SPI 接收数据采样沿控制寄存器。

Bits	Name	R/W	Description	Default
0	rxedge	RW	Rx 接收沿控制寄存器，0 表示在 SCK 上升沿时刻接收数据，1 表示在 SCK 下降沿时刻接收数据。	0x0

9.2.17 SPI_TXMARK

寄存器 offset：0x050

寄存器描述：SPI 发送 FIFO 水位寄存器。

Bits	Name	R/W	Description	Default
0:2	txmark	RW	该域的值表示 TX-FIFO 产生中断的阈值 （Watermark）。	0x0

9.2.18 SPI_RXMARK

寄存器 offset：0x054

寄存器描述：SPI 接收 FIFO 水位寄存器。

Bits	Name	R/W	Description	Default
0:2	rxmark	RW	该域的值表示 RX-FIFO 产生中断的阈值 （Watermark）。	0x0

9.2.19 SPI_IE

寄存器 offset：0x070

寄存器描述：SPI 中断使能寄存器。

Bits	Name	R/W	Description	Default
0	txie	RW	如果 txie 域为 1，则表示使能 SPI 的发送中断。 如果 txie 域为 0，则表示不使能 SPI 的发送中断。	0x0
1	rxie	RW	如果 rxie 域为 1，则表示使能 SPI 的接收中断。 如果 rxie 域为 0，则表示不使能 SPI 的接收中断。	0x0

9.2.20 SPI_IP

寄存器 offset：0x070

寄存器描述：SPI 中断使能寄存器。

Bits	Name	R/W	Description	Default
0	txip	RW	如果该域为 1，则表示当前正在产生发送中断。 如果该域为 0，则表示当前没有产生发送中断。	0x0
1	rxip	RW	如果该域为 1，则表示当前正在产生接收中断。 如果该域为 0，则表示当前没有产生接收中断。	0x0

9.3 UART

UART 全称为 Universal Asynchronous Receiver-Transmitter（通用异步接收-发射器），提供了一个灵活方便的串行数据交换接口，数据帧可以通过异步的方式进行传输。 UART 提供了可编程的波特率发生器，默认波特率 38400bps，能分频产生 UART 发送和接收所需的特定频率。本 SoC 有两个独立的 UART，两个 UART 均复用 GPIO 的顶层引脚与外界通信。

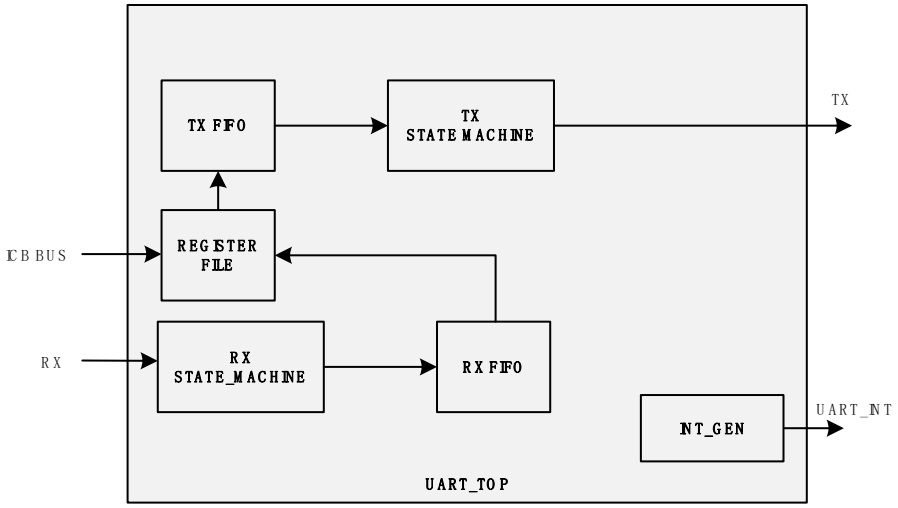

图 9-3 UART 数据流

9.3.1 UART 特性

- 全双工异步通信
- 发送器或接收器的单独使能位
- 波特率可配置

- 可配置的奇偶校验生成（奇/偶）和检查
- 可配置的数据字宽（5/6/7/8/9）
- 可配置停止位（支持 0.5/1/1.5/2 停止位）
- RX 和 TX 的 FIFO 大小可在 RTL 级别配置

9.3.2 UART 中断请求

Interrupt Source	Interrupt Flag	Interrupt Control bit
tx fifo water mark	tx_ip	tx_ie
rx fifo water mark	rx_ip	rx_ie
rx fifo overflow detect	rx_error_overflow	overflow_error_en
parity checking error	rx_error_parity	parity_error_en

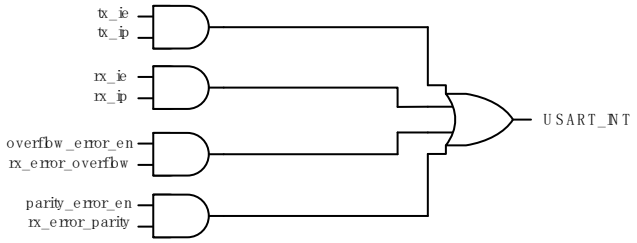

图 9-4 Uart 中断映射

9.3.3 存储器映射

UART 控制器的存储器映射如表 9-3 所示，UART 存储器映射仅支持自然对其的 32 位内存访问。

表 9-3 UART 寄存器映射关系表

Register Offset	Register Name	Description
0x00	UART_TXDATA	发送数据寄存器
0x04	UART_RXDATA	接收数据寄存器
0x08	UART_TXCTRL	发送控制寄存器
0x0c	UART_RXCTRL	接收控制寄存器
0x10	UART_IE	UART 中断使能寄存器
0x14	UART_IP	UART 中断等待寄存器
0x18	UART_DIV	波特率除数寄存器
0x1c	UART_STATUS	RX/TX busy 状态寄存器
0x20	UART_SETUP	UART setup 寄存器
0x24	UART_ERROR	UART 接收错误状态寄存器
0x28	UART_IRQ_EN	UART 中断请求使能寄存器

9.3.4 UART_TXDATA

寄存器 offset：0x00

寄存器描述：发送数据寄存器。如果 FIFO 可以接收新数据，则写入 txdata 寄存器会将数字段中包含的字符依次写入到发送 FIFO 中；从 txdata 寄存器读数返回当前的满标志和数据域；满标志表明发送 FIFO 是否可以接收新的数据，当满标志有效之后，再写入的数据会被忽略。

Bits	Name	R/W	Description	Default
0:8	txdata	RW	发送数据	0x0
9:30	/	/	保留	/
31	full	RW	发送 FIFO 满	0x0

9.3.5 UART_RXDATA

寄存器 offset：0x04

寄存器描述：接收数据寄存器。读 rxdata 寄存器时返回数据域的值。空标志表明接收 FIFO 是否为空，如果有效，则数据域中没有有效数据。

Bits	Name	R/W	Description	Default
0:8	rxdata	RW	发送数据	0x0
9:30	/	/	保留	/
31	full	RW	接收数据空	0x0

9.3.6 UART_TXCTRL

寄存器 offset：0x08

寄存器描述：发送控制寄存器。读写 txctrl 寄存器控制发送通道的操作。txen 位控制 tx 通道是否处于有效状态。清零时，将禁止发送 tx FIFO 中的数据，并将 txd 引脚驱动为高电平。

Bits	Name	R/W	Description	Default
0	txen	RW	发送使能	0x0
1:2	nstop	RW	停止位配置： 00：1bit 01：2bit 10：0.5bit 11：1.5bit	0x0
3:15	/	/	保留	/
16:19	txcnt	RW	发送水印级别	0x0
20:31	/	/	保留	/

9.3.7 UART_RXCTRL

寄存器 offset：0x0C

寄存器描述：接收控制寄存器。读写 rxctrl 寄存器控制接收通道的操作。rxen 位控制 rx channel 是否有效。当清零时，rxd 引脚的状态会被忽略，数据不会写入 rx FIFO 中。

Bits	Name	R/W	Description	Default
0	txen	RW	发送使能	0x0
1:15	/	/	保留	/
16:19	rxcnt	RW	接收水印级别	0x0
20:31	/	/	保留	/

9.3.8 UART_IE

寄存器 offset：0x10

寄存器描述：UART 中断使能寄存器。

Bits	Name	R/W	Description	Default
0	txie	RW	发送水印中断使能	0x0
1	rxie	RW	接收水印中断使能	0x0
2:31	/	/	保留	/

9.3.9 UART_IP

寄存器 offset：0x14

寄存器描述：UART 中断等待寄存器。

Bits	Name	R/W	Description	Default
0	txip	RW	发送水印中断等待	0x0
1	rxip	RW	接收水印中断等待	0x0
2:31	/	/	保留	/

9.3.10 UART_DIV

寄存器 offset：0x18

寄存器描述：波特率除数寄存器。通过读写 baud div 寄存器指定 tx 和 rx 通道中用于产生波特率的除数。输入时钟和波特率之间可通过如下公式进行换算：$f_{baud} = f_{clk\_in}/(div+1)$。

Bits	Name	R/W	Description	Default
0:15	baud div	RW	波特率除数	0x21e
16:31	/	/	保留	/

9.3.11 UART_STATUS

寄存器 offset：0x1C

寄存器描述：Uart 状态寄存器。

Bits	Name	R/W	Description	Default
0	rx_busy	RW	正在接收数据	0x0
1	tx_busy	RW	正在发送数据	0x0
2:31	/	/	保留	/

9.3.12 UART_SETUP

寄存器 offset：0x20

寄存器描述：Uart 建立寄存器。

Bits	Name	R/W	Description	Default
0	parity_en	RW	奇偶校验位生成和检查配置域： 0：禁止 1：启用	0x0
1	parity_sel	RW	奇偶校验模式选择： 0：偶校验 1：奇校验	0x0
2	/	/	保留	/
3	clean_fifo	RW	清零 rx FIFO，设置为 0/1 以实现 FIFO 复位： 0：停止清零 RX FIFO。 1：清零 RX FIFO。	0x0
4:6	bit_length	RW	字符长度位域： 0x0：5 bits 0x1：6 bits 0x2：7 bits 0x3：8 bits 0x4：9 bits	0x0
7:31	/	/	保留	/

9.3.13 UART_ERROR

寄存器 offset：0x24

寄存器描述：Uart 接收错误状态寄存器。错误寄存器为只读寄存器，错误状态由硬件设置。软件可通过发送读命令得到 rx fifo 溢出和奇偶校验结果。如果发生溢出或奇偶校验错误，软件可以通过再次发送读命令清掉错误状态。

Bits	Name	R/W	Description	Default
0	rx_error_overflow	RO	RX FIFO 溢出错误标志： 0：正确 1：发生 RX FIFO 溢出错误	0x0
1	rx_error_parity	RO	RX 奇偶校验错误标志： 0：没错	0x0

			1：发生 RX 奇偶校验错误	
2:31	/	/	保留	/

9.3.14 UART_IRQ_EN

寄存器 offset：0x28

寄存器描述：Uart 中断请求使能寄存器。

Bits	Name	R/W	Description	Default
0	rx_error_overflow	RW	RX FIFO 溢出错误标志： 0：正确 1：发生 RX FIFO 溢出错误	0x0
1	rx_error_parity	RW	RX 奇偶校验错误标志： 0：没错 1：发生 RX 奇偶校验错误	0x0
2:31	/	/	保留	/

9.4 DMA

DMA 控制器是一种硬件方法，用于在外设/内存和内存之间直接进行数据传输而无需 CPU 干预，旨在从需要的地方转移大量数据块的负载。

9.4.1 功能简介

- 两个 SICB（标准 ICB）用于数据传输，一个 FSICB（简化版 ICB）用于 DMA 配置；
- 支持 1-16 beat 递增突发类型的内存访问；
- 支持可配置的数据宽度 8、16、32、64、128 位用于内存访问；
- 支持固定或增加地址生成算法；
- 支持连续或重复可配置 N 次传输方式；
- 支持独立的源和目的基地址设置，字节，半字或字对齐；
- 支持（1M-1）字节以内的任意传输块大小；
- 支持中断产生，屏蔽和清除。

9.4.2 寄存器映射

下表列出了 DMA 控制器部分的整体存储器映射（偏移），可以从相应的外设文档中找到不同外设的 DMA 相关寄存器定义（基地址，传输大小，启用等）。 所有的寄存器访问都是 32 位字对齐的。

DMA 控制器部分的寄存器映射（偏移）关系如下表所示：

Register Offset	Register Name	Description
0x008+0x14	DMA_CFG_MSRCADDR	源数据基址
0x00C+0x14	DMA_CFG_MDSTADDR	目的地址基址
0x010+0x14	DMA_CFG_MCTRL	控制寄存器
0x014+0x14	DMA_CFG_RPT	传输重复次数
0x018+0x14	DMA_CFG_MSIZE	传输大小
0x100+0xC	DMA_CHX_IRQ_EN	M2M 中断使能
0x104+0xC	DMA_CHX_IRQ_STAT	M2M 中断状态状态
0x108+0xC	DMA_CHX_IRQ_CLR	M2M 中断清除

9.4.3 DMA_CFG_MSRCADDR

寄存器 offset：0x008+0x14

寄存器描述：源地址基址寄存器。

Bits	Name	R/W	Description	Default
0:31	src_base	RW	源数据块的基地址	0x0

9.4.4 DMA_CFG_MDSTADDR

寄存器 offset：0x00C+0x14

寄存器描述：目的数据基址寄存器。

Bits	Name	R/W	Description	Default
0:31	dst_base	RW	目的数据块的基地址	0x0

9.4.5 DMA_CFG_MCTRL

寄存器 offset：0x010+0x14

寄存器描述：控制寄存器。

Bits	Name	R/W	Description	Default
0	trans_en	RW	DMA 传输使能，断言该位将启动 mem2mem DMA 传输，可以通过 SW 将其清除，并且在当前突发结束后传输将停止。 当完成一次完整的传输或发生传输错误响应时，硬件也可以将其清除，在这种情况下，当前传输将被中止。 注意，仅当所有其他配置位都已正确配置时，才可以设置该位。	0x0
1:2	trans_type	RW	mem2mem 固定为 2’b00	0x0

3:5	trans_per_sel	RW	保留	0x0
6:7	trans_mode	RW	传输模式选择 2’b00：单模式传输； 2’b01：连续模式传输，在这种模式下，完成当前传输后，将以相同的传输配置自动启动新的传输； 2’b10：传输重复模式，使用相同的传输配置，传输将连续 N 次（在“传输重复编号寄存器”中定义） 2’b11：保留	0x0
8:9	priority	RW	保留	0x0
12	mdna	RW	下一地址生成算法，用于将数据传输到目标存储器 1’b0：增加地址模式 1’b1：固定地址 如果配置了固定地址，则强制将起始目标地址对齐	0x0
13	msna	RW	用于从源存储器中获取数据的下一个地址生成算法 1’b0：增加地址模式 1’b1：固定地址 如果配置了固定地址，则强制将起始源地址对齐	0x0
16:18	mdwidth	RW	用于将数据传输到目标的传输宽度： 3’b000：8 位 3’b001：16 位 3’b010：32 位 3’b011：64 位 3’b100：128 位 其他：保留	0x0
21:23	mswidth	RW	从源获取的传输宽度 3’b000：8 位 3’b001：16 位 3’b010：32 位 3’b011：64 位 3’b100：128 位 其他：保留	0x0
24:27	mdburst	RW	突发中用于将数据传输到目标存储器的传输次数 4’b0000：1 次转移 4’b0001：2 次传输 4’b0010：3 次转移 4’b0011：4 次转移 4’b0100：5 次转帐 4’b0101：6 次转移 4’b0110：7 次传输 … 4’b1111：16 次传输	0x0
28:31	msburst	RW	突发中用于将数据传输到目标存储器的传输次数 4’b0000：1 次转移 4’b0001：2 次传输 4’b0010：3 次转移 4’b0011：4 次转移	0x0

			4’b0100：5 次转帐 4’b0101：6 次转移 4’b0110：7 次转移 … 4’b1111：16 次传输	

9.4.6 DMA_CFG_RPT

寄存器 offset：0x014+0x14

寄存器描述：传输重复次数寄存器。

Bits	Name	R/W	Description	Default
0:11	trans_rpt	RW	当 DMA_CFG_MCTRL 寄存器的 trans_mode 配置成 b10 传输重复模式时，此域定义重复循环次数。	0x0

9.4.7 DMA_CFG_MSIZE

寄存器 offset：0x018+0x14

寄存器描述：传输大小寄存器。

Bits	Name	R/W	Description	Default
0:19	tsize	RW	一次 DMA 传输的传输大小。 不允许 0 开始传输，这样做可能会导致意外结果 在 DMA 传输期间，这些位指示要传输的剩余字节。 如果 DMA_CFG_MCTRL 寄存器的 trans_mode 配置为 b01 或 b10（即连续或重复模式），则当当前传输完成时，它将自动重新加载到原始值以进行新的传输；dma 传输错误可能会使寄存器冻结到上一次成功传输的值，但是它将重新开始并自动记录下一次新传输的正确剩余数据号。	0x0

9.4.8 DMA_CHX_IRQ_EN

寄存器 offset：0x100+0xC

寄存器描述：中断使能寄存器。

Bits	Name	R/W	Description	Default
0	ftrans_irq_en	RW	满输出的中断使能。	0x0
1	htrans_irq_en	RW	半传输中断使能。	0x0
2	rsp_err_irq_en	RW	dma 访问错中断使能。	0x0
3:31	/	/	保留	/

9.4.9 DMA_CHX_IRQ_STAT

寄存器 offset：0x104+0xC

寄存器描述：中断状态寄存器。

Bits	Name	R/W	Description	Default
0	ftrans_irq_stat	RO	满传输的中断状态标志。	0x0
1	htrans_irq_stat	RO	半传输的中断状态标志。	0x0
2	rsp_err_irq_stat	RO	dma 访问错误的中断状态标志。	0x0
3:31	/	/	保留。	/

9.4.10 DMA_CHX_IRQ_CLR

寄存器 offset：0x108+0xC

寄存器描述：中断清除寄存器。

Bits	Name	R/W	Description	Default
0	ftrans_irq_clr	WO	满传输的中断状态标志。	0x0
1	htrans_irq_clr	WO	半传输的中断状态标志。	0x0
2	rsp_err_irq_clr	WO	dma 访问错误的中断状态标志。	0x0
3:31	/	/	保留。	/

9.4.11 访问 switch 寄存器

若要将 DMA 当做访问 switch 寄存器的 table DMA，需要先配置好 switch 侧的寄存器，然后再开启 DMA。switch 侧的寄存器有三个，分别是告警、配置、启动。告警寄存器是告警信号的标志位，配置寄存器用于配置 DMA 传输相关的参数，启动寄存器用于 DMA 开启。

Switch 侧告警寄存器

寄存器偏移		0x50 + TOP_CFG_REG + 0x6000_0000（TOP_CFG_REG =0）		
位域	域名	读写特性	复位值	描述
[1:0]	axi_cp_alm	RO		1 写错误，0 读错误

switch 侧配置寄存器

寄存器偏移		0x54 + TOP_CFG_REG + 0x6000_0000（TOP_CFG_REG =0）		
位域	域名	读写特性	复位值	描述
[5：0]	seg_limit	WO		表项实际宽度(=实际值-1)
[11:6]	seg_num	WO		表项地址宽度(2^n-1，如 1，3，7，15 等）。e.g：seg_limit=69，则 seg_num=127。
[12]	tx_cnt_endian_swap	WO		发送到寄存器的数据，大小端调换
[13]	rx_cnt_endian_swap	WO		寄存器发送到上端的数据，大小端调换

[31:14]	reg_dma_count	RO		只读寄存器， start 以后，操作了多少个寄存器

switch 侧启动寄存器

寄存器偏移		0x58 + TOP_CFG_REG + 0x6000_0000（TOP_CFG_REG =0）		
位域	域名	读写特性	复位值	描述
[0]	reg_dma_start	WO		写 1 表示 dma 开始

配置操作流程：

1. 配置好寄存器（0x54），然后配置 reg_dma_start（0x58）为 1
2. 配置并开启 DMA，开始传输数据。

9.5 I2C

I2C 寄存器映射关系如下表所示：

Register Offset	Register Name	Description
0x00	I2C_PRERlo	预分频寄存器低 8 字节
0x01	I2C_PRERhi	预分频寄存器高 8 字节
0x02	I2C_CTR	控制寄存器
0x03	I2C_TXR	发送寄存器
0x05	I2C_RXR	接收寄存器
0x06	I2C_CR	命令寄存器
0x04	I2C_SR	状态寄存器
0x07	I2C_TRISE	scl 延迟寄存器
0x08	I2C_FLTER	滤波寄存器

9.5.1 I2C_PRERlo

寄存器 offset：0x00

寄存器描述：预分频 SCL 时钟线。由于 I2C 接口的结构，内核在内部使用 4 * SCL 时钟。预分频寄存器必须设置为 4 * SCL 比特率。仅当清除“EN”位时，才更改预分频寄存器的值。

Bits	Name	R/W	Description	Default
0:7	prerlo	RW	预分频寄存器低字节。	0xffff

9.5.2 I2C_PRERhi

寄存器 offset：0x01

寄存器描述：预分频 SCL 时钟线。由于 I2C 接口的结构，内核在内部使用 4 * SCL 时钟。预分频寄存器必须设置为 4 * SCL 比特率。仅当清除“EN”位时，才更改预分频寄存器的值。

Bits	Name	R/W	Description	Default
0:7	prerhi	RW	预分频寄存器高字节。	0xffff

9.5.3 I2C_CTR

寄存器 offset：0x02

寄存器描述：控制寄存器。内核仅在将“EN”位置于 1 时，才响应新命令。待处理命令已完成。 仅当没有传输正在进行时（即在 STOP 命令之后）或命令寄存器的 STO 位置 1 时，才清除“ EN”位。 在传输过程中暂停时，内核可以挂起 I2C 总线。

Bits	Name	R/W	Description	Default
0:5	/	RW	保留。	/
6	ctr_en	RW	I2C 内核使能位： 1：使能 0：禁止	0x00
7	ctr_in	RW	I2C 内核中断使能位： 1：使能 0：禁止	0x00

9.5.4 I2C_TXR

寄存器 offset：0x03

寄存器描述：发送寄存器。

Bits	Name	R/W	Description	Default
0	txr	WO	下一个要传输的字节	0x00
1:7	txr_data	WO	进行数据传输时，该位代表数据的 LSB。 如果是从机地址传输，则该位代表 RW 位。 1:读 slave 0:写 slave	0x00

9.5.5 I2C_RXR

寄存器 offset：0x05

寄存器描述：接收寄存器。

Bits	Name	R/W	Description	Default
0:7	rxr	RO	接收到的最后一个字节	0x00

9.5.6 I2C_CR

寄存器 offset：0x06

寄存器描述：命令寄存器。

Bits	Name	R/W	Description	Default
0	lack	RW	中断响应，置位时，清除挂起的中断。	0x00
1:2	/	RW	保留	/
3	ack	RW	接收方发送 ACK（ACK ='0'）或 NACK（ACK ='1'）	0x00
4	wr	RW	写 slave	0x00
5	rd	RW	读 slave	0x00
6	sto	RW	生成停止条件	0x00
7	sta	RW	生成（重复）开始条件	0x00

9.5.7 I2C_SR

寄存器 offset：0x04

寄存器描述：状态寄存器。

Bits	Name	R/W	Description	Default
0	if	RO	中断标志。 当中断待处理时，该位置 1，如果 IEN 位置 1，将引起处理器中断请求。在以下情况下设置中断标志：完成一个字节传输。	0x00
1	tip	RO	正在进行传输。 1：正在传输 0：传输完成	
2:5	/	RO	保留	/
6	busy	RO	写 slave I2C 总线 busy 信号。 检测到 START 信号后为“ 1” 检测到 STOP 信号后为“ 0”	0x00
7	rxack	RO	收到从机的响应。 该标志表示被寻址的 slave 的响应。 1：未收到响应 0：收到响应	0x00

9.5.8 I2C_TRISE

寄存器 offset：0x07

寄存器描述：延迟寄存器，用于屏蔽 slave 等待时间带来的分频误差。

Bits	Name	R/W	Description	Default
0:7	trise	RW	用于屏蔽 slave 等待时间带来的分频误差。	0x00

9.5.9 I2C_FLTER

寄存器 offset：0x08

寄存器描述：滤波寄存器，用于调整配置滤波频率大小。

Bits	Name	R/W	Description	Default
0:4	flter	RW	用于调整配置滤波频率大小。	0x00

注：

1. 寄存器之间应满足如下关系：

I2C_FLTR < I2C_PRER;

I2C_TRISE > I2C_FLTR*2 + 3;

(scl 频率) scl = (I2C_PRER * (5 + 1) + I2C_TRISE) (主频)

2. 发送寄存器和命令寄存器都映射到地址 0x02。发送寄存器是 MSB，命令寄存器是 LSB。

3. 接收寄存器和状态寄存器都映射到地址 0x03。接收寄存器是 MSB，状态寄存器是 LSB。

10 修订信息

修订时间	版本	描述
2022.12.23	**V1.0**	**初始版本。**