FSL91030(M) chip

SoC User Manual

Manual version: V1.0

Wuhan Feisiling Microelectronics Technology Co., Ltd.

December 2022

Table of contents

1. Function Overview

The SoC functions of the FSL91030M chip are summarized as follows:

ÿ Use Nuclei core, a 64-bit reduced instruction set based on RISC-V architecture.

ÿ Equipped with on-chip SRAM as LM:

ÿ 64KB tightly coupled instruction memory

ÿ 64KB tightly coupled data memory

ÿ Equipped with customized SoC bus.

ÿ Equipped with various peripheral IPs

ÿ 2 asynchronous serial transceiver interfaces

ÿ One I2C interface

ÿ General GPIO: Supports 24 general input and output.

ÿ Dedicated reset:

ÿ 1 System reset interface;

ÿ 1 POR reset interface;

ÿ QSPI Flash interface

ÿ Support SPI FLASH boot via XIP

1.1 Functional Block Diagram

The block diagram of the FSL91030M SoC is shown in Figure 1-1. In addition to the RSIC V core and bus, the figure also includes related peripherals. For detailed description of each peripheral, refer to Chapter 9.

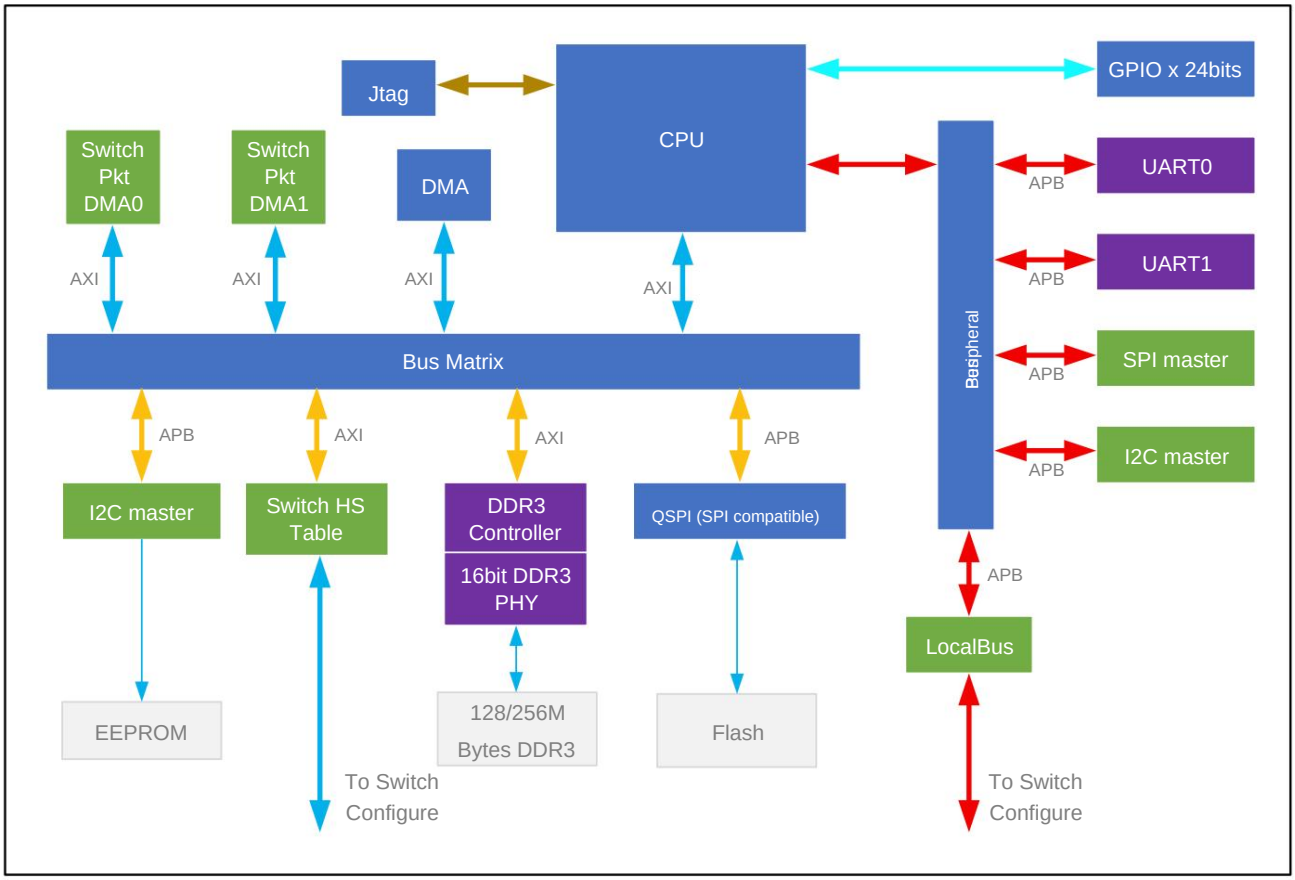

Figure 1-1 SoC block diagram

The clock of SoC is shown in Figure 1-2.

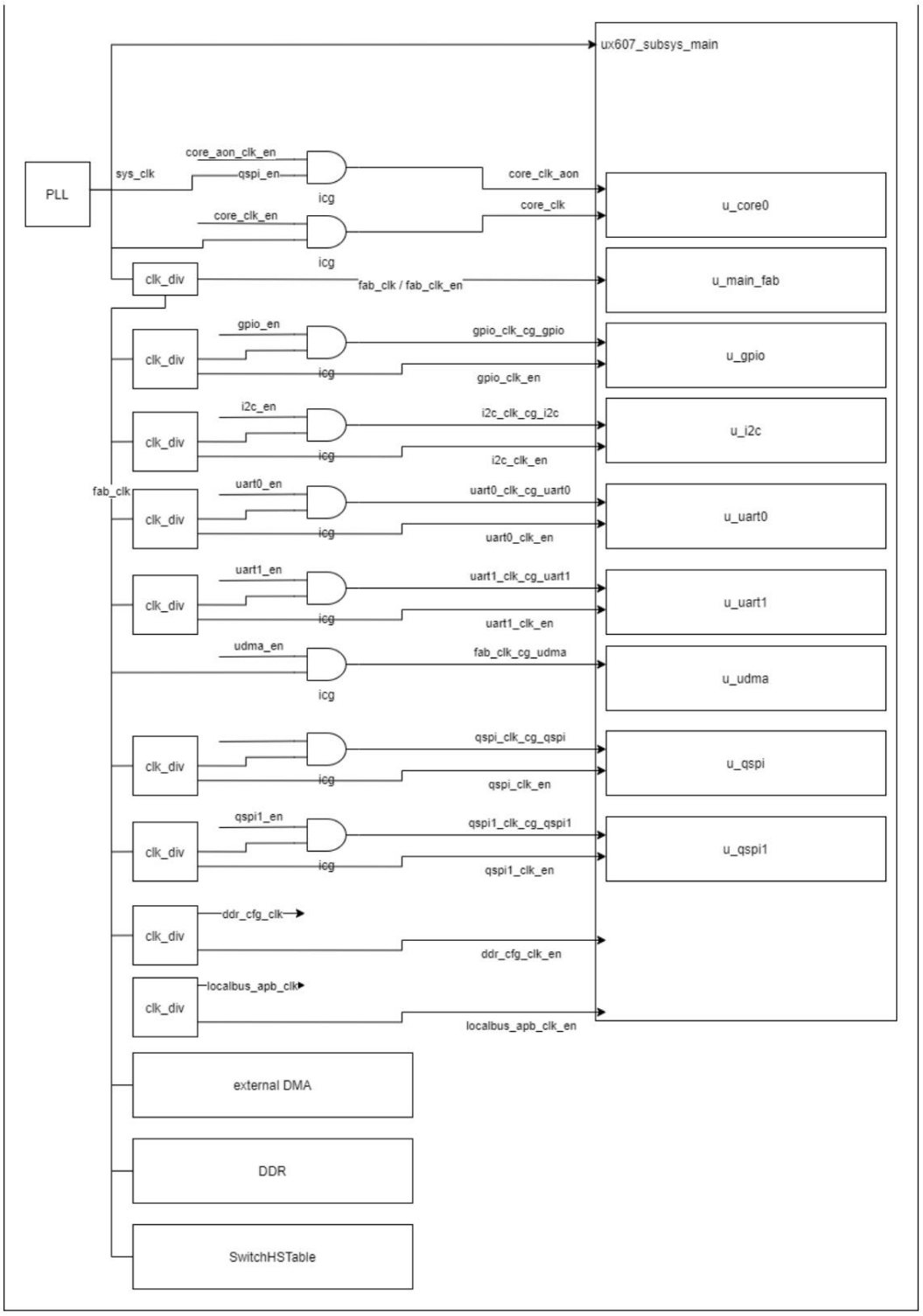

Figure 1-2 SoC clock

The frequency division unit used is shown in Figure 1-3.

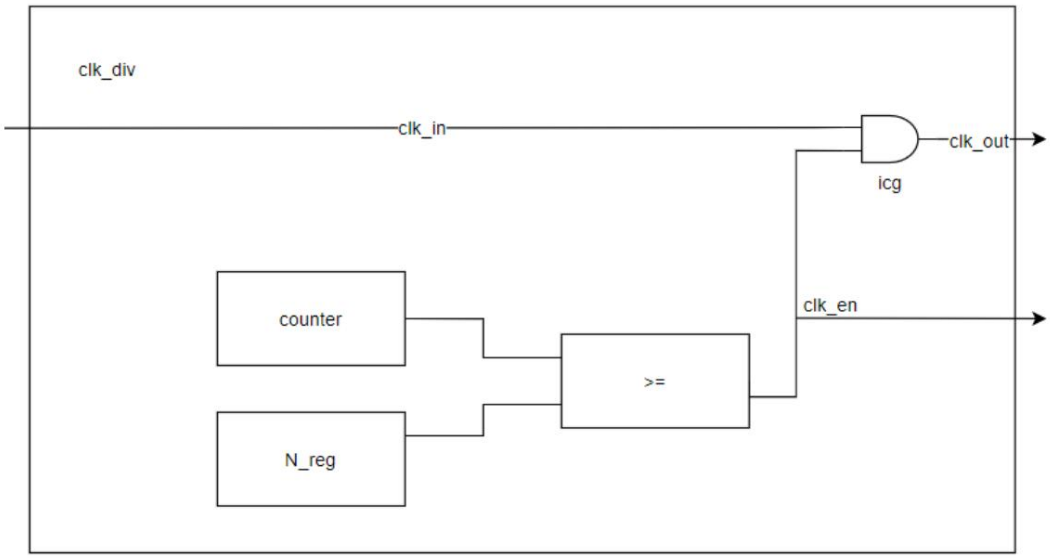

Figure 1-3 Frequency division unit used

The SoC reset is shown in Figure 1-4.

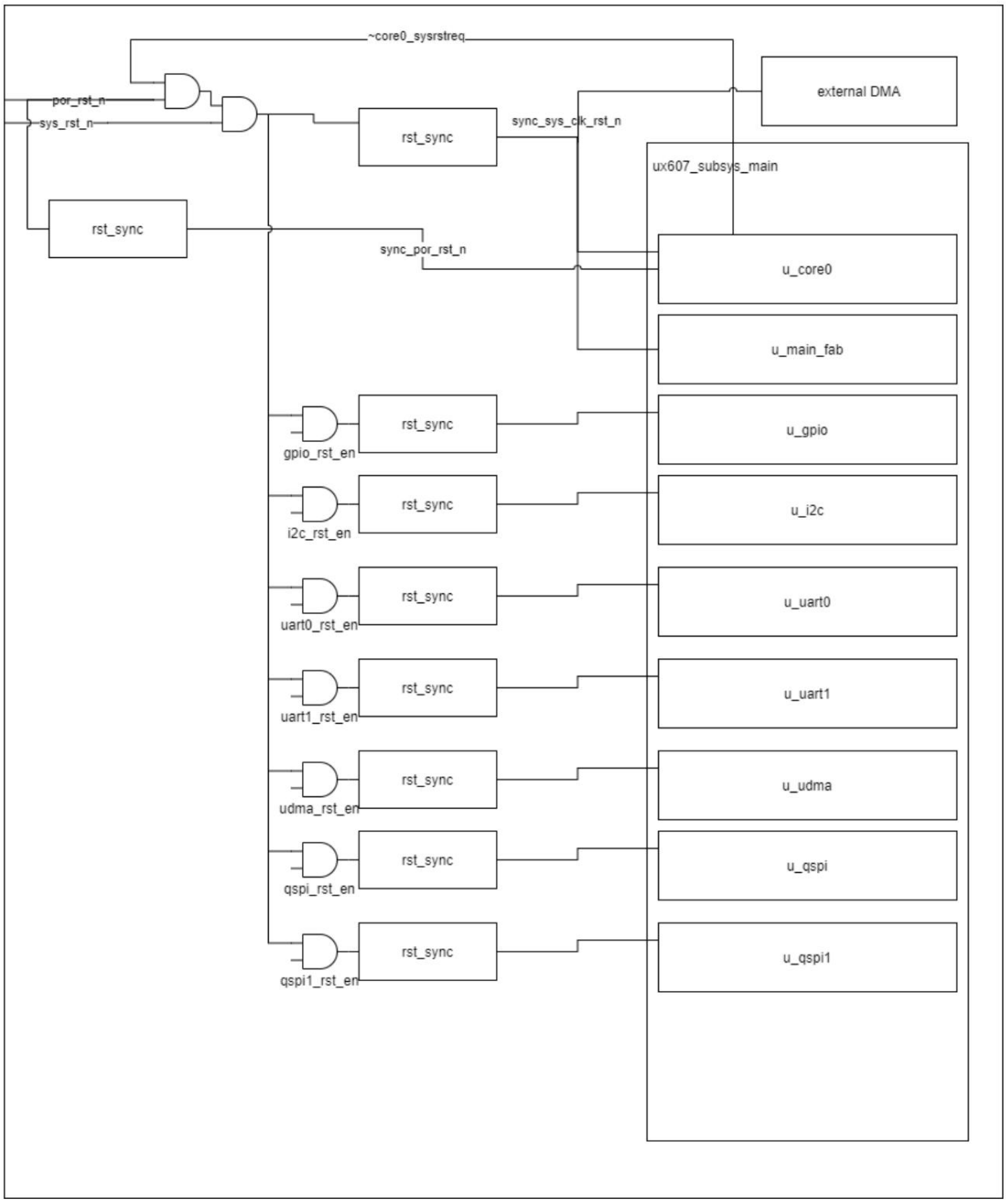

Figure 1-4 SoC reset

1.2 Bus Characteristics

The SoC of the FSL91030M chip defines a custom bus protocol ICB (Internal Chip Bus). The original intention of the ICB bus is
In order to combine the advantages of AXI bus and AHB bus as much as possible, and to have both high speed and ease of use, it has the following characteristics:

ÿ Compared with AXI and AHB, ICB's protocol control is simpler, with only two independent channels, as shown in Figure 1-5. Read and write operations share the address channel and the result return channel.

ÿ Uses separate address and data phases like the AXI bus.

ÿ Like the AXI bus, it uses address range addressing and supports any number of masters and slaves, such as one master and one slave, one master and multiple slaves, multiple masters and one slave, multiple masters and multiple slaves, etc.

ÿ Like the AHB bus, each read or write operation generates an address on the address channel, instead of just generating a start address like in AXI.

ÿ Like the AXI bus, it supports address non-aligned data access and uses a byte mask (Write Mask) to control some write operations.

ÿ Like the AXI bus, it supports multiple outstanding transactions.

ÿ Like the AHB bus, it does not support out-of-order return and out-of-order completion. The feedback channel must return the results in order.

ÿ As with the AXI bus, it is very easy to add pipeline stages to obtain high-frequency timing.

ÿ The protocol is very simple and can be easily bridged to other bus types, such as AXI, AHB, APB or TileLink.

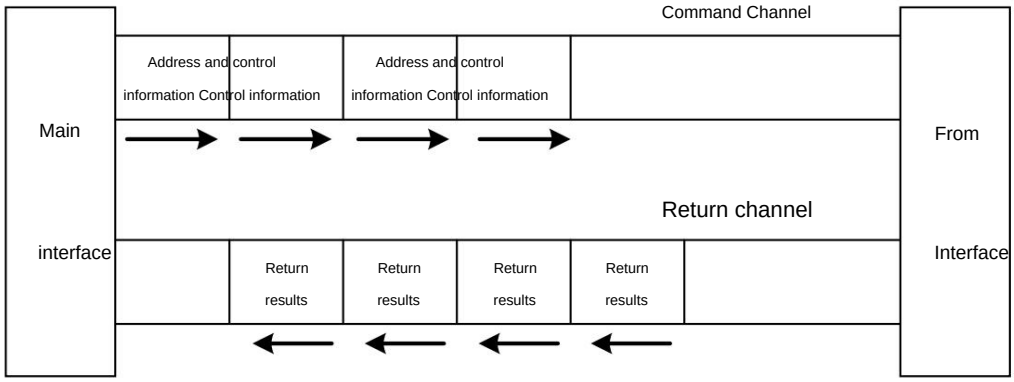

Figure 1-5 ICB bus diagram

2 Introduction to RISC V Processor Core

The features of the RISC V core are listed below:

ÿ CPU Core

ÿ 6-stage variable-length pipeline architecture adopts first-class processor architecture design to achieve industry-leading energy efficiency and comprehensive cost.

ÿ Support dynamic branch prediction.

ÿ Configurable instruction prefetch unit that can prefetch two instructions in sequence, thereby hiding the memory access latency of instructions.

ÿ Supports machine mode, configurable supervisor mode and configurable user Mode (User Mode).

ÿ Supports non-aligned access to memory.

ÿ Two-level nested exception recovery.

ÿ Support hardware single-cycle multiplier.

ÿ Support hardware multi-cycle divider.

ÿ Support instruction set architecture (ISA)

The processor core supports 64-bit RISC-V instruction set architecture and RV64I/M/A/C

IMAC is a combination of command subsets that are fixedly supported.

ÿ Bus interface

ÿ Supports 64-bit standard AXI system bus interface for accessing external instructions and data.

ÿ Supports 64-bit wide instruction local memory (ILM) bus interface (supports SRAM Interface protocol) for connecting to private instruction local memory.

ÿ Supports 64-bit wide Data Local Memory (DLM) bus interface (supports SRAM interface protocol) for connecting private data local memory.

ÿ Supports 64-bit wide private peripheral interface (PPI) and standard APB interface Port protocol used to connect private peripherals.

ÿ Support debugging function

ÿ Support standard four-wire JTAG interface.

ÿ Support RISC-V debugging standard.

ÿ Supports a configurable number of hardware breakpoints.

ÿ Supports configurable debugger connection timeout function.

ÿ Support mature interactive debugging tools.

ÿ Support low power consumption management

ÿ Support WFI (Wait For Interrupt) and WFE (Wait For Event) to enter sleep mode.

ÿ Support two levels of sleep mode: light sleep and deep sleep.

ÿ Support Enhanced Core Level Interrupt Controller (ECLIC)

ÿ Support software interrupts, timer interrupts and external interrupts defined by the RISC-V standard.

ÿ Supports a configurable number of external interrupts.

ÿ Supports configurable number of interrupt levels and priorities, and supports software dynamic programmable modification of the number of interrupt levels and interrupt priorities.
value.

ÿ Supports interrupt nesting based on interrupt levels.

ÿ Support fast vector interrupt processing mechanism.

ÿ Support fast interruption tail biting mechanism.

ÿ Support NMI (Non-Maskable Interrupt).

3 Storage Resources

As shown in Figure 1-1, the memory resources in the FSL91030M SoC are divided into on-chip memory resources and off-chip Flash memory resources, which are introduced in this section.

3.1 On-chip storage resources

The on-chip storage resources of the FSL91030M SoC mainly include ILM, DLM and system RAM.

ÿ ILM is the instruction memory of the processor core, and its characteristics are as follows:

- ÿ Size: 64KB.
- ÿ Configurable address range (default starting address is 0x8000_0000), see Chapter 4 for the complete address allocation of the SoC.
- ÿ In this supporting SoC, although the ILM SRAM is mainly used to store instructions, its address range can also be used to store data. The processor core can access the ILM through the system memory bus of the SoC.

ÿ DLM is the data memory of the processor core, and its characteristics are as follows:

- ÿ Size: 64KB.
- ÿ Configurable address range (default start address is 0x8001_0000), see Chapter 4 for complete address allocation of SoC.
- ÿ In this supporting SoC, although DLM SRAM is mainly used to store data, its address range can also be used to store The processor core can access the DLM through the SoC's system memory bus.

ÿ System DDR RAM is the data storage of SOC bus space, and its characteristics are as follows:

- ÿ Size: 256MB.
- ÿ Configurable address range (default start address is 0x4000_0000), see Chapter 4 for complete address allocation of SoC.
- ÿ In this supporting SoC, although DLM SRAM is mainly used to store data, its address range can also be used to store instructions. The processor core can access DLM through the system memory bus of SoC.

3.2 Off-chip **Flash** storage resources

The SoC off-chip storage resources of the FSL91030M chip are mainly Flash memory, the main points of which are as follows:

- ÿ External Flash can be mapped into a read-only address area through QSPI0 using its XiP (Execution in Place) mode. The address range is 0x2000_0000 ~ 0x2FFF_FFFF.
- ÿ Users can use the debugger to burn the developed program into Flash, and then use the XiP mode of Flash to directly Executed from Flash.

3.3 On-chip **SRAM** List

3.3.1 ICache

Tag Single Port SRAM: 8 blocks

Width: 54

Depth: 128

Valid Write-Mask: 1 bit

Data Single Port SRAM: 8 blocks

Width: 64

Depth: 512

Valid 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 blocks

Width: 23

Depth: 128

Valid Write-Mask: 1 bit

Data Single Port SRAM: 4 blocks

Width: 32

Depth: 512

Valid Write-Mask: 4 bits

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 block

Width: 64

Depth: 64KB

Valid 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 blocks

Width: 32

Depth: 64KB in total

Valid Write-Mask: 4 bits

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 Address Allocation

The SoC bus address allocation of the FSL91030M chip is shown in Table 4-1.

Table 4-1 SoC address allocation table

Bus Grouping	Components	Address range	describe
Core internal private	DEBUG	0x0000_0000 ~ 0x0000_0FFF	Note: The debug module is mainly used for debugging Ordinary software programs should not use this interval
	ECLIC	0x0C00_0000 ~ 0x0C00_FFFF	ECLIC Interrupt Management Unit Address Space
	PLIC	0x0800_0000 ~ 0x0800_FFFF	PLIC Interrupt Management Unit Address Space
System Storage (MEM) Interface	ILM	0x8000_0000 ~ 0x8000_FFFF	ILM address range, the range size depends on the size of the ILM configuration Small.
	DLM	0x8001_0000 ~ 0x8001_FFFF	DLM address range, the range size depends on the size of the ILM configuration Small.
System peripherals	GPIO	0x1001_1000 ~ 0x1001_1FFF	GPIO module register address range
	UART 0	0x1001_3000 ~ 0x1001_3FFF	The first UART module register address range
	UART 1	0x1001_2000 ~ 0x1001_2FFF	Second UART module register address range
	I2C	0x1001_8000 ~ 0x1001_8FFF	I2C module register address range
	QSPI	0x1001_4000 ~ 0x1001_4FFF	XPI QSPI module register address range
	QSPI0	0x1001_6000 ~ 0x1001_6FFF	QSPI0 module register address range
	DMA	0x1001_7000 ~ 0x1001_7FFF	DMA module register address range
	Off-Chip QSPI Flash Read	0x2000_0000 ~ 0x2FFF_FFFF	When QSPI is in Flash XiP mode, the external Flash is mapped Read-only address range.
External AXI interface ground Address Space	SwitchHStable	0x5000_0000 ~ 0x5FFF_FFFF	Table DMA address range when accessing switch registers
	DDR CFG	0x1001_9000~ 0x1001_9FFF	Ddr configuration register address range
	Local Bus	0x6000_0000 ~ 0x6FFF_FFFF	Address range when directly accessing the switch register
	DDR RAM	0x4000_0000 ~ 0x4FFF_FFFF	DDR RAM address range. The range size depends on the size of the DDR RAM configuration.
Other address ranges	The address ranges not used in the table are ignored for writes and return 0 for reads.		

5 Power-on process control

After the processor core in the FSL91030M SoC is powered on and reset, different addresses can be selected for execution using the reset_vector input. Usually the address can be set as follows:

ÿ Start execution from external Flash

ÿ Since the address range of the mapped external Flash (Off-Chip QSPI0 Flash Read) is 0x2000_0000 ~ 0x2FFF_FFFF, if execution starts from the external Flash, the PC reset value of the RISC-V processor core is 0x20000000.

ÿ Users can use the debugger to burn the developed program into the Flash. Using the XiP mode of the Flash, the program can be directly Executed from Flash.

ÿ Start execution from DDR address.

ÿ Users can burn the developed program into DDR through the debugger, and the program can be executed directly from DDR.

6 Reset Processing

There are 2 reset domains in the SoC of the FSL91030M chip, including:

1. PoR reset domain: Debug module and JTAG DMI module
2. System reset domain: including other system modules except PoR reset domain

There are several sources for resetting the SoC.

(1) From the POR (Power-On-Reset) nPOR_RST chip pin.

After the chip is powered on, the external POR outputs a reset signal until the voltage reaches the stable threshold, ensuring that the chip can be automatically reset correctly. This reset source can act on the POR and system reset domains.

(2) From chip pin nSYS_RST.

The nSYS_RST pin can be used for external reset. This reset source only works in the system reset domain.

(3) Reset from Core, including Reset triggered by Debug Module. This reset source only works on the system reset domain.

Any of the above global reset sources can trigger a system reset.

7 Top level pin table

The top-level pins of the FSL91030M chip SoC are shown in Table 7-1.

Table 7-1 Top-level pin table

	Name Function System clock input		Auxiliary function 1	Auxiliary function 2
Clock	syst_clk			
clock	rtc_clk	RTC clock input		
system	nPOR_RST	System power-on reset signal input		
system	nSYS_RST	System normal operation reset signal input		
system	JTAG_TCK	JTAG debug port		
system	JTAG_TDI			
system	JTAG_TDO			
system	JTAG_TMS			
system	ext_nmi	NMI Input		
	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	sd			
	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		

8Interrupt handling

This chapter will introduce the interrupt module.

8.1 IRQC Interrupt Management

IRQC can support multiple external interrupt sources. The number of interrupt sources connected in a specific SoC may be different.

GPIO interrupts and other peripheral interrupts are connected as external interrupt sources, and their interrupt allocation is shown in Table 8-1.

Table 8-1 IRQC interrupt management

IRQC interrupt source number	Current new version
19	gpio_0~23
20	uart0
twenty one	uart1
twenty two	I2C
twenty three	QSPI
twenty four	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 Interrupt Management

NMI interrupt is on the external port of this SoC.

9Peripheral Introduction

This chapter will introduce the peripheral modules mounted on the private device bus of the SoC.

9.1 GPIO

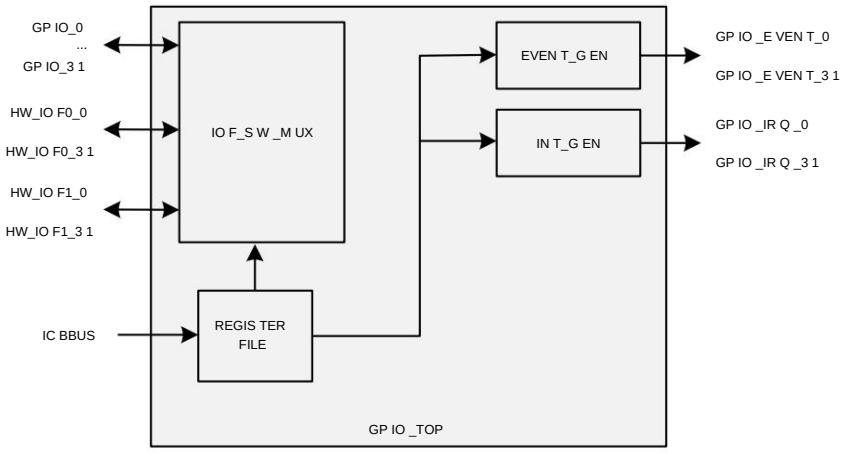

Figure 9-1 GPIO data flow

9.1.1 **GPIO** Function Description

GPIO stands for General Purpose I/O, and its main points are as follows:

ÿ GPIO is used to provide a set of I/O general input and output interfaces. Each I/O can be configured as input or output by software. If it is output, the specific output value can be set.

ÿ Each GPIO I/O is connected to the interrupt source interface of IRQC as an interrupt source.

ÿ Each I/O can also be configured as an IOF (Hardware I/O Functions), which means that the I/O is provided to other modules inside the SoC. Reuse.

ÿ For example, SPI, UART, etc. The multiplexing and allocation of GPIO by SoC internal modules is shown in Figure 9-1, where each I/O can be multiplexed by two internal modules, and the software can select the signal source by configuring each I/O to select IOF0 or IOF1.

The SoC GPIO pins of the FSL91030M chip are the main general interface for connecting the SoC with the outside world.

9.1.2 Port Operation Description

Each bit of the GPIO port can be configured into multiple modes by software:

ÿ Input pull-up

ÿ Input pull-down

ÿ Open drain output

ÿ Push-pull output

ÿ Drive strength output

9.1.3 GPIO Interrupt Request

Each GPIO bit can generate an interrupt bit. The interrupt can be driven by the rising or falling edge, or by the level value, and each
Each interrupt can be enabled individually.

The input is synchronized before being sampled by the interrupt logic, so the input pulse width must be long enough to be detected by the synchronization logic.

To enable interrupts, set the corresponding bits in rise_ie and/or fall_ie to 1. If the corresponding bits in rise_ip or fall_ip are set
is 1, the interrupt pin will go high.

Table 9-1 GPIO interrupt request

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

9.1.4 Hardware I/O Function (IOF)

Each GPIO pin can implement up to 2 HW driver functions (IOF) enabled via the IOF_EN register.
and IOF_SEL1 registers select which IOF to use.

When the pin is set to perform IOF, the software registers IEN, OEN, OVAL, PUE, DS, PDE, OD,
PUP can directly control the pins. However, the pins can be controlled by the hardware driver IOF. Which functions are controlled by IOF and which functions are controlled by software?
register control, fixed in hardware on a per IOF basis. If a pin [x] configured with IOF_EN [x] does not have iof_en, then the pin
Will revert to full software control.

9.1.5 Output Inversion

When configured as an output (controlled by SW or IOF), the XOR register of SW can be written in combination with the output to invert it.

9.1.6 Memory Map

The memory map of the GPIO registers is shown in the following table. The GPIO memory map is a 32-bit memory access that requires native access to it.

Table 9-2 GPIO memory address mapping register list

Register Offset	Register Name	Description

0x00	IVAL	Pin Value
0x04	IEN	Pin input enable
0x08	OEN	Output enable of Pin
0x0C	OVAL	Output port value
0x10	PUE	Pull-up enable
0x14	DS	Drive Strength
0x18	PDE	Pull-down enable
0x1C	OPEN DRAIN	Open drain enable
0x20	PUP	Push-pull enable
0x24	RISE_IE	Rising edge interrupt enable
0x28	RISE_IP	Rising edge interrupt wait flag
0x2C	FALL_IE	Falling edge interrupt enable
0x30	FALL_IP	Falling edge interrupt wait flag
0x34	HIGH_IE	High level interrupt enable
0x38	HIGH_IP	High level interrupt wait flag
0x3C	LOW_IE	Low level interrupt enable
0x40	LOW_IP	Low level interrupt wait flag
0x44	IOF_EN	HW I/O function enable
0x48	IOF_SEL0	HW I/O function select 0
0x4C	IOF_SEL1	HW I/O function select 1
0x50	EVENT_RISE_EN	Rising edge event enable
0x54	EVENT_FALL_EN	Falling edge event enable
0x58	OUT_XOR	Output XOR
0x5C	SW_FILTER_EN	Active Schmitt trigger input

9.1.7 IVAL

Register offset: 0x00

Register Description: Pin Value

Bits	Name	R/W	DescriptionInput	Default
[0:31]	ival	RO	value	0x0

9.1.8 IEN

Register offset: 0x04

Register Description: Pin input enable

Bits	Name	R/W	Description	Default

0:31	ien	R W	Input Enable 0: Input disabled 1: Input Enable	0x0

9.1.9 OEN

Register offset: 0x08

Register Description: Pin output enable

Bits	Name	R/W	Description	Default
0:31	oen	R W	Output Enable 0: Output disabled 1: Output Enable	0x0

9.1.10 OVAL

Register offset: 0x0C

Register Description: Output Port Value

Bits	Name	R/W		Default
0:31	oval	R W	**DescriptionOutput** value	0x0

9.1.11 PUE

Register offset: 0x10

Register Description: Pull-up Enable

Bits	Name	R/W	DescriptionPull	Default
0:31	pue	R W	-up mode: 0: Disable 1: Enable	0x0

9.1.12 DS

Register offset: 0x14

Register Description: Drive Strength

Bits	Name R/W		DescriptionWhen	Default
0:31	ds	R W	configured as an output, each pin has a SW controllable drive strength 0: Disable 1: Enable	0x0

9.1.13 PDE

Register offset: 0x18

Register Description: Pull-down Enable

Bits	Name	R/W	Description	Default
[0:31]	pde	R W	drop-down mode: 0: Disable 1: Enable	0x0

9.1.14 OPEN DRAIN

Register offset: 0x1C

Register Description: Open Drain Enable

Bits	Name	R/W	DescriptionOpen	Default
0:31	open drain	R W	drain mode: 0: Disable 1: Enable	0x0

9.1.15 PUP

Register offset: 0x20

Register Description: Push-Pull Enable

Bits	Name	R/W	DescriptionPull	Default
0:31	pull-up enable RW		-up mode: 0: Disable 1: Enable	0x0

9.1.16 RISE_IE

Register offset: 0x24

Register Description: Rising edge interrupt enable

Bits	Name	R/W	DescriptionRising	Default
0:31	rise ie	R W	edge interrupt mode: 0: Disable 1: Enable	0x0

9.1.17 RISE_IP

Register offset: 0x28

Register Description: Rising edge interrupt wait flag

Bits	Name	R/W	Description	Default
0:31	rise ip	R W	Rising edge interrupt pending: 0: Rising edge interrupt driven by GPIO rising edge 1: Rising edge interrupt driven by software	0x0

9.1.18 FALL_IE

Register offset: 0x2C

Register Description: Falling Edge Interrupt Enable

Bits	Name	R/W	Description	Default
0:31	fall ie	R W	Falling edge interrupt enable: 0: Disable 1: Enable	0x0

9.1.19 FALL_IP

Register offset: 0x30

Register Description: Falling edge interrupt wait flag

Bits	Name	R/W		Default
0:31	fall ip	R W	**DescriptionFalling** edge interrupt wait flag: 0: Falling edge interrupt driven by the falling edge of GPIO 1: Falling edge interrupt driven by software	0x0

9.1.20 HIGH_IE

Register offset: 0x34

Register Description: High level interrupt enable

Bits	Name	R/W	DescriptionHigh	Default
0:31	high ie	R W	level interrupt enable: 0: Disable 1: Enable	0x0

9.1.21 HIGH_IP

Register offset: 0x38

Register Description: High level interrupt wait flag

Bits	Name	R/W	DescriptionHigh	Default
0:31	high ip	R W	level interrupt waiting flag: 0: Falling edge interrupt driven by the falling edge of GPIO 1: Falling edge interrupt driven by software	0x0

9.1.22 LOW_IP

Register offset: 0x40

Register Description: Low level interrupt wait flag

Bits	Name	R/W	DescriptionLow	Default
0:31	low ip	R W	level interrupt wait flag: 0: Low level interrupt driven by the falling edge of GPIO 1: Low level interrupt driven by software	0x0

9.1.23 IOF_EN

Register offset: 0x44

Register Description: HW I/O function enable

Bits	Name	R/W	Description	Default
[0:31]	iof en	R W	HW I/O function enable	0x0

9.1.24 IOF_SEL0

Register offset: 0x48

Register Description: HW I/O Function Selection 0

Bits	Name	R/W	DescriptionEach	Default
0:31	iof sel0	R W	GPIO pin can implement up to 4 HW driver functions (based on at {iof_sel1,iof_sel0}) 00: Select HW IOF0 01: Select HW IOF1 10/11: Reversal	0x0

9.1.25 LOW_IE

Register offset: 0x3C

Register Description: Low level interrupt enable

Bits	Name	R/W	DescriptionLow	Default
0:31	low ie	R W	level interrupt enable: 0: Disable 1: Enable	0x0

9.1.26 IOF_SEL1

Register offset: 0x4C

Register Description: HW I/O Function Selection 1

Bits	Name	R/W	DescriptionEach	Default
0:31	iof sel1	R W	GPIO pin can implement up to 4 HW driver functions (based on at {iof_sel1,iof_sel0}) 00: Select HW IOF0 01: Select HW IOF1 10/11: Reversal	0x0

9.1.27 EVENT_RISE_EN

Register offset: 0x50

Register Description: Rising Edge Event Enable

Bits	Name	R/W		Default
0:31	event rise en RW		**DescriptionRising** edge event enable: 0: Disable 1: Enable	0x0

9.1.28 EVENT_FALL_EN

Register offset: 0x54

Register Description: Falling Edge Event Enable

Bits	Name	R/W	Description	Default
0:31	event fell in RW		Falling edge event enable: 0: Disable 1: Enable	0x0

9.1.29 OUT_XOR

Register offset: 0x58

Register Description: Output Inversion Enable

Bits	Name	R/W	DescriptionOutput	Default
0:31	out xor	R W	inversion enable: 0: Disable 1: Enable	0x0

9.1.30 SW_FILTER_EN

Register offset: 0x5C

Register Description: Valid Schmitt Trigger Input

Bits	Name	R/W	Description	Default
0:31	sw filter en	R W	Effective Schmitt trigger input in PAD interface: 0: Disable 1: Enable	0x0

9.2 QSPI

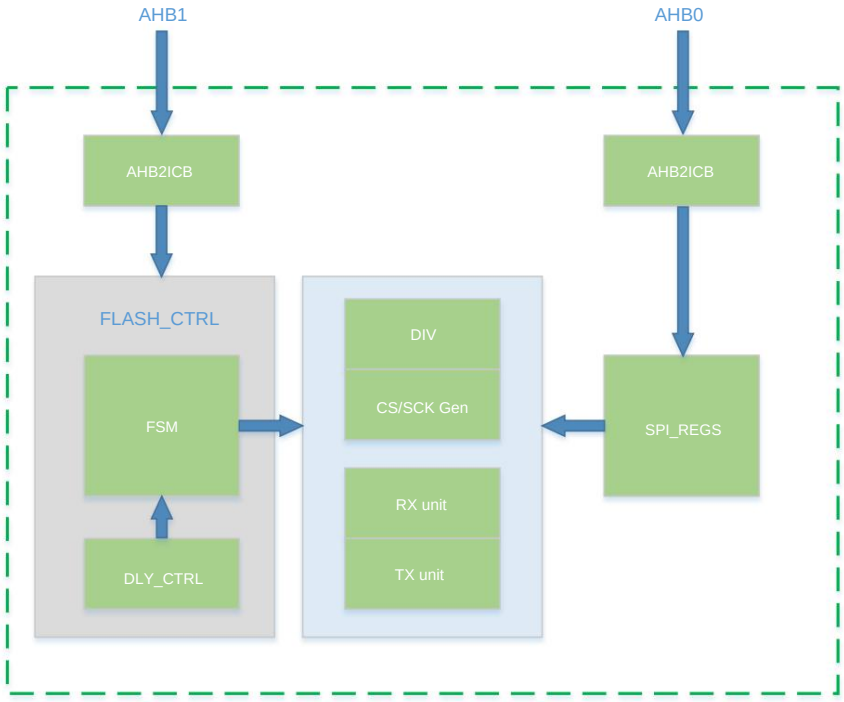

Figure 9-2 QSPI data flow

In this SoC, there is a QSPI master interface. As a SPI master, it supports sending and receiving data.
4 data lines, but can be configured as single-line (Single-SPI), dual-line (Dual-SPI) and quad-line (Quad-SPI) mode through registers
Supports sending and receiving FIFO buffers, and supports software programmable thresholds (Watermark) to generate interrupts. Supports registering
The device configures the polarity and phase of the SPI clock signal SCK.

ÿ Quad-SPI (QSPI) interface dedicated to connecting external Flash, with dedicated SoC top-level pins.

ÿ The QSPI interface can also be configured by software to be in eXecute-In-Place (XIP) mode. In this mode, the Flash
It can be treated as a read-only section and read directly as memory. After power-on by default, QSPI is in this mode.
Since Flash does not lose data when power is off, the system startup program can be stored in an external Flash, and then the processor
The core directly accesses the external Flash loader to start the boot program through the QSPI interface in eXecute-In-Place mode.

9.2.1 Register Description

The configurable register of QSPI is the memory address mapped register (Memory Address Mapped).
The SPI is loaded on the private device bus of the SoC. The configurable register list of SPI and its offset address are shown in the following table.

Register Offset	Register Name	Description
0x000	SPI_SCKDIV	SCK clock frequency division factor register (xip/normal mode)
0x004	SPI_SCKMODE	SCK mode configuration register (xip/normal mode)
0x00C	SPI_FORCE	SPI unused interface forced output 1 enabled
0x010	SPI_CSID	CS strobe identification (ID) register (xip/normal mode)
0x014	SPI_CSDEF	CS Idle Value Register
0x018	SPI_CSMODE	CS Mode Register
0x01C	SPI_VISION	SPI Version Register Version 1.0
0x028	SPI_DELAY0	XIP transmission delay control register 0 (xip/normal mode)
0x02C	SPI_DELAY1	XIP transmission delay control register 1 (xip/normal mode)
0x040	SPI_FMT	Transfer parameter configuration register (xip/normal mode)
0x07C	SPI_STATUS	Transfer Status Register (Normal Mode)
0x048	SPI_TXDATA	Send data register (normal mode)
0x04C	SPI_RXDATA	Receive data register (normal mode)
0x060	SPI_FCTRL	XIP Mode Control Register (xip)
0x064	SPI_FFMT	XIP transfer parameter control register (xip)
0x078	SPI_FFMT1	XIP transfer parameter control register 1 (xip)
0x080	SPI_RXEDGE	SPI receive data sampling edge control register
0x050	SPI_TXMARK	SPI transmit fifo water level register
0x054	SPI_RXMARK	SPI receive fifo water level register
0x070	SPI_IE	SPI Interrupt Enable Register
0x074	SPI_IP	SPI interrupt pending register

9.2.2 SPI_SCKDIV

Register offset: 0x000

Register Description: Used to set the SCK clock frequency of SPI

Bits	Name	R/W	Description is used	Default
0:11	div	R W	to configure the frequency division coefficient of the SCK signal.	0x3

9.2.3 SPI_SCKMODE

Register offset: 0x004

Register Description: Used to set the SCK clock frequency of SPI

Bits	Name	R/W	Description	Default
0	pha	R W	Used to configure	0x0
1	pol	R W	CPHA. Used to configure CPOL.	0x0

9.2.4 SPI_CSID

Register offset: 0x00C

Register Description: The SPI interface can have multiple enable signals. Multiple enable signals are used to connect multiple SPI slave devices on the same bus. It is possible, but only one SPI slave device can be enabled at a time. The SPI_CSID register is used to select the enable signal for the SPI.

Bits	Name	R/W	DescriptionThe	Default
0:1	csid	R W	value of this field is used to select the index of the enable signal. 00: No selection 01: Select SS0 10: Choose SS1 11: Choose SS2	0x0

9.2.5 SPI_CSDEF

Register offset: 0x014

Register Description: CS Idle Register.

Bits	Name	R/W	DescriptionThe	Default
0	cs0def	R W	value of this field indicates the idle value of the CS0 enable	0x1
1	cs1def	R W	signal.The value of this field indicates the idle value of the CS1	0x1
2	cs2def	R W	enable signal.The value of this field indicates the idle value of the	0x1
3	cs3def	R W	CS2 enable signal.The value of this field indicates the idle value of the CS3 enable signal.	0x1

9.2.6 SPI_CSMODE

Register offset: 0x018

Register Description: CS Mode Register.

Bits	Name	R/W	Description	Default

0:1	mode	RW	Assume that the value of this field is 0, indicating that the configuration enable signal is in AUTO mode. Assume that the value of this field is 2, indicating that the configuration enable signal is in HOLD mode. Assume that the value of this field is 3, indicating that the configuration enable signal is in OFF mode.	0x0

9.2.7 SPI_DELAY0

Register offset: 0x028

Register Description: Used to configure the delay period parameters.

Bits	Name	R/W	Description	Default
0:7	cssck	R W	The value of this field specifies the first SCK before starting to send data. How many cycles before the clock leading edge will The enable signal (SS) is set to a valid value.	0x0
16:23	sckcs	R W	The value of this field specifies the last How many cycles will the SCK clock continue to Keep the enable signal (SS) at a valid value.	0x0

9.2.8 SPI_DELAY1

Register offset: 0x02C

Register Description: Used to configure several delay cycle parameters.

Bits	Name	R/W	Description The value of	Default
0:7	intercs	R W	this field specifies whether the enable signal can be restored from the "valid value to the idle value" (de-assertion)" to "reset to a valid value The minimum number of idle cycles that should last between "assertions" (Minimum CS inactive time).	0x0

9.2.9 SPI_FMT

Register offset: 0x040

Register Description: In FIFO transmit/receive mode, the SPI_TXDATA and SPI_RXDATA registers can be used to send or receive data. When using SPI_TXDATA and SPI_RXDATA to send and receive data, the SPI_FMT register can be used

to configure several transmission parameters.

Bits	Name	R/W	Description If	Default
0:1	proto	R W	the value of this field is 2, the transmission protocol is configured as Quad-SPI. The four data lines DQ0, DQ1, DQ2, and DQ3 are working. If the value of this field is 1, the transmission protocol is configured as Dual-SPI. Two data lines DQ0 and DQ1 are operational. If the value of this field is 0, the transmission protocol is configured as Single-SPI. There are two data lines DQ0 (working as MOSI) and DQ1 (working as MISO).	0x0
2	Endian RW	If the value of this field is 1, the data is sent low bit first (LSB first).		0x0

			If the value of this field is 0, the data is sent high bit first (MSB first).	
3	dir	R W	If the value of this field is 1, it means TX, that is, sending. In this mode, the RX-FIFO No data will be received. If the value of this field is 0, it means RX, that is, receiving. In this mode, the RX-FIFO Will receive data: If the proto domain is configured for Dual or Quad-SPI protocol, all DQ numbers The data lines are in the input state of accepting data. If the proto domain is configured with the Single-SPI protocol, then according to the common SPI protocol, DQ0 (MOSI) will still be output, and DQ1 (MISO) will receive data as input.	0x0
16:19	len	R W	The value of this field specifies the number of bits (length value) to send a frame of data. The valid length value The range is 0 to 8.	0x1

9.2.10 SPI_STATUS

Register offset: 0x07C

Register Description: Used to indicate the current transmission status.

Bits	Name	R/W	Description: The	Default
0	tip	RO	value of this field is used to indicate the current transmission status.	0x0

9.2.11 SPI_TXDATA

Register offset: 0x048

Register Description: In FIFO transmit/receive mode, data can be sent through the SPI_TXDATA register.

Bits Name R/W 0:7 txdata			Description /	Default
RO				0x0
31	full RO		This bit is a read-only field used to indicate whether the SPI TX-FIFO is full. If the full bit is 1, it means that the current SPI-TX-FIFO is full and the txdata field is written. The data in the txdata field will be ignored; otherwise, if it is not full, the data written into the txdata field will be received.	0x0

9.2.12 SPI_RXDATA

Register offset: 0x04C

Register Description: In FIFO transmit/receive mode, data can be received through the SPI_RXDATA register.

Bits Name R/W			Description If	Default
0:7 rxdata RO			the empty field is 0, the data in the txdata field read by the software is valid data If the empty field is 1, the data read from the txdata field by the software is invalid data. This	0x0
31 empty RO			bit is a read-only field and is used to indicate whether the SPI RX-FIFO status is full. If the full bit is 1, it means that the current SPI-TX-FIFO is full and is written to the rxdata domain. The data in the rxdata field will be ignored; otherwise, it is not full and the data written to the rxdata field will be received.	0x0

9.2.13 SPI_FCTRL

Register offset: 0x060

Register Description: Enable Flash Xip mode of QSPI0 through SPI_FCTRL register.

Bits	Name	R/W	Description If	Default
0	flash_en	R W	this field is 1, it means that the Flash XiP mode of QSPI0 is enabled. If this field is 0, it means that the Flash XiP mode of QSPI0 is not enabled. In normal FIFO transmit/receive mode.	0x0
1	flash_wen RW		If this field is 1, it enables the Flash XiP write mode of QSPI0. If this field is 0, it means that the Flash XiP write mode of QSPI0 is not enabled. The Flash XiP mode can only be used in read mode.	0x0
3	flash_burst_en RW		If this field is 1, it means that the burst mode of Flash XiP of QSPI0 is enabled. If this field is 0, it means that the burst mode of Flash XiP of QSPI0 is not enabled. Only Flash XiP single mode is supported.	0x1

9.2.14 SPI_FFMT

Register offset: 0x064

Register Description: When QSPI0 is in Flash Xip mode, the entire QSPI0 (external Flash) is mapped to a read-only address
Reading data or fetching instructions directly from this interval will automatically trigger QSPI0 to read external
Flash. The specific SPI protocol behavior of QSPI0 reading external Flash through the SPI interface is controlled by the SPI_FFMT register.

Bits	Name R/W	Description Whether to send the command (Command)	Default
0	consists of the	cmd_en RW addr_len RW The address bit	0x1
1:3	number of bytes (0 to 4)	The default is 3 bytes (ie 24 bits). 0x3	
4:7	pad_cnt RW	How many Dummy read cycles are sent.	0x0
8:9	cmd_proto RW	The SPI protocol used in the command sending stage, see SPI_FMT register The definition of the proto field of the device.	0x0
10:11	addr_proto RW	The SPI protocol used in the address sending phase, see SPI_FMT register The definition of the proto domain.	0x0
12:13	data_proto RW	The SPI protocol used in the data transmission phase is described in the SPI_FMT register. proto field definition.	0x0
14	data_endian RW	If the value of this field is 1, the data is sent low bit first (LSB first). If the value of this field is 0, the data is sent high bit first (MSB first).	0x0
16:23	cmd_code RW	Specific command value. The default value is 0x3. It is the commonly used Winbond/Numonx Flash serial READ command (0x03).	0x3
24:31	pad_code RW	The first 8 bits sent in Dummay Cycles.	0x0

9.2.15 SPI_FFMT1

Register offset: 0x078

Register Description: XIP transfer parameter control register 1 (xip).

Bits	Name	R/W	Description	Default
0:7	wcmd_code	R W	Flash XiP mode specific write command value. How many	0x2
8:11	wpad_cnt	R W	Dummy write cycles are sent?	0x0

9.2.16 SPI_RXEDGE

Register offset: 0x080

Register Description: SPI receive data sampling edge control register.

Bits	Name	R/W	Description	Default
0	rxedge	RW	Rx Receive edge control register. 0 means receiving data at the rising edge of SCK 1 means receiving data at the falling edge of SCK.	0x0

9.2.17 SPI_TXMARK

Register offset: 0x050

Register Description: SPI transmit FIFO water level register.

Bits	Name	R/W	Description	Default
0:2	txmark	RW	The value of this field indicates the threshold (Watermark) for TX-FIFO to generate an interrupt.	0x0

9.2.18 SPI_RXMARK

Register offset: 0x054

Register Description: SPI receive FIFO water level register.

Bits	Name	R/W	Description	Default
0:2	rxmark	RW	The value of this field indicates the threshold (Watermark) for RX-FIFO to generate an interrupt.	0x0

9.2.19 SPI_IE

Register offset: 0x070

Register Description: SPI interrupt enable register.

Bits	Name	R/W	Description	Default
0	txie	R W	If the txie field is 1, it means enabling the SPI transmit interrupt. If the txie field is 0, it means that the SPI transmit interrupt is disabled.	0x0
1	rxie	R W	If the rxie field is 1, it means that the SPI receive interrupt is enabled. If the rxie field is 0, it means that the SPI receive interrupt is not enabled.	0x0

9.2.20 SPI_IP

Register offset: 0x070

Register Description: SPI interrupt enable register.

Bits	Name	R/W	**Description** If this	Default
0	txip	R W	field is 1, it indicates that a transmit interrupt is currently being generated. If this field is 0, it means that no transmit interrupt is currently generated.	0x0
1	rxip	R W	If this field is 1, it means that a receive interrupt is currently generated. If this field is 0, it means that no receive interrupt has occurred.	0x0

9.3 UART

UART stands for Universal Asynchronous Receiver-Transmitter, which provides a
Flexible and convenient serial data exchange interface, data frames can be transmitted in an asynchronous manner. UART provides a programmable baud rate generator
The default baud rate is 38400bps, which can generate the specific frequency required for UART transmission and reception. This SoC has two independent UARTs.
Both UARTs multiplex the top-level GPIO pins to communicate with the outside world.

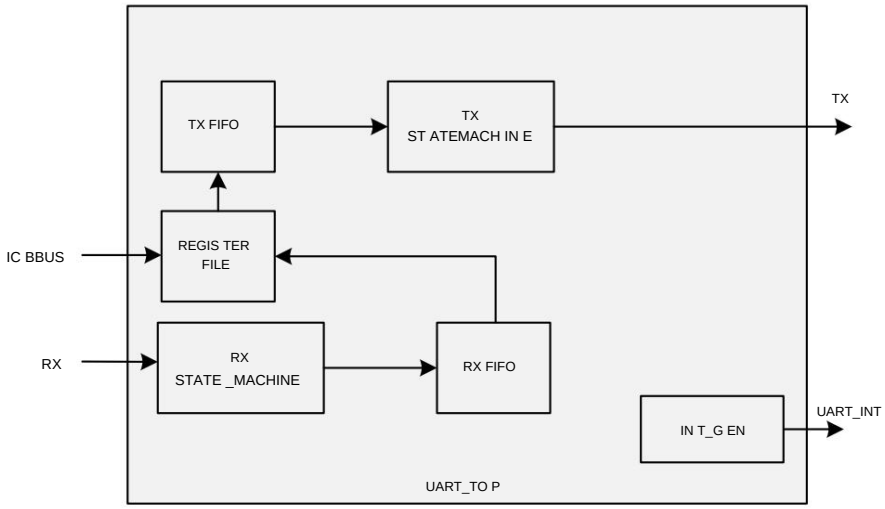

Figure 9-3 UART data flow

9.3.1 **UART** Features

ÿ Full-duplex asynchronous communication

ÿ Individual enable bits for transmitter or receiver

ÿ Baud rate configurable

ÿ Configurable parity generation (odd/even) and checking

ÿ Configurable data word width (5/6/7/8/9)

ÿ Configurable stop bits (support 0.5/1/1.5/2 stop bits)

ÿ RX and TX FIFO sizes are configurable at the RTL level

9.3.2 **UART** Interrupt Request

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

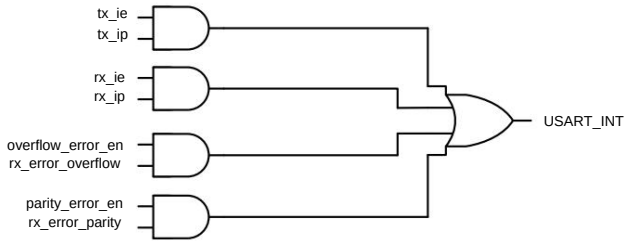

Figure 9-4 Uart interrupt mapping

9.3.3 Memory Map

The memory map of the UART controller is shown in Table 9-3. The UART memory map only supports 32-bit memory accesses.

Table 9-3 UART register mapping table

Register Offset	**Register Name**	**Description**
0x00	UART_TXDATA	Send data register
0x04	UART_RXDATA	Receive data register
0x08	UART_TXCTRL	Transmit Control Register
0x0c	UART_RXCTRL	Receive Control Register
0x10	UART_IE	UART Interrupt Enable Register
0x14	UART_IP	UART interrupt wait register
0x18	UART_DIV	Baud Rate Divisor Register
0x1c	UART_STATUS	RX/TX busy status register
0x20	UART_SETUP	UART setup register
0x24	UART_ERROR	UART Receive Error Status Register
0x28	UART_IRQ_EN	UART Interrupt Request Enable Register

9.3.4 UART_TXDATA

Register offset: 0x00

Register Description: Transmit data register. If the FIFO can receive new data, writing to the txdata register will
The characters are written into the transmit FIFO in sequence; the full flag and data field are returned from the txdata register; the full flag indicates that the transmit FIFO is full.
Whether new data can be received. When the full flag is valid, the data written will be ignored.

Bits	Name	R/W	DescriptionSend	Default
0:8	txdata	R W	data is	0x0
9:30	/	/		/
31	full	R W	reserved. Transmit FIFO is full.	0x0

9.3.5 UART_RXDATA

Register offset: 0x04

Register Description: Receive data register. Reading the rxdata register returns the value of the data field. The empty flag indicates whether the receive FIFO is empty.
If valid, there is no valid data in the data field.

Bits	Name	R/W	Description Send	Default
0:8	rxdata	R W	data	0x0
9:30	/	/		/
31	full	R W	Reserved Receive data Empty	0x0

9.3.6 UART_TXCTRL

Register offset: 0x08

Register Description: Transmit control register. Reading and writing the txctrl register controls the operation of the transmit channel. The txen bit controls whether the tx channel is in
Active state. When cleared, disables transmission of data in the tx FIFO and drives the txd pin high.

Bits	Name	R/W	DescriptionSend	Default
0	txen	R W	enable stop	0x0
1:2	nstop	R W	bit configuration: 00: 1 bit 01: 2bit 10: 0.5bit 11: 1.5bit	0x0
3:15	/	/	Retain	/
16:19	txcnt	R W	Send Watermark Level	0x0
20:31	/	/	Retain	/

9.3.7 UART_RXCTRL

Register offset: 0x0C

Register Description: Receive control register. Reading and writing the rxctrl register controls the operation of the receive channel. The rxen bit controls whether the rx channel When cleared, the state of the rxd pin is ignored and data is not written to the rx FIFO.

Bits	Name	R/W	Description Send	Default
0	txen	R W	Enable	0x0
1:15	/	/		/
16:19	rxcnt	R W	Reserve Receive	0x0
20:31	/	/	Watermark Level Reserve	/

9.3.8 UART_IE

Register offset: 0x10

Register Description: UART interrupt enable register.

Bits	Name	R/W	DescriptionSend	Default
0	txie	R W	watermark interrupt	0x0
1	rxie	R W	enableReceive watermark	0x0
2:31	/	/	interrupt enableReserve	/

9.3.9 UART_IP

Register offset: 0x14

Register Description: UART interrupt wait register.

Bits	Name	R/W	DescriptionSend	Default
0	txip	R W	watermark interrupt	0x0
1	rxip /	R W	waitingReceive watermark	0x0
2:31		/	interrupt waiting reserved	/

9.3.10 UART_DIV

Register offset: 0x18

Register Description: Baud rate divisor register. The baud rate divisor used to generate the baud rate in the tx and rx channels is specified by reading and writing the baud div register. The input clock and baud rate can be converted using the following formula: fbaud = fclk_in/(div+1).

Bits	Name	R/W	DescriptionBaud	Default
0:15	baud div	R W	rate divisor	0x21e
16:31	/	/	reserved	/

9.3.11 UART_STATUS

Register offset: 0x1C

Register Description: Uart status register.

Bits	Name	R/W		Default
0	rx_busy	R W		0x0
1	tx_busy /	R W		0x0
2:31		/	**DescriptionReceiving** dataSending dataReserved	/

9.3.12 UART_SETUP

Register offset: 0x20

Register Description: Uart setup register.

Bits	Name	R/W	DescriptionParity	Default
0	parity_en	R W	bit generation and checking configuration fields: 0: Disable 1: Enable	0x0
1	parity_sel	R W	Parity mode selection: 0: Even parity 1: Odd parity	0x0
2	/	/		/
3	clean_fifo	R W	Reserved to clear rx FIFO, set to 0/1 to reset FIFO: 0: Stop clearing the RX FIFO. 1: Clear RX FIFO.	0x0
4:6	bit_length	R W	Character length field: 0x0: 5 bits 0x1: 6 bits 0x2: 7 bits 0x3: 8 bits 0x4: 9 bits	0x0
7:31	/	/	reserve	/

9.3.13 UART_ERROR

Register offset: 0x24

Register Description: Uart receive error status register. The error register is a read-only register, and the error status is set by hardware. Software can The software can obtain the overflow and parity check results of the rx fifo by sending a read command. If an overflow or parity check error occurs, the software can command to clear the error status.

Bits	Name	R/W	Description	Default
0	rx_error_overflow	RO	RX FIFO overflow error flag: 0: True 1: RX FIFO overflow error occurred	0x0
1	rx_error_parity	RO	RX parity error flag: 0: Yes	0x0

			1: RX parity error occurred	
2:31	/	/	reserve	/

9.3.14 UART_IRQ_EN

Register offset: 0x28

Register Description: Uart interrupt request enable register.

Bits	Name	R/W	Description	Default
0	rx_error_overflow	R W	RX FIFO overflow error flag: 0: True 1: RX FIFO overflow error occurred	0x0
1	rx_error_parity	R W	RX parity error flag: 0: Yes 1: RX parity error occurred	0x0
2:31	/	/	reserve	/

9.4 DMA

A DMA controller is a hardware method for transferring data directly between peripherals/memory and memory without CPU intervention. Shift the load of large chunks of data from where they are needed.

9.4.1 Function Overview

ÿ Two SICBs (standard ICBs) are used for data transmission, and one FSICB (simplified ICB) is used for DMA configuration;

ÿ Support 1-16 beat incremental burst type memory access;

ÿ Supports configurable data width of 8, 16, 32, 64, 128 bits for memory access;

ÿ Support fixed or additional address generation algorithms;

ÿ Support continuous or repeated configurable N-times transmission mode;

ÿ Support independent source and destination base address settings, byte, half-word or word alignment;

ÿ Support any transmission block size within (1M-1) bytes;

ÿ Support interrupt generation, masking and clearing.

9.4.2 Register Map

The following table lists the overall memory map (offset) of the DMA controller section. The memory map of different peripherals can be found in the corresponding peripheral documentation. DMA related register definitions (base address, transfer size, enable, etc.) All register accesses are 32-bit word aligned.

The register mapping (offset) relationship of the DMA controller part is shown in the following table:

Register Offset	Register Name	Description
0x008+0x14	DMA_CFG_MSRCADDR	Source data base address
0x00C+0x14	DMA_CFG_MDSTADDR	Destination base address
0x010+0x14	DMA_CFG_MCTRL	Control Register
0x014+0x14	DMA_CFG_RPT	Transmission repetition times
0x018+0x14	DMA_CFG_MSIZE	Transfer size
0x100+0xC	DMA_CHX_IRQ_EN	M2M interrupt enable
0x104+0xC	DMA_CHX_IRQ_STAT	M2M Interrupt Status
0x108+0xC	DMA_CHX_IRQ_CLR	M2M interrupt clear

9.4.3 DMA_CFG_MSRCADDR

Register offset: 0x008+0x14

Register Description: Source address base register.

Bits	Name	R/W	**DescriptionThe** base	Default
0:31	src_base	R W	address of the source data block	0x0

9.4.4 DMA_CFG_MDSTADDR

Register offset: 0x00C+0x14

Register Description: Destination data base address register.

Bits	Name	R/W	**DescriptionThe** base	Default
0:31	dst_base	R W	address of the destination data block	0x0

9.4.5 DMA_CFG_MCTRL

Register offset: 0x010+0x14

Register Description: Control register.

Bits	Name	R/W	Description	Default
0	trans_en	R W	DMA transfer enable, asserting this bit will start mem2mem DMA transfer. It can be cleared by SW and after the current burst ends The transfer will stop. When a complete transfer is completed or a transfer error occurs The hardware can also clear it when responding, in which case the current The transfer will be aborted. Note that this is only possible if all other configuration bits are set correctly. This bit can only be set during configuration.	0x0
1:2	trans_type	R W	mem2mem is fixed to 2'b00	0x0

3:5	trans_per_sel RW		Keep	0x0
6:7	trans_mode	R W	transfer mode selection 2'b00: single mode transmission; 2'b01: Continuous mode transmission, in this mode, complete the current transmission After that, a new transfer will be automatically started with the same transfer configuration; 2'b10: Transmission repeat mode, using the same transmission configuration, the transmission will N consecutive times (defined in the "Transfer Repeat Number Register") 2'b11: Reserved	0x0
8:9	priority	R W		0x0
12	mdna	R W	Reserves the next address generation algorithm for transferring data to the target memory 1'b0: Add address mode 1'b1: Fixed address If fixed addresses are configured, force the start target address to be aligned	0x0
13	msna	R W	Next address generation algorithm for fetching data from source memory 1'b0: Add address mode 1'b1: Fixed address If a fixed address is configured, the starting source address is forced to be aligned	0x0
16:18	mdwidth	R W	The transfer width used to transfer data to the destination: 3'b000: 8 bits 3'b001: 16 bits 3'b010: 32 bits 3'b011: 64 bits 3'b100: 128 bits Others: Reserved	0x0
21:23	mswidth	R W	The transfer width obtained from the source 3'b000: 8 bits 3'b001: 16 bits 3'b010: 32 bits 3'b011: 64 bits 3'b100: 128 bits Others: Reserved	0x0
24:27	mdburst	R W	The number of transfers in a burst used to transfer data to the destination memory 4'b0000: 1 transfer 4'b0001: 2 transfers 4'b0010: 3 transfers 4'b0011: 4 transfers 4'b0100: 5 transfers 4'b0101: 6 transfers 4'b0110: 7 transmissions … 4'b1111: 16 transfers	0x0
28:31	msburst	R W	The number of transfers in a burst used to transfer data to the destination memory 4'b0000: 1 transfer 4'b0001: 2 transfers 4'b0010: 3 transfers 4'b0011: 4 transfers	0x0

			4'b0100: 5 transfers 4'b0101: 6 transfers 4'b0110: 7 transfers … 4'b1111: 16 transfers	

9.4.6 DMA_CFG_RPT

Register offset: 0x014+0x14

Register Description: Transmission repetition count register.

Bits	Name	R/W	DescriptionWhen	Default
0:11	trans_rpt	R W	the trans_mode configuration of the DMA_CFG_MCTRL register When b10 transmits a repeat mode, this field defines the number of repeat cycles.	0x0

9.4.7 DMA_CFG_MSIZE

Register offset: 0x018+0x14

Register Description: Transfer size register.

Bits	Name	R/W	DescriptionThe	Default
0:19	tsize	R W	transfer size of a DMA transfer. Do not allow transmission to start with 0; doing so may lead to unexpected results. During a DMA transfer, these bits indicate the remaining bytes to be transferred. If the trans_mode of the DMA_CFG_MCTRL register is configured Set to b01 or b10 (continuous or repeat mode), then when the current When the transfer is complete, it will automatically reload to the original value for the new transfer; DMA transfer errors may freeze the registers to the previous The value of the successful transmission, but it will restart and automatically record The correct remaining data number for a new transmission.	0x0

9.4.8 DMA_CHX_IRQ_EN

Register offset: 0x100+0xC

Register Description: Interrupt enable register.

Bits	Name	R/W	DescriptionFull	Default
0	ftrans_irq_en RW		output interrupt enable. Half	0x0
1	htrans_irq_en RW		transfer interrupt enable.	0x0
2	rsp_err_irq_en RW		DMA access error interrupt	0x0
3:31	/	/	enable. Reserved	/

9.4.9 DMA_CHX_IRQ_STAT

Register offset: 0x104+0xC

Register Description: Interrupt Status Register.

Bits	Name	R/W		Default
0	ftrans_irq_stat RO		**DescriptionInterrupt** status flag	0x0
1	htrans_irq_stat RO		for full transfer.Interrupt status	0x0
2	rsp_err_irq_stat RO		flag for half transfer.Interrupt status flag for	0x0
3:31	/	/	DMA access error. Reserved.	/

9.4.10 DMA_CHX_IRQ_CLR

Register offset: 0x108+0xC

Register Description: Interrupt clear register.

Bits	Name	R/W		Default
0	ftrans_irq_clr WO		**DescriptionInterrupt** status flag	0x0
1	htrans_irq_clr WO		for full transfer.Interrupt status	0x0
2	rsp_err_irq_clr WO		flag for half transfer.Interrupt status flag for	0x0
3:31	/	/	DMA access error. Reserved.	/

9.4.11 Accessing **switch** registers

If you want to use DMA as a table DMA to access switch registers, you need to configure the registers on the switch side first, and then turn on DMA. There are three registers on the switch side, namely alarm, configuration, and startup. The alarm register is the flag bit of the alarm signal, and the configuration register is The register is used to configure DMA transfer related parameters, and the start register is used to enable DMA.

Switch side alarm register

Register offset		0x50 + TOP_CFG_REG + 0x6000_0000 (TOP_CFG_REG = 0)		
Bit	domain	Read and write characteristic reset values		describe
field [1:0]	name axi_cp_alm	RO		1 write error, 0 read error

Switch side configuration register

Register offset		0x54 + TOP_CFG_REG + 0x6000_0000 (TOP_CFG_REG = 0)		
Bit		Read and write characteristic reset values		describe
field	domain	WO		Actual width of the table item (=actual value - 1)
[5:0] [11:6]	name seg_limit seg_num	WO		Table entry address width (2^n-1, such as 1, 3, 7, 15, etc.). seg_limit=69, then seg_num=127.
[12]	tx_cnt_endian_swap WO			Data sent to register, big and small endian swap
[13]	rx_cnt_endian_swap WO			The data sent by the register to the upper end is swapped

[31:14]	reg_dma_count	RO		Read-only registers, how many registers are operated after start

Switch side startup register

Register offset		0x58 + TOP_CFG_REG + 0x6000_0000 (TOP_CFG_REG = 0)		
Bit		Read and write characteristic reset values		describe
field [0]	domain name reg_dma_start	WO		Write 1 to start DMA

Configuration operation process:

1. Configure the register (0x54), and then configure reg_dma_start (0x58) to 1

2. Configure and enable DMA to start transferring data.

9.5 I2C

The I2C register mapping relationship is shown in the following table:

Register Offset 0x00	**Register Name**	**Description**
	I2C_PRERlo	Prescaler register lower 8 bytes
0x01	I2C_PRERhi	Prescaler register high 8 bytes
0x02	I2C_CTR	Control Register
0x03	I2C_TXR	Send Register
0x05	I2C_RXR	Receive Register
0x06	I2C_CR	Command Register
0x04	I2C_SR	Status Register
0x07	I2C_TRISE	scl delay register
0x08	I2C_FLTER	Filter register

9.5.1 I2C_PRERlo

Register offset: 0x00

Register Description: Prescaler SCL clock line. Due to the structure of the I2C interface, the core uses 4 * SCL clock internally.
The value of the prescaler register should only be changed if the 'EN' bit is cleared.

Bits	**Name**	**R/W**		**Default**
0:7	prerlo	R W	**DescriptionPrescaler** register low byte.	0xffff

9.5.2 I2C_PRERhi

Register offset: 0x01

Register Description: Prescaler SCL clock line. Due to the structure of the I2C interface, the core uses 4 * SCL clock internally.
The value of the prescaler register should only be changed if the 'EN' bit is cleared.

Bits	Name	R/W		Default
0:7	prerhi	R W	**DescriptionPrescaler** register high byte.	0xffff

9.5.3 I2C_CTR

Register offset: 0x02

Register Description: Control register. The core will respond to new commands only when the "EN" bit is set to 1. Pending commands have completed. The "EN" bit is cleared when no transmission is in progress (i.e. after a STOP command) or when the STO bit of the Command Register is set to 1.

When the process is suspended, the core can suspend the I2C bus.

Bits	Name /	R/W	Description	Default
0:5		R W	reserved.	/
6	ctr_en	R W	I2C core enable bits: 1: Enable 0: Disable	0x00
7	ctr_in	R W	I2C core interrupt enable bit: 1: Enable 0: Disable	0x00

9.5.4 I2C_TXR

Register offset: 0x03

Register Description: Send register.

Bits	Name	R/W	**Description** The	Default
0	txr	WO	next byte to be transmitted	0x00
1:7	txr_data	WO	during data transmission, this bit represents the LSB of the data. If it is a slave address transmission, this bit represents the RW bit. 1: Read slave 0: write slave	0x00

9.5.5 I2C_RXR

Register offset: 0x05

Register Description: Receive register.

Bits	Name	R/W	Description	Default
0:7	rxr	RO	The last byte received	0x00

9.5.6 I2C_CR

Register offset: 0x06

Register Description: Command register.

Bits	Name	R/W	DescriptionInterrupt	Default
0	lack	R W	response, when set, clear the pending interrupt.	0x00
1:2	/	R W	Reserved	/
3	ack	RW	The receiver sends ACK (ACK = '0') or NACK (ACK = '1')	0x00
4	wr	RW	Write slave Read slave Generate stop condition	0x00
5	rd	R W	Generate	0x00
6	sto	R W	(repeated) start	0x00
7	sta	R W	condition	0x00

9.5.7 I2C_SR

Register offset: 0x04

Register Description: Status Register.

Bits	Name	R/W		Default
0	if	RO	**DescriptionInterrupt** flag. When an interrupt is pending, this position is 1. If IEN When the bit is set to 1, it will cause the processor to interrupt. Set interrupt flag: complete one byte transfer.	0x00
1	tip	RO	Transfer in progress. 1: Transmitting 0: Transfer completed	
2:5	/	RO		/
6	busy	RO	Reserved to write slave I2C bus busy signal. "1" after detecting START signal After the STOP signal is detected, it is "0"	0x00
7	rxack	RO	Receive a response from the slave. This flag indicates the response of the addressed slave. 1: No response received 0: Received a response	0x00

9.5.8 I2C_TRISE

Register offset: 0x07

Register Description: Delay register, used to shield the frequency division error caused by slave waiting time.

Bits	Name	R/W	Description	Default
0:7	trise	RW	Used to shield the frequency division error caused by slave waiting time.	0x00

9.5.9 I2C_FLTER

Register offset: 0x08

Register Description: Filter register, used to adjust the filter frequency.

Bits	Name	R/W	**Description** is used	Default
0:4	flter	R W	to adjust the configuration filter frequency size.	0x00

Note:

1. The registers should satisfy the following relationship:

I2C_FLTR < I2C_PRER;

I2C_TRISE > I2C_FLTR*2 + 3;

(scl frequency) scl = (I2C_PRER * (5 + 1) + I2C_TRISE) (main frequency)

2. Both the transmit register and the command register are mapped to address 0x02. The transmit register is the MSB and the command register is the LSB.

3. Both the receive register and the status register are mapped to address 0x03. The receive register is the MSB and the status register is the LSB.

10 Revision Information

Revision time	Version	describe
2022.12.23	**V1.0**	initial version.