K210技术参考手册

KENDRYTE
勘智

提升社会运行效率 | 改善人类生活方式

关于本文档

本文档为用户提供Kendryte 210 寄存器介绍使用手册。

发布说明

日期	版本	发布说明
2018-08-01	V0.1.0	初始版本
2018-09-13	V0.1.1	修正 SPI 与 GPIO 中错误的描述
2018-09-14	V0.1.2	修正第一章节出现的错别字
2018-09-17	V0.1.3	修正第二章节引脚描述错误
2018-09-18	V0.1.4	增加 Kendryte 系统架构图
2018-09-19	V0.1.5	修正关于定时器的错误描述
2020-05-06	V0.1.6	修正部分单元细节参数

免责声明

本文中的信息,包括参考的 URL 地址, 如有变更, 恕不另行通知。文档 “按现状” 提供,不负任何担保责任, 包括对适销性、适用于特定用途或非侵权性的任何担保, 和任何提案、规格或样品在他处提到的任何担保。本文档不负任何责任, 包括使用本文档内信息产生的侵犯任何专利权行为的责任。本文档在此未以禁止反言或其他方式授予任何知识产权使用许可, 不管是明示许可还是暗示许可。文中提到的所有商标名称、商标和注册商标均属其各自所有者的财产, 特此声明。

版权公告

目 录

1 概述

Kendryte K210 是一款集成了机器视觉与机器听觉能力的系统级芯片(SoC)，使用台积电 (TSMC) 超低功耗的28nm先进制程,具有双核 64 位CPU,拥有较好的功耗性能，稳定性与可靠性。该方案力主为用户提供零门槛开发，可在最短时效内部署于用户的产品中，赋予产品人工智能功能。

Kendryte K210是定位于AI和IoT领域的SoC，同时也是使用非常方便的MCU。Kendryte 中文含义为勘智，而勘智取自勘物探智。

该芯片主要应用领域为物联网领域，因此为勘物；主要提供人工智能解决方案，在人工智能领域探索，因此为探智。其主要特性包括：

- 具备机器视觉能力；
- 具备机器听觉能力；
- 更好的低功耗视觉处理速度与准确率；
- 具备卷积人工神经网络硬件加速器 KPU，可高性能进行卷积运算；
- TSMC 28nm 先进制程，温度范围-40° C 到 125° C，稳定可靠；
- 支持固件加密，难以使用普通方法破解；
- 独特的可编程 IO 阵列，使产品设计更加灵活；
- 低电压，与相同处理能力的系统相比具有更低功耗；
- 3.3V/1.8V 双电压支持，无需电平转换，节约成本。

1.1 AI 解决方案

1.1.1 机器视觉

Kendryte K210 具备机器视觉能力，是零门槛机器视觉嵌入式解决方案。它可以在低功耗情况下进行卷积神经网络计算。该芯片可以实现以下机器视觉能力：

- 基于卷积神经网络的一般目标检测；
- 基于卷积神经网络的图像分类任务；
- 人脸检测和人脸识别；
- 实时获取被检测目标的大小与坐标；
- 实时获取被检测目标的种类。

1.1.2 机器听觉

Kendryte K210 具备机器听觉能力。芯片上自带高性能麦克风阵列音频处理器，可以进行实时声源定向与波束形成。该芯片可以实现以下机器听觉能力：

- 声源定向；
- 声场成像；
- 波束形成；
- 语音唤醒；
- 语音识别。

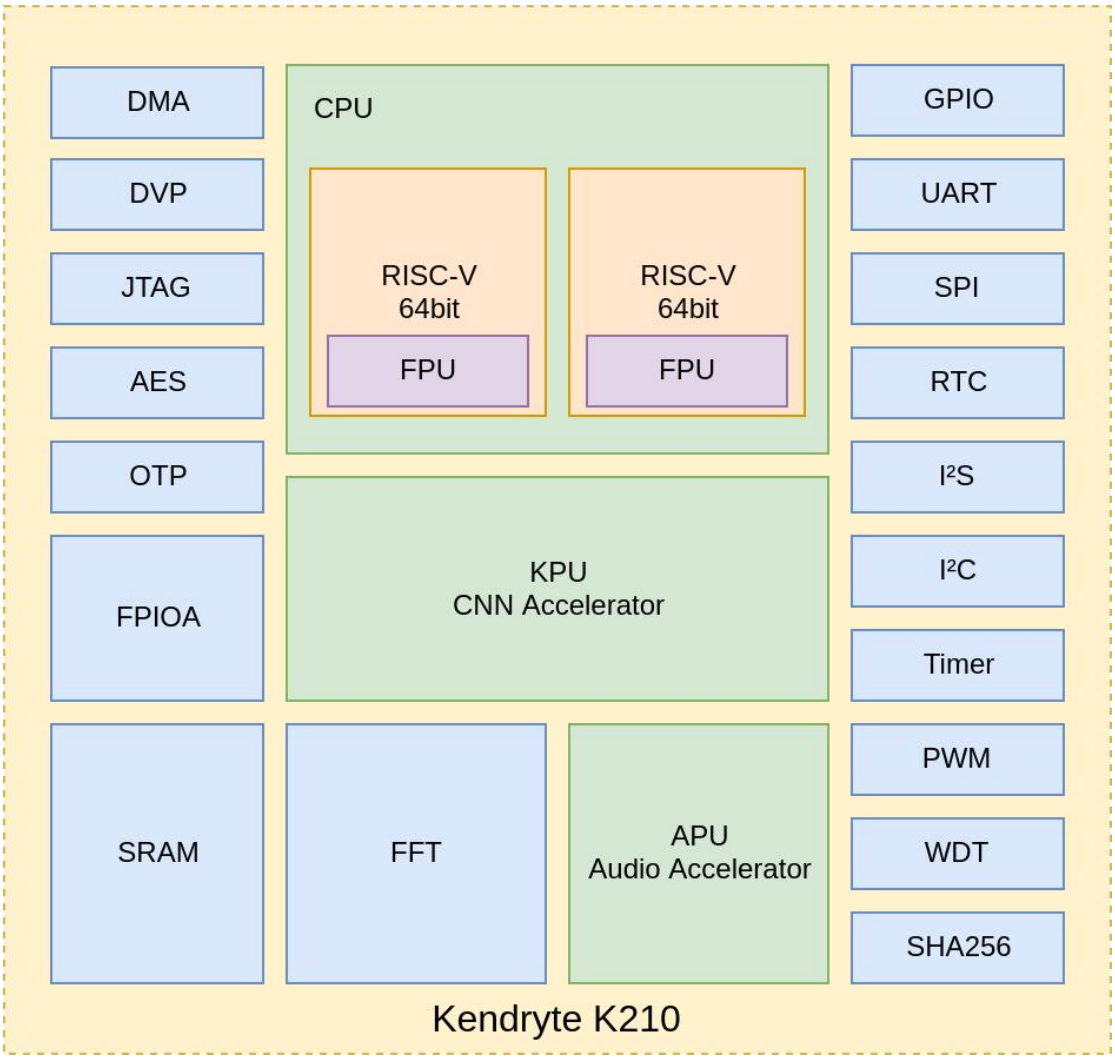
1.1.3 视觉/听觉混合解决方案

Kendryte K210 可结合机器视觉和机器听觉能力，提供更强大的功能。一方面，在应用中既可以通过声源定位和声场成像辅助机器视觉对目标的跟踪，又可以通过一般目标检测获得目标的方位后辅助机器 听觉对该方位进行波束形成。另一方面，可以通过摄像头传来的图像获得人的方向后，使得麦克风阵列

通过波束形成指向该人。同时也可以根据麦克风阵列确定一个说话人的方向，转动摄像头指向该人。

1.2 系统架构

1.2.1 功能框图

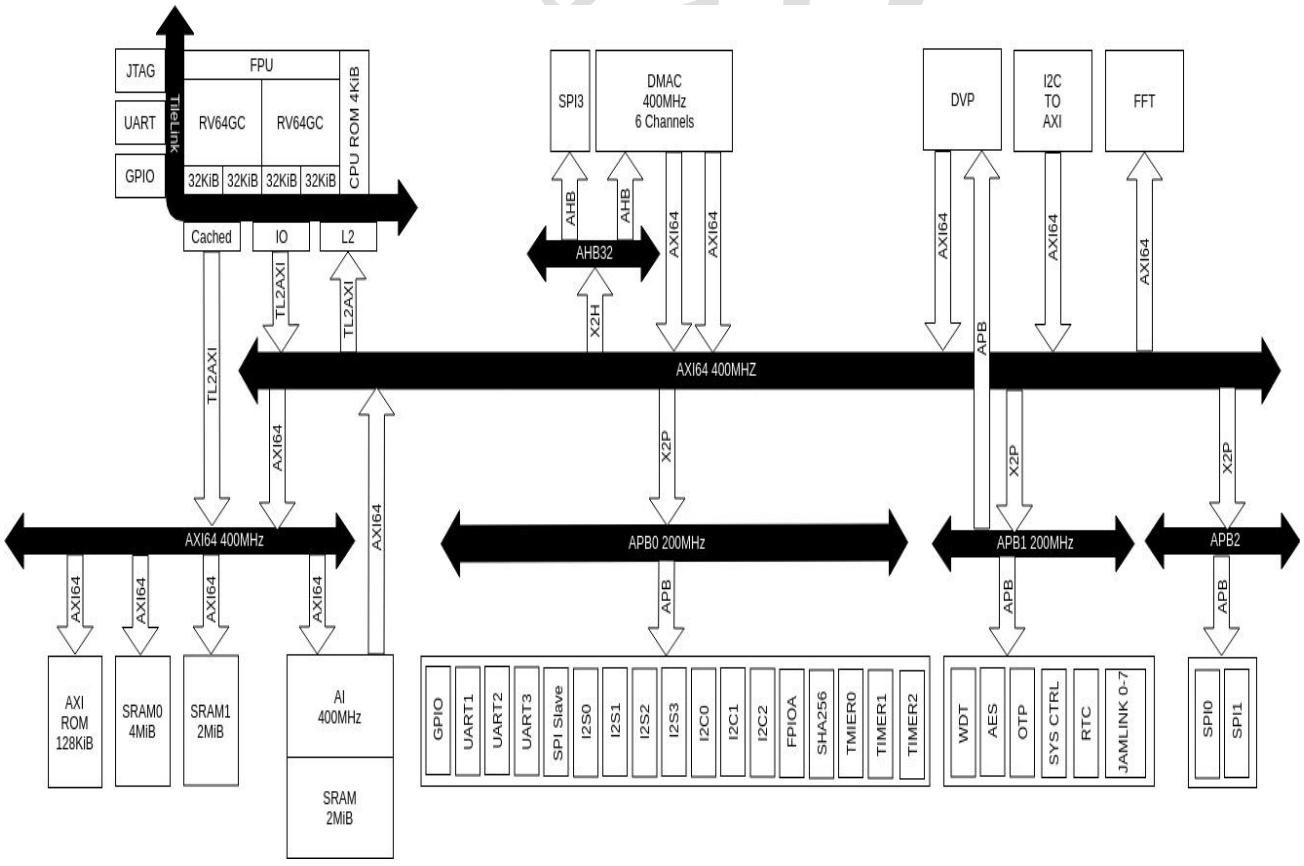


1.2.2 框图概述

K210 包含 RISC-V 64 位双核 CPU，每个核心内置独立 FPU. K210 的核心功能是机器视觉与听觉，其包含用于计算卷积人工神经网络的 KPU 与用于处理麦克风阵列输入的 APU. 同时 K210 具备快速傅里叶变换加速器，可以进行高

性能复数 FFT 计算。因此对于大多数机器学习算法，K210 具备高性能处理能力。K210 内嵌 AES 与 SHA256 算法加速器，为用户提供基本安全功能。K210 拥有高性能、低功耗的 SRAM，以及功能强大的 DMA，在数据吞吐能力方面性能优异。K210 具备丰富的外设单元，分别是：DVP、JTAG、OTP、FPIOA、GPIO、UART、SPI、RTC、I²S、I²C、WDT、Timer 与 PWM，可满足海量应用场景。

1.2.3 系统架构

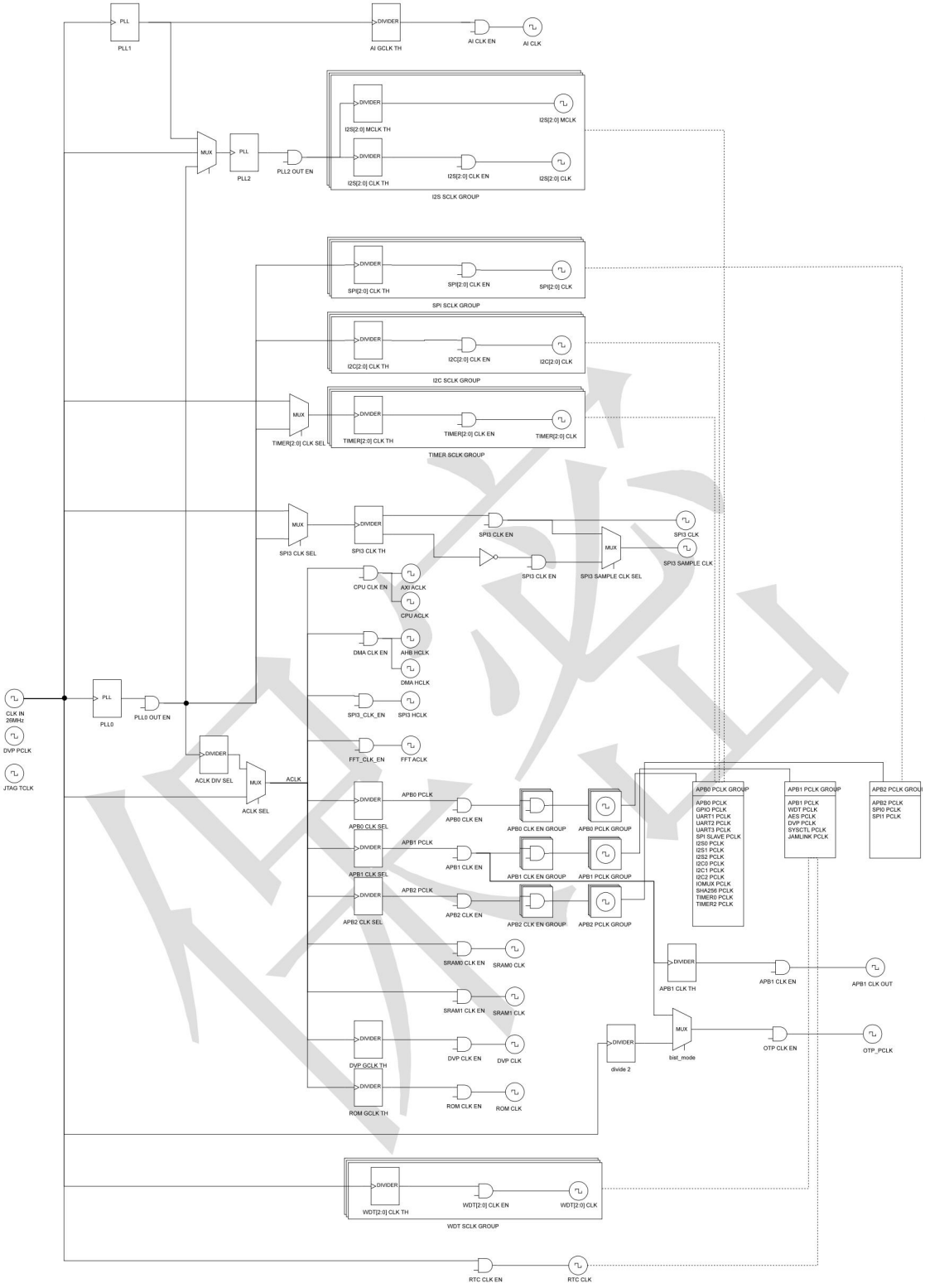


1.2.4 内存映射

Bus	Start Address	Length	Peripherals
AXI0	0x50000000	0x200000	To AHB
	0x50200000	0x200000	To APB0
	0x50400000	0x200000	To APB1
	0x80000000	0x10000000	TO TILELINK
	0x40000000	0x10000000	TO AXI1
	0x40800000	0xc00000	To AI
	0x42000000	0x400000	To FFT
AXI1	0x80000000	0x400000	MEM0 CACHE
	0x40000000	0x400000	MEM0 IO
	0x80400000	0x200000	MEM1 CACHE
	0x40400000	0x200000	MEM1 IO
	0x80600000	0x200000	AI CACHE
	0x40600000	0x600000	AI IO
	0x88000000	0x1000000	ROM
AHB	0x50000000	0xc00	DMAC
	0x54000000	0x2000000	SPI3
APB0	0x50200000	0x10000	GPIO
	0x50210000	0x10000	UART1
	0x50220000	0x10000	UART2
	0x50230000	0x10000	UART3
	0x50240000	0x10000	SPI SLAVE
	0x50250000	0x10000	I2S0
	0x50260000	0x10000	I2S1
	0x50270000	0x10000	I2S2
	0x50280000	0x10000	I2C0
	0x50290000	0x10000	I2C1
	0x502a0000	0x10000	I2C2
	0x502b0000	0x10000	FPIOA
	0x502c0000	0x10000	SHA256
	0x502d0000	0x10000	TIMER0

	0x502e0000	0x10000	TIMER1
	0x502f0000	0x10000	TIMER2
APB1	0x50400000	0x10000	WDT0
	0x50410000	0x10000	WDT1
	0x50430000	0x10000	DVP
	0x50440000	0x10000	SYSCTL
	0x50450000	0x10000	AES
	0x50460000	0x10000	RTC
	0x50470000	0x10000	EMPTY
APB2	0x52000000	0x1000000	SPI0
	0x53000000	0x1000000	SPI1
TINELINK	0x38000000	0x1000	UART0
	0x38001000	0x1000	GPIOHS

1.2.5 时钟树


PLL
PLL1
DIVIDER
AI GCLK TH
AI CLK EN
AI CLK
I2S[2:0] MCLK TH
I2S[2:0] MCLK
MUX
PLL
PLL2
PLL2 OUT EN
I2S[2:0] CLK TH
I2S[2:0] CLK EN
I2S[2:0] CLK
I2S SCLK GROUP
SPI[2:0] CLK TH
SPI[2:0] CLK EN
SPI[2:0] CLK
SPI SCLK GROUP
I2C[2:0] CLK TH
I2C[2:0] CLK EN
I2C[2:0] CLK
I2C SCLK GROUP
TIMER[2:0] CLK SEL
TIMER[2:0] CLK TH
TIMER[2:0] CLK EN
TIMER[2:0] CLK
TIMER SCLK GROUP
SPI3 CLK SEL
SPI3 CLK TH
SPI3 CLK EN
SPI3 CLK
SPI3 SAMPLE CLK
SPI3 SAMPLE CLK SEL
CPU CLK EN
AXI ACLK
CPU ACLK
DMA CLK EN
AHB HCLK
DMA HCLK
SPI3_CLK_EN
SPI3 HCLK
FFT_CLK_EN
FFT ACLK
CLK IN 26MHz
DVP PCLK
JTAG TCLK
PLL0
PLL0 OUT EN
ACLK DIV SEL
ACLK SEL
ACLK
APB0 CLK SEL
APB0 PCLK
APB0 CLK EN
APB0 CLK EN GROUP
APB0 PCLK GROUP
APB1 CLK SEL
APB1 PCLK
APB1 CLK EN
APB1 CLK EN GROUP
APB1 PCLK GROUP
APB2 CLK SEL
APB2 PCLK
APB2 CLK EN
APB2 CLK EN GROUP
APB2 PCLK GROUP
SRAM0 CLK EN
SRAM0 CLK
SRAM1 CLK EN
SRAM1 CLK
DVP GCLK TH
DVP CLK EN
DVP CLK
ROM GCLK TH
ROM CLK EN
ROM CLK
divide 2
bist_mode
APB1 CLK TH
APB1 CLK EN
APB1 CLK OUT
OTP CLK EN
OTP_PCLK
WDT[2:0] CLK TH
WDT[2:0] CLK EN
WDT[2:0] CLK
WDT SCLK GROUP
RTC CLK EN
RTC CLK

使用一路外部低频时钟，并使用三路PLL提供高频时钟。

PLL0为CPU与大部分外设提供时钟

PLL1为AI加速核提供时钟

PLL2为I2S音频提供时钟

其中，AI 只能选择PLL1作为时钟源，I2S MCLK只能选择PLL2作为时钟源，其余外设可以在外部时钟与PLL0之间进行选择(通过PLL BYPASS来选择外部时钟)。而PLL2可以与PLL0级联，以获得更精准的音频采样频率。

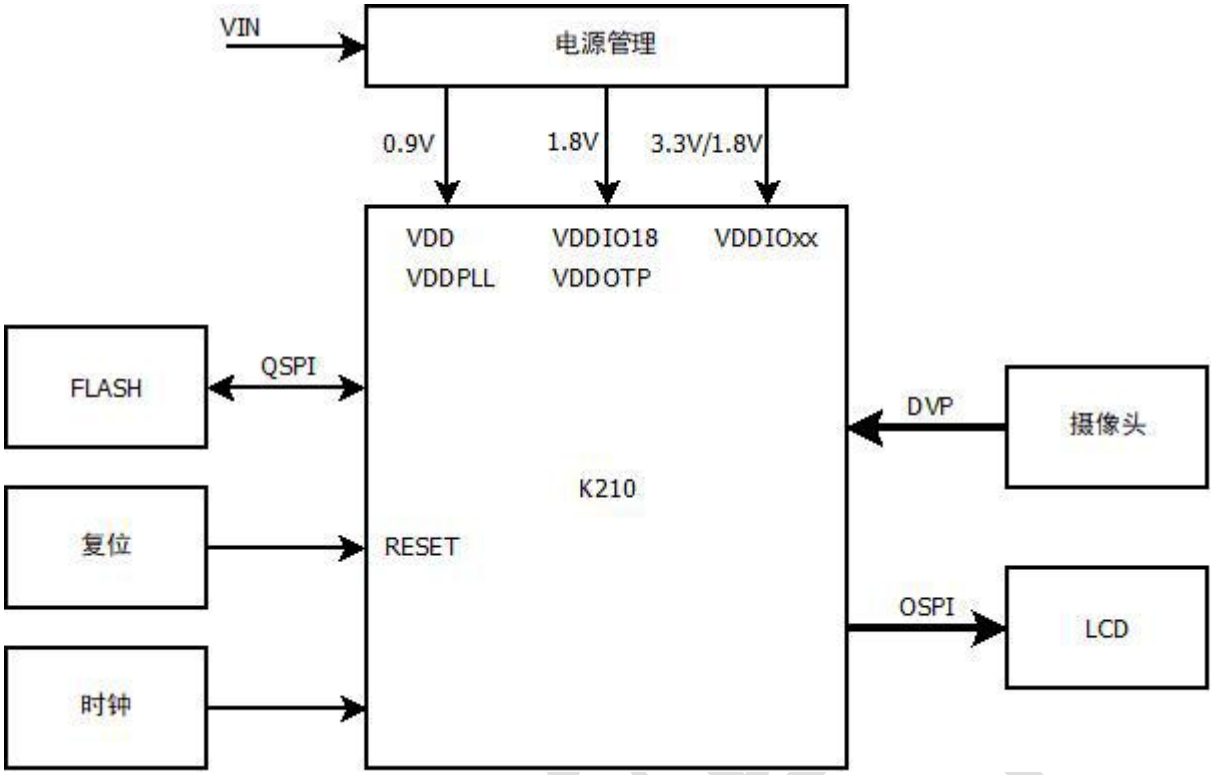
1.3 应用

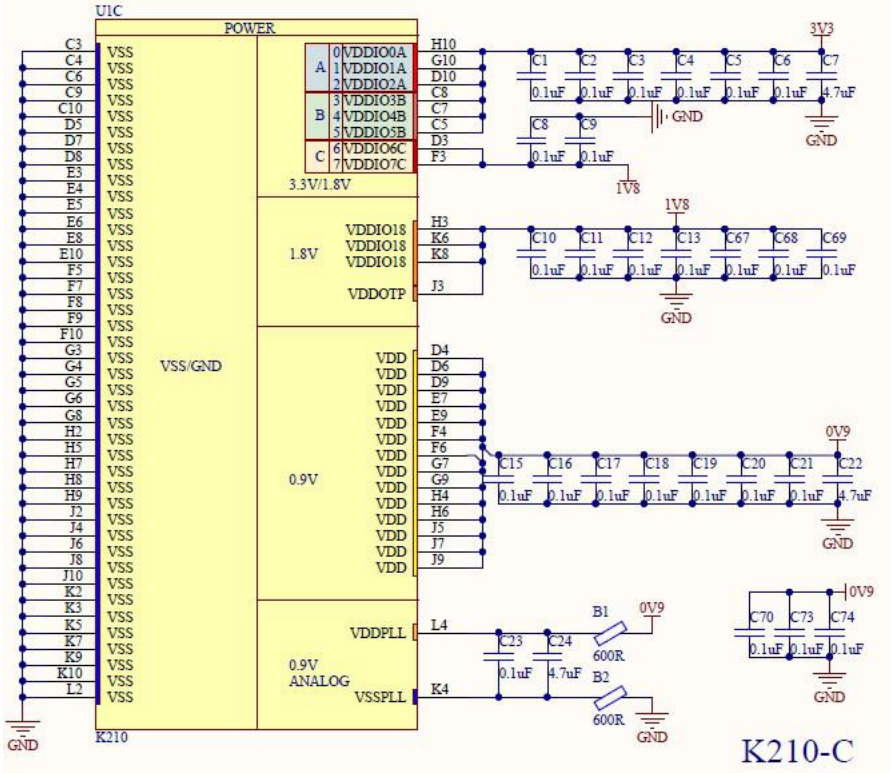
1.3.1 应用场景

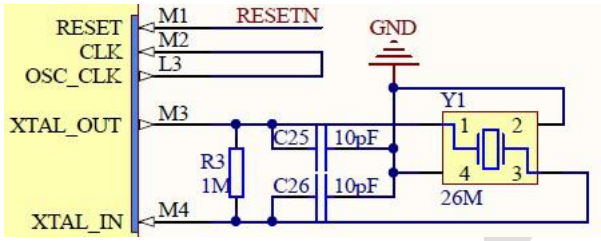
K210可应用于人脸识别、图像识别场景和声源定位场景：

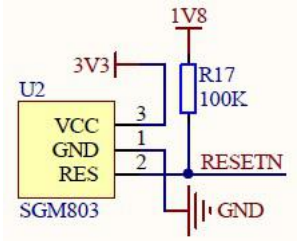
- 智能家居：智能门锁；
- 智慧楼宇
  - ✓ 人脸门禁识别
  - ✓ 无感门禁
- 智慧工厂：生产线包装袋计数
- 智慧抄表：图像识别抄表
- 智慧医疗【戴口罩场景】
- STEAM教育
- 等等

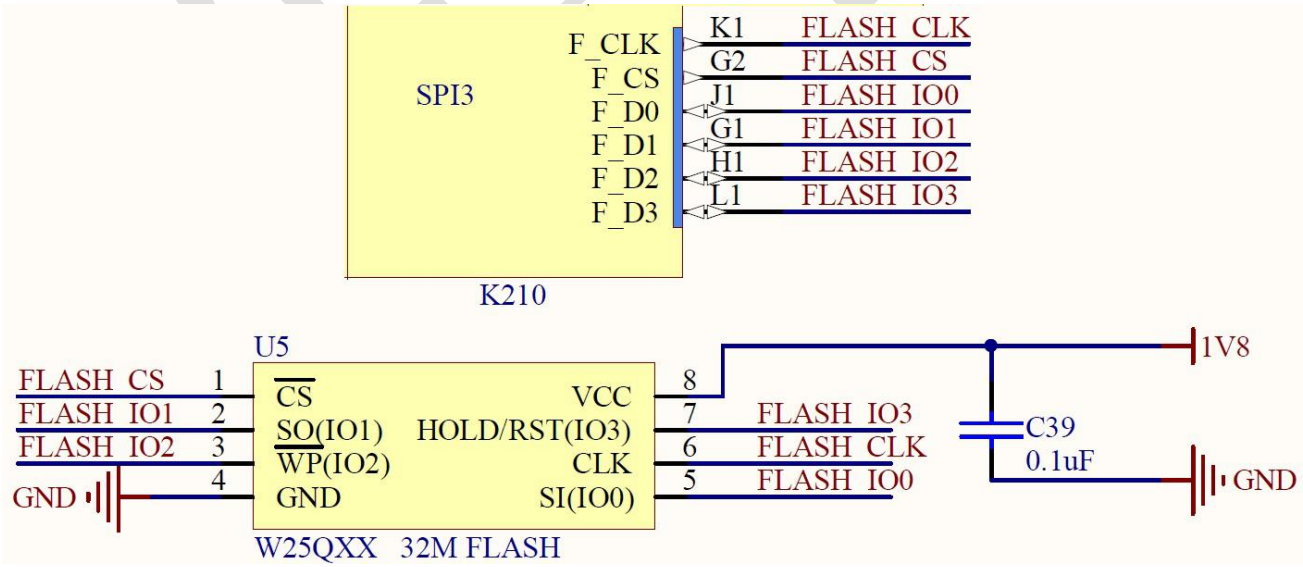
1.3.2 应用电路示例

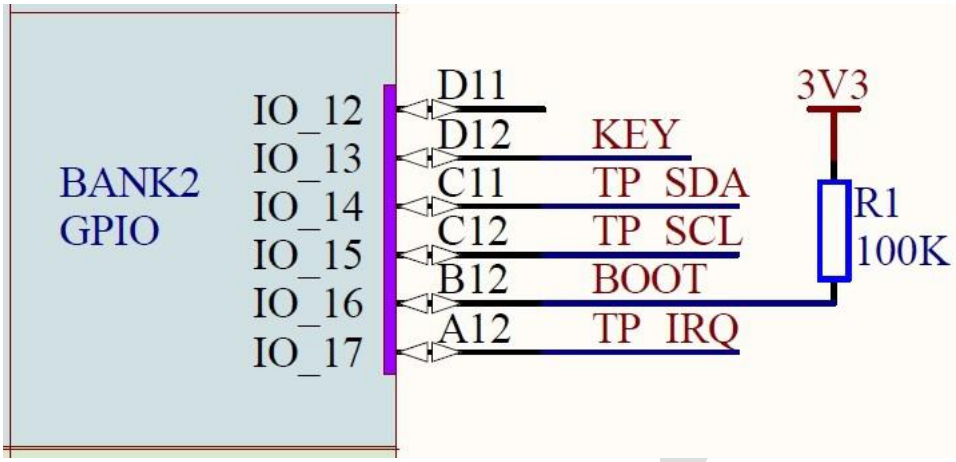

VIN
电源管理
0.9V
1.8V
3.3V/1.8V
VDD
VDDIO18
VDDIOxx
VDDPLL
VDDOTP
FLASH
QSPI
DVP
摄像头
K210
复位
RESET
OSPI
LCD
时钟

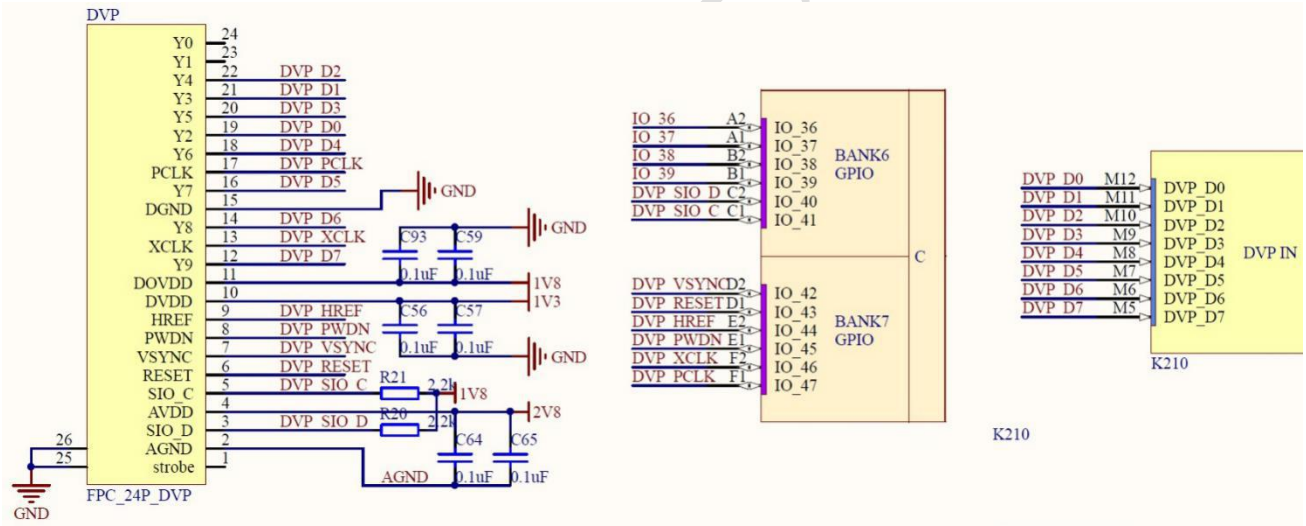

U1C
POWER
VSS/GND
3.3V/1.8V
1.8V
0.9V
0.9V ANALOG
VDDIO0A
VDDIO1A
VDDIO2A
VDDIO3B
VDDIO4B
VDDIO5B
VDDIO6C
VDDIO7C
VDDIO18
VDDOTP
VDD
VDDPLL
VSSPLL
3V3
1V8
0V9
GND
600R
K210
K210-C

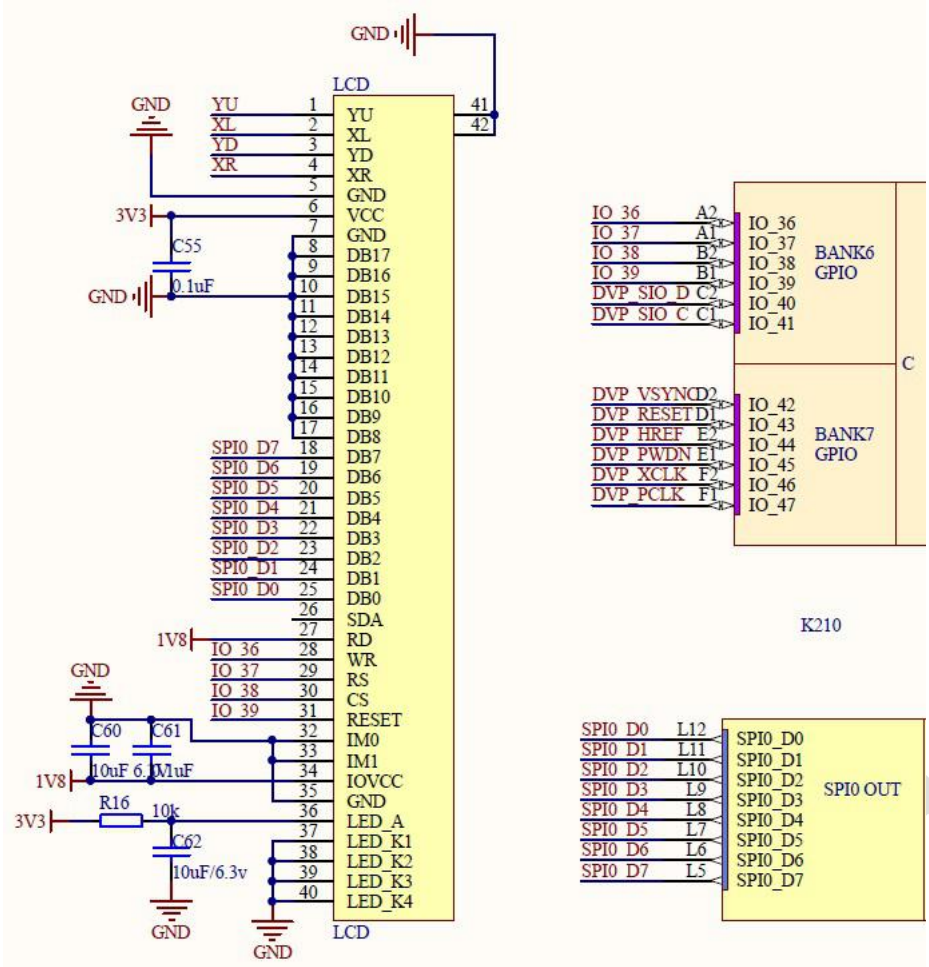

RESET
CLK
OSC_CLK
XTAL_OUT
XTAL_IN
M1
M2
L3
M3
M4
RESETN
GND
Y1
C25
10pF
C26
10pF
R3
1M
26M


1V8
3V3
U2
VCC
GND
RES
SGM803
R17
100K
RESETN
GND


SPI3
F_CLK
F_CS
F_D0
F_D1
F_D2
F_D3
K1 FLASH_CLK
G2 FLASH_CS
J1 FLASH_IO0
G1 FLASH_IO1
H1 FLASH_IO2
L1 FLASH_IO3
K210
U5
FLASH_CS 1
FLASH_IO1 2
FLASH_IO2 3
GND 4
CS
SO(IO1)
WP(IO2)
GND
VCC
HOLD/RST(IO3)
CLK
SI(IO0)
8
7
6
5
FLASH_IO3
FLASH_CLK
FLASH_IO0
1V8
C39
0.1uF
GND
W25QXX 32M FLASH


BANK2
GPIO
IO_12
IO_13
IO_14
IO_15
IO_16
IO_17
D11
D12
C11
C12
B12
A12
KEY
TP_SDA
TP_SCL
BOOT
TP_IRQ
3V3
R1
100K


DVP
FPC_24P_DVP
DVP_D2
DVP_D1
DVP_D3
DVP_D0
DVP_D4
DVP_PCLK
DVP_D5
DVP_D6
DVP_XCLK
DVP_D7
DVP_HREF
DVP_PWDN
DVP_VSYNC
DVP_RESET
DVP_SIO_C
DVP_SIO_D
GND
1V8
1V3
2V8
AGND
C93
C59
C56
C57
C64
C65
R21
R20
2.2k
0.1uF
IO_36
IO_37
IO_38
IO_39
DVP_SIO_D
DVP_SIO_C
IO_40
IO_41
BANK6
GPIO
DVP_VSYNC
DVP_RESET
DVP_HREF
DVP_PWDN
DVP_XCLK
DVP_PCLK
IO_42
IO_43
IO_44
IO_45
IO_46
IO_47
BANK7
GPIO
K210
DVP IN



1.3.3 推荐接入外设

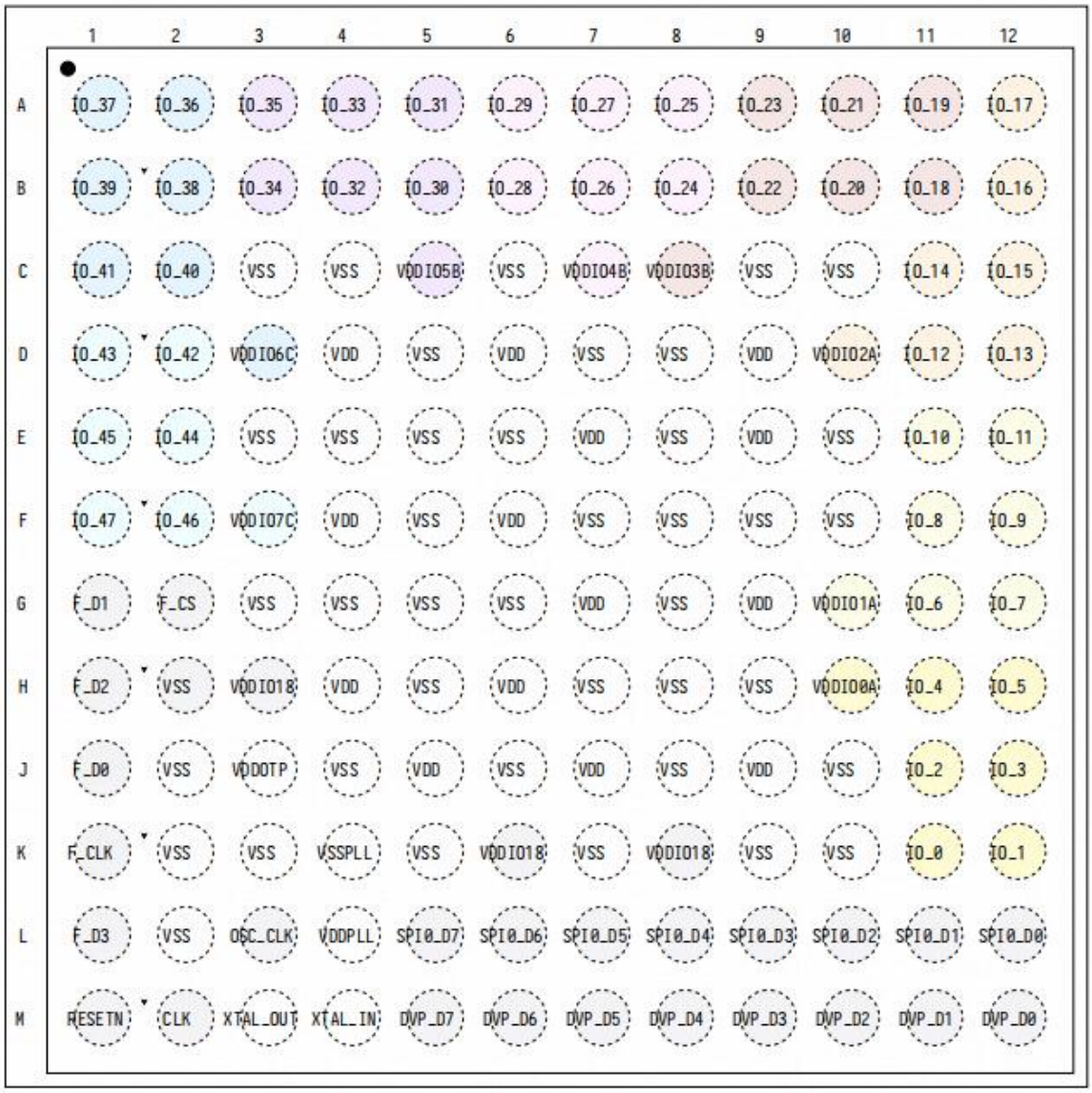
芯片	外设	推荐型号
K210芯片	摄像头	OV7725
		GC0308
	电源单元	TPS65266
		RY1303
	Wifi单元	W600【SPI接口】

2 引脚定义

K210 使用精心设计的引脚布局，确保信号都在 BGA 外圈，以方便 PCB 工程师进行扇出与布线，提升电气性能，降低设计难度。由于 K210 包含多种电源域的 IO 信号，并且不同电源域可能会有不同的电压，以下将会对使用的电源域进行列表说明：

电源域组	电源域	支持电压（V）	互联特性	电源名称
A	0	3.3 或 1.8	组内互联，组间独立	VDDIO0A
A	1	3.3 或 1.8	组内互联，组间独立	VDDIO1A
A	2	3.3 或 1.8	组内互联，组间独立	VDDIO2A
B	3	3.3 或 1.8	组内互联，组间独立	VDDIO3B
B	4	3.3 或 1.8	组内互联，组间独立	VDDIO4B
B	5	3.3 或 1.8	组内互联，组间独立	VDDIO5B
C	6	3.3 或 1.8	组内互联，组间独立	VDDIO6C
C	7	3.3 或 1.8	组内互联，组间独立	VDDIO7C
低压 IO	低压 IO	1.8	无特殊要求	VDDIO18
OTP	OTP	1.8	无特殊要求	VDDOTP
PLL	PLL	0.9	无特殊要求	VDDPLL
数字核心	数字核心	0.9	无特殊要求	VDD

2.1 引脚布局



注意：芯片的引脚定义如上图（顶视图，锡球朝向下方）。该芯片使用 BGA144 封装，正方形，每一边有 12 个引脚。芯片宽度为 8mm，长度为 8mm，高度为 0.953mm。

2.2 引脚描述

编号	名称	类型	功能	复位后初始状态
A1	IO_37	I/O	可编程 IO 阵列（FPIOA）的多功能 IO（电源域 6，组 C）	GPIOHS21
A2	IO_36	I/O	可编程 IO 阵列（FPIOA）的多功能 IO（电源域 6，组 C）	GPIOHS20
A3	IO_35	I/O	可编程 IO 阵列（FPIOA）的多功能 IO（电	GPIOHS19

			源域 5，组 B）	
A4	IO_33	I/O	可编程 IO 阵列（FPIOA）的多功能 IO（电源域 5，组 B）	GPIOHS17
A5	IO_31	I/O	可编程 IO 阵列（FPIOA）的多功能 IO（电源域 5，组 B）	GPIOHS15
A6	IO_29	I/O	可编程 IO 阵列（FPIOA）的多功能 IO（电源域 4，组 B）	GPIOHS13
A7	IO_27	I/O	可编程 IO 阵列（FPIOA）的多功能 IO（电源域 4，组 B）	GPIOHS11
A8	IO_25	I/O	可编程 IO 阵列（FPIOA）的多功能 IO（电源域 4，组 B）	GPIOHS9
A9	IO_23	I/O	可编程 IO 阵列（FPIOA）的多功能 IO（电源域 3，组 B）	GPIOHS7
A10	IO_21	I/O	可编程 IO 阵列（FPIOA）的多功能 IO（电源域 3，组 B）	GPIOHS5
A11	IO_19	I/O	可编程 IO 阵列（FPIOA）的多功能 IO（电源域 3，组 B）	GPIOHS3
A12	IO_17	I/O	可编程 IO 阵列（FPIOA）的多功能 IO（电源域 2，组 A）	GPIOHS1
B1	IO_39	I/O	可编程 IO 阵列（FPIOA）的多功能 IO（电源域 6，组 C）	GPIOHS23
B2	IO_38	I/O	可编程 IO 阵列（FPIOA）的多功能 IO（电源域 6，组 C）	GPIOHS22
B3	IO_34	I/O	可编程 IO 阵列（FPIOA）的多功能 IO（电源域 5，组 B）	GPIOHS18
B4	IO_32	I/O	可编程 IO 阵列（FPIOA）的多功能 IO（电源域 5，组 B）	GPIOHS16
B5	IO_30	I/O	可编程 IO 阵列（FPIOA）的多功能 IO（电源域 5，组 B）	GPIOHS14
B6	IO_28	I/O	可编程 IO 阵列（FPIOA）的多功能 IO（电源域 4，组 B）	GPIOHS12
B7	IO_26	I/O	可编程 IO 阵列（FPIOA）的多功能 IO（电	GPIOHS10

			源域 4，组 B）	
B8	IO_24	I/O	可编程 IO 阵列（FPIOA）的多功能 IO（电源域 4，组 B）	GPIOHS8
B9	IO_22	I/O	可编程 IO 阵列（FPIOA）的多功能 IO（电源域 3，组 B）	GPIOHS6
B10	IO_20	I/O	可编程 IO 阵列（FPIOA）的多功能 IO（电源域 3，组 B）	GPIOHS4
B11	IO_18	I/O	可编程 IO 阵列（FPIOA）的多功能 IO（电源域 3，组 B）	GPIOHS2
B12	IO_16	I/O	可编程 IO 阵列（FPIOA）的多功能 IO（电源域 2，组 A）	GPIOHS0（ISP）
C1	IO_41	I/O	可编程 IO 阵列（FPIOA）的多功能 IO（电源域 6，组 C）	GPIOHS25
C2	IO_40	I/O	可编程 IO 阵列（FPIOA）的多功能 IO（电源域 6，组 C）	GPIOHS24
C3	VSS	S	接地	VSS
C4	VSS	S	接地	VSS
C5	VDDIO5B	S	3.3V/1.8V 电源，为 FPIOA 多功能 IO 供电（电源域 5，组 B）	VDDIO33
C6	VSS	S	接地	VSS
C7	VDDIO4B	S	3.3V/1.8V 电源，为 FPIOA 多功能 IO 供电（电源域 4，组 B）	VDDIO33
C8	VDDIO3B	S	3.3V/1.8V 电源，为 FPIOA 多功能 IO 供电（电源域 3，组 B）	VDDIO33
C9	VSS	S	接地	VSS
C10	VSS	S	接地	VSS
C11	IO_14	I/O	可编程 IO 阵列（FPIOA）的多功能 IO（电源域 2，组 A）	GPIO6
C12	IO_15	I/O	可编程 IO 阵列（FPIOA）的多功能 IO（电源域 2，组 A）	GPIO7
D1	IO_43	I/O	可编程 IO 阵列（FPIOA）的多功能 IO（电源域 7，组 C）	GPIOHS27

D2	IO_42	I/O	可编程 IO 阵列（FPIOA）的多功能 IO（电源域 7，组 C）	GPIOHS26
D3	VDDIO6C	S	3.3V/1.8V 电源，为 FPIOA 多功能 IO 供电（电源域 6，组 C）	VDDIO33
D4	VDD	S	0.9V 电源，为芯片数字核心供电	VDD
D5	VSS	S	接地	VSS
D6	VDD	S	0.9V 电源，为芯片数字核心供电	VDD
D7	VSS	S	接地	VSS
D8	VSS	S	接地	VSS
D9	VDD	S	0.9V 电源，为芯片数字核心供电	VDD
D10	VDDIO2A	S	3.3V/1.8V 电源，为 FPIOA 多功能 IO 供电（电源域 2，组 A）	VDDIO33
D11	IO_12	I/O	可编程 IO 阵列（FPIOA）的多功能 IO（电源域 2，组 A）	GPIO4
D12	IO_13	I/O	可编程 IO 阵列（FPIOA）的多功能 IO（电源域 2，组 A）	GPIO5
E1	IO_45	I/O	可编程 IO 阵列（FPIOA）的多功能 IO（电源域 7，组 C）	GPIOHS29
E2	IO_44	I/O	可编程 IO 阵列（FPIOA）的多功能 IO（电源域 7，组 C）	GPIOHS28
E3	VSS	S	接地	VSS
E4	VSS	S	接地	VSS
E5	VSS	S	接地	VSS
E6	VSS	S	接地	VSS
E7	VDD	S	0.9V 电源，为芯片数字核心供电	VDD
E8	VSS	S	接地	VSS
E9	VDD	S	0.9V 电源，为芯片数字核心供电	VDD
E10	VSS	S	接地	VSS
E11	IO_10	I/O	可编程 IO 阵列（FPIOA）的多功能 IO（电源域 1，组 A）	GPIO2
E12	IO_11	I/O	可编程 IO 阵列（FPIOA）的多功能 IO（电源域 1，组 A）	GPIO3

F1	IO_47	I/O	可编程 IO 阵列（FPIOA）的多功能 IO（电源域 7，组 C）	GPIOHS31
F2	IO_46	I/O	可编程 IO 阵列（FPIOA）的多功能 IO（电源域 7，组 C）	GPIOHS30
F3	VDDIO7C	S	3.3V/1.8V 电源，为 FPIOA 多功能 IO 供电（电源域 7，组 C）	VDDIO33
F4	VDD	S	0.9V 电源，为芯片数字核心供电	VDD
F5	VSS	S	接地	VSS
F6	VDD	S	0.9V 电源，为芯片数字核心供电	VDD
F7	VSS	S	接地	VSS
F8	VSS	S	接地	VSS
F9	VSS	S	接地	VSS
F10	VSS	S	接地	VSS
F11	IO_8	I/O	可编程 IO 阵列（FPIOA）的多功能 IO（电源域 1，组 A）	GPIO0
F12	IO_9	I/O	可编程 IO 阵列（FPIOA）的多功能 IO（电源域 1，组 A）	GPIO1
G1	F_D1	I/O	SPI 专用 GPIO（支持电平为 1.8V，不可切换）	F_D1
G2	F_CS	O	SPI 专用 GPIO（支持电平为 1.8V，不可切换）	F_CS
G3	VSS	S	接地	VSS
G4	VSS	S	接地	VSS
G5	VSS	S	接地	VSS
G6	VSS	S	接地	VSS
G7	VDD	S	0.9V 电源，为芯片数字核心供电	VDD
G8	VSS	S	接地	VSS
G9	VDD	S	0.9V 电源，为芯片数字核心供电	VDD
G10	VDDIO1A	S	3.3V/1.8V 电源，为 FPIOA 多功能 IO 供电（电源域 1，组 A）	VDDIO33
G11	IO_6	I/O	可编程 IO 阵列（FPIOA）的多功能 IO（电源域 1，组 A）	(FLOAT*)

G12	IO_7	I/O	可编程 IO 阵列（FPIOA）的多功能 IO（电源域 1，组 A）	(FLOAT*)
H1	F_D2	I/O	SPI 专用 GPIO（支持电平为 1.8V，不可切换）	F_D2
H2	VSS	S	接地	VSS
H3	VDDIO18	S	1.8V 电源，为低压 GPIO 供电	VDDIO18
H4	VDD	S	0.9V 电源，为芯片数字核心供电	VDD
H5	VSS	S	接地	VSS
H6	VDD	S	0.9V 电源，为芯片数字核心供电	VDD
H7	VSS	S	接地	VSS
H8	VSS	S	接地	VSS
H9	VSS	S	接地	VSS
H10	VDDIO0A	S	3.3V/1.8V 电源，为 FPIOA 多功能 IO 供电(电源域 0，组 A)	VDDIO33
H11	IO_4	I/O	可编程 IO 阵列（FPIOA）的多功能 IO（电源域 0，组 A）	UARTHS_RX（ISP）
H12	IO_5	I/O	可编程 IO 阵列（FPIOA）的多功能 IO（电源域 0，组 A）	UARTHS_TX（ISP）
J1	F_D0	I/O	SPI 专用 GPIO（支持电平为 1.8V，不可切换）	F_D0
J2	VSS	S	接地	VSS
J3	VDDOTP	S	1.8V 电源，为一次性可编程存储器（OTP）供电	VDDOTP
J4	VSS	S	接地	VSS
J5	VDD	S	0.9V 电源，为芯片数字核心供电	VDD
J6	VSS	S	接地	VSS
J7	VDD	S	0.9V 电源，为芯片数字核心供电	VDD
J8	VSS	S	接地	VSS
J9	VDD	S	0.9V 电源，为芯片数字核心供电	VDD
J10	VSS	S	接地	VSS
J11	IO_2	I/O	可编程 IO 阵列（FPIOA）的多功能 IO（电源域 0，组 A）	JTAG_TMS

J12	IO_3	I/O	可编程 IO 阵列（FPIOA）的多功能 IO（电源域 0，组 A）	JTAG_TDO
K1	F_CLK	O	SPI 专用 GPIO（支持电平为 1.8V，不可切换）	F_CLK
K2	VSS	S	接地	VSS
K3	VSS	S	接地	VSS
K4	VSSPLL	S	接模拟地，锁相环（PLL）使用，噪声敏感	VSSPLL
K5	VSS	S	接地	VSS
K6	VDDIO18	S	1.8V 电源，为低压 GPIO 供电	VDDIO18
K7	VSS	S	接地	VSS
K8	VDDIO18	S	1.8V 电源，为低压 GPIO 供电	VDDIO18
K9	VSS	S	接地	VSS
K10	VSS	S	接地	VSS
K11	IO_0	I/O	可编程 IO 阵列（FPIOA）的多功能 IO（电源域 0，组 A）	JTAG_TCLK
K12	IO_1	I/O	可编程 IO 阵列（FPIOA）的多功能 IO（电源域 0，组 A）	JTAG_TDI
L1	F_D3	I/O	SPI 专用 GPIO（支持电平为 1.8V，不可切换）	F_D3
L2	VSS	S	接地	VSS
L3	OSC_CLK	O	有源振荡器输出，时钟来源于外部晶体振荡器	OSC_CLK
L4	VDDPLL	S	0.9V 模拟电源，为锁相环（PLL）供电	VDDPLL
L5	SPIO_D7	O	输出专用引脚，用于 SPIO D7 输出	(FLOAT*)
L6	SPIO_D6	O	输出专用引脚，用于 SPIO D6 输出	(FLOAT*)
L7	SPIO_D5	O	输出专用引脚，用于 SPIO D5 输出	(FLOAT*)
L8	SPIO_D4	O	输出专用引脚，用于 SPIO D4 输出	(FLOAT*)
L9	SPIO_D3	O	输出专用引脚，用于 SPIO D3 输出	(FLOAT*)
L10	SPIO_D2	O	输出专用引脚，用于 SPIO D2 输出	(FLOAT*)
L11	SPIO_D1	O	输出专用引脚，用于 SPIO D1 输出	(FLOAT*)
L12	SPIO_D0	O	输出专用引脚，用于 SPIO D0 输出	(FLOAT*)

M1	RESET	I	系统复位引脚，低电平复位	RESET
M2	CLK	I	系统时钟输入	CLK
M3	XTAL_OUT	O	无源晶体振荡器输出脚。非失效安全，禁止灌入有源信号	XTAL_OUT
M4	XTAL_IN	I	无源晶体振荡器输入脚。非失效安全，禁止灌入有源信号	XTAL_IN
M5	DVP_D7	I	输入专用引脚，用于 DVP D7 输入	(FLOAT*)
M6	DVP_D6	I	输入专用引脚，用于 DVP D6 输入	(FLOAT*)
M7	DVP_D5	I	输入专用引脚，用于 DVP D5 输入	(FLOAT*)
M8	DVP_D4	I	输入专用引脚，用于 DVP D4 输入	(FLOAT*)
M9	DVP_D3	I	输入专用引脚，用于 DVP D3 输入	(FLOAT*)
M10	DVP_D2	I	输入专用引脚，用于 DVP D2 输入	(FLOAT*)
M11	DVP_D1	I	输入专用引脚，用于 DVP D1 输入	(FLOAT*)
M12	DVP_D0	I	输入专用引脚，用于 DVP D0 输入	(FLOAT*)

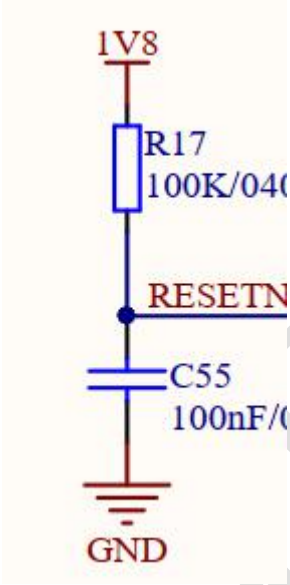
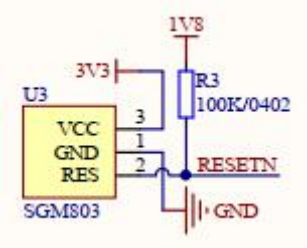
2.3 电源分配

电源域	电源名称	额定电压（V）	最大电流（mA）
I/O 3.3V/1.8V	VDDIO0A	3.3 或 1.8V*1	200
I/O 3.3V/1.8V	VDDIO1A	3.3 或 1.8V	200
I/O 3.3V/1.8V	VDDIO2A	3.3 或 1.8V	200
I/O 3.3V/1.8V	VDDIO3B	3.3 或 1.8V	200
I/O 3.3V/1.8V	VDDIO4B	3.3 或 1.8V	200
I/O 3.3V/1.8V	VDDIO5B	3.3 或 1.8V	200
I/O 3.3V/1.8V	VDDIO6C	3.3 或 1.8V	200
I/O 3.3V/1.8V	VDDIO7C	3.3 或 1.8V	200
I/O 1.8V	VDDIO18	1.8	200
OTP 1.8V	VDDOTP	1.8	50
Core 0.9V	VDD	0.9	2000
SoC	VSS	0	-
PLL 0.9V	VDDPLL	0.9	15
PLL	VSSPLL	0	-

注意：组 A、B、C 之间的 IO 电源相互不互联，电压可以不一致；相同组内的 IO 电源 互联，电压一致。

2.4 复位电路

复位电路建议采用 1.8V 输出的 MCU 专用电源监控芯片，在上电、断电和欠压条件下保证稳定复位。复位时间：上电稳定时间到时钟稳定时间大于10ms;

RC复位	专用电源监控芯片复位
电阻 R = 100 KΩ 电容 C = 100 nF	推荐芯片SMG803
	

2.5 特殊引脚

IO_16 用于 boot 模式选择，上电复位时，拉高进入 FLASH 启动，拉低进入 ISP 模式。复位后，IO_0、IO_1、IO_2、IO_3 为 JTAG 引脚。IO_4、IO_5 为 ISP 引脚。

3 寄存器说明

3.1 中央处理器 (CPU)

3.1.1 概述

Kendryte K210搭载 RISC-V ISA 的双核心 64 位的高性能低功耗CPU，配置了4x 64位缓存一致性RISC-V应用程序核心。每个K210核心都有一个高性能的单问题顺序64位执行管道，峰值持续执行率为每个时钟周期一个结构,频率可达400MHz。K210核心支持综合动态分支预测方案，包括BTB、BHT和利用本地和全局历史来提高性能的返回地址堆栈,核心支持标准RV64IMAFDC ISA，包括对单精度和双精度IEEE 754-2008浮点的全硬件支持，全流水线的融合乘加、硬件除法和平方根单元，以及对子数字的全硬件支持。还提供了一种硬件整数乘法器和分法器。K210核心支持标准的C压缩扩展以减少代码大小。K210核心使用Sv39虚拟ad- dress转换方案实现了高达512 GiB的虚拟地址空间，并带有一个用于地址转换缓存填充的硬件页表walker。

3.1.2 功能描述

- 高级中断管理能力

该 RISC-V CPU 的 PLIC 控制器支持非常灵活的高级中断管理，可分 7 个优先级灵活配置 64 个外部中断源，两个核心都可独立进行配置:

1) 可对两个核心独立进行中断管理与中断路由控制；
2) 支持软件中断，并且双核心可以相互触发核间中断；
3) 支持 CPU 内置定时器中断，两个核心都可自由配置；
4) 高级外部中断管理，支持 64 个外部中断源，每个中断源可配置 7 个优先级。

- FPU与浮点计算能力

1) FPU 满足 IEEE754-2008 标准，计算流程以流水线方式进行，具备很强的运算能力；

2) 核心 0 与核心 1 各具备独立 FPU，两个核心皆可胜任高性能硬件浮点计算；

3) 支持 F 扩展，即单精度浮点扩展，CPU 内嵌的 FPU 支持单精度浮点硬件加速；

4) 支持 D 扩展，即双精度浮点扩展，CPU 内嵌的 FPU 支持双精度浮点硬件加速；

5) FPU 具备除法器，支持单精度、双精度的浮点的硬件除法运算；

6) FPU 具备平方根运算器，支持单精度、双精度的浮点的硬件平方根运算。

- 调试能力

1) 支持性能监控指令，可统计指令执行周期；

2) 具备用以调试的高速 UART 与 JTAG 接口；

3) 支持 DEBUG 模式以及硬件断点。

- 存储系统

每个中央处理器的私有一级指令缓存和数据缓存配置为8路 set associative 32 KiB 缓存。监控核心具有双向集关联的 4kibll 指令缓存。所有芯片上的内存结构都使用奇偶校验或ECC进行保护， 并且RISC-V核心IP中的所有核心都具有物理内存保护(PMP)单元。

- 外部总线接口

中央处理器具有系统、内存和外设端口，可用于访问 Core Complex 地址空间。系统端口符合 TL-UH 规范，可用于访问高速离开核心的复杂设备。内存端口通过 TL-C 规范支持可缓存事务，并连接到主内存。外设端口支持 TL-UL 规范，通常连接到低速外设。还有一个 TL-C 总线接口，称为前端端口，它允许离开核心的复杂主机访问核心的复杂设备，如数据和指令紧密集成的存储器。

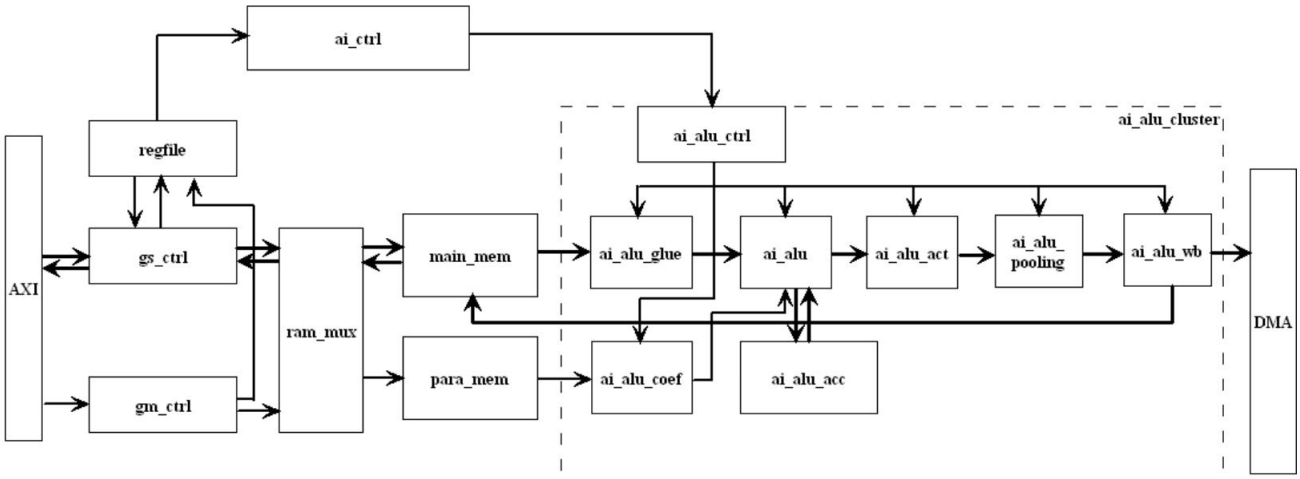
3.2 神经网络处理器 (KPU)

3.2.1 概述

KPU是通用的神经网络处理器，它可以在低功耗的情况下实现卷积神经网络计算，实时获取被检测目标的大小、坐标和种类，对人脸或者物体进行检测和分类。 主要特征包含：

- 支持计算多层卷积神经网络，每层卷积神经网络的控制参数可单独配置；
- 支持中断模式，可配置加速器在每层卷积结束后是否拉起中断；
- 支持输入图像片上存储，存储容量大小为2M字节，卷积结果可由DMA读出；
- 支持输入输出通道数目、输入输出图像行高列宽可配，其中通道数目范围在（1~1024）之间，输出行高列宽与输入相同，或者是输入除以2或者4且向下取整；
- 支持两种卷积内核1x1和3x3，卷积步长为1，十种池化方式，包括bypass、步长为1且大小为2x2的均值/最大值、步长为2且大小为2x2的均值/最大值/左上值/右上值、步长为4且大小为4x4的均值/最大值/左上值；
- 支持两种padding方式：任意填充和取最近值；
- 支持在输入图像行高超过256时，自动对卷积结果进行抽样，仅保留奇数行奇数列结果；
- 支持卷积参数、批归一化参数、激活参数可配，AI加速器主动读取，读取地址可配；
- 支持卷积参数片上存储，存储容量为72K字节，可以边卷积边读取卷积参数，每层网络最多可以读取64次；
- 支持mobilenet-V1；
- 实时工作时最大支持神经网络参数大小为 5.5MiB 到 5.9MiB；
- 非实时工作时最大支持网络参数大小为（Flash 容量-软件体积）。

3.2.3 功能描述



KPU包括了main_ctrl（顶层控制单元）、gs_ctrl（gs控制单元）、gm_ctrl（gm控制单元）、ram_mux（ram多路选择单元）、main_mem（主存储器）、para_ram（参数存储器）、regfile(参数解析单元)、alu_cluster（alu运算单元）等子单元。配置参数、原始图像数据由gs接口从AXI bus（AXI总线）上读入，另外alu运算所需的卷积参数、批归一化参数、激活参数、池化参数则由gm接口从AXI bus（AXI总线）中读入；接下来，配置参数由regfile（参数解析单元）解析分配给main_ctrl（顶层控制单元），由main_ctrl（顶层控制单元）来控制整个AI数据处理流程，原始图像数据经由ram_mux（ram多路选择单元）存储到main_mem（主存储器）中，供alu运算单元使用，卷积、批归一化、激活、池化参数也经由ram_mux（ram多路选择单元）存储到para_ram（参数存储器）中，为后续alu运算随时取用；待数据和参数准备完毕，main_ctrl（顶层控制单元）控制addr_gen（地址生成器）产生读地址，将读取的图像数据、卷积参数等与基本配置信息一起送入alu运算单元，再由alu运算单元依次执行卷积、批归一化、激活、池化操作，alu运算单元中的所有操作由alu_ctrl（算法控制单元）统一控制，最终处理结果由write_back（写回单元）写回到main_mem（主存储器）中，以便之后各卷积层的alu运算。最后一个卷积层结束运算且数据写回main_mem（主存储器）后，DMA可以读取最终处理结果。

3.2.3.1 gs模块

该模块实现的功能是从AXI总线上接收配置参数，每一层需要12个配置参数，gs模块最多支持一次接收13层的配置参数。另外，gs模块还会接收图像原始数据。该模块包含两个fifo（先入先出存储器），第一个fifo是cmd fifo(指令fifo)，也就是说所有的读写指令会存储到cmd fifo中，第二个fifo是resp fifo(响应fifo)，也就是说对读写指令的响应会存储到resp fifo中。如果AXI总线写，那么写数据会先写入cmd fifo中，然后送给regfile模块解析，或者ram_mux写主存储器，同时会写响应到resp fifo中；如果AXI总线读，那么读数据会先写入resp fifo中，然后由AXI总线读走。

3.2.3.2 gm 模块

该模块实现的功能是从AXI总线上读取卷积权重参数、批归一化计算参数、激活计算参数。

3.2.3.3 ram_mux模块

该模块实现的功能是对图像数据或者各类权重参数的读写操作进行分类，然后将读写命令分别发送给mem32Kx64B（主存储器）和para_ram（主存储器）。其中，输入输出图像数据将会被送到mem32Kx64B（主存储器）模块；与卷积权重、批归一化计算参数相关的读写操作将会被送到para_ram（主存储器）模块。

3.2.3.4 mem32Kx64B模块

该模块实现的功能是对输入和输出图像数据进行存储，这个RAM位宽是66Byte，下标定义为0~65，其中，下标1~64的位置可以存放图像数据，下标0和下标65位置存储的像素点稍有特殊，下标0位置可以拷贝上一行（前一地址）下标64位置的像素，下标65位置可以拷贝下一行（后一个地址）下标1位置的像素。这个RAM的深度是32K，该RAM大小为32K*64Byte，原始图像数据和alu运算后数据将会写入这个RAM的不同地址空间中。

3.2.3.5 para_ram模块

该模块实现的功能是对gm模块接收的卷积权重参数、批归一化计算参数进行存储。该模块包括两个深度4096*宽度144bit的RAM，以及一个深度1056*宽度

60bit的RAM。其中前两个RAM乒乓读写，实现卷积权重参数存储，第三个RAM用于存储批归一化计算参数。

3.2.3.6 ai_ctrl模块

该模块实现的功能是对读取配置参数、卷积参数、批归一化参数、激活参数、alu运算的整个流程进行控制，流程图如下：

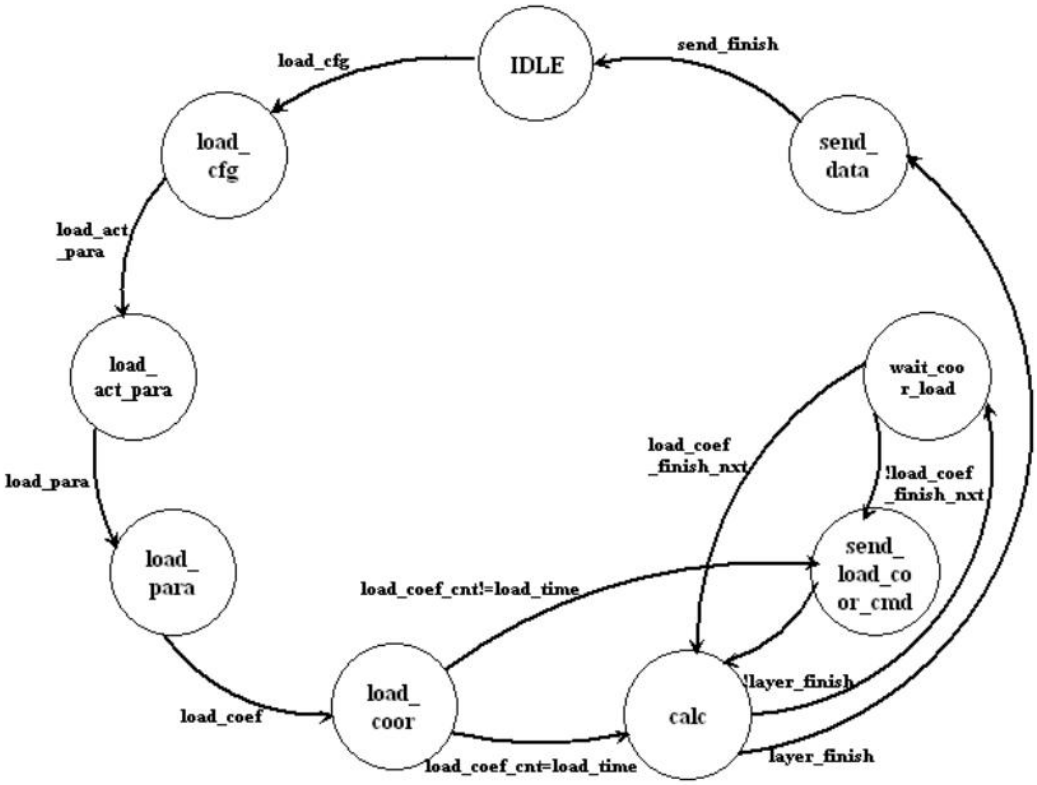


图　状态图

各个状态的解释如下：

- load_cfg状态：执行读取cfg参数，即配置参数；
- load_act_para状态：执行读取ACT参数，即激活计算参数；
- load_para状态：执行读取BN参数，即批归一化计算参数；
- load_coor状态：执行读取kernel参数，即卷积权重参数；
- Calc状态：进行卷积、批归一化、激活、池化运算；
- send_load_coor_cmd状态：在发送读卷积权重参数指令的同时，启动calc运算；
- wait_coor_load状态：在读取的卷积权重数量不满足本次卷积要求时，先跳转到send_load_coor_cmd状态，发送读卷积指令，再执行calc运算；

在读取的卷积权重数量满足本次卷积要求时，可以直接执行calc运算；

- send_data状态：可以将运算结果写入主存储器，同时在所有的卷积层运算结束后，将数据写入AXI总线。

3.2.3.7 alu_cluster模块

3.2.3.7.1 ai_alu_ctrl 模块

该模块实现的功能是对整个alu运算单元进行控制，控制读取输入图像和运算参数，执行alu运算，写回主存储器，写AXI的整个流程。

3.2.3.7.2 ai_alu_coef 模块

该模块实现的功能是对读取的卷积权重参数进行简单运算，例如累加求和，该结果将参与ai_alu的卷积运算。

3.2.3.7.3 ai_alu_acc 模块

该模块实现的功能是对每个输入通道的卷积结果进行缓存。由于多个输入通道卷积结果的累加和是一个输出通道的卷机值。每个输入通道的卷积结果会写入ai_alu_acc的RAM中，当前输入通道的卷积计算结果加上前一个输入通道的卷积结果，该求和结果才会写入ai_alu_acc的RAM中。

3.2.3.7.4 ai_alu_glue 模块

该模块实现的功能是读取主存储器中的数据，每一个时钟周期读取64Byte有效数据，同时将该数据分为64组，每组包含3Byte（3个像素点），这64组数据会被送入64个alu运算单元中进行并行处理。

3.2.3.7.5 ai_alu 模块

该模块实现的功能是对输入图像进行卷积、批归一化运算。每个输入通道的卷积结果与前一个输入通道的卷积结果求和后，会写入ai_alu_acc的RAM。当所有输入通道的卷积运算完成后，将会从ai_alu_acc中读取最终卷积结果，再执行批归一化运算。

3.2.3.7.6 ai_act 模块

该模块实现的功能是对数据进行激活运算。采用16个分段线性拟合激活函数，支持leaky/leaky relu等激活函数。

3.2.3.7.7 ai_pool 模块

该模块实现的功能是对数据进行池化，可以支持十种池化类型，包括bypass、步长为1且大小为2x2的均值/最大值、步长为2且大小为2x2的均值/最大值/左上值/右上值、步长为4且大小为4x4的均值/最大值/左上值。

3.2.3.7.8 ai_alu_wb 模块

该模块实现的功能是将每个输出通道的运算结果写入主存储器。另外，在最后一层运算结束后，可以将主存储器中已经写入的运算结果读出，由DMA读走该笔数据并写到AXI总线上。

3.2.4 KPU限制说明

卷积

- 只支持1x1,3x3 stride=1卷积，也可以实现3x3 stride =2卷积，实现方式是：卷积采用 3x3, stride= 1，pooling采用2x2, stride = 2 left_top.
- 可以bypass卷积，实现方式是：采用3x3 normal conv，kernel参数设置为常值 0，0，0；0，1，0；0，0，0.
- 卷积padding 仅支持padding一圈，可以一圈常值，或者mirror一圈.
- depth wise conv目前仅支持3x3, stride=1,如果是3x3,stride=2, 实现方式是：当前层只做 3x3,stride=1,下一层做2x2 ,stride=2 left_top pooling.

池化

- 2x2,stride=2仅支持输入行宽列高为2的倍数情况；4x4，stride=4 仅支持输入行宽列高为4的倍数情况.

Kernel 参数

- 16bit mode下，kernel参数每层size最大为4.5MByte = 64 * 72KByte；
- 8bit mode 下，kernel 参数每层size 最大为 4.46 MByte,同时需要保证最后一次加载参数size小于36KByte.

图像尺寸：

- 输入输出最小尺寸是 宽4*高4。

3.2.5 寄存器列表

Base Address：KPU_BASE_ADDR (0x40800000)

Name	Description	Offset address
KPU_INTERRUPT_ENABLE	set ping pong flag, interrupt enable	0x00
KPU_IMAGE_ADDR	addr for image input & output	0x00
KPU_CHANNEL_NUM	image channel number for input and output	0x00
KPU_IMAGE_SIZE	image size for input and output image	0x00
KPU_KERNEL_POOL_TYPE_CFG	pooling config	0x00
KPU_KERNEL_LOAD_CFG	pooling kernel loading configure	0x00
KPU_KERNEL_OFSET	pooling kernel data offset	0x00
KPU_KERNEL_CALC_TYPE_CFG	pooling switch addr	0x00
KPU_WRITE_BACK_CFG	write back	0x00
KPU_CONV_VALUE	conv parameter	0x00
KPU_CONV_VALUE2	conv parameter2	0x00
KPU_DMA_PARAMETER	dma_parameter	0x00
KPU_INTERRUPT_STATUS	interrupt_status	0x08
KPU_INTERRUPT_RAW	interrupt_raw	0x10
KPU_INTERRUPT_MASK	interrupt_mask	0x18
KPU_INTERRUPT_CLEAR	interrupt_clear	0x20
KPU_FIFO_THRESH	fifo_threshold	0x28
KPU_DMA_OUTPUT_PORT	fifo_data_out	0x30
KPU_FIFO_CTRL	fifo_ctrl	0x38
KPU_EIGHT_BIT_MODE	eight_bit_mode	0x40

注：上述表格中 DMA 表示直接存储器访问。

3.2.6 寄存器设置

3.2.6.1 KPU_INTERRUPT_ENABLE (0x00)

Reset value：0x00

Bits	Name	Description	Memory Access
63:4	RESERVED	Reserved	W
3	DEPTH_WISE_LAYER	set to 1: indicate depth_wise_layer; set to 0 : indicate normal layer	W
2	FULL_ADD	set to 1: indicate when calculate finish, dma output send accumulated value out, not the original output channel data, one accumulated channel data occupy 64bits, which is the accumulated value of all the byte data of one output channel	W
1	RESERVED	Reserved	W
0	INT_EN	set to 1: indicate pull up the interrupt when calculation finish	W

注：DEPTH_WISE_LAYER 表示 MOBILENET 网络中的 DEPTHWISE 卷积层

3.2.6.2 KPU_IMAGE_ADDR (0x00)

Reset value：0x00

Bits	Name	Description	Memory Access
63:47	RESERVED		W
46:32	IMAGE_DST_ADDR	set image dst address address * 64 is physical address	W
31:15	RESERVED		W
14:0	IMAGE_SRC_ADDR	set image src address, address * 64 is physical address;	W

3.2.6.3 KPU_CHANNEL_NUM (0x00)

Reset value：0x00

Bits	Name	Description	Memory Access
63:58	RESERVED		W
57:48	O_CH_NUM_COEF	output channel number corresponding to one weight parameter memory, since the weight is large for certain layer, accelerator need to load weight parameter from cpu memory for several times, which is set by paramter load_time in kernel_load_cfg register. The paramter o_ch_num_coef indicate one weight memory load can support how many output channel's calculation. Start from 0, 0 means one output channel	W
47:42	RESERVED		W
41:32	O_CH_NUM	output channel number, start from 0, 0 means 1 output channel	W
31:10	RESERVED		W
9:0	I_CH_NUM	input channel number, start from 0, 0 means 1 input channel	W

3.2.6.4 KPU_IMAGE_SIZE (0x00)

Reset value：0x00

Bits	Name	Description	Memory Access
63:51	RESERVED		W
50:42	O_COL_HIGH	output image column pixel num, start from 0, 0 means 1 row height	W
41:32	O_ROW_WID	output image row pixel number, start from 0, 0 means 1 pixel in one row	W
31:19	RESERVED		W
18:10	I_COL_HIGH	input image column height, start from 0, 0 means 1 row height	W

9:0	I_ROW_WID	Input image row pixel number, start from 0, 0 means 1 pixel in one row	W

3.2.6.5 KPU_KERNEL_POOL_TYPE_CFG (0x00)

Reset value：0x00

Bits	Name	Description	Memory Access
63:32	BWSX_BASE_ADDR	BN paramter load start address, 8 byte aligned, this is the 32bit address in memory map	W
31:24	PAD_VALUE	when pad_type set to 0, use this value as pad value	W
23:16	DMA_BURST_SIZE	kernal data dma burst size: 0-255	W
15:11	RESERVED		W
10	LOAD_PARA	load BN parameter when set to 1	W
9	BYPASS_CONV	bypass_conv: bypass convolution and activation when set to 1, the function is not available	W

Bits	Name	Description	Memory Access
8	FIRST_STRIDE	first layer stride, if the input column height is larger than 255(count from 0), must set this bit to 1	W
7:4	POOL_TYPE	pool type, 0: bypass pooling; 1: max pooling 2x2; 2: mean pooling 2x2; 3: max pooling 4x4; 4: mean pooling 4x4; 5: left pooling 2x2; 6: right pooling 2x2; 7: left pooling 4x4; 8 : mean pooling 2x2 stride 1; 9 : max pooling 2x2 stride 1	W
3	PAD_TYPE	pad type, 0: use the pad_value parameter to fill pad position; 1: use the nearest data to fill pad position	W
2:0	KERNEL_TYPE	kernel type, 0: 1x1 mode; 1: 3x3 mode	W

3.2.6.6 KPU_KERNEL_LOAD_CFG (0x00)

Reset value：0x00

Bits	Name	Description	Memory Access
63:32	PARA_START_ADDR	weight parameter load star address, 8 byte aligned, indicate the 32bit address in memory map	W
31:15	PARA_SIZE	indicate how many bytes of weight parameter the accelerator need to load for one load, start from 1, 1 means 1 byte	W
14:7	RESERVED		W
6:1	LOAD_TIME	indicate how many times we need to load the weight ram, start from 0, 0 means load weight ram one time, if weight parameter is larger than 36KByte, weight ram need to load several times	W
0	LOAD_COOR	load weight paramter, set to 1 indicate that weight parameter need to load	W

3.2.6.7 KPU_KERNEL_OFFSET(0x00)

Reset value：0x00

Bits	Name	Description	Memory Access
63:16	RESERVED		W
15:4	COEF_ROW_OFFSET	para row offset, SET TO 0!!	W
3:0	COEF_COLUMN_OFFSET	weight parameter column offset, SET TO 0!!	W

3.2.6.8 KPU_KERNEL_CALC_TYPE_CFG (0x00)

Reset value：0x00

Bits	Name	Description	Memory Access
63:32	ACT_ADDR	active parameter start address, 256byte aligned, this is the 32 bit address in memory map	W
31	LOAD_ACT	Indicate accelerator need to load activate parameter when set to 1	W
30:28	COEF_GROUP	indicate how many groups of weight value accelerator need to use for one calculation, if input row width is no more than 16byte, set to 4, no more than 32 bytes, set to 2, larger than 32 bytes, set to 1	W
27:20	RESERVED		W
19:16	ROW_SWITCH_ADDR	indicate one row of the input data occupy how many rows in the AI ram, as AI ram is 64Byte width, if the input image data’s row width is X Byte, set this register to ceil(X/64)	W
15	RESERVED		W
14:0	CHANNEL_SWITCH_ADDR	Indicate input channel switch addr in AI ram, the value is (i_row_wid+1) * (i_col_high+1)/64, facilitate accelerator to easily switch to next input channel	W

3.2.6.9 KPU_WRITE_BACK_CFG (0x00)

Reset value：0x00

Bits	Name	Description	Memory Access
63:23	RESERVED		W

22:20	WB_GROUP	indicate how many output channel we have for one AI ram row, if the output row width is no more than 16, set to 4, no more than 32 set to 2, otherwise set to 1	W
19:16	WB_ROW_SWITCH_ADDR	write back row switch address in AI ram, the value is ceil((o_row_wid+1)/64)	W
15	RESERVED		W
14:0	WB_CHANNEL_SWITCH_ADDR	write back channel switch address in AI ram, the value is ceil((o_row_wid+1)* (o_col_height+1)/64)	W

3.2.6.10 KPU_CONV_VALUE (0x00)

Reset value：0x00

Bits	Name	Description	Memory Access
63:56	RESERVED	reserved	W
55:32	ARG_X	multiply value for input data	W
31:8	ARG_W	multiply value for weight	W
7:4	SHR_X	shift value for input data in convolution calculation	W
3:0	SHR_W	shift value for weight in convolution calculation	W

3.2.6.11 KPU_CONV_VALUE2 (0x00)

Reset value：0x00

Bits	Name	Description	Memory Access
63:40	RESERVED	reserved	W
39:0	ARG_ADD	arg_add value in convolution calculation	W

3.2.6.12 KPU_DMA_PARAMETER (0x00)

Reset value：0x00

Bits	Name	Description	Memory Access
63:32	DMA_TOTAL_BYTE	the total byte dma transfer, start from 0 , 0 means 1 byte	W
31:16	CHANNEL_BYTE_NUM	the byte number of one output channel, start from 0 , 0 means 1 byte	W
15:1	RESERVED		W
0	SEND_DATA_OUT	indicate whether need to use dma to send data out or not	W

3.2.6.13 KPU_INTERRUPT_STATUS (0x08)

Reset value：0x00

Bits	Name	Description	Memory Access
63:3	RESERVED		R
2	CFG_ALMOST_FULL_STS	0x1:layer_cfg_almost_full	R
1	CFG_ALMOST_EMPTY_STS	0x1:layer_cfg_almost_empty	R
0	CALC_DONE_STS	calc_done_int	R

3.2.6.14 KPU_INTERRUPT_RAW (0x10)

Reset value：0x00

Bits	Name	Description	Memory Access
63:3	RESERVED		R
2	CFG_ALMOST_FULL	0x1:layer_cfg_almost_full	R
1	CFG_ALMOST_EMPTY	0x1:layer_cfg_almost_empty	R
0	CALC_DONE	calc_done_int	R

3.2.6.15 KPU_INTERRUPT_MASK (0x18)

Reset value：0x00

Bits	Name	Description	Memory Access
63:3	RESERVED		R/W
2	CFG_ALMOST_FULL_MASK	0x1:layer_cfg_almost_full mask	R/W
1	CFG_ALMOST_EMPTY_MASK	0x1:layer_cfg_almost_empty mask	R/W
0	CALC_DONE_MASK	0x1:calc_done_int mask	R/W

3.2.6.16 KPU_INTERRUPT_CLEAR (0x20)

Reset value：0x00

Bits	Name	Description	Memory Access
63:3	RESERVED		W
2	CFG_ALMOST_FULL_CLR	0x1:layer_cfg_almost_full_clear	W
1	CFG_ALMOST_EMPTY_CLR	0x1:layer_cfg_almost_empty_clear	W
0	CALC_DONE_CLR	0x1:calc_done_clear	W

3.2.6.17 KPU_FIFO_THRESHOLD (0x28)

Reset value：0x0C

Bits	Name	Description	Memory Access
63:16	RESERVED		R/W
15:8	FIFO_ALMOST_EMPTY_TH	fifo_empty_threshold, default value is 0	R/W
7:0	FIFO_ALMOST_FULL_TH	fifo full threshold, default value is 12	R/W

3.2.6.18 KPU_DMA_OUTPUT_PORT (0x30)

Reset value：0x00

Bits	Name	Description	Memory Access

63:0	DMA_OUTPUT_PORT	dma output port	R

3.2.6.19 KPU_FIFO_CTRL (0x38)

Reset value：0x00

Bits	Name	Description	Memory Access
63: 5	RESERVED		R/W
4	RESP_FIFO_FLUSH_N	0x1: flush fifo	R/W
3	CMD_FIFO_FLUSH_N	0x1: flush fifo	R/W
2	CFG_FIFO_FLUSH_N	0x1: flush fifo	R/W
1	GS_FIFO_FLUSH_N	0x1: flush fifo	R/W
0	DMA_FIFO_FLUSH_N	0x1: flush fifo	R/W

3.2.6.20 KPU_EIGHT_BIT_MODE (0x40)

Reset value：0x00

Bits	Name	Description	Memory Access
63:1	RESERVED		W
0	EIGHT_BIT_MODE	0: 16 bit mode 1: 8 bit mode	W

3.3 音频处理器 (APU)

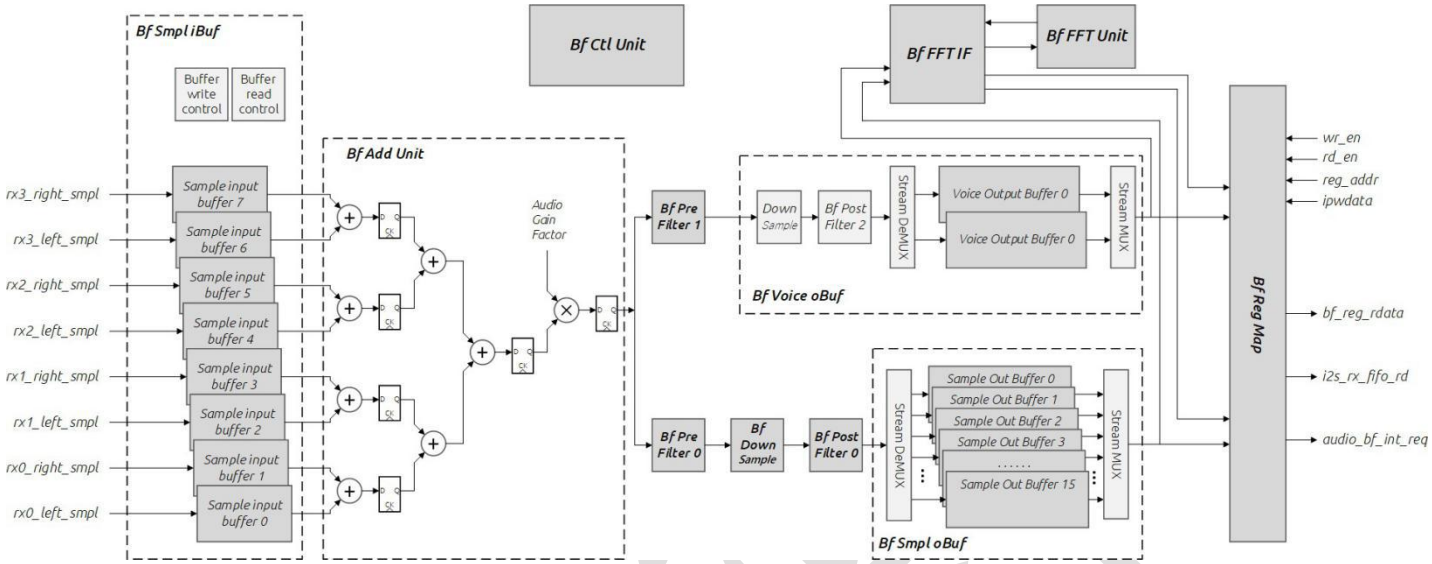
3.3.1 概述

APU用来协助驱动软件完成语音方向检测和有效语音方向的原始语音数据的预处理工作。该模块可同时获取16个方向的声音数据，这些声音数据会被写入到APU内部的声音样点输出缓冲区中（总计有16个声音sample输出 buffers），驱动软件通过系统DMA将预处理后的数据搬运到系统内存中并进行后续的处理获知有效的语音方向。在获得有效语音方向的情况下，驱动软件可以配置APU内部的寄存器来选定有效方向的语音输出数据流，APU会将这个有效方向的语音数据流输出到其内部的一个语音输出数据缓冲内（由2个以乒乓方式工作的sample输出 buffers 构成），驱动软件可以通过系统 DMA 将数据从这个缓冲区搬移到系统内存中，并进行后续的语音数据处理。另外，APU模块内部还集成了一个可支持512点的时域到频域变换的FFT Unit。如果使能FFT变换功能，APU所获取的语音数据会输入到FFT内部进行时域到频域变换处理，然后再将数据从FFT内部的RAM通过系统DMA搬移到系统内存中。

3.3.2 主要特性

- 可以支持最多8路音频输入数据流，即4路双声道；
- 可以支持多达16个方向的声源同时扫描预处理与波束形成；
- 可以支持一路有效的语音数据流输出；
- 内部音频信号处理精度达到16-位；
- 输入音频信号支持 12-位，16-位，24-位，32-位精度；
- 支持多路原始信号直接输出；
- 可以支持高达192K采样率的音频输入；
- 内置FFT变换单元，可对音频数据提供512点快速傅里叶变换；
- 利用系统DMAC将输出数据存储到SoC的系统内存中。

3.3.3 功能描述



APU模块构成：

Audio_bf_reg_map：子模块通过一个内部的寄存器交互接口从 I2S0 host 获取软件的配置参数，这些参数包括 low-pass 滤波器的系数，对应于不同方向的音频采样输入缓冲区的起始读地址，声音采样流的 downsampling 配置等。

Audio_bf_ctl_unit：子模块是 beamforming 数据处理流程的控制模块。

Audio_bf_add_unit：子模块负责从8个声音采样输入缓冲区中读出一组数据并进行求平均值操作（这个过程相当于beam forming）。

Audio_bf_smpl_ibufAudio_bf_dwn_siz：子模块负责对声音方向搜索处理通路上的音频采样流做 downsampling 处理（进行抽样）。

Audio_bf_filter0：子模块是用来处理声音方向搜索处理通路上数据的低通滤波器，这个filter被实例化两次，分别用来对downsampling前的数据流做低通滤波和对downsampling后的数据流做低通滤波。

Audio_bf_filter1：子模块用来处理语音输出通路上的数据低通滤波，也需要实例化两次，分别对 downsampling 前和 downsampling 后的数据流做低通滤波。

Audio_bf_smpl_obuf：子模块用来存储 16 个声音搜索方向的音频数据。

Audio_bf_fft_if：Audio_bf_fft 的控制接口。

Audio_bf_fft 是一个可以支持 512 点时域到频域变换的 FFT Unit。

Audio_bf_voice_obuf：子模块用来存储语音输出通路产生的音频数据。

3.3.3.1 Autio_bf_ctl_unit

- Audio_bf_ctl_unit 是 audio beam forming 数据处理流程的总控制模块。这个模块负责产生一组时序控制信号，这些时序信号负责控制在声音方向搜索及语音流输出时读取输入 sample buffers 的数据，启动信号低通滤波器，数据写入输出 buffer 以及从 I2S0 RX channel 接收新的输入语音采样。时序控制信号的产生是通过一个控制状态机来实现的。

3.3.3.2 Audio_bf_smpl_ibuf

audio_bf_smpl_ibuf 模块是 audio beam forming的输入音频数据缓冲区，这个模块内置了 8 个 音频sample输入buffer，每个 buffer 的规格是64x16bit，存储的数据是16-bit有符号数据。这 8 个音频输入buffers的数据来自于*I2S0*的四个RX Channels。Sample ibuffer 0对应于 RX channel 0 left sample输出，Sample ibuffer 1对应于 RX channel 0 right sample输出，Sample ibuffer 2对应于 RX channel 1 left sample输出，Sample ibuffer 3对应于 RX channel 1 right sample输出，Sample ibuffer 4对应于 RX channel 2 left sample输出，Sample ibuffer 5对应于 RX channel 2 right sample输出，Sample ibuffer 6对应于 RX channel 3 left sample输出，Sample ibuffer 7对应于 RX channel 3 right sample输出。

由于所有的sample ibuffers都是使用单口Register File实现的，要实现16个方向的搜索及语音输出，必须以分时复用的方式读取sample ibuffers。在audio_bf_smpl_ibuf模块内部，维护了17组sample ibuffers的读取地址计数器，每一组读取地址计数器由8个地址寄存器构成，这8个地址寄存器分别控制8个ibuffer的数据读取访问。其中16组读取地址计数器用来实现16个声音方向的数据读取，第17个用来实现语音输出时的数据读取。当检测到para_dir_search_en由0变为1后，对应于声音方向搜索的16组地址计数器会被分别初始化为软件所配置的起始偏移地址。当检测到para_stream_gen_en由0变

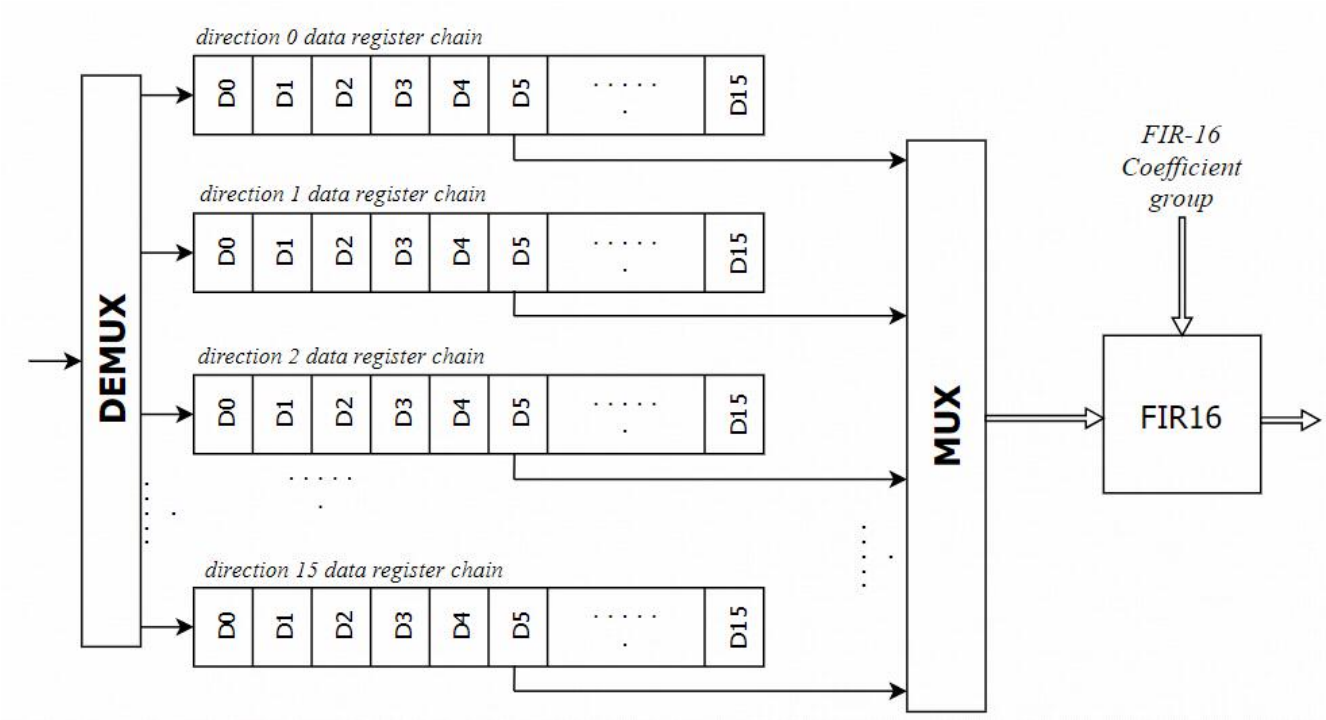
为1后，对应于语音输出的那组地址计数器会被初始化为所选定的方向上的那组起始偏移地址。当BFU的FSM处于SMPL_RD0状态时，就会针对声音方向搜索进行数据读取，并更新相应的一组地址计数器；当BFU的FSM处于SMPL_RD1状态时，就会针对语音输出进行数据读取，并更新相应的地址计数器。

3.3.3.3 Audio_bf_add_unit

Audio_bf_add_unit 模块用来对从输入 sample buffer 中读取的一组音频数据做平均值处理。这个模块由 3 级加法器组构成，并成两级寄存器处理以改善 timing。获取平均值的操作是通过算数右移来实现的。audio_bf_add_unit 模块的结构可以参考前面结构框图中的 bf_add_unit 部分。

3.3.3.4 Audio_bf_filter0

Audio_bf_filter0 模块用来对声音方向搜索通路的数据流做前置和后置低通滤波。前置低通滤波指的是在 DownSizing 处理之前的低通滤波，后置低通滤波指的是在 DownSizing 处理之后的低通滤波。因此 audio_bf_filter0 模块实例化了两次，分别对应于前置低通滤波器和后置低通滤波器。audio_bf_filter0 模块由一个 16 阶的 FIR 滤波器和 16 组音频数据寄存器链构成。audio_bf_filter0 模块的结构如下图所示：



注意：由于音频数据输入的速率比较低，因此不需要一个并行的告诉FIR处理模块。为了节省面积，这里采用了穿行累加的方式来实现这个16阶FIR滤波器，完成一次数据处理需要17个cycle。

3.3.3.4 Audio_bf_dwn_siz

audio_bf_dwn_siz 模块负责对声音搜索处理通路上的前置滤波器输出的数据流进行降采样，降采样率可有软件配置，最多支持 15:1 的down sampling。在实施降采样的时候，在每一个循环降采样周期内总是保留第一个数据sample。举个具体的例子：将定软件将降采样率设置为 4:1，那么每四个sample构成一个循环降采样周期，这这个周期内，第一个sample会被保留下来。

3.3.3.5 Audio_bf_smpl_obuf

audio_bf_smpl_obuf 模块负责存储声音方向搜索处理通路所产生的数据。audio_bf_smpl_obuf 模块内部包含16个sample输出buffer，每一个buffer的规格是512x16-bit。每一个output buffer对应一个声音搜索方向。audio_bf_smpl_obuf 模块内的16个obuffers可工作在写入和读出两种模式下。当软件将para_dir_search_en配置为1时，输出buffer被设置为写入模式，此时来自声音搜索处理通路的数据会交替的写入这16个obuffers中，当所有这16个obuffers都已经填满，就会自动切换到读出模式。在读出模式下，软件可利用系统的DMA从这些obuffers中将数据读出。

3.3.3.6 Audio_bf_voice_obuf

audio_bf_voice_obuf 模块负责存储语音输出通过所产生的数据。audio_bf_voice_obuf 模块内包含用来对输入音频数据做down sizing的逻辑（也就是数据抽样），其作用于audio_bf_dwn_siz一致。audio_bf_voice_obuf 模块内还使用一个低通滤波器，用来对down sizing后的数据流做低通滤波。这个低通滤波器即为 audio_bf_filter1。audio_bf_voice_obuf 模块还包含两个乒乓模式工作的输出buffers，这两个buffer的规格都是512x16-bit。当硬件处理通路向其中一个buffer写入数据时，软件可以从另外一个buffer中读出数据。

3.3.3.7 Audio_bf_filter1

audio_bf_filter1 模块是针对语音输出通路所设置的低通滤波器。audio_bf_filter1 模块由一个17级的audio sample register chain和一个FIR-16滤波器构成。

3.3.4 寄存器列表

Base Address：APU_BASE_ADDR (0x50250200)

Name	Description	Offset address
I2S0_BF_CH_CFG	I2S0 Audio Beam-forming Channel configuration register	0x00
I2S0_BF_CTL	I2S0 Audio Beam-forming control register	0x04
I2S0_BF_DIR0_BIDX0	I2S0 Audio Beam-forming direction 0 sample ibuffer read index configure register 0.	0x08
I2S0_BF_DIR0_BIDX1	I2S0 Audio Beam-forming direction 0 sample ibuffer read index configure register 1.	0x0C
I2S0_BF_DIR1_BIDX0	I2S0 Audio Beam-forming direction 1 sample ibuffer read index configure register 0.	0x10
I2S0_BF_DIR1_BIDX1	I2S0 Audio Beam-forming direction 1 sample ibuffer read index configure register 1.	0x14
I2S0_BF_DIR2_BIDX0	I2S0 Audio Beam-forming direction 2 sample ibuffer read index configure register 0.	0x18
I2S0_BF_DIR2_BIDX1	I2S0 Audio Beam-forming direction 2 sample ibuffer read index configure register 1.	0x1C
I2S0_BF_DIR3_BIDX0	I2S0 Audio Beam-forming direction 3 sample ibuffer read index configure register 0.	0x20
I2S0_BF_DIR3_BIDX1	I2S0 Audio Beam-forming direction 3 sample ibuffer read index configure register 1.	0x24
I2S0_BF_DIR4_BIDX0	I2S0 Audio Beam-forming direction 4 sample ibuffer read index configure register 0.	0x28
I2S0_BF_DIR4_BIDX1	I2S0 Audio Beam-forming direction 4 sample ibuffer read index configure register 1.	0x2C

I2S0_BF_DIR5_BIDX0	I2S0 Audio Beam-forming direction 5 sample ibuffer read index configure register 0.	0x30

Name	Description	Offset address
I2S0_BF_DIR5_BIDX1	I2S0 Audio Beam-forming direction 5 sample ibuffer read index configure register 1.	0x34
I2S0_BF_DIR6_BIDX0	I2S0 Audio Beam-forming direction 6 sample ibuffer read index configure register 0.	0x38
I2S0_BF_DIR6_BIDX1	I2S0 Audio Beam-forming direction 6 sample ibuffer read index configure register 1.	0x3C
I2S0_BF_DIR7_BIDX0	I2S0 Audio Beam-forming direction 7 sample ibuffer read index configure register 0.	0x40
I2S0_BF_DIR7_BIDX1	I2S0 Audio Beam-forming direction 7 sample ibuffer read index configure register 1.	0x44
I2S0_BF_DIR8_BIDX0	I2S0 Audio Beam-forming direction 8 sample ibuffer read index configure register 0.	0x48
I2S0_BF_DIR8_BIDX1	I2S0 Audio Beam-forming direction 8 sample ibuffer read index configure register 1.	0x4C
I2S0_BF_DIR9_BIDX0	I2S0 Audio Beam-forming direction 9 sample ibuffer read index configure register 0.	0x50
I2S0_BF_DIR9_BIDX1	I2S0 Audio Beam-forming direction 9 sample ibuffer read index configure register 1.	0x54
I2S0_BF_DIR10_BIDX0	I2S0 Audio Beam-forming direction 10 sample ibuffer read index configure register 0.	0x58
I2S0_BF_DIR10_BIDX1	I2S0 Audio Beam-forming direction 10 sample ibuffer read index configure register 1.	0x5C
I2S0_BF_DIR11_BIDX0	I2S0 Audio Beam-forming direction 11 sample ibuffer read index configure register 0.	0x60
I2S0_BF_DIR11_BIDX1	I2S0 Audio Beam-forming direction 11 sample ibuffer read index configure register 1.	0x64
I2S0_BF_DIR12_BIDX0	I2S0 Audio Beam-forming direction 12 sample ibuffer read index configure register 0.	0x68
I2S0_BF_DIR12_BIDX1	I2S0 Audio Beam-forming direction 12 sample ibuffer read index configure register 1.	0x6C

I2S0_BF_DIR13_BIDX0	I2S0 Audio Beam-forming direction 13 sample ibuffer read index configure register 0.	0x70
I2S0_BF_DIR13_BIDX1	I2S0 Audio Beam-forming direction 13 sample ibuffer read index configure register 1.	0x74
I2S0_BF_DIR14_BIDX0	I2S0 Audio Beam-forming direction 14 sample ibuffer read index configure register 0.	0x78

Name	Description	Offset address
I2S0_BF_DIR14_BIDX1	I2S0 Audio Beam-forming direction 14 sample ibuffer read index configure register 1.	0x7C
I2S0_BF_DIR15_BIDX0	I2S0 Audio Beam-forming direction 15 sample ibuffer read index configure register 0.	0x80
I2S0_BF_DIR15_BIDX1	I2S0 Audio Beam-forming direction 15 sample ibuffer read index configure register 1.	0x84
I2S0_BF_PRE_FIR0_COEF0	I2S host beam-forming preposition Filter 0 FIR16 Coefficient Register0	0x88
I2S0_BF_PRE_FIR0_COEF1	I2S host beam-forming preposition Filter 0 FIR16 Coefficient Register1	0x8C
I2S0_BF_PRE_FIR0_COEF2	I2S host beam-forming preposition Filter 0 FIR16 Coefficient Register2	0x90
I2S0_BF_PRE_FIR0_COEF3	I2S host beam-forming preposition Filter 0 FIR16 Coefficient Register3	0x94
I2S0_BF_PRE_FIR0_COEF4	I2S host beam-forming preposition Filter 0 FIR16 Coefficient Register4	0x98
I2S0_BF_PRE_FIR0_COEF5	I2S host beam-forming preposition Filter 0 FIR16 Coefficient Register5	0x9C
I2S0_BF_PRE_FIR0_COEF6	I2S host beam-forming preposition Filter 0 FIR16 Coefficient Register6	0xA0
I2S0_BF_PRE_FIR0_COEF7	I2S host beam-forming preposition Filter 0 FIR16 Coefficient Register7	0xA4
I2S0_BF_PRE_FIR0_COEF8	I2S host beam-forming preposition Filter 0 FIR16 Coefficient Register8	0xA8
I2S0_BF_POS_FIR0_COEF0	I2S host beam-forming postposition Filter 0 FIR16 Coefficient Register0	0xAC

I2S0_BF_POS_FIR0_COEF1	I2S host beam-forming postposition Filter 0 FIR16 Coefficient Register1	0xB0
I2S0_BF_POS_FIR0_COEF2	I2S host beam-forming postposition Filter 0 FIR16 Coefficient Register2	0xB4
I2S0_BF_POS_FIR0_COEF3	I2S host beam-forming postposition Filter 0 FIR16 Coefficient Register3	0xB8
I2S0_BF_POS_FIR0_COEF4	I2S host beam-forming postposition Filter 0 FIR16 Coefficient Register4	0xBC
I2S0_BF_POS_FIR0_COEF5	I2S host beam-forming postposition Filter 0 FIR16 Coefficient Register5	0xC0

Name	Description	Offset address
I2S0_BF_POS_FIR0_COEF6	I2S host beam-forming postposition Filter 0 FIR16 Coefficient Register6	0xC4
I2S0_BF_POS_FIR0_COEF7	I2S host beam-forming postposition Filter 0 FIR16 Coefficient Register7	0xC8
I2S0_BF_POS_FIR0_COEF8	I2S host beam-forming postposition Filter 0 FIR16 Coefficient Register8	0xCC
I2S0_BF_PRE_FIR1_COEF0	I2S host beam-forming preposition Filter 1 FIR16 Coefficient Register0	0xD0
I2S0_BF_PRE_FIR1_COEF1	I2S host beam-forming preposition Filter 1 FIR16 Coefficient Register1	0xD4
I2S0_BF_PRE_FIR1_COEF2	I2S host beam-forming preposition Filter 1 FIR16 Coefficient Register2	0xD8
I2S0_BF_PRE_FIR1_COEF3	I2S host beam-forming preposition Filter 1 FIR16 Coefficient Register3	0xDC
I2S0_BF_PRE_FIR1_COEF4	I2S host beam-forming preposition Filter 1 FIR16 Coefficient Register4	0xE0
I2S0_BF_PRE_FIR1_COEF5	I2S host beam-forming preposition Filter 1 FIR16 Coefficient Register5	0xE4
I2S0_BF_PRE_FIR1_COEF6	I2S host beam-forming preposition Filter 1 FIR16 Coefficient Register6	0xE8
I2S0_BF_PRE_FIR1_COEF7	I2S host beam-forming preposition Filter 1 FIR16 Coefficient Register7	0xEC

I2S0_BF_PRE_FIR1_COEF8	I2S host beam-forming preposition Filter 1 FIR16 Coefficient Register8	0xF0
I2S0_BF_POS_FIR1_COEF0	I2S host beam-forming postposition Filter 1 FIR16 Coefficient Register0	0xF4
I2S0_BF_POS_FIR1_COEF1	I2S host beam-forming postposition Filter 1 FIR16 Coefficient Register1	0xF8
I2S0_BF_POS_FIR1_COEF2	I2S host beam-forming postposition Filter 1 FIR16 Coefficient Register2	0xFC
I2S0_BF_POS_FIR1_COEF3	I2S host beam-forming postposition Filter 1 FIR16 Coefficient Register3	0x100
I2S0_BF_POS_FIR1_COEF4	I2S host beam-forming postposition Filter 1 FIR16 Coefficient Register4	0x104
I2S0_BF_POS_FIR1_COEF5	I2S host beam-forming postposition Filter 1 FIR16 Coefficient Register5	0x108

Name	Description	Offset address
I2S0_BF_POS_FIR1_COEF6	I2S host beam-forming postposition Filter 1 FIR16 Coefficient Register6	0x10C
I2S0_BF_POS_FIR1_COEF7	I2S host beam-forming postposition Filter 1 FIR16 Coefficient Register7	0x110
I2S0_BF_POS_FIR1_COEF8	I2S host beam-forming postposition Filter 1 FIR16 Coefficient Register8	0x114
I2S0_VF_DWSZ_CFG	I2S Host beam-forming Down-Sizing Configure Register.	0x118
I2S0_BF_FFT_CFG	I2S Host beam-forming FFT Configure Register.	0x11C
I2S0_BF_SDMA	I2S Host beam-forming Sample obuffer DMA Register.	0x120
I2S0_BF_VDMA	I2S Host beam-forming Voice obuffer DMA Register.	0x124
I2S0_BF_INT_STAT	I2S Host beam-forming Interrupt status register.	0x128
I2S0_BF_INT_MASK	I2S Host beam-forming Interrupt Mask register.	0x12C

I2S0_BF_SAT_CNT	I2S Host beam-forming Sample Saturation Counting Register.	0x130
I2S0_BF_SAT_CFG	I2S Host beam-forming Sample Saturation Configure Register.	0x134

3.3.5 寄存器设置

3.3.5.1 I2S0_BF_CH_CFG(0x00)

Reset value: 0x10

Bits	Parameter Name	Description	Memory Access
31:19	RSV		RO
18	we_bf_shift_factor	write enable for bf_shift_factor parameter. 0x1: allowing updates made to bf_shift_factor .	WO
17	we_bf_target_dir	write enable for bf_target_dir parameter. 0x1: allowing updates made to bf_target_dir .	WO
16	we_bf_sound_ch_en	write enable for bf_sound_ch_en parameter. 0x1: allowing updates made to bf_sound_ch_en .	WO
15:12	shift_factor	Voice strength average value right shift factor. When performing sound direction detect, the average value of samples from different channels is required, this right shift factor is used to perform division. 0x0: no right shift; 0x1: right shift by 1-bit; 0xF: right shift by 14-bit.	RW
11:8	bf_target_dir	Target direction select for valid voice output. When the source voice direaction searching is done, software can use this field to select one from 16 sound directions for the following voice recognition. 0x0: select sound direction 0; 0x1: select sound direction 1; 0xF:	RW

		select sound direction 15.	
7:0	bf_sound_ch_en	BF unit sound channel enable control bits. Bit x corresponds to enable bit for sound channel x (x = 0, 1, 2, . . ., 7). BF sound channels are related with I2S host RX channels. BF sound channel 0/1 correspond to the left/right channel of I2S RX0; BF channel 2/3 correspond to left/right channels of I2S RX1; and things like that. 0x1: writing '1' to enable the corresponding BF sound channel. 0x0: writing '0' to close the corresponding BF sound channel.	RW

3.3.5.2 I2S0_BF_CTL(0x04)

Reset value: 0x00

Bits	Name	Description	Memory Access
31:30	RSV		RO
29:24	dir0_rd_idx3	sample ibuffer 3 start read index for sound direction 0. When performing direction search, this index is used at the start read address to ibuffer 3.	RW
23:22	RSV		RO
21:16	dir0_rd_idx2	sample ibuffer 2 start read index for sound direction 0. When performing direction search, this index is used at the start read address to ibuffer 2.	RW
15:14	RSV		RO
13:8	dir0_rd_idx1	sample ibuffer 1 start read index for sound direction 0. When performing direction search, this index is used at the start read address to ibuffer 1.	RW

7:6	RSV		RO
5:0	dir0_rd_idx0	sample ibuffer 0 start read index for sound direction 0. When performing direction search, this index is used at the start read address to ibuffer 0.	RW

3.3.5.3 I2S0_BF_DIRn_BIDX0(0x08 + (8 * n))

Reset value: 0x00

Bits	Name	Description	Memory Access
31:30	RSV		RO
29:24	dir0_rd_idx7	sample ibuffer 3 start read index for sound direction 0. When performing direction search, this index is used at the start read address to ibuffer 7.	RW
23:22	RSV		RO
21:16	dir0_rd_idx6	sample ibuffer 2 start read index for sound direction 0. When performing direction search, this index is used at the start read address to ibuffer 6.	RW
15:14	RSV		RO
13:8	dir0_rd_idx5	sample ibuffer 1 start read index for sound direction 0. When performing direction search, this index is used at the start read address to ibuffer 5.	RW
7:6	RSV		RO
5:0	dir0_rd_idx4	sample ibuffer 0 start read index for sound direction 0. When performing direction search, this index is used at the start read address to ibuffer 4.	RW

3.3.5.4 I2S0_BF_DIRn_BIDX1(0x0C + (8*n))

Reset value: 0x00

Bits	Name	Description	Memory Access
31:30	RSV		RO

29:24	dir0_rd_idx7	sample ibuffer 3 start read index for sound direction 0. When performing direction search, this index is used at the start read address to ibuffer 7.	RW
23:22	RSV		RO
21:16	dir0_rd_idx6	sample ibuffer 2 start read index for sound direction 0. When performing direction search, this index is used at the start read address to ibuffer 6.	RW
15:14	RSV		RO
13:8	dir0_rd_idx5	sample ibuffer 1 start read index for sound direction 0. When performing direction search, this index is used at the start read address to ibuffer 5.	RW
7:6	RSV		RO
5:0	dir0_rd_idx4	sample ibuffer 0 start read index for sound direction 0. When performing direction search, this index is used at the start read address to ibuffer 4.	RW

3.3.5.5 I2S0_BF_PRE_FIR0_COEFn(0x88 + (4 * n))

Reset value: 0x00

Bits	Name	Description	Memory Access
31:16	pre_fir0_tap(2n+1)_coef*	Pre Low-Pass Filter 0 FIR16 tap 1 coefficient.	RW
15:0	pre_fir0_tap(2n)_coef*	Pre Low-Pass Filter 0 FIR16 tap 0 coefficient.	RW

3.3.5.6 I2S0_BF_POS_FIR0_COEFn(0xAC + 4*n)

Reset value: 0x00

Bits	Name	Description	Memory Access
31:16	pos_fir0_tap1_coef	Low-Pass Filter 0 FIR16 tap 1 + 2*n coefficient.	RW
15:0	pos_fir0_tap0_coef	Low-Pass Filter 0 FIR16 tap 0 + 2*n coefficient.	RW

3.3.5.7 I2S0_BF_POS_FIR0_COEF8(0xCC)

Reset value: 0x00

Bits	Name	Description	Memory Access
31:16	RSRV		RO
15:0	pos_fir0_tap16_coef	Low-Pass Filter 0 FIR16 tap 16 coefficient.	RW

3.3.5.8 I2S0_BF_PRE_FIR1_COEFn(0xD0 + (4 * n))

Reset value: 0x00

Bits	Name	Description	Memory Access
31:16	pre_fir1_tap1_coef	Low-Pass Filter 0 FIR16 tap 1 + 2*n coefficient.	RW
15:0	pre_fir1_tap0_coef	Low-Pass Filter 0 FIR16 tap 0 + 2*n coefficient.	RW

3.3.5.9 I2S0_BF_PRE_FIR1_COEF8(0xF0)

Reset value: 0x00

Bits	Name	Description	Memory Access
31:16	RSRV		RO
15:0	pre_fir0_tap16_coef	Low-Pass Filter 0 FIR16 tap 16 coefficient.	RW

3.3.5.10 I2S0_BF_POS_FIR1_COEFn(0xF4 + 4*n)

Reset value: 0x00

Bits	Name	Description	Memory Access
31:16	pos_fir1_tap1_coef	Low-Pass Filter 0 FIR16 tap 1 + 2*n coefficient.	RW

15:0	pos_fir1_tap0_coef	Low-Pass Filter 0 FIR16 tap 0 + 2*n coefficient.	RW

3.3.5.11 I2S0_BF_POS_FIR1_COEF8(0x114)

Reset value: 0x00

Bits	Name	Description	Memory Access
31:16	RSRV		RO
15:0	pos_fir1_tap16_coef	Low-Pass Filter 0 FIR16 tap 16 coefficient.	RW

3.3.5.12 I2S0_BF_DWSZ_CFG(0x118)

Reset value: 0x00

Bits	Name	Description	Memory Access
31:13	RSV		RO
12:8	smpl_shift_bits	This bit field is used to perform sample precision reduction when the source sound sample (from I2S0 host receiving channels) precision is 20/24/32 bits. 0x0: take bits 15~0 from the source sound sample; 0x1: take bits 16~1 from the source sound sample; 0x2: take bits 17~2 from the source sound sample; . . . . . 0x10: take bits 31~16 from the source sound sample;	RW
7:4	voc_dwn_siz_rate	The down-sizing ratio used for voice stream generation. 0x0: no down-sizing; 0x1: 1/2 down sizing; 0x2: 1/3 down sizing; 0xF: 1/16 down sizing.	RW
3:0	dir_dwn_siz_rate	The down-sizing ratio used for direction searching. 0x0: no down-sizing; 0x1: 1/2 down sizing; 0x2: 1/3 down sizing; 0xF: 1/16 down sizing.	RW

3.3.5.13 I2S0_BF_FFT_CFG(0x11C)

Reset value: 0x00

Bits	Name	Description	Memory Access
31:8	RSV		RO
12	fft_enable	FFT function enable control parameter. 0x1: enable FFT operation; 0x0: disable FFT operation.	RW
11:9	RSV		RO
8:0	fft_shift_factor	FFT computation data shift factor parameter.	RW

3.3.5.14 I2S0_BF_SDMA(0x120)

Reset value: 0x00

Bits	Name	Description	Memory Access
31:13	RSV		RO
12	fft_enable	FFT function enable control parameter. 0x1: enable FFT operation; 0x0: disable FFT operation.	RW
11:9	RSV		RO
8:0	fft_shift_factor	FFT computation data shift factor parameter.	RW

3.3.5.15 I2S0_BF_VDMA(0x124)

Reset value: 0x00

Bits	Name	Description	Memory Access
31:0	vobuf_dma_rdata	This is the read register for system DMA to read data stored in voice out buffers (the voice out buffers are used for voice recognition). Each data contains two sound samples.	RO

3.3.5.16 I2S0_BF_INT_STAT(0x128)

Reset value: 0x00

Bits	Name	Description	Memory Access
31:2	RSV		RO
1	voc_buf_data_rdy	voice output stream buffer data ready interrupt event. When a block of 512 voice samples are collected, this interrupt event is asserted. Writing '1' to clear this interrupt event. 0x1: voice output stream buffer data is ready; 0x0: no event.	RW1C
0	dir_search_data_rdy	sound direction searching data ready interrupt event. Writing '1' to clear this interrupt event. 0x1: data is ready for sound direction detect; 0x0: no event.	RW1C

3.3.5.17 I2S0_BF_INT_MASK(0x12C)

Reset value: 0x00

Bits	Name	Description	Memory Access
31:2	RSV		RO
1	voc_buf_rdy_msk	This is the interrupt mask to voice output stream buffer ready interrupt. 0x1: mask off this interrupt; 0x0: enable this interrupt.	RW
0	search_data_rdy_msk	This is the interrupt mask to dir searching data ready interrupt. 0x1: mask off this interrupt; 0x0: enable this interrupt.	RW

3.3.5.18 I2S0_BF_SAT_CNT(0x130)

Reset value: 0x00

Bits	Name	Description	Memory Access
31	voc_sat_cnt_rst	This bit is used to reset the Voice sample saturation counter. When this bit is asserted, the value of sat_smpl_cnt and voc_smpl_cnt will be cleared to 0.	WO

30:16	sat_smpl_cnt	This field provides the number of voice sample whose value exceed the limit set by software.	RO
15:0	voc_smpl_cnt	This field provides the total number of voice sample generated by voice output path.	RO

3.3.5.19 I2S0_BF_SAT_CFG(0x134)

Reset value: 0x80007FFF

Bits	Name	Description	Memory Access
31:16	sat_bottom_limit	A sample is deemed bottom saturated if it s value is negative and less than the value of sat_bottom_limit. The value here must be a negative number!	WO
15:0	sat_upper_limit	A sample is deemed upper saturated if it's value is positive and greater than the value of sat_upper_limit. The value here must be a positive number!	RO

3.4 静态随机存取存储器 (SRAM)

3.4.1 概述

片上存储器为静态随机存取存储器(SRAM)，SRAM包含两个部分，分别是6MiB的片上通用SRAM 存储器与2MiB的片上AI SRAM存储器，共计8MiB（1MiB为1兆字节）。其中，AI SRAM存储器是专为KPU分配的存储器。它们分布在连续的地址空间中，不仅可以通过CPU的缓存接口访问，而且可以通过非缓存接口直接访问。通用SRAM存储器在芯片正常工作的任意时刻都可以访问。该存储器分为两个Bank，分别为MEM0 与MEM1，并且DMA控制器可操作不同Bank。CPU使用经CPU缓存的地址的访问延迟较小，其他外设使用非经CPU缓存的地址的访问延迟较小；经CPU缓存的访问会先经过CPU的cache，非经CPU缓存的访问不会经过cache;不同设备访问同一处SRAM地址时需注意cache一致性，最好使用相同的访问路径，或者用CPU把cache的内容刷新到SRAM中来保持cache一致性。

3.4.2 功能描述

- **地址映射**

模块名称	映射类型	子模块	开始地址	结束地址	空间大小
通用SRAM存储器	经CPU缓存	MEM0	0x80000000	0x803FFFFF	0x400000
		MEM1	0x80400000	0x805FFFFF	0x200000
AI SRAM存储器	经CPU缓存		0x80600000	0x807FFFFF	0x200000
通用SRAM存储器	非经CPU缓存	MEM0	0x40000000	0x403FFFFF	0x400000
		MEM1	0x40400000	0x405FFFFF	0x200000
AI SRAM存储器	非经CPU缓存		0x40600000	0x407FFFFF	0x200000

- **寄存器设置**

AI SRAM存储器仅在以下条件都满足时才可访问：

1） PLL1 已使能，并时钟系统配置正确KPU 没有进行神经网络计算；

2） KPU 没有进行神经网络计算。

3.5 系统控制器 (SYSCTL)

3.5.1 概述

芯片集成了3个PLL，支持灵活的时钟配置，包含四大功能：

- PLL0输出时钟作为芯片的主要时钟提供给CPU和除I2S外的各个外设，输入是外部的26M时钟；
- PLL1输出时钟作为AI的工作时钟，输入是外部的26M时钟；
- PLL2输出时钟作为I2S的工作时钟，输入有三个选择：外部的26M时钟或者PLL0/1的输出时钟；
- PLL的输入参考时钟支持1～16分频，其内部反馈时钟支持1~64分频，其输出时钟上支持1~16分频。

芯片有各种外部及内部的复位和时钟，主要功能包括：

- 芯片有一个外部输入的复位信号，还有一个软复位以及两个看门狗复位，做为全局复位信号；
- 芯片内部各主要模块都有独立的复位控制；
- 芯片有一个外部输入的26M时钟，并使用三路PLL提供高频时钟；
- 芯片各模块的时钟都有丰富的分频及时钟使能控制。

3.5.2 功能描述

3.5.2.1PLL输出频率配置

1） CLKR：4bit参考时钟分频系数，支持1-16分频（NR），NR = CLKR[3:0] + 1

例如：

/1 配置值： 0000

/4 配置值： 0011

/8 配置值： 0111

2） CLKF：6bit输出倍频系数，支持1-64倍频（NF），NF = CLKF[5:0] + 1

例如：

X1 配置值： 000000

X2 配置值： 000001

X64 配置值：111111

3） CLKOD：4bit的输出分频系数，支持1-16分频（OD），OD = CLKOD[3:0] + 1

例如：

/1 配置值： 0000

/4 配置值： 0011

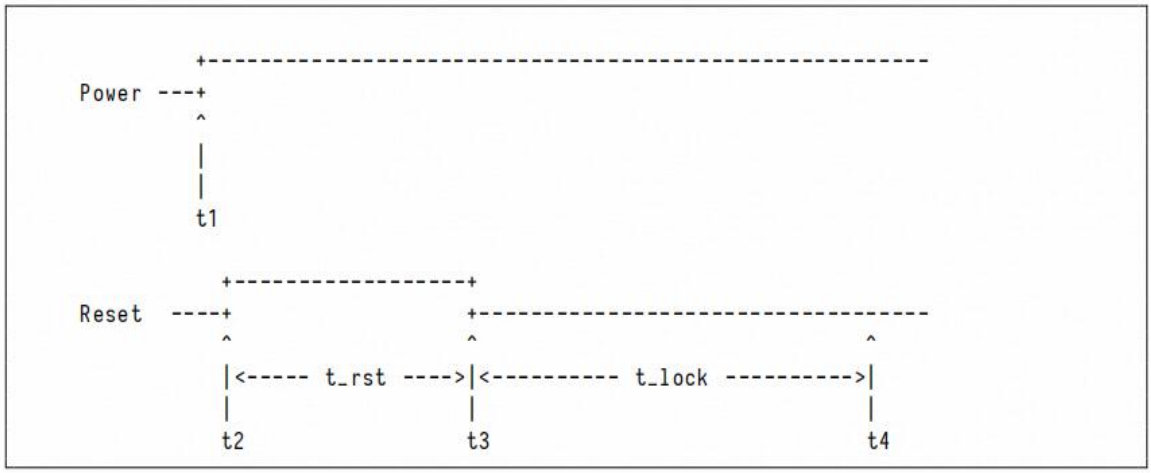
/8 配置值： 0111

4） BWADJ：6bit的带宽调节分频系数，配置值应与CLKF相同（NB）NB = BWADJ[5:0] + 1，Fout = (Fref × NF)/ (NR * OD)。BWADJ应和CLKR相同。建议使用参考设计里的脚本来配置PLL参数。

3.5.2.2 PLL2输入时钟选择

PLL2的输入时钟可以在26M输入时钟，PLL0/1输出时钟之间进行选择，输出更精准的I2S时钟， PLL_clkin_sel2配置寄存器主要用于PLL2输入时钟的选择。

3.5.2.3 PLL初始化



PLL上电及复位前应先配置好所有的配置寄存器。PLL初始化时应把pwr置1，同时或之后把Reset先置1，100ns后再置0。在检测到PLL 锁定信号的bit1变为1时PLL的输出时钟应该已经稳定，但如果此时如果检查到锁定信号的bit0是0，可以对寄存器PLL_SLIP_CLEAR写1，再次检查锁定信号的bit0，如果反复多次发现锁定信号的bit0仍是0，表示PLL的状态异常。

3.5.2.4 系统复位

- 芯片内部有三种全局复位信号:

1. 外部复位信号，会对全芯片进行复位；

2. 看门狗复位，芯片内部有两个看门狗模块，均可复位除看门狗模块之外的所有模块；

3. 软复位，由软件控制，可对全芯片进行复位

- 软件可以对DMA, DVP, AI以及各种外设模块的任意时刻进行复位，复位前应先把该模块的时钟关闭,复位完成后再把时钟打开
- 上电默认不开启的clk的模块有：WDT0, WDT1, RTC

3.5.2.5 系统时钟

- PLL0为CPU与大部分外设提供时钟；

- PLL1为AI加速核提供时钟；
- PLL2为I2S音频提供时钟。

其中，I2S MCLK，I2S CLK都只能选择PLL2为时钟源。其余外设可以在外部时钟与PLL0之间进行选择(通过PLL BYPASS来选择外部时钟)。而PLL2的输入可以在26M输入时钟以及PLL0/1输出时钟之间选择，以获得更精准的音频采样频率。

3.5.3 寄存器列表

Base Address：PLL_BASE_ADDR (0x50440000)

Name	Description	Offset address
GIT_ID	Git short commit id	0x00
CLK_FREQ	System clock base frequency	0x04
PLL0_CTRL	PLL0 configuration register	0x08
PLL1_CTRL	PLL1 configuration register	0x0C
PLL2_CTRL	PLL2 configuration register	0x10
PLL0_LOCK	PLL configuration register	0x18
SYS_CLK_SEL0	Clock selection control register0	0x20
SYS_CLK_SEL1	Clock selection control register1	0x24
SYS_CLK_EN	Module clock enable register	0x2C
SYS_SOFT_RST_CTRL	Soft reset control register	0x30
SYS_RST_CTRL	Module reset control register	0x34
SYS_CLK_TH0	Clock division setting register 0	0x38
SYS_CLK_TH1	Clock division setting register 1	0x3C
SYS_CLK_TH2	Clock division setting register 2	0x40
SYS_CLK_TH3	Clock division setting register 3	0x44
SYS_CLK_TH4	Clock division setting register 4	0x48

SYS_CLK_TH5	Clock division setting register 5	0x4C
SYS_CLK_TH6	Clock division setting register 6	0x50
SYS_MISC	Miscellaneous controller	0x54
SYS_PERI	Peripheral controller	0x58
SYS_SPI_SLEEP	SPI sleep controller	0x5C
SYS_RESET_STATUS	Reset source status	0x60
SYS_DMA_SEL0	DMA handshake selector	0x64
SYS_DMA_SEL 1	DMA handshake selector	0x68
SYS_POWER_SEL	IO Power Mode Select controller	0x6C

3.5.4 寄存器设置

3.5.4.1 Git short commit id (0x00)

Bits	Name	Description	Memory Access
31:0	git_id	Git commit ID	R

3.5.4.2 System clock base frequency (0x04)

Bits	Name	Description	Memory Access
31:0	clk_freq	Clock base frequency	R

3.5.4.3 PLL0 controller(0x08)

Reset value：0x458560

Bits	Name	Description	Memory Access
3:0	CLKR0	PLL0 reference clock division factor	R/W

9:4	CLKF0	PLL0 output octave factor	R/W
13:10	CLKOD0	PLL0 output clock division coefficient	R/W
19:14	BWADJ0	PLL0 frequency division coe ffi cient of bandwidth adjustment	R/W
20	PLL_reset0	PLL0 reset control, 1 for PLL reset, 0 for release reset	R/W
21	PLL_pwr0	PLL0 power off control, 0 for PLL power off, 1 for open	R/W
22	PLL_intfb0	PLL0 feedback loop configuration, must be set to 1	R/W
23	PLL_bypass0	PLL0 outputs reference clock directly when 1 is con fi gured	R/W
25	PLL_out_en0	PLL0 output enable	R/W

3.5.4.4 PLL1 controller(0x0C)

Reset value：0x458560

Bits	Name	Description	Memory Access
3:0	CLKR1	PLL1 reference clock division factor	R/W
9:4	CLKF1	PLL1 output octave factor	R/W
13:10	CLKOD1	PLL1 output clock division coefficient	R/W
19:14	BWADJ1	PLL1 frequency division coe ffi cient of bandwidth adjustment	R/W
20	PLL_reset1	PLL1 reset control, 1 for PLL reset, 0 for release reset	R/W
21	PLL_pwr1	PLL1 power off control, 0 for PLL power off, 1 for open	R/W
22	PLL_intfb1	PLL1 feedback loop configuration, must be set to 1	R/W
23	PLL_bypass1	PLL1 outputs reference clock directly when 1 is con fi gured	R/W

25	PLL_out_en1	PLL1 output enable	R/W

3.5.4.5 PLL2 controller(0x10)

Reset value：0x458560

Bits	Name	Description	Memory Access
3:0	CLKR2	PLL2 reference clock division factor	R/W
9:4	CLKF2	PLL2 output octave factor	R/W
13:10	CLKOD2	PLL2 output clock division coefficient	R/W
19:14	BWADJ2	PLL2 frequency division coefficient of bandwidth adjustment	R/W
20	PLL_RESET2	PLL2 reset control, 1 for PLL reset, 0 for release reset	R/W
21	PLL_PWR2	PLL2 power off control, 0 for PLL power off, 1 for open	R/W
22	PLL_INTFB2	PLL2 feedback loop configuration, must be set to 1	R/W
23	PLL_BYPASS2	PLL2 outputs reference clock directly when 1 is configured	R/W
25	PLL_OUT_EN2	PLL2 output enable	R/W
27:26	PLL_CKIN_SEL2	PLL2 Input reference clock selection 0:IN0 1:PLL0 2:PLL1	R/W

3.5.4.6 PLL LOCK(0x18)

Reset value：-

Bits	Name	Description	Memory Access

1:0	PLL_LOCK0	PLL0 Lock flag 0: No lock 1: Lock timer complete 2: PLL No loss of lock 3: PLL lock	R
2	PLL_SLIP_CLEAR0	If pll0 is not locked after pll0 lock timing, write 1 in this bit, clear the pll0 lock loss flag, and check whether pll0 is locked again	W
9:8	PLL_LOCK1	PLL1 Lock flag 0: No lock 1: Lock timer complete 2: PLL No loss of lock 3: PLL lock	R
10	PLL_SLIP_CLEAR1	If pll1 is not locked after pll0 lock timing, write 1 in this bit, clear the pll0 lock loss flag, and check whether pll0 is locked again	W
17:16	PLL_LOCK2	PLL2 Lock flag 0: No lock 1: Lock timer complete 2: PLL No loss of lock 3: PLL lock	R
18	PLL_SLIP_CLEAR2	If pll2 is not locked after pll0 lock timing, write 1 in this bit, clear the pll0 lock loss flag, and check whether pll0 is locked again	W

3.4.4.7 时钟选择控制 0(0x20)

Reset value：0xE248

Bits	Name	Description	Memory Access
0	aclk_sel	ACLK select 0:in0 1:pll0	R/W
2:1	aclk_divider_sel	When ACLK from pll0，The frequency division 0:2 frequency division, 1:4 frequency division 2:8 frequency division, 3:16 frequency division	R/W

5:3	apb0_clk_sel	Apb0 clock is obtained from aclk frequency division 0:aclk, 1:aclk/2 2:aclk/3 ... 7:aclk/8 The recommended value is 1，aclk/2	R/W

Bits	Name	Description	Memory Access
8:6	apb1_clk_sel	APB1 clock is obtained from aclk frequency division 0:aclk, 1:aclk/2 2:aclk/3 ... 7:aclk/8 The recommended value is 1，aclk/2	R/W
11:9	apb2_clk_sel	APB2 clock is obtained from aclk frequency division 0:aclk 1:aclk/2 2:aclk/3 ... 7:aclk/8 The recommended value is 1，aclk/2	R/W
12	spi3_clk_sel	SPI3 reference clock selection 0: clock in0 1: pll0	R/W
13	timer0_clk_sel	timer0 reference clock selection 0: clock in0 1: pll0	R/W
14	timer1_clk_sel	timer1 reference clock selection 0: clock in0, 1: pll0	R/W
15	timer2_clk_sel	timer2 reference clock selection 0: clock in0 1: pll0	R/W

3.5.4.8 时钟选择控制 1(0x24)

Reset value：0x0

Bits	Name	Description	Memory Access
0	spi3_sample_clk_sel	Input data sampling clock edge selection 0: Rising edge 1: Falling edge	R/W

3.5.4.9 模块时钟使能 (0x2C)

Reset value：0xCFFFFFF

Bits	Name	Description	Memory Access
0	rom_clk_en	ROM clock enable	1
1	dma_clk_en	DMA clock enable	1
2	ai_clk_en	AI clock enable	1
3	dvp_clk_en	DVP clock enable	1
4	fft_clk_en	FFT clock enable	1
5	gpio_clk_en	GPIO clock enable	1
6	spi0_clk_en	SPI0 clock enable	1
7	spi1_clk_en	SPI1 clock enable	1
8	spi2_clk_en	SPI2 clock enable	1
9	spi3_clk_en	SPI3 clock enable	1
10	i2s0_clk_en	I2S0 clock enable	1
11	i2s1_clk_en	I2S1 clock enable	1
12	i2s2_clk_en	I2S2 clock enable	1
13	i2c0_clk_en	I2C0 clock enable	1
14	i2c1_clk_en	I2C1 clock enable	1
15	i2c2_clk_en	I2C2 clock enable	1
16	uart1_clk_en	UART1 clock enable	1
17	uart2_clk_en	UART2 clock enable	1
18	uart3_clk_en	UART3 clock enable	1
19	aes_clk_en	AES clock enable	1
20	fpioa_clk_en	FPIOA clock enable	1
21	timer0_clk_en	TIMER0 clock enable	1
22	timer1_clk_en	TIMER1 clock enable	1
23	timer2_clk_en	TIMER2 clock enable	1
24	wdt0_clk_en	WDT0 clock enable	0
25	wdt1_clk_en	WDT1 clock enable	0

26	sha_clk_en	SHA clock enable	1
27	otp_clk_en	OTP clock enable	1
29	rtc_clk_en	RTC clock enable	0

3.5.4.10 软复位控制 (0x30)

Reset value: -

Bits	Name	Description	Memory Access
0	soft_reset	Software reset trigger signal, write 1 trigger system reset	W

3.5.4.11 模块复位控制 (0x34)

Reset value: 0x00

Bits	Name	Description	Memory Access
0	rom_reset	ROM reset control, set 1 for reset, set 0 for end of reset	R/W
1	dma_reset	DMA reset control, set 1 for reset, set 0 for end of reset	R/W
2	ai_reset	AI reset control, set 1 for reset, set 0 for end of reset	R/W
3	dvp_reset	DVP reset control, set 1 for reset, set 0 for end of reset	R/W
4	fft_reset	FFT reset control, set 1 for reset, set 0 for end of reset	R/W
5	gpio_reset	GPIO reset control, set 1 for reset, set 0 for end of reset	R/W
6	spi0_reset	SPI0 reset control, set 1 for reset, set 0 for end of reset	R/W
7	spi1_reset	SPI1 reset control, set 1 for reset, set 0 for end of reset	R/W
8	spi2_reset	SPI2 reset control, set 1 for reset, set 0 for end of reset	R/W

9	spi3_reset	SPI3 reset control, set 1 for reset, set 0 for end of reset	R/W
10	i2s0_reset	I2S0 reset control, set 1 for reset, set 0 for end of reset	R/W
11	i2s1_reset	I2S1 reset control, set 1 for reset, set 0 for end of reset	R/W
12	i2s2_reset	I2S2 reset control, set 1 for reset, set 0 for end of reset	R/W
13	i2c0_reset	I2C0 reset control, set 1 for reset, set 0 for end of reset	R/W
14	i2c1_reset	I2C1 reset control, set 1 for reset, set 0 for end of reset	R/W
15	i2c2_reset	I2C2 reset control, set 1 for reset, set 0 for end of reset	R/W
16	uart1_reset	UART1 reset control, set 1 for reset, set 0 for end of reset	R/W
17	uart2_reset	UART2 reset control, set 1 for reset, set 0 for end of reset	R/W

Bits	Name	Description	Memory Access
18	uart3_reset	UART3 reset control, set 1 for reset, set 0 for end of reset	R/W
19	aes_reset	AES reset control, set 1 for reset, set 0 for end of reset	R/W
20	fpioa_reset	FPIOA reset control, set 1 for reset, set 0 for end of reset	R/W
21	timer0_reset	TIMER0 reset control, set 1 for reset, set 0 for end of reset	R/W
22	timer1_reset	TIMER1 reset control, set 1 for reset, set 0 for end of reset	R/W
23	timer2_reset	TIMER2 reset control, set 1 for reset, set 0 for end of reset	R/W
24	wdt0_reset	WDT0 reset control, set 1 for reset, set 0 for end of reset	R/W

25	wdt1_reset	WDT1 reset control, set 1 for reset, set 0 for end of reset	R/W
26	sha_reset	SHA reset control, set 1 for reset, set 0 for end of reset	R/W

3.5.4.12 时钟分频设置 0(0x38)

Reset value：0x100

Bits	Name	Description	Memory Access
3:0	sram0_gclk_threshold	Frequency division based on aclk clock as reference clock 0:aclk 1:aclk/2 2:aclk/3 ..., 15:aclk/16	R/W
7:4	sram1_gclk_threshold	Frequency division based on aclk clock as reference clock 0:aclk 1:aclk/2 2:aclk/3, ..., 15:aclk/16	R/W
15:12	dvp_gclk_threshold	Frequency division based on aclk clock as reference clock 0:aclk 1:aclk/2 2:aclk/3, ..., 15:aclk/16	R/W
19:16	rom_gclk_threshold	Frequency division based on aclk clock as reference clock 0:aclk 1:aclk/2 2:aclk/3 ..., 15:aclk/16	R/W

3.5.4.13 时钟分频设置 1(0x3C)

Reset value：0x00

Bits	Name	Description	Memory Access

7:0	spi0_clk_threshold	Frequency division based on PLL0 clock as reference clock 0:clk/2 1:clk/4 2:clk/6 3:clk/8 ..., 255:clk/512	R/W

15:8	spi1_clk_threshold	Frequency division based on PLL0 clock as reference clock 0:clk/2 1:clk/4 2:clk/6 3:clk/8 ..., 255:clk/512	R/W
23:16	spi2_clk_threshold	Frequency division based on PLL0 clock as reference clock 0:clk/2 1:clk/4 2:clk/6 3:clk/8 ..., 255:clk/512	R/W
31:24	spi3_clk_threshold	Frequency division based on the selected spi3 clock as the reference clock 0:clk/2 1:clk/4 2:clk/6 3:clk/8 ..., 255:clk/512	R/W

3.5.4.14 时钟分频设置 2(0x40)

Reset value：0x00

Bits	Name	Description	Memory Access
7:0	timer0_clk_threshold	Frequency division based on the selected timer0 clock as the reference clock 0:clk/2 1:clk/4 2:clk/6 3:clk/8 ..., 255:clk/512	R/W
15:8	timer1_clk_threshold	Frequency division based on the selected timer1 clock as the reference clock 0:clk/2 1:clk/4 2:clk/6 3:clk/8 ..., 255:clk/512	R/W

23:16	timer2_clk_threshold	Frequency division based on the selected timer2 clock as the reference clock 0:clk/2 1:clk/4 2:clk/6 3:clk/8 ..., 255:clk/512	R/W

3.5.4.15 时钟分频设置 3(0x44)

Reset value：0x00

Bits	Name	Description	Memory Access
15:0	i2s0_clk_threshold	Frequency division based on pll2 output clock as reference clock 0:clk/2 1:clk/4 2:clk/6 3:clk/8 ..., 32767:clk/65536	R/W
31:16	i2s1_clk_threshold	Frequency division based on pll2 output clock as reference clock 0:clk/2 1:clk/4 2:clk/6 3:clk/8 ... 32767:clk/65536	R/W

3.5.4.16 时钟分频设置 4(0x48)

Reset value：0x00

Bits	Name	Description	Memory Access
15:0	i2s2_clk_threshold	Frequency division based on pll2 output clock as reference clock 0:clk/2 1:clk/4 2:clk/6 3:clk/8 ... 32767:clk/65536	R/W
23:16	i2s0_mclk_threshold	Frequency division based on pll2 output clock as reference clock 0:clk/2 1:clk/4 2:clk/6 3:clk/8 ... 255:clk/512	R/W
31:24	i2s1_mclk_threshold	Frequency division based on pll2 output clock as reference clock 0:clk/2 1:clk/4 2:clk/6 3:clk/8 ... 255:clk/512	R/W

3.5.4.17 时钟分频设置 5(0x4C)

Reset value：0x00

Bits	Name	Description	Memory Access
7:0	i2s2_mclk_threshold	Frequency division based on pll2 output clock as reference clock 0:clk/2 1:clk/4 2:clk/6 3:clk/8 ... 255:clk/512	R/W
15:8	i2c0_clk_threshold	Frequency division based on pll2 output clock as reference clock 0:clk/2 1:clk/4 2:clk/6 3:clk/8 ... 255:clk/512	R/W
23:16	i2c1_clk_threshold	Frequency division based on pll2 output clock as reference clock 0:clk/2 1:clk/4 2:clk/6 3:clk/8 ... 255:clk/512	R/W
31:24	i2c2_clk_threshold	Frequency division based on pll2 output clock as reference clock 0:clk/2 1:clk/4 2:clk/6 3:clk/8 ... 255:clk/512	R/W

3.5.4.18 时钟分频设置 6(0x50)

Reset value：0x00

Bits	Name	Description	Memory Access
7:0	wdt0_clk_threshold	Frequency division based on IN0 output clock as reference clock 0:clk/2 1:clk/4 2:clk/6 3:clk/8 ... 255:clk/512	R/W
15:8	wdt1_clk_threshold	Frequency division based on IN0 output clock as reference clock 0:clk/2 1:clk/4 2:clk/6 3:clk/8 ... 255:clk/512	R/W

3.5.4.19 Miscellaneous controller(0x54)

Reset value：0x00

Bits	Name	Description	Memory Access
5:0	debug_sel	Debug selector	RW
9:6	i2c2axi_spike_threshold	Threshold of clk count for stable logic. Glitch lesser than this threshold will be filtered.	RW
10	jamlink_io_sel	0: Extra 1.8V IO is JAMLINK clock, 1: Extra 1.8V IO is SPI0 with I2S0	RW
31:11	reserved	Reserved	RW

3.5.4.20 Peripheral controller (0x58)

Reset value：0x00

Bits	Name	Description	Memory Access
0	timer0_pause	Pause timer0	RW
1	timer1_pause	Pause timer1	RW
2	timer2_pause	Pause timer2	RW
3	timer3_pause	Pause timer3	RW
4	timer4_pause	Pause timer4	RW
5	timer5_pause	Pause timer5	RW
6	timer6_pause	Pause timer6	RW
7	timer7_pause	Pause timer7	RW
8	timer8_pause	Pause timer8	RW
9	timer9_pause	Pause timer9	RW
10	timer10_pause	Pause timer10	RW
11	timer11_pause	Pause timer11	RW
12	spi0_xip_en	SPI0 exec in place enable	RW

Bits	Name	Description	Memory Access
13	spi1_xip_en	SPI1 exec in place enable	RW
14	spi2_xip_en	SPI2 exec in place enable	RW
15	spi3_xip_en	SPI3 exec in place enable	RW
16	spi0_clk_bypass	SPI0 clock output bypass iomux	RW
17	spi1_clk_bypass	SPI1 clock output bypass iomux	RW
18	spi2_clk_bypass	SPI2 clock output bypass iomux	RW
19	i2s0_clk_bypass	I2S0 clock output bypass iomux	RW
20	i2s1_clk_bypass	I2S1 clock output bypass iomux	RW
21	i2s2_clk_bypass	I2S2 clock output bypass iomux	RW
22	jtag_clk_bypass	JTAG clock output bypass iomux	RW
23	dvp_clk_bypass	DVP clock input and output bypass iomux	RW
24	debug_clk_bypass	Debug clock output bypass iomux	RW
25	jamlink_clk_bypass	Jamlink clock output bypass iomux	RW
31:26	reserved	Reserved	RW

3.5.4.21 SPI sleep controller (0x5c)

Reset value：0x00

Bits	Name	Description	Memory Access
0	ssi0_sleep	SPI0 controller sleep	R
1	ssi1_sleep	SPI1 controller sleep	R
2	ssi2_sleep	SPI2 controller sleep	R

Bits	Name	Description	Memory Access
3	ssi3_sleep	SPI3 controller sleep	R
31:4	reserved	Reserved	R

3.5.4.22 Reset source status (0x60)

Reset value：0x00

Bits	Name	Description	Memory Access
0	reset_sts_clr	Reset status clear	W
1	pin_reset_sts	pin reset status	R
2	wdt0_reset_sts	wdt0 reset status	R
3	wdt1_reset_sts	wdt1 reset status	R
4	soft_reset_sts	soft reset status	R
31:5	reserved	Reserved	R

3.5.4.23 DMA handshake selector (0x64)

Reset value：0x00

Bits	Name	Description	Memory Access
5:0	dma_sel0		RW
11:6	dma_sel1		RW
17:12	dma_sel2		RW
23:18	dma_sel3		RW
29:24	dma_sel4		RW
31:30	reserved	Reserved	R

3.5.4.24 DMA handshake selector (0x68)

Reset value：0x00

Bits	Name	Description	Memory Access
5:0	dma_sel5		RW
31:6	reserved	Reserved	R

3.5.4.25 IO Power Mode Select controller (0x6c)

Reset value：0x00

Bits	Name	Description	Memory Access
0	power_mode_sel0	0: 3.3V IO mode, 1: 1.8V IO mode. Initial status must be 3.3V.	RW
1	power_mode_sel1	0: 3.3V IO mode, 1: 1.8V IO mode. Initial status must be 3.3V.	RW
2	power_mode_sel2	0: 3.3V IO mode, 1: 1.8V IO mode. Initial status must be 3.3V.	RW
3	power_mode_sel3	0: 3.3V IO mode, 1: 1.8V IO mode. Initial status must be 3.3V.	RW
4	power_mode_sel4	0: 3.3V IO mode, 1: 1.8V IO mode. Initial status must be 3.3V.	RW
5	power_mode_sel5	0: 3.3V IO mode, 1: 1.8V IO mode. Initial status must be 3.3V.	RW
6	power_mode_sel6	0: 3.3V IO mode, 1: 1.8V IO mode. Initial status must be 3.3V.	RW

Bits	Name	Description	Memory Access
7	power_mode_sel7	0: 3.3V IO mode, 1: 1.8V IO mode. Initial status must be 3.3V.	RW
31:8	reserved	Reserved	R

3.6 现场可编程 IO 阵列 (FPIOA/IOMUX)

3.6.1 概述

FPIOA（现场可编程IO阵列）允许用户将255个内部功能映射到芯片外围的48个自由IO上。主要功能包括：

- 支持IO的可编程功能选择；
- 支持IO输出的8种驱动能力选择；
- 支持IO的内部上拉电阻选择；
- 支持IO的内部下拉电阻选择；
- 支持IO输入的内部施密特触发器设置；
- 支持IO输出的斜率控制；
- 支持内部输入逻辑的电平设置。

3.6.2 功能描述

- I/O Pad供电

部分I/O默认输出3.3V，可切换为1.8V。

- 配置的一般流程

1）初始化使能模块；

2）配置功能管脚到需要的IO管脚。

- FPIOA pin Function table

Function	Name	Description
0	JTAG_TCLK	JTAG Test Clock
1	JTAG_TDI	JTAG Test Data In
2	JTAG_TMS	JTAG Test Mode Select
3	JTAG_TDO	JTAG Test Data Out
4	SPI0_D0	SPI0 Data 0
5	SPI0_D1	SPI0 Data 1
6	SPI0_D2	SPI0 Data 2
7	SPI0_D3	SPI0 Data 3
8	SPI0_D4	SPI0 Data 4
9	SPI0_D5	SPI0 Data 5
10	SPI0_D6	SPI0 Data 6
11	SPI0_D7	SPI0 Data 7
12	SPI0_SS0	SPI0 Chip Select 0
13	SPI0_SS1	SPI0 Chip Select 1
14	SPI0_SS2	SPI0 Chip Select 2
15	SPI0_SS3	SPI0 Chip Select 3
16	SPI0_ARB	SPI0 Arbitration
17	SPI0_SCLK	SPI0 Serial Clock
18	UARTHS_RX	UART High speed Receiver
19	UARTHS_TX	UART High speed Transmitter
20	RESV6	Reserved function
21	RESV7	Reserved function
22	CLK_SPI1	Clock SPI1
23	CLK_I2C1	Clock I2C
24	GPIOHS0	GPIO High speed 0
25	GPIOHS1	GPIO High speed 1

26	GPIOHS2	GPIO High speed 2
27	GPIOHS3	GPIO High speed 3
28	GPIOHS4	GPIO High speed 4
29	GPIOHS5	GPIO High speed 5

Function	Name	Description
30	GPIOHS6	GPIO High speed 6
31	GPIOHS7	GPIO High speed 7
32	GPIOHS8	GPIO High speed 8
33	GPIOHS9	GPIO High speed 9
34	GPIOHS10	GPIO High speed 10
35	GPIOHS11	GPIO High speed 11
36	GPIOHS12	GPIO High speed 12
37	GPIOHS13	GPIO High speed 13
38	GPIOHS14	GPIO High speed 14
39	GPIOHS15	GPIO High speed 15
40	GPIOHS16	GPIO High speed 16
41	GPIOHS17	GPIO High speed 17
42	GPIOHS18	GPIO High speed 18
43	GPIOHS19	GPIO High speed 19
44	GPIOHS20	GPIO High speed 20
45	GPIOHS21	GPIO High speed 21
46	GPIOHS22	GPIO High speed 22
47	GPIOHS23	GPIO High speed 23
48	GPIOHS24	GPIO High speed 24
49	GPIOHS25	GPIO High speed 25
50	GPIOHS26	GPIO High speed 26
51	GPIOHS27	GPIO High speed 27

52	GPIOHS28	GPIO High speed 28
53	GPIOHS29	GPIO High speed 29
54	GPIOHS30	GPIO High speed 30
55	GPIOHS31	GPIO High speed 31
56	GPIO0	GPIO pin 0
57	GPIO1	GPIO pin 1
58	GPIO2	GPIO pin 2
59	GPIO3	GPIO pin 3

Function	Name	Description
60	GPIO4	GPIO pin 4
61	GPIO5	GPIO pin 5
62	GPIO6	GPIO pin 6
63	GPIO7	GPIO pin 7
64	UART1_RX	UART1 Receiver
65	UART1_TX	UART1 Transmitter
66	UART2_RX	UART2 Receiver
67	UART2_TX	UART2 Transmitter
68	UART3_RX	UART3 Receiver
69	UART3_TX	UART3 Transmitter
70	SPI1_D0	SPI1 Data 0
71	SPI1_D1	SPI1 Data 1
72	SPI1_D2	SPI1 Data 2
73	SPI1_D3	SPI1 Data 3
74	SPI1_D4	SPI1 Data 4
75	SPI1_D5	SPI1 Data 5
76	SPI1_D6	SPI1 Data 6
77	SPI1_D7	SPI1 Data 7

78	SPI1_SS0	SPI1 Chip Select 0
79	SPI1_SS1	SPI1 Chip Select 1
80	SPI1_SS2	SPI1 Chip Select 2
81	SPI1_SS3	SPI1 Chip Select 3
82	SPI1_ARB	SPI1 Arbitration
83	SPI1_SCLK	SPI1 Serial Clock
84	SPI_SLAVE_D0	SPI Slave Data 0
85	SPI_SLAVE_SS	SPI Slave Select
86	SPI_SLAVE_SCLK	SPI Slave Serial Clock
87	I2S0_MCLK	I2S0 Master Clock
88	I2S0_SCLK	I2S0 Serial Clock(BCLK)
89	I2S0_WS	I2S0 Word Select(LRCLK)

Function	Name	Description
90	I2S0_IN_D0	I2S0 Serial Data Input 0
91	I2S0_IN_D1	I2S0 Serial Data Input 1
92	I2S0_IN_D2	I2S0 Serial Data Input 2
93	I2S0_IN_D3	I2S0 Serial Data Input 3
94	I2S0_OUT_D0	I2S0 Serial Data Output 0
95	I2S0_OUT_D1	I2S0 Serial Data Output 1
96	I2S0_OUT_D2	I2S0 Serial Data Output 2
97	I2S0_OUT_D3	I2S0 Serial Data Output 3
98	I2S1_MCLK	I2S1 Master Clock
99	I2S1_SCLK	I2S1 Serial Clock(BCLK)
100	I2S1_WS	I2S1 Word Select(LRCLK)
101	I2S1_IN_D0	I2S1 Serial Data Input 0
102	I2S1_IN_D1	I2S1 Serial Data Input 1

103	I2S1_IN_D2	I2S1 Serial Data Input 2
104	I2S1_IN_D3	I2S1 Serial Data Input 3
105	I2S1_OUT_D0	I2S1 Serial Data Output 0
106	I2S1_OUT_D1	I2S1 Serial Data Output 1
107	I2S1_OUT_D2	I2S1 Serial Data Output 2
108	I2S1_OUT_D3	I2S1 Serial Data Output 3
109	I2S2_MCLK	I2S2 Master Clock
110	I2S2_SCLK	I2S2 Serial Clock(BCLK)
111	I2S2_WS	I2S2 Word Select(LRCLK)
112	I2S2_IN_D0	I2S2 Serial Data Input 0
113	I2S2_IN_D1	I2S2 Serial Data Input 1
114	I2S2_IN_D2	I2S2 Serial Data Input 2
115	I2S2_IN_D3	I2S2 Serial Data Input 3
116	I2S2_OUT_D0	I2S2 Serial Data Output 0
117	I2S2_OUT_D1	I2S2 Serial Data Output 1
118	I2S2_OUT_D2	I2S2 Serial Data Output 2
119	I2S2_OUT_D3	I2S2 Serial Data Output 3

Function	Name	Description
120	RESV0	Reserved function
121	RESV1	Reserved function
122	RESV2	Reserved function
123	RESV3	Reserved function
124	RESV4	Reserved function
125	RESV5	Reserved function
126	I2C0_SCLK	I2C0 Serial Clock
127	I2C0_SDA	I2C0 Serial Data

128	I2C1_SCLK	I2C1 Serial Clock
129	I2C1_SDA	I2C1 Serial Data
130	I2C2_SCLK	I2C2 Serial Clock
131	I2C2_SDA	I2C2 Serial Data
132	CMOS_XCLK	DVP System Clock
133	CMOS_RST	DVP System Reset
134	CMOS_PWDN	DVP Power Down Mode
135	CMOS_VSYNC	DVP Vertical Sync
136	CMOS_HREF	DVP Horizontal Reference output
137	CMOS_PCLK	Pixel Clock
138	CMOS_D0	Data Bit 0
139	CMOS_D1	Data Bit 1
140	CMOS_D2	Data Bit 2
141	CMOS_D3	Data Bit 3
142	CMOS_D4	Data Bit 4
143	CMOS_D5	Data Bit 5
144	CMOS_D6	Data Bit 6
145	CMOS_D7	Data Bit 7
146	SCCB_SCLK	SCCB Serial Clock
147	SCCB_SDA	SCCB Serial Data
148	UART1_CTS	UART1 Clear To Send
149	UART1_DSR	UART1 Data Set Ready

Function	Name	Description
150	UART1_DCD	UART1 Data Carrier Detect
151	UART1_RI	UART1 Ring Indicator
152	UART1_SIR_IN	UART1 Serial Infrared Input

153	UART1_DTR	UART1 Data Terminal Ready
154	UART1_RTS	UART1 Request To Send
155	UART1_OUT2	UART1 User-designated Output 2
156	UART1_OUT1	UART1 User-designated Output 1
157	UART1_SIR_OUT	UART1 Serial Infrared Output
158	UART1_BAUD	UART1 Transmit Clock Output
159	UART1_RE	UART1 Receiver Output Enable
160	UART1_DE	UART1 Driver Output Enable
161	UART1_RS485_EN	UART1 RS485 Enable
162	UART2_CTS	UART2 Clear To Send
163	UART2_DSR	UART2 Data Set Ready
164	UART2_DCD	UART2 Data Carrier Detect
165	UART2_RI	UART2 Ring Indicator
166	UART2_SIR_IN	UART2 Serial Infrared Input
167	UART2_DTR	UART2 Data Terminal Ready
168	UART2_RTS	UART2 Request To Send
169	UART2_OUT2	UART2 User-designated Output 2
170	UART2_OUT1	UART2 User-designated Output 1
171	UART2_SIR_OUT	UART2 Serial Infrared Output
172	UART2_BAUD	UART2 Transmit Clock Output
173	UART2_RE	UART2 Receiver Output Enable
174	UART2_DE	UART2 Driver Output Enable
175	UART2_RS485_EN	UART2 RS485 Enable
176	UART3_CTS	UART3 Clear To Send
177	UART3_DSR	UART3 Data Set Ready
178	UART3_DCD	UART3 Data Carrier Detect
179	UART3_RI	UART3 Ring Indicator

Function	Name	Description
180	UART3_SIR_IN	UART3 Serial Infrared Input
181	UART3_DTR	UART3 Data Terminal Ready
182	UART3_RTS	UART3 Request To Send
183	UART3_OUT2	UART3 User-designated Output 2
184	UART3_OUT1	UART3 User-designated Output 1
185	UART3_SIR_OUT	UART3 Serial Infrared Output
186	UART3_BAUD	UART3 Transmit Clock Output
187	UART3_RE	UART3 Receiver Output Enable
188	UART3_DE	UART3 Driver Output Enable
189	UART3_RS485_EN	UART3 RS485 Enable
190	TIMER0_TOGGLE1	TIMER0 Toggle Output 1
191	TIMER0_TOGGLE2	TIMER0 Toggle Output 2
192	TIMER0_TOGGLE3	TIMER0 Toggle Output 3
193	TIMER0_TOGGLE4	TIMER0 Toggle Output 4
194	TIMER1_TOGGLE1	TIMER1 Toggle Output 1
195	TIMER1_TOGGLE2	TIMER1 Toggle Output 2
196	TIMER1_TOGGLE3	TIMER1 Toggle Output 3
197	TIMER1_TOGGLE4	TIMER1 Toggle Output 4
198	TIMER2_TOGGLE1	TIMER2 Toggle Output 1
199	TIMER2_TOGGLE2	TIMER2 Toggle Output 2
200	TIMER2_TOGGLE3	TIMER2 Toggle Output 3
201	TIMER2_TOGGLE4	TIMER2 Toggle Output 4
202	CLK_SPI2	Clock SPI2
203	CLK_I2C2	Clock SPI2
204	INTERNAL0	Internal function 0
205	INTERNAL1	Internal function 1

206	INTERNAL2	Internal function 2
207	INTERNAL3	Internal function 3
208	INTERNAL4	Internal function 4
209	INTERNAL5	Internal function 5

Function	Name	Description
210	INTERNAL6	Internal function 6
211	INTERNAL7	Internal function 7
212	INTERNAL8	Internal function 8
213	INTERNAL9	Internal function 9
214	INTERNAL10	Internal function 10
215	INTERNAL11	Internal function 11
216	INTERNAL12	Internal function 12
217	INTERNAL13	Internal function 13
218	INTERNAL14	Internal function 14
219	INTERNAL15	Internal function 15
220	INTERNAL16	Internal function 16
221	INTERNAL17	Internal function 17
222	CONSTANT	Constant function
223	INTERNAL18	Internal function 18
224	DEBUG0	Debug function 0
225	DEBUG1	Debug function 1
226	DEBUG2	Debug function 2
227	DEBUG3	Debug function 3
228	DEBUG4	Debug function 4
229	DEBUG5	Debug function 5
230	DEBUG6	Debug function 6

231	DEBUG7	Debug function 7
232	DEBUG8	Debug function 8
233	DEBUG9	Debug function 9
234	DEBUG10	Debug function 10
235	DEBUG11	Debug function 11
236	DEBUG12	Debug function 12
237	DEBUG13	Debug function 13
238	DEBUG14	Debug function 14
239	DEBUG15	Debug function 15

Function	Name	Description
240	DEBUG16	Debug function 16
241	DEBUG17	Debug function 17
242	DEBUG18	Debug function 18
243	DEBUG19	Debug function 19
244	DEBUG20	Debug function 20
245	DEBUG21	Debug function 21
246	DEBUG22	Debug function 22
247	DEBUG23	Debug function 23
248	DEBUG24	Debug function 24
249	DEBUG25	Debug function 25
250	DEBUG26	Debug function 26
251	DEBUG27	Debug function 27
252	DEBUG28	Debug function 28
253	DEBUG29	Debug function 29
254	DEBUG30	Debug function 30
255	DEBUG31	Debug function 31

- FPIOA default IO Function table

IO	FUNC_NAME	FUNC	ST	DI_INV	IE_INV	IE_EN	SL	SPU	PD	PU	DO_INV	DO_SEL	OE_INV	OE_EN	DS
IO0	JTAG_TCLK	0	1	0	0	1	0	0	0	0	0	0	0	0	0000
IO1	JTAG_TDI	1	1	0	0	1	0	0	0	0	0	0	0	0	0000
IO2	JTAG_TMS	2	1	0	0	1	0	0	0	0	0	0	0	0	0000
IO3	JTAG_TDO	3	0	0	0	0	0	0	0	0	0	0	0	1	1111
IO4	UARTHS_RX	18	1	0	0	1	0	0	0	0	0	0	0	0	0000
IO5	UARTHS_TX	19	0	0	0	0	0	0	0	0	0	0	0	1	1111
IO6	I2C2AXI_SCLK	217	1	0	0	1	0	0	0	1	0	0	0	0	0000
IO7	I2C2AXI_SDA	218	1	0	0	1	1	0	0	1	0	0	0	1	1111
IO8	GPIO0	56	1	0	0	1	0	0	0	0	0	0	0	0	1111
IO9	GPIO1	57	1	0	0	1	0	0	0	0	0	0	0	0	1111
IO10	GPIO2	58	1	0	0	1	0	0	0	0	0	0	0	0	1111
IO11	GPIO3	59	1	0	0	1	0	0	0	0	0	0	0	0	1111
IO12	GPIO4	60	1	0	0	1	0	0	0	0	0	0	0	0	1111
IO13	GPIO5	61	1	0	0	1	0	0	0	0	0	0	0	0	1111
IO14	GPIO6	62	1	0	0	1	0	0	0	0	0	0	0	0	1111
IO15	GPIO7	63	1	0	0	1	0	0	0	0	0	0	0	0	1111
IO16	GPIOHS0	24	1	0	0	1	0	0	0	0	0	0	0	0	1111
IO17	GPIOHS1	25	1	0	0	1	0	0	0	0	0	0	0	0	1111
IO18	GPIOHS2	26	1	0	0	1	0	0	0	0	0	0	0	0	1111
IO19	GPIOHS3	27	1	0	0	1	0	0	0	0	0	0	0	0	1111

IO20	GPIOHS4	28	1	0	0	1	0	0	0	0	0	0	0	1111
IO21	GPIOHS5	29	1	0	0	1	0	0	0	0	0	0	0	1111
IO22	GPIOHS6	30	1	0	0	1	0	0	0	0	0	0	0	1111
IO23	GPIOHS7	31	1	0	0	1	0	0	0	0	0	0	0	1111
IO24	GPIOHS8	32	1	0	0	1	0	0	0	0	0	0	0	1111
IO25	GPIOHS9	33	1	0	0	1	0	0	0	0	0	0	0	1111
IO26	GPIOHS10	34	1	0	0	1	0	0	0	0	0	0	0	1111
IO27	GPIOHS11	35	1	0	0	1	0	0	0	0	0	0	0	1111
IO28	GPIOHS12	36	1	0	0	1	0	0	0	0	0	0	0	1111
IO29	GPIOHS13	37	1	0	0	1	0	0	0	0	0	0	0	1111
IO30	GPIOHS14	38	1	0	0	1	0	0	0	0	0	0	0	1111
IO31	GPIOHS15	39	1	0	0	1	0	0	0	0	0	0	0	1111
IO32	GPIOHS16	40	1	0	0	1	0	0	0	0	0	0	0	1111
IO33	GPIOHS17	41	1	0	0	1	0	0	0	0	0	0	0	1111
IO34	GPIOHS18	42	1	0	0	1	0	0	0	0	0	0	0	1111
IO35	GPIOHS19	43	1	0	0	1	0	0	0	0	0	0	0	1111
IO36	GPIOHS20	44	1	0	0	1	0	0	0	0	0	0	0	1111
IO3	GPIOHS21	45	1	0	0	1	0	0	0	0	0	0	0	1111

7														
IO38	GPIOHS22	46	1	0	0	1	0	0	0	0	0	0	0	1111
IO39	GPIOHS23	47	1	0	0	1	0	0	0	0	0	0	0	1111
IO40	GPIOHS24	48	1	0	0	1	0	0	0	0	0	0	0	1111
IO41	GPIOHS25	49	1	0	0	1	0	0	0	0	0	0	0	1111
IO42	GPIOHS26	50	1	0	0	1	0	0	0	0	0	0	0	1111
IO43	GPIOHS27	51	1	0	0	1	0	0	0	0	0	0	0	1111
IO44	GPIOHS28	52	1	0	0	1	0	0	0	0	0	0	0	1111
IO45	GPIOHS29	53	1	0	0	1	0	0	0	0	0	0	0	1111
IO46	GPIOHS30	54	1	0	0	1	0	0	0	0	0	0	0	1111
IO47	GPIOHS31	55	1	0	0	1	0	0	0	0	0	0	0	1111

3.6.3 寄存器列表

FPIOA IO Pin RAM Layout

Base Address：FPIOA_BASE_ADDR (0x502B0000)

Name	Description	Offset address
IO0	FPIOA GPIO multiplexer io 0	0x00
IO1	FPIOA GPIO multiplexer io 1	0x04
IO2	FPIOA GPIO multiplexer io 2	0x08
IO3	FPIOA GPIO multiplexer io 3	0x0C
IO4	FPIOA GPIO multiplexer io 4	0x10

IO5	FPIOA GPIO multiplexer io 5	0x14
IO6	FPIOA GPIO multiplexer io 6	0x18
IO7	FPIOA GPIO multiplexer io 7	0x1C
IO8	FPIOA GPIO multiplexer io 8	0x20
IO9	FPIOA GPIO multiplexer io 9	0x24
IO10	FPIOA GPIO multiplexer io 10	0x28
IO11	FPIOA GPIO multiplexer io 11	0x2C
IO12	FPIOA GPIO multiplexer io 12	0x30
IO13	FPIOA GPIO multiplexer io 13	0x34
IO14	FPIOA GPIO multiplexer io 14	0x38
IO15	FPIOA GPIO multiplexer io 15	0x3C
IO16	FPIOA GPIO multiplexer io 16	0x40
IO17	FPIOA GPIO multiplexer io 17	0x44
IO18	FPIOA GPIO multiplexer io 18	0x48
IO19	FPIOA GPIO multiplexer io 19	0x4C
IO20	FPIOA GPIO multiplexer io 20	0x50
IO21	FPIOA GPIO multiplexer io 21	0x54
IO22	FPIOA GPIO multiplexer io 22	0x58
IO23	FPIOA GPIO multiplexer io 23	0x5C
IO24	FPIOA GPIO multiplexer io 24	0x60
IO25	FPIOA GPIO multiplexer io 25	0x64
IO26	FPIOA GPIO multiplexer io 26	0x68
IO27	FPIOA GPIO multiplexer io 27	0x6C
IO28	FPIOA GPIO multiplexer io 28	0x70
IO29	FPIOA GPIO multiplexer io 29	0x74

Name	Description	Offset address

IO30	FPIOA GPIO multiplexer io 30	0x78
IO31	FPIOA GPIO multiplexer io 31	0x7C
IO32	FPIOA GPIO multiplexer io 32	0x80
IO33	FPIOA GPIO multiplexer io 33	0x84
IO34	FPIOA GPIO multiplexer io 34	0x88
IO35	FPIOA GPIO multiplexer io 35	0x8C
IO36	FPIOA GPIO multiplexer io 36	0x90
IO37	FPIOA GPIO multiplexer io 37	0x94
IO38	FPIOA GPIO multiplexer io 38	0x98
IO39	FPIOA GPIO multiplexer io 39	0x9C
IO40	FPIOA GPIO multiplexer io 40	0xA0
IO41	FPIOA GPIO multiplexer io 41	0xA4
IO42	FPIOA GPIO multiplexer io 42	0xA8
IO43	FPIOA GPIO multiplexer io 43	0xAC
IO44	FPIOA GPIO multiplexer io 44	0xB0
IO45	FPIOA GPIO multiplexer io 45	0xB4
IO46	FPIOA GPIO multiplexer io 46	0xB8
IO47	FPIOA GPIO multiplexer io 47	0xBC

3.6.4 寄存器设置

3.6.4.1 FPIOA function tie bits RAM Layout

Name	Description	Offset address
TIE_EN31:0	Input tie enable bits 31:0	0xC0
TIE_EN63:32	Input tie enable bits 63:32	0xC4
TIE_EN95:64	Input tie enable bits 95:64	0xC8
TIE_EN127:96	Input tie enable bits 127:96	0xCC
TIE_EN159:128	Input tie enable bits 159:128	0xD0

TIE_EN191:160	Input tie enable bits 191:160	0xD4
TIE_EN223:192	Input tie enable bits 223:192	0xD8
TIE_EN255:224	Input tie enable bits 255:224	0xDC
TIE_VAL31:0	Input tie value bits 31:0	0xE0
TIE_VAL63:32	Input tie value bits 63:32	0xE4
TIE_VAL95:64	Input tie value bits 95:64	0xE8
TIE_VAL127:96	Input tie value bits 127:96	0xEC
TIE_VAL159:128	Input tie value bits 159:128	0xE0
TIE_VAL191:160	Input tie value bits 191:160	0xE4
TIE_VAL223:192	Input tie value bits 223:192	0xE8
TIE_VAL255:224	Input tie value bits 255:224	0xEC

3.6.4.2 FPIOA register bits Layout

Reset value：0x00

Bit	Name	Description	Access
31	PAD_DI	Read current IO s data input. 0:low 1:high	RO
30:24	RESV1	Reserved bits	-
23	ST	Schmitt trigger.0:disable 1:enable	WR
22	DI_INV	Invert Data input. 0:disable 1:invert	WR
21	IE_INV	Invert the input enable signal.0:disable 1:enable	WR
20	IE_EN	Input enable. 0:disable IO input 1:enable	WR
19	SL	Slew rate control enable.0:disable 1:enable	WR
18	RESV0	Reserved bits	-
17	PD	Pull down enable. 0 for nothing, 1 for pull down	WR
16	PU	Pull up enable. 0 for nothing, 1 for pull up	WR
15	DO_INV	Invert the result of data output select (DO_SEL)	WR

14	DO_SEL	Data output select: 0 for DO, 1 for OE	WR
13	OE_INV	Invert the output enable signal.0:disable 1:enable	WR
12	OE_EN	Output enable.0:disable IO output. 1:enable output.	WR
11:8	DS	Driving selector	WR
7:0	CH_SEL	Channel select from 256 input	WR

3.6.4.3 Driving selector Table

- Level Output Current

DS3:0	Min(mA)	Typ(mA)	Max(mA)
0000	3.2	5.4	8.3
0001	4.7	8.0	12.3
0010	6.3	10.7	16.4
0011	7.8	13.2	20.2
0100	9.4	15.9	24.2
0101	10.9	18.4	28.1
0110	12.4	20.9	31.8
0111	13.9	23.4	35.5

- High Level Output Current

DS3:0	Min(mA)	Typ(mA)	Max(mA)
0000	5.0	7.6	11.2
0001	7.5	11.4	16.8
0010	10.0	15.2	22.3
0011	12.4	18.9	27.8
0100	14.9	22.6	33.3
0101	17.4	26.3	38.7
0110	19.8	30.0	44.1

0111	22.3	33.7	49.5

3.7 看门狗定时器 (WDT 0/1)

3.7.1 概述

该模块包含2个看门狗定时器,分别为WDT0、WDT1。

3.7.2 功能描述

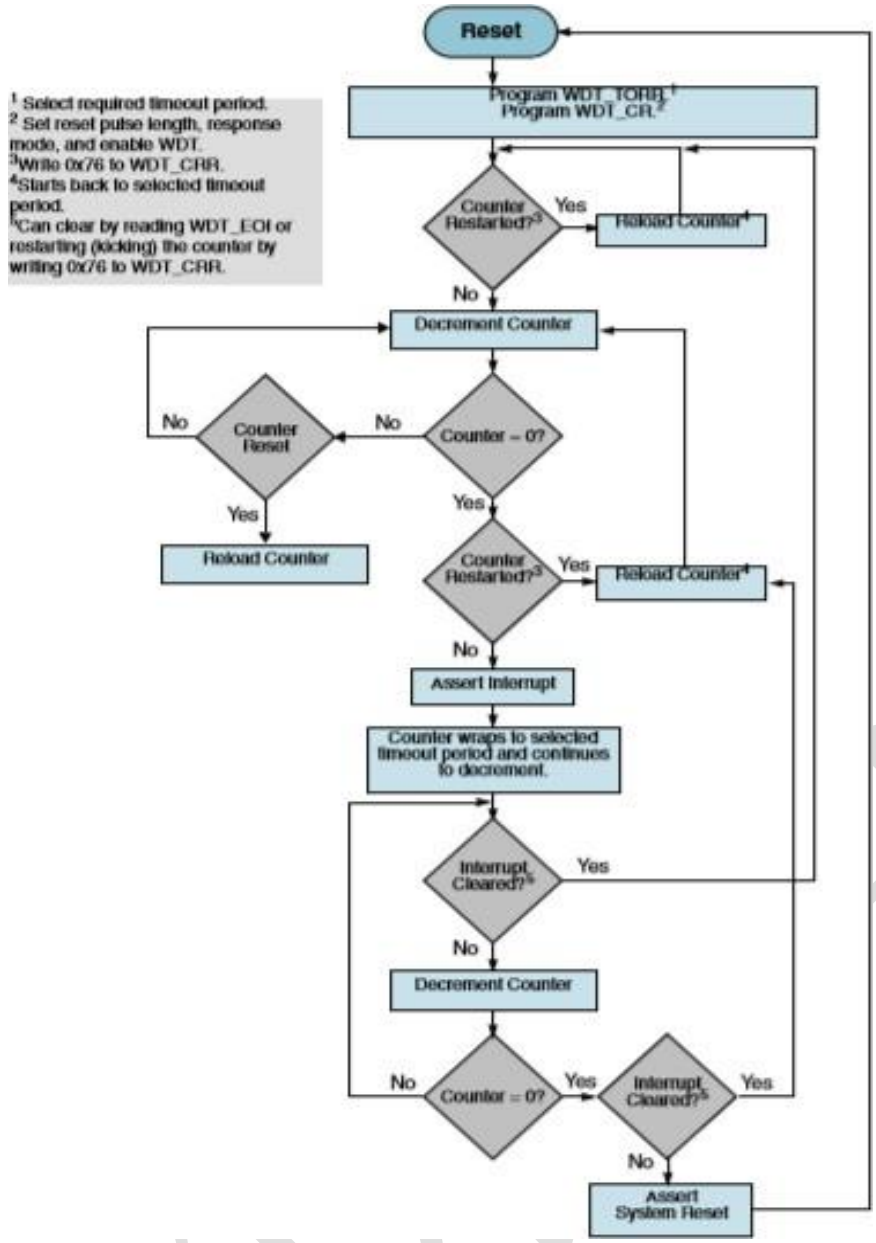
WDT是一种从外设，主要包含模块有：

- 一个APB从接口；
- 一个与当前计数器同步的寄存器模块；
- 一个随着计数器递减而中断或者系统重置的模块和逻辑控制电路；
- 一个同步时钟域来为异步时钟同步做支持的模块。

看门狗定时器支持如下设置：

- APB总线宽度可配置为8、16和32位；
- 时钟计数器从某一个设定的值递减到0来指示时间的计时终止；
- 当一个时钟超时，WDT可以执行以下任务：1.产生一个系统复位信号；2.产生一个中断。如果没有清除中断，第二次中断它会产生一个系统复位信号；
- 占空比可调节；
- 可编程超时周期；
- 外部异步时钟支持。当该项功能启用时，也可以产生时钟中断和系统复位信号，即使在APB总线时钟关闭的情况下。

配置操作流程：



3.7.3 寄存器列表

Base Address：WDT0_BASE_ADDR (0x50400000)

Base Address：WDT1_BASE_ADDR (0x50410000)

Name	Description	Offset address
WDT_CR	Control Register	0x0
WDT_TORR	Timeout Range Register	0x4
WDT_CCVR	Current Counter Value Register	0x8
WDT_CRR	Counter Restart Register	0xc

WDT_STAT	Interrupt Status Register	0x10
WDT_EOI	Interrupt Clear Register	0x14
WDT_COMP_PARAM_5	Component Parameters Register 5	0xe4
WDT_COMP_PARAM_4	Component Parameters Register 4	0xe8
WDT_COMP_PARAM_3	Component Parameters Register 3	0xec
WDT_COMP_PARAM_2	Component Parameters Register 2	0xf0
WDT_COMP_PARAM_1	Component Parameters Register 1	0xf4
WDT_COMP_VERSION	Component Version Register	0xf8
WDT_COMP_TYPE	Component Type Register	0xfc

3.7.4 寄存器设置

3.7.4.1 WDT_CR(0x0)

Bits	Name	Description	Memory Access
31:6	RSVD_WDT_CR	WDT_CR[31:6] Reserved bits and read as zero (0).	R
5	NO_NAME	Redundant R/W bit. Included for ping test purposes, as it is the only R/W register bit that is in every configuration of the apb_wdt.	R/W
4:2	RPL	Reset pulse length. Writes have no effect when the configuratio parameter WDT_HC_RPL is 1, making the regis read-only. This is used to select the numbe cycles for which the system reset stays ass The range of values available is 2 to 256 p cycles. Values: ■0x0 (PCLK_CYCLES2): 2 pclk cycles ■0x1 (PCLK_CYCLES4): 4 pclk cycles ■0x2 (PCLK_CYCLES8): 8 pclk cycles ■0x3 (PCLK_CYCLES16): 16 pclk cycles ■0x4 (PCLK_CYCLES32): 32 pclk cycles ■0x5 (PCLK_CYCLES64): 64 pclk cycles	R/W

		■0x6 (PCLK_CYCLES128): 128 pclk cycles ■0x7 (PCLK_CYCLES256): 256 pclk cycles	
1	RMOD	Response mode. Values: ■0x0 (RESET): Generate a system reset ■0x1 (INTERRUPT): First generate an interrup even if it is cleared by the time a second	R/W
0		WDT enable. Values: ■0x0 (DISABLED): Watchdog timer disabled ■0x1 (ENABLED): Watchdog timer enabled	R/W

3.7.4.2 WDT_TORR(0x4)

Bits	Name	Description	Memory Access
31:8	Reserved	WDT_TORR[31:24] Reserved and read as zero (0).	R
7:4	RSVD_TOP_INIT	WDT_TORR[7:4] Reserved field for TOP_INIT	R
3:0	TOP	Timeout period. Writes have no effect when the configuration parameter WDT_HC_TOP = 1, thus making this register read-only. This field is used to select the timeout period from which the watchdog counter restarts. A change of the timeout period takes effect only after the next counter restart (kick).The range of values is limited by the WDT_CNT_WIDTH. If TOP is programmed to select a range that is	R/W

greater than the counter width, the timeout period is truncated to fit to the counter width.
This affects only the non-user specified values as users are limited to these boundaries during configuration.The range of values available for a 32-bit watchdog counter are:Where i = TOP and t = timeout periodFor i = 0 to 15if WDT_USE_FIX_TOP==1t = 2(16 + i)elset = WDT_USER_TOP_(i)
Reset Value: WDT_DFLT_TOP Values:
0x0 (USER0_OR_64K): Time out of WDT_USER_TOP_0 or 64K Clocks 0x1 (USER1_OR_128K): Time out of WDT_USER_TOP_1 or 128K Clocks 0x2 (USER2_OR_256K): Time out of WDT_USER_TOP_2 or 256K Clocks 0x3 (USER3_OR_512K): Time out of WDT_USER_TOP_3 or 512K Clocks 0x4 (USER4_OR_1M): Time out of WDT_USER_TOP_4 or 1M Clocks 0x5 (USER5_OR_2M): Time out of WDT_USER_TOP_5 or 2M Clocks 0x6 (USER6_OR_4M): Time out of WDT_USER_TOP_6 or 4M Clocks 0x7 (USER7_OR_8M): Time out of WDT_USER_TOP_7 or 8M Clocks 0x8 (USER8_OR_16M): Time out of WDT_USER_TOP_8 or 16M Clocks 0x9 (USER9_OR_32M): Time out of WDT_USER_TOP_9 or 32M Clocks 0xa (USER10_OR_64M): Time out of WDT_USER_TOP_10 or 64M Clocks 0xb (USER11_OR_128M): Time out of WDT_USER_TOP_11 or 128M Clocks 0xc (USER12_OR_256M): Time out of WDT_USER_TOP_12 or 256M Clocks 0xd (USER13_OR_512M): Time out of WDT_USER_TOP_13 or 512M Clocks 0xe (USER14_OR_1G): Time out of WDT_USER_TOP_14 or 1G Clocks

		0xf (USER15_OR_2G): Time out of WDT_USER_TOP_15 or 2G Clocks	

3.7.4.3 WDT_CCVR(0x8)

Bits	Name	Description	Memory Access
31:0	WDT_CCVR	WDT Current Counter Value Register. This register, when read, is the current value of the internal counter. This value is read coherently when ever it is read, which is relevant when the APB_DATA_WIDTH is less than the counter width. Reset Value: WDT_CNT_RST Volatile: true	R

3.7.4.4 WDT_CRR(0xc)

Bits	Name	Description	Memory Access
31:8	RSVD_WDT_CRR	WDT_CRR[31:24] Reserved bits - Write Only	W
7:0	WDT_CRR	Counter Restart Register. This register is used to restart the WDT counter. As a safety feature to prevent accidental restarts, the value 0x76 must be written. A restart also clears the WDT interrupt. Reading this register returns zero. Reset Value: 0 Values: 0x76 (RESTART): Watchdog timer restart command	W

3.7.4.5 WDT_STAT(0x10)

Bits	Name	Description	Memory Access
31:1	RSVD_WDT_STAT	WDT_STAT[31] Reserved bits - Read Only Volatile: true	R

0	WDT_STAT	Interrupt status registerThis register shows the interrupt status of the WDT. Reset Value 0 Values: 0x0 (INACTIVE): Interrupt is inactive 0x1 (ACTIVE): Interrupt is active regardless of polarity Volatile: true	R

3.7.4.6 WDT_EOI(0x14)

Bits	Name	Description	Memory Access
31:1	RSVD_WDT_EOI	RSVD_WDT_EOI[31] Reserved bits and read as zero (0). Volatile: true	R
0	WDT_EOI	Interrupt Clear Register. Clears the watchdog interrupt. This can be used to clear the interrupt without restarting the watchdog counter. Reset Value: 0 Volatile: true	R

3.7.4.6 WDT_COMP_PARAM_5(0xe4)

Bits	Name	Description	Memory Access
31:0	CP WDT USER TOP MAX	Upper limit of Timeout Period parameters.The value of this register is derived from the WDT_USER_TOP_* coreConsultant parameters. Value After Reset: 0x0	R

3.7.4.7 WDT_COMP_PARAM_4(0xe8)

Bits	Name	Description	Memory Access

31:0	CP_WDT_USER_TOP_INIT_MAX	Upper limit of Initial Timeout Period parameters. The value of this register is derived from the WDT_USER_TOP_INIT_* coreConsultant parameters. Value After Reset: 0x0	R

3.7.4.8 WDT_COMP_PARAM_3(0xec)

Bits	Name	Description	Memory Access
31:0	CD_WDT_TOP_RST	The value of this register is derived from the WDT_TOP_RST coreConsultant parameter. Value After Reset: 0x0	R

3.7.4.9 WDT_COMP_PARAM_2(0xf0)

Bits	Name	Description	Memory Access
31:0	CP_WDT_CNT_RST	The value of this register is derived from the WDT_RST_CNT coreConsultant parameter. Value After Reset: 0xffff	R

3.7.4.10 WDT_COMP_PARAM_1(0xf4)

Bits	Name	Description	Memory Access
31:29	RSVD_31_29	WDT_COMP_PARAM_1[31:29]Reserved bits and read as zero (0). Value After Reset: 0x0	R
28:24	WDT_CNT_WIDTH	The Watchdog Timer counter width. Value After Reset: 0x10	R
23:20	WDT_DFLT_TOP_INIT	Describes the initial timeout period that is available directly after reset. It controls the reset value of the register. If WDT_HC_TOP is 1, then the default initial time period is the only possible period. Value After Reset: 0x0	R

19:16	WDT_DFLT_TOP	Selects the timeout period that is available directly after reset. It controls the reset value of the register. If WDT_HC_TOP is set to 1, then the default timeout period is the only possible timeout period. Can choose one of 16 values. Value After Reset: 0x0	R
15:13	RSVD_15_13	WDT_COMP_PARAM_1[15:13] Reserved bits and read as zero (0). Value After Reset: 0x0	R
12:10	WDT_DFLT_RPL	The reset pulse length that is available directly after reset. Value After Reset: 0x0	R

Bits	Name	Description	Memory Access
9:8	APB_DATA_WIDTH	Width of the APB Data Bus to which this component is attached. Values: 0x0 (APB_8BITS): APB data width is 8 bits 0x1 (APB_16BITS): APB data width is 16 bits 0x2 (APB_32BITS): APB data width is 32 bits Value After Reset: 0x2	R
7	WDT_PAUSE	Configures the peripheral to have a pause enable signal (pause) on the interface that can be used to freeze the watchdog counter during pause mode. Values: 0x0 (DISABLED): Pause enable signal is non existent 0x1 (ENABLED): Pause enable signal is included Value After Reset: 0x0	R

6	WDT_USE_FIX_TOP	When this parameter is set to 1, timeout period range is fixed. The range increments by the power of 2 from 2^16 to 2^(WDT_CNT_WIDTH-1). When this parameter is set to 0, the user must define the timeout period range (2^8 to 2^(WDT_CNT_WIDTH)-1) using the WDT_USER_TOP_(i) parameter. Values: 0x0 (USERDEFINED): User must define timeout values 0x1 (PREDEFINED): Use predefined timeout values Value After Reset: 0x1	R
5	WDT_HC_TOP	When set to 1, the selected timeout period(s) is set to be hard coded. Values: 0x0 (PROGRAMMABLE): Timeout period is programmable 0x1 (HARDCODED): Timeout period is hard coded Value After Reset: 0x0	R

Bits	Name	Description	Memory Access
4	WDT_HC_RPL	Configures the reset pulse length to be hard coded. Values: 0x0 (PROGRAMMABLE): Reset pulse length is programmable 0x1 (HARDCODED): Reset pulse length is hardcoded Value After Reset: 0x0	R

3	WDT_HC_RMOD	Configures the output response mode to be hard coded. Values: 0x0 (PROGRAMMABLE): Output response mode is programmable 0x1 (HARDCODED): Output response mode is hard coded Value After Reset: 0x0	R
2	WDT_DUAL_TOP	When set to 1, includes a second timeout period that is used for initialization prior to the first kick. Values: 0x0 (DISABLED): Second timeout period is not present 0x1 (ENABLED): Second timeout period is present Value After Reset: 0x0	R
1	WDT_DFLT_RMOD	Describes the output response mode that is available directly after reset. Indicates the output response the WDT gives if a zero count is reached; that is, a system reset if equals 0 and an interrupt followed by a system reset, if equals 1. If WDT_HC_RMOD is 1, then default response mode is the only possible output response mode. Values: 0x0 (DISABLED): System reset only 0x1 (ENABLED): Interrupt and system reset Value After Reset: 0x0	R

0	WDT_ALWAYS_EN	Configures the WDT to be enabled from reset. If this setting is 1, the WDT is always enabled and a write to the WDT_EN field (bit 0) of the Watchdog Timer Control Register (WDT_CR) to disable it has no effect. Values: 0x0 (DISABLED): Watchdog timer disabled on reset 0x1 (ENABLED): Watchdog timer enabled on reset Value After Reset: 0x0	R

3.7.4.11 WDT_COMP_VERSION(0xf8)

Bits	Name	Description	Memory Access
31:0	WDT_COMP_VERSION	ASCII value for each number in the version.	R

3.7.4.12 WDT_COMP_TYPE(0xfc)

Bits	Name	Description	Memory Access
31:0	WDT_COMP_TYPE	Designware Component Type number = 0x44_57_01_20. This assigned unique hex value is constant, and is derived from the two ASCII letters DW followed by a 16-bit unsigned number.	R

3.8 高级加密加速器 (AES Accelerater)

3.8.1 概述

K210内置AES(高级加密加速器)，相对于软件可以极大的提高AES运算速度。AES加速器支持多种加密/解密模式(ECB,CBC,GCM)，多种长度的KEY(ECB,CBC,GCM)的运算。主要功能包含：

- 支持ECB-128加解密运算；
- 支持ECB-192加解密运算；
- 支持ECB-256加解密运算；
- 支持CBC-128加解密运算；
- 支持CBC-192加解密运算；
- 支持CBC-256加解密运算；
- 支持GCM-128加解密运算；
- 支持GCM-192加解密运算；
- 支持GCM-256加解密运算；
- KEY可以同过软件配置，也可以通过OTP直接硬件加载(安全效果更好，软件层面无法破解)；
- 支持DMA数据输入；
- 支持4种密钥字节序和4种文本字节序。

3.8.2 功能描述

- 运算模式选择：AES加速器支持ECB/CBC/GCM三种加密模式，并支持128/192/256三种长度的KEY，通过配置gb_encrpt_sel来选择加密或者解密，通过配置gb_cipher_mode[2:0]来配置加密模式，通过配置kmode[1:0]来选择所用的KEY的长度。

gb_encrpt_sel	加解密选择
0	加密
1	解密

gb_cipher_mode[2:0]	模式选择
0	ECB
1	CBC
2	GCM

kmode[1:0]	KEY 的长度
0	128
1	192
2	256

- 配置的一般流程

1. 选择数据大小端模式；
2. 写入16字节的key；
3. 写入16字节的IV值；
4. 选择加密模式（cbc,ecb,gcm），以及加解密模式；
5. 写入数据；
6. 等待加解密完成；
7. 读回结果。

3.8.3 寄存器列表

Base Address：AES_BASE_ADDR (0x50450000)

Name	Description	Offset address
AES_KEY_SW_0	KEY[31:0]bits	0x00
AES_KEY_SW_1	KEY[63:32]bits	0x04

AES_KEY_SW_2	KEY[95:64]bits	0x08
AES_KEY_SW_3	KEY[127:96]bits	0x0c
AES_KEY_SW_4	KEY[159:128]bits	0x84
AES_KEY_SW_5	KEY[191:160]bits	0x88
AES_KEY_SW_6	KEY[223:192]bits	0x8c
AES_KEY_SW_7	KEY[255:224]bits	0x90
ENCRYPT_SEL	[31:1]retain，bit0： 0：encrypt 1：decrypt	0x10
AES_MODE_REG	Used to set AES encryption and decryption mode, key length, and byte order	0x14
AES_IV_0	This register is used to set the [31:0] bit of IV input in the CBC and GCM modes	0x18
AES_IV_1	This register is used to set the [63:32] bit of IV input in the CBC and GCM modes	0x1C
AES_IV_2	This register is used to set the [95:64] bit of IV input in the CBC and GCM modes	0x20
AES_IV_3	This register is used to set the [127:96] bit of IV input in the CBC and GCM modes	0x24
AES_ENDIAN	The [31:1] of this register is reserved, and bit 0 is used to set the Endian of input data (0: Big Endian; 1: Little Endian)	0x28
CAL_FINISH	The [31:1] of this register is reserved，bit0 is the finish flag.0：not finish 1：finish	0x2c
DMA_SEL	The [31:1] of this register is reserved, and bit 0 is used to set whether the calculated data is moved by DMA or directly read by CPU 0：CPU 1：DMA	0x30
AES_AAD_NUM	AAD data number	0x34
AES_PC_NUM	Used to set the amount of data entered for encryption or decryption	0x3c

Name	Description	Offset address

AES_TEXT_DATA	Encrypted or decrypted data is input from this register	0x40
AES_AAD_DATA	The input of AAD is input from this register	0x44
TAG_CHK	The [31:2] of the register is reserved, and the [1:0] bit is used to mark the check result of the tag： 00：Not finished 01：Incorrect tag 10：tag ok 11：reserved	0x48
DATA_IN_FLAG	The [31:1] of this register is reserved, and whether bit0 flag allows data input: 0，Input not allowed 1：Input allowed	0x4c
GCM_IN_TAG_0	GCM TAG[31:0]	0x50
GCM_IN_TAG_1	GCM TAG的[63:32]	0x54
GCM_IN_TAG_2	GCM TAG[95:64]	0x58
GCM_IN_TAG_3	GCM TAG[127:96]	0x5c
AES_OUT_DATA	get result data	0x60
AES_ENABLE	The [31:1] of this register is reserved, and bit 0 is used to enable AES calculation	0x64
DATA_OUT_FLAG	The [31:1] of this register is reserved, and bit 0 is used to mark whether the output data is valid	0x68
TAG_IN_FLAG	The [31:1] of this register is reserved, and bit 0 is used to allow the input of tag	0x6c
TAG_CLR	The [31:1] of this register is reserved, and bit 0 is used to clear the tag_ clk	0x70
GCM_OUT_TAG_0	GCM output TAG[31:0]	0x74
GCM_OUT_TAG_1	GCM output TAG[63:32]	0x78
GCM_OUT_TAG_2	GCM output TAG[95:64]	0x7c
GCM_OUT_TAG_3	GCM output TAG[127:96]	0x80

3.8.4 寄存器设置

寄存器设置说明

- 选择完解密和解密模式后，需要进行key的选择，可以通过配置AES_KEY_SW_0-7来进行软件配置，也可以在otp模块内通过配置OTP_KEY_EN，选择使用otp内部的key(otp内部的可以只可以选择128bit)；
- 如果是配置成GCM和CBC模式，还需要通过AES_iv模块来配置IV；
- 通过配置AES_AAD_NUM来配置GCM模式下AAD的数据的长度，通过配置AES_PC_NUM来设置需要解密和解密的明文和密文的长度；(这两个寄存器的设置需要配置成：总长度-1，例如输入数据 总长度是10，那么配置成9)；
- 通过配置DMA_SEL选择输入的数据是通过dma写进还是通过总线写入；基本配置完成之后通过使能AES_ENABLE，来使能这一次的运算；
- AAD数据需要通过寄存器AES_AAD_DATA[31：0]写入，明文或者密文通过AES_TEST_DATA[31:0]写入；
- 计算完的数据会通过AES_OUT_DATA[31:0]返回，通过DATA_OUT_FLAG来判读是否可以读走处理完的数据。GCM模式下的tag会通过GCM_OUT_TAG返回；

3.8.4.1 AES_KEY_SW_0-7(0X50450000-0X50450090)

Reset value: 0x00

Bits	Name	Description	Memory Access
31:0	AES_KEY_SW_0	User key，KEY[31:0]	W

Bits	Name	Description	Memory Access
31:0	AES_KEY_SW_0	User key，KEY[31:0]	W

Bits	Name	Description	Memory Access
31:0	AES_KEY_SW_2	User key，KEY[95:64]	W

Bits	Name	Description	Memory Access
31:0	AES_KEY_SW_3	User key，KEY[127:96]	W

Bits	Name	Description	Memory Access
31:0	AES_KEY_SW_4	User key，KEY[159:128]	W

Bits	Name	Description	Memory Access
31:0	AES_KEY_SW_5	User key，KEY[191:160]	W

Bits	Name	Description	Memory Access
31:0	AES_KEY_SW_6	User key，KEY[223:192]	W

Bits	Name	Description	Memory Access
31:0	AES_KEY_SW_7	User key，KEY[255:224]	W

备注：AES_KEY_SW_0-7,用来设置用于加密或者解密的KEY。如使用128bit的KEY只需要设置AES_KEY_SW_0-3;如使用192bit的KEY只需要设置AES_KEY_SW_0-5,如使用256bit的KEY只需要设置AES_KEY_SW_0-7;

3.8.4.2 ENCRYPT_SEL(0X50450010)

Reset value: 0x00

Bits	Name	Description	Memory Access
31:1	RSRV	reserved	R
0	encrpty_sel	Used to select encryption and decryption 0：encryption 1：decryption	R/W

3.8.4.3 AES_MODE_REG(0x14)

Reset value: 0x00

Bits	Name	Description	Memory Access
31:11	RSRV	reserved	R
10:9	AES_OUTPUT_ORDER	AES_OUTPUT_ORDER	R/W
8:7	AES_INPUT_ORDER	AES_INPUT_ORDER	R/W
6:5	AES_KEY_ORDER	AES_KEY_ORDER	R/W

4:3	AES_KEY_MODE	The length of key 00:128 01:192 10:256 11:保留	R/W
1:0	AES_CIPHER_MODE	Select mode 000:ecb 001:cbc 010：aes_gcm	R/W

3.8.4.4 AES_IV_0-3(0x18-0X24)

Reset value: 0x00

Bits	Name	Description	Memory Access
31:0	AES_IV_0	This register is used to set the [31:0] bit of IV input in the CBC and GCM modes	W

Bits	Name	Description	Memory Access
31:0	AES_IV_1	This register is used to set the [63:32] bit of IV input in the CBC and GCM modes	W

Bits	Name	Description	Memory Access
31:0	AES_IV_2	This register is used to set the [95:64] bit of IV input in the CBC and GCM modes	W

Bits	Name	Description	Memory Access
31:0	AES_IV_3	This register is used to set the [127:96] bit of IV input in the CBC and GCM modes	W

备注：AES_IV_0-3用来配置GCM和CBC模式下使用的IV，GCM模式下只需要配置AES_IV_0~2，CBC模式需要配置AES_IV_0~3；

3.8.4.5 AES_ENDIAN(0x28)

Reset value: 0x00

Bits	Name	Description	Memory Access
31:1	RSRV	Reserved	R
0	AES_ENDIAN	Set the Endian of input data (0: Big Endian; 1: Little Endian)	R

3.8.4.6 CAL_FINISH(0x2c)

Reset value: 0x00

Bits	Name	Description	Memory Access
31:1	RSRV	Reserved	R
0	CAL_FINISH	Finish flag 0：not finish 1：finish	R

3.8.4.7 DMA_SEL(0x30)

Reset value: 0x00

Bits	Name	Description	Memory Access
31:1	RSRV	Reserved	R
0	DMA_SEL	the calculated data is moved by DMA or directly read by CPU 0：CPU 1：DMA	W

3.8.4.8 AES_AAD_NUM(0x34)

Reset value: 0x00

Bits	Name	Description	Memory Access
31:0	AES_AAD_NUM	AAD number	R/W

备注：AES_AAD_NUM在设置的时候要等于真实的数值-1，例如输入的AAD为3，那么AES_AAD_NUMJ设置为2。

3.8.4.9 AES_PC_NUM(0x3c)

Reset value: 0x00

Bits	Name	Description	Memory Access
31:0	AES_PC_NUM	Used to set the amount of data entered for encryption or decryption	R/W

备注：AES_PC_NUM在设置的时候要等于真实的数值-1，例如输入的PC为5，那么AES_PC_NUMJ设置为4。

3.8.4.10 AES_TEXT_DATA(0x40)

Reset value: 0x00

Bits	Name	Description	Memory Access
31:0	AES_TEXT_DATA	input data	R/W

3.8.4.11 AES_AAD_DATA(0x44)

Reset value: 0x00

Bits	Name	Description	Memory Access
31:0	AES_AAD_DATA	AAD input	R/W

3.8.4.12 TAG_CHK(0x48)

Reset value: 0x00

Bits	Name	Description	Memory Access
31:2	RSRV	Reserved	R
1:0	TAG_CHK	Check result used to mark tag 00: not finished 01: tag incorrect 10: tag correct 11: reserved	R

3.8.4.13 DATA_IN_FLAG(0x4c)

Reset value: 0x00

Bits	Name	Description	Memory Access

31:1	RSRV	Reserved	R
0	DATA_IN_FLAG	Used to mark whether encryption or decryption data is allowed to enter 0:not allowed to enter 1: allowed to enter	R

3.8.4.14 AES_ENABLE(0x64)

Reset value: 0x00

Bits	Name	Description	Memory Access
31:1	RSRV	Reserved	R
0	AES_ENABLE	Set 1 to enable AES module	W

3.8.4.15 DATA_OUT_FLAG(0x68)

Reset value: 0x00

Bits	Name	Description	Memory Access
31:1	RSRV	Reserved	R
0	DATA_OUT_FLAG	Used to mark whether there is data output	R

3.8.4.16 TAG_IN_FLAG(0x6c)

Reset value: 0x00

Bits	Name	Description	Memory Access
31:1	RSRV	Reserved	R
0	TAG_IN_FLAG	Used to mark whether the tag for comparison is allowed	R

3.8.4.17 TAG_CLR(0x70)

Reset value: 0x00

Bits	Name	Description	Memory Access

31:1	RSRV	Reserved	R
0	TAG_CLR	Clear TAG_CHK	W

3.8.4.18 GCM_IN_TAG_0-3(0x50-0x5c)

Reset value: 0x00

Bits	Name	Description	Memory Access
31:0	GCM_IN_TAG_0	IN TAG[31:0]	R/W

Bits	Name	Description	Memory Access
31:0	GCM_IN_TAG_1	IN TAG[63:32]	R/W

Bits	Name	Description	Memory Access
31:0	GCM_IN_TAG_2	IN TAG[95:64]	R/W

Bits	Name	Description	Memory Access
31:0	GCM_IN_TAG_3	IN TAG[127:96]	R/W

备注：GCM_IN_0-3用来输入用于比对的GCM的TAG

3.8.4.19 GCM_OUT_TAG_0-3(0x74-0x80)

Reset value: 0x00

Bits	Name	Description	Memory Access
31:0	GCM_OUT_TAG_0	TAG[31:0]	R

Bits	Name	Description	Memory Access
31:0	GCM_OUT_TAG_1	TAG[63:32]	R

Bits	Name	Description	Memory Access
31:0	GCM_OUT_TAG_2	TAG[95:64]	R

Bits	Name	Description	Memory Access
31:0	GCM_OUT_TAG_3	TAG[127:96]	R

备注：GCM_OUT_0-3用来输出用于比对的GCM的TAG

3.9 中断（Interrupt）

3.9.1 概述

可以将任一外部中断源单独分配到每个 CPU 的外部中断上。这提供了强大的灵活性，能适应不同的应用需求。主要特性包括：

- 接受 65 个外部中断源作为输入；
- 每个中断都有唯一的中断号，优先级可配置。

3.9.2 功能描述

3.9.2.1 Interrupt Sources

芯片共有 65 个外部中断源，都可以分配给两个 CPU。

Interrupt number	Name	Description
0	IRQN_NO_INTERRUPT	The non-existent interrupt
1	IRQN_SPI0_INTERRUPT	SPI0 interrupt
2	IRQN_SPI1_INTERRUPT	SPI1 interrupt
3	IRQN_SPI_SLAVE_INTERRUPT	SPI_SLAVE interrupt
4	IRQN_SPI3_INTERRUPT	SPI3 interrupt
5	IRQN_I2S0_INTERRUPT	I2S0 interrupt
6	IRQN_I2S1_INTERRUPT	I2S1 interrupt
7	IRQN_I2S2_INTERRUPT	I2S2 interrupt
8	IRQN_I2C0_INTERRUPT	I2C0 interrupt
9	IRQN_I2C1_INTERRUPT	I2C1 interrupt
10	IRQN_I2C2_INTERRUPT	I2C2 interrupt
11	IRQN_UART1_INTERRUPT	UART1 interrupt
12	IRQN_UART2_INTERRUPT	UART2 interrupt
13	IRQN_UART3_INTERRUPT	UART3 interrupt
14	IRQN_TIMER0A_INTERRUPT	TIMER0 channel 0 or 1 interrupt
15	IRQN_TIMER0B_INTERRUPT	TIMER0 channel 2 or 3 interrupt
16	IRQN_TIMER1A_INTERRUPT	TIMER1 channel 0 or 1 interrupt
17	IRQN_TIMER1B_INTERRUPT	TIMER1 channel 2 or 3 interrupt
18	IRQN_TIMER2A_INTERRUPT	TIMER2 channel 0 or 1 interrupt
19	IRQN_TIMER2B_INTERRUPT	TIMER2 channel 2 or 3 interrupt
20	IRQN_RTC_INTERRUPT	RTC tick and alarm interrupt

21	IRQN_WDT0_INTERRUPT	Watching dog timer0 interrupt
22	IRQN_WDT1_INTERRUPT	Watching dog timer1 interrupt
23	IRQN_APB_GPIO_INTERRUPT	APB GPIO interrupt
24	IRQN_DVP_INTERRUPT	Digital video port interrupt
25	IRQN_AI_INTERRUPT	AI accelerator interrupt
26	IRQN_FFT_INTERRUPTFFT	FFT accelerator interrupt
27	IRQN_DMA0_INTERRUPT	DMA channel0 interrupt
28	IRQN_DMA1_INTERRUPT	DMA channel1 interrupt
29	IRQN_DMA2_INTERRUPT	DMA channel2 interrupt
30	IRQN_DMA3_INTERRUPT	DMA channel3 interrupt
31	IRQN_DMA4_INTERRUPT	DMA channel4 interrupt
32	IRQN_DMA5_INTERRUPT	DMA channel5 interrupt
33	IRQN_UARTHS_INTERRUPT	Hi-speed UART0 interrupt
34	IRQN_GPIOHS0_INTERRUPT	Hi-speed GPIO0 interrupt
35	IRQN_GPIOHS1_INTERRUPT	Hi-speed GPIO1 interrupt
36	IRQN_GPIOHS2_INTERRUPT	Hi-speed GPIO2 interrupt
37	IRQN_GPIOHS3_INTERRUPT	Hi-speed GPIO3 interrupt
38	IRQN_GPIOHS4_INTERRUPT	Hi-speed GPIO4 interrupt
39	IRQN_GPIOHS5_INTERRUPT	Hi-speed GPIO5 interrupt
40	IRQN_GPIOHS6_INTERRUPT	Hi-speed GPIO6 interrupt
41	IRQN_GPIOHS7_INTERRUPT	Hi-speed GPIO7 interrupt
42	IRQN_GPIOHS8_INTERRUPT	Hi-speed GPIO8 interrupt
43	IRQN_GPIOHS9_INTERRUPT	Hi-speed GPIO9 interrupt
44	IRQN_GPIOHS10_INTERRUPT	Hi-speed GPIO10 interrupt
45	IRQN_GPIOHS11_INTERRUPT	Hi-speed GPIO11 interrupt
46	IRQN_GPIOHS12_INTERRUPT	Hi-speed GPIO12 interrupt
47	IRQN_GPIOHS13_INTERRUPT	Hi-speed GPIO13 interrupt
48	IRQN_GPIOHS14_INTERRUPT	Hi-speed GPIO14 interrupt
49	IRQN_GPIOHS15_INTERRUPT	Hi-speed GPIO15 interrupt
50	IRQN_GPIOHS16_INTERRUPT	Hi-speed GPIO16 interrupt
51	IRQN_GPIOHS17_INTERRUPT	Hi-speed GPIO17 interrupt
52	IRQN_GPIOHS18_INTERRUPT	Hi-speed GPIO18 interrupt
53	IRQN_GPIOHS19_INTERRUPT	Hi-speed GPIO19 interrupt
54	IRQN_GPIOHS20_INTERRUPT	Hi-speed GPIO20 interrupt
55	IRQN_GPIOHS21_INTERRUPT	Hi-speed GPIO21 interrupt
56	IRQN_GPIOHS22_INTERRUPT	Hi-speed GPIO22 interrupt
57	IRQN_GPIOHS23_INTERRUPT	Hi-speed GPIO23 interrupt
58	IRQN_GPIOHS24_INTERRUPT	Hi-speed GPIO24 interrupt

59	IRQN_GPIOHS25_INTERRUPT	Hi-speed GPIO25 interrupt
60	IRQN_GPIOHS26_INTERRUPT	Hi-speed GPIO26 interrupt
61	IRQN_GPIOHS27_INTERRUPT	Hi-speed GPIO27 interrupt
62	IRQN_GPIOHS28_INTERRUPT	Hi-speed GPIO28 interrupt
63	IRQN_GPIOHS29_INTERRUPT	Hi-speed GPIO29 interrupt
64	IRQN_GPIOHS30_INTERRUPT	Hi-speed GPIO30 interrupt
65	IRQN_GPIOHS31_INTERRUPT	Hi-speed GPIO31 interrupt

7.3.2Interrupt Priorities

Each individual interrupt can have its own priority selected among 7 possible levels (1 is the lowest priority and 7 the highest priority). Only the three least significant bits of each address are used for priority select, the remaining bits should be kept in “0”.

3.9.2.2 Interrupt Threshold

There is an interrupt threshold register which controls current interrupt prior- ity level, ignoring interrupts with priority lower than specified.

3.9.3 寄存器列表

Base Address：INTERRUPT_BASE_ADDR (0x0C000000)

Name	Description	Offset address
PLIC_SOURCE_PRIORITY_1	set source 1 priority	0x04
PLIC_SOURCE_PRIORITY_2	set source 2 priority	0x08
……	……	……
PLIC_SOURCE_PRIORITY_65	set source 3 priority	0x104
PLIC_TARGET_ENABLE0	CPU0 PLIC enables	0x2000
PLIC_TARGET_ENABLE1	CPU1 PLIC enables	0x2080
PLIC_PRIORITY_THRESHOLD0	set CPU0 PLIC priority threshold	0x200000
PLIC_CLAIM_COMPLETE0	CPU0 claim/complete	0x200004
PLIC_PRIORITY_THRESHOLD1	set CPU1 PLIC priority threshold	0x201000
PLIC_CLAIM_COMPLETE1	CPU1 claim/complete	0x201004

3.9.4 寄存器设置

3.9.4.1 PLIC_SOURCE_PRIORITY(1-65)(0x04-0x104)

Reset value：0x00

Bits	Name	Description	Memory Access
31:0	PRIORITY	对应中断优先级值	读/写

3.9.4.2 PLIC_TARGET_ENABLE0（0-31）(0x2000-0x207C)

Reset value：0x00

Bits	Name	Description	Memory Access
31:0	ENABLE	每位对应中断使能	读/写

注：寄存器 0 中的第 0 位为 0，其余 1023 位对应 1023 个中断，本芯片使用 1 至 65，66 之后的保留。

3.9.4.3 PLIC_TARGET_ENABLE1（0-31）(0x2080-0x20FC)

Reset value：0x00

Bits	Name	Description	Memory Access
31:0	ENABLE	每位对应中断使能	读/写

3.9.4.4 PLIC_PRIORITY_THRESHOLD0(0x200000)

Reset value：0x00

Bits	Name	Description	Memory Access
31:0	THRESHOLD	CPU中断优先级阀值，中断优先级大于该值才会产生中断	读/写

3.9.4.5 PLIC_CLAIM_COMPLETE0(0x200004)

Reset value：0x00

Bits	Name	Description	Memory Access
31:0	CLAIM	产生中断的中断号，通过读取该寄存器判断中断号，读取该寄存器时会同步的清除中断标志	读/写

3.9.4.6 PLIC_PRIORITY_THRESHOLD1(0x200100)

Reset value：0x00

Bits	Name	Description	Memory Access
31:0	THRESHOLD	CPU中断优先级阀值，中断优先级大于该值才会产生中断	读/写

3.9.4.7 PLIC_CLAIM_COMPLETE1(0x200104)

Reset value：0x00

Bits	Name	Description	Memory Access
31:0	CLAIM	产生中断的中断号，通过读取该寄存器判断中断号，读取该寄存器时会同步的清除中断标志	读/写

3.10 实时时钟 (RTC)

3.10.1 概述

RTC是用来计时的单元，在设置时间后具备计时功能。

3.10.2 功能描述

RTC主要功能包括：

- 可使用外部高频晶振进行计时；
- 可配置外部晶振频率与分频；
- 支持万年历配置，可配置的项目包含世纪、年、月、日、时、分、秒与星期；
- 可按秒进行计时，并查询当前时刻；
- 支持设置一组闹钟，可配置的项目包含年、月、日、时、分、秒，闹钟到达时触发中断；
- 中断可配置，支持每日、每时、每分、每秒触发中断；
- 可读出小于1秒的计数器计数值，最小刻度单位为外部晶振的单个周期；
- 上电/复位后数据清零。

RTC有一个可编程的二进制计数器，用户为其指定宽度（rtc_CNT_width）。计数器在输入计数器时钟rtc_clk的连续正边缘递增。当计数器加载寄存器（RTC_CLR）编程时，计数器加载一个允许计数器递增的起始值。当计数器达到其最大值（所有位都很高）时，它将变到0，然后继续递增。根据用户配置的计数器时钟和APB总线时钟之间的关系，加载到计数器中的值可能需要跨时钟域传输。一旦值被传输，它就被加载到计数器中。新的值限定符信号与加载值一起传输，以生成计数器的启用加载。

计数器的事件序列为：

1. 用户通过写入RTC_CLR来编程新的加载值。
2. RTC_CLR被传输到计数器时钟域。
3. 传输的值加载到计数器中。
4. 计数器从计数器时钟正边缘的加载值开始递增。

3.10.3 寄存器列表

Base Address：RTC_BASE_ADDR (0x50460000)

Name	Description	Offset address
RTC_CCVR	Current Counter Value Register	0x0
RTC_CMR	Counter Match Register	0x4
RTC_CLR	Counter Load Register	0x8
RTC_CCR	Counter Control Register. Note: If the RTC_RSTAT register indicates	0xc
RTC_STAT	Interrupt Status Register	0x10
RTC_RSTAT	Interrupt Raw Status Register	0x14
RTC_EOI onpage	End of Interrupt Register	0x18
RTC_COMP_VERSION	Component Version Register	0x1c
RTC_CPSR	Counter PreScaler Register	0x20
RTC_CPCVR	Current Prescaler Counter Value Register	0x24

3.10.4 寄存器设置

3.10.4.1 RTC_CCVR(0x0)

Bits	Name	Description	Memory Access
31:y	RSVD_CCVR	RTC_CCVR 31toRTC_CNT_WIDTH Reserved bits - Read Only Exists:Always Volatile:true Range Variable[y]:RTC_CNT_WIDTH	R
x:0	Current_Counter_Value	When read, this register is the current value of the internal counter. This value is always read coherently. Bits from RTC_CNT_WIDTH to 31 are read as 0 when RTC_CNT_WIDTH is less than 31. Reset	R

		Value:0x0Exists:AlwaysVolatile:true Range Variable[x]:RTC_CNT_WIDTH - 1	

3.10.4.2 RTC_CMR(0x4)

Bits	Name	Description	Memory Access
31:y	RSVD_CMR	RTC_CMR 31toRTC_CNT_WIDTH Reserved bits - Read Only Exists:Always Range Variable[y]:RTC_CNT_WIDTH	R
x:0	Counter_Match	Interrupt Match Register. When the internal counter matches this register, an interrupt is generated, provided interrupt generation is enabled. When appropriate, this value is written coherently. Only when all the bytes are written is the register used by the interrupt detection logic. Bits from RTC_CNT_WIDTH and above are read and written as 0 when RTC_CNT_WIDTH is less than 31. Reset Value:0x0 Exists:Always Range Variable[x]:RTC_CNT_WIDTH - 1	R/W

3.10.4.3 RTC_CLR(0x8)

Bits	Name	Description	Memory Access
31:y	RSVD_CLR	RTC_CLR 31toRTC_CNT_WIDTH Reserved bits - Read Only	R
x:0	Counter_Load	Loaded into the counter as the loaded value,which is written coherently. Bits from RTC_CNT_WIDT Hand above are read and written as 0 when RTC_CNT_WIDTH is less than31. Reset Value:0x0	R/W

3.10.4.4 RTC_CCR(0xc)

Bits	Name	Description	Memory Access
31:8	RSVD_CCR	RTC_CCR 31to8 Reserved and read as 0.	R
7:5	rtc_prot_level	This field holds the protection level value of apb_rtc.	R/W
4	rtc_psclr_en	Optional. Allows user to control the usage of RTC Prescaler feature. Reset Value:0x0 Values: 0x0 (DISABLED): Disables the Prescalercounter 0x1 (ENABLED): Enables the Prescalercounter	R/W
3	rtc_wen	Optional. Allows the user to force the counter to wrap when a match occurs instead of waiting until the maximum count is reached. 0 = Wrap disabled, 1 = Wrap enabled, This bit is writable only when RTC_WRAP_MODE =1 Reset Value:0x0 Values: 0x0 (DISABLED): Disables theWRAP 0x1 (ENABLED): Enables theWRAP	* Varies
2	rtc_en	Optional. Allows the user to control counting in the counter. This bit does not exist if RTC_EN_MODE = 0. Internally, the counter is always enabled. Reset Value:0x0 Values: 0x0 (DISABLED): Disables thecounter 0x1 (ENABLED): Enables thecounter	R/W
1	rtc_mask	Allows the user to mask interrupt generation. Reset Value:0x0	R/W
0	rtc_ien	Allows the user to disable interrupt generation. Reset Value:0x0 Values: 0x0 (DISABLED): Disables the interruptgeneration 0x1 (ENABLED): Enables the interruptgeneration	R/W

3.10.4.5 RTC_STAT(0x10)

Bits	Name	Description	Memory Access

31:1	RSVD_RTC_STAT	RTC_STAT 31to1 Reserved and read as 0.	R
0	rtc_stat	This register is the masked raw status. Reset Value:0x0 Values: 0x0 (INACTIVE): Interrupt isinactive 0x1 (ACTIVE): Interrupt is active (regardless ofpolarity)	R

3.10.4.6 RTC_RSTAT(0x14)

Bits	Name	Description	Memory Access
31:1	RSVD_RTC_RSTAT	RTC_RSTAT 31to1 Reserved and read as 0.	R
0	rtc_rstat	Raw Status ResetValue:0x0Values: 0x0 (INACTIVE): Interrupt isinactive 0x1 (ACTIVE): Interrupt is active (regardless ofpolarity)	R

3.10.4.7 RTC_EOI(0x18)

Bits	Name	Description	Memory Access
31:1	RSVD_RTC_EOI	RTC_EOI 31to1 Reserved and read as 0.	R
0	rtc_eoi	By reading this location, the match interrupt is cleared. Performing read-to-clear on interrupt, the interrupt is cleared at the end of the read. Reset Value:0x0	R

3.10.4.8 RTC_COMP_VERSION(0x1c)

Bits	Name	Description	Memory Access
31:0	rtc_comp_version	ASCII value for each number in the version, followed by. For example, 32_30_31_2A represents the version 2.01. ResetValue:See the Releases table in the apb_rtc Release Notes.	R

3.10.4.9 RTC_CPSR(0x20)

Bits	Name	Description	Memory Access
31:y	RSVD_CPSR	RTC_CPSR 31toRTC_PRESCLR_WIDTH Reserved bits - Read Only	R
x:0	Counter_Prescaler_ Value	Counter Prescaler Register. The RTC counter will be updating at the rate of rtc_clk, rtc_clk_en, or pclk based on the configuration. This register is used to prescale the rate at which the RTC counter updates.When appropriate, this register is written coherently. Only when all the bytes are written, the register used by the prescaler counter logic.Reset Value:RTC_PRESCLR_VAL	R/W

3.10.4.10 RTC_CPCVR(0x24)

Bits	Name	Description	Memory Access
31:y	RSVD_CPCVR	RTC_CPCVR 31toRTC_PRESCLR_WIDTH Reserved bits - Read Only	R
x:0	Current_Prescaler_ Counter_Valu e	When read, this register provides the current value of the internal prescaler counter. This value always is read coherently. Bits from RTC_PRESCLR_WIDTH to 31 are read as 0 when RTC_PRESCLR_WIDTH is less than 31. Reset Value:0x0	R

3.11 安全散列算法加速器 (SHA256 Accelerater)

3.11.1 概述

SHA256模块是用来计算SHA-256的计算单元，主要功能包括：支持SHA-256的计算；支持DMA输入数据。

3.11.2 功能描述

- 配置sha_endian选择大小端，配置dma_en选择数据输入的来源，配置sha_data_num输入数据的数量（配置成1就代表输入1个512bit数据，配置成5就代表输入2个512bit数据）；
- 使能sha_en开始计算；
- 通过sha_data_in写入需要计算的数据，输入的数据会必须补齐成512的整数倍；
- 当计算完成时会通过置位sha_en来标志计算完成，可以通过sha_result_dw0~7将数据读走。

操作流程

1. 通过寄存器sha_function_reg_0、sha_function_reg_1、shar_num_reg初始化模块；
2. 通过寄存器sha_data_in输入数据；
3. 等待sha_status的bit0等于1，表示计算完成；
4. 通过寄存器sha_result_dw读取结果。

3.11.3 寄存器列表

Base Address： SHA256_BASE_ADDR (0x502C0000)

Name	Description	Offset address
sha_result_dw0	result	0x00
sha_result_dw1		0x04
sha_result_dw2		0x08
sha_result_dw3		0x0C
sha_result_dw4		0x10

sha_result_dw5		0x14
sha_result_dw6		0x18
sha_result_dw7		0x1c
sha_data_in1	input data	0x20
sha_data_in2		0x24
sha_data_num	Count register of sha256	0x28
sha_status	Status register	0x2C
double_sha	Enable double_sha	0x34
input_ctrl	Control register	0x38

3.11.4 寄存器设置

3.11.4.1 sha_result_dw0(0x00)

Reset value：0x00

Bits	Name	Description	Memory Access
31:0	sha_result_dw0	result[31:0]	R

3.11.4.2 sha_result_dw1(0x04)

Reset value：0x00

Bits	Name	Description	Memory Access
31:0	sha_result_dw1	result[63:32]	R

3.11.4.3 sha_result_dw2(0x08)

Reset value：0x00

Bits	Name	Description	Memory Access
31:0	sha_result_dw2	result[95:64]	R

3.11.4.4 sha_result_dw3(0x0C)

Reset value：0x00

Bits	Name	Description	Memory Access
31:0	sha_result_dw3	result[127:96]	R

3.11.4.5 sha_result_dw4(0x10)

Reset value：0x00

Bits	Name	Description	Memory Access
31:0	sha_result_dw4	result[159:128]	R

3.11.4.6 sha_result_dw5(0x14)

Reset value：0x00

Bits	Name	Description	Memory Access
31:0	sha_result_dw5	result[191:160]	R

3.11.4.7 sha_result_dw6(0x18)

Reset value：0x00

Bits	Name	Description	Memory Access
31:0	sha_result_dw6	result[223:192]	R

3.11.4.8 sha_result_dw7(0x1c)

Reset value：0x00

Bits	Name	Description	Memory Access
31:0	sha_result_dw7	result[255:224]	R

3.11.4.9 sha_data_in1(0x20)

Reset value：0x00

Bits	Name	Description	Memory Access
31:0	sha_data_in1	Input data of sha256 from this register	W

3.11.4.10 sha_data_in2(0x24)

Reset value：0x00

Bits	Name	Description	Memory Access
31:0	sha_data_in2	Input data of sha256 from this register	W

3.11.4.11 sha_data_num(0x28)

Reset value：0x00

Bits	Name	Description	Memory Access
31:16	sha_data_num	How many 512blocks are currently calculated	R/W
15:0	sha_data_cnt	The total amount of data calculated by sha256 is set by this register, and the minimum unit is 512bit	R/W

3.11.4.12 sha_status(0x2C)

Reset value：0x00

Bits	Name	Description	Memory Access
31:17	RSRV		R
16	sha_endian	0：Little Endian 1：Big Endian	R/W
15:9	RSRV		R
8	sha_overflow	Sha256 calculation overflow flag register	R
7:1	RSRV		R
0	sha_en	Sha256 enable register / sha256 calculation end flag bit	R/W

3.12 数字视频接口 (DVP)

3.12.1 概述

DVP是摄像头接口模块，支持把摄像头输入图像数据转发给AI模块或者内存。主要功能包括：

- 支持DVP接口的摄像头；
- 支持SCCB协议配置摄像头寄存器；
- 最大支持640x480每帧的图像输入，每帧图像大小可配置；
- 支持YUV422和RGB565格式的图像输入；
- 支持图像同时输出到AI模块和显示屏:
- 输出到AI模块的格式可选RGB888，或YUV422输入时的Y分量；
- 输出到显示屏的格式为RGB565；
- 图像输出接口为AXI总线，支持4 beats burst和1 beat burst；
- 检测到一帧开始或一帧图像传输完成时可向CPU发送中断；

3.12.2 功能描述

根据需要的输入的图像格式配置DVP的内部寄存器，并用DVP的SCCB接口配置摄像头的工作模式，摄像头PLL频率等设置，然后用DVP的中断信号配合来接收一帧一帧的图像输入，送到AI模块或者内存中。

操作流程：

1. 初始化配置时钟；
2. 使能burst；
3. 选择使能输出位置（显示、AI）；
4. 配置图像格式和大小；

3.12.3 寄存器列表

Base Address：DVP_BASE_ADDR(0x50430000)

Name	Description	Offset address

DVP_CFG	DVP config	0x00
DVP_R_ADDR	Start Address Register of R Component in Data Output to AI	0x04
DVP_G_ADDR	Start Address Register of G Component in Data Output to AI	0x08
DVP_B_ADDR	Start Address Register of B Component in Data Output to AI	0x0C
DVP_CMOS	CMOS Control Register	0x10
SCCB_CFG	SCCB Control Register	0x14
SCCB_W	SCCB Write Data Register	0x18
DVP_AXI	AXI Interface Register	0x1C
DVP_STS	DVP Status Register	0x20
DVP_DISPLAY_ADDR	Display Output Address Register	0x28

3.12.4 寄存器设置

3.12.4.1 DVP_CFG(0x00)

Reset value：0x01E01410

Bits	Name	Description	Memory Access
31:30	RSRV	Reserved	—
29:20	RLINE_NUM	Line number in a frame	R/W
19:12	RHREF_BURST_SIZE	Burst number in a line. For 640x480 frame and in burst 4 beats mode, RHREF_BURST_SIZE = 640/4/8 = 0x14	R/W
11	RSRV	Reserved	—
10	RCMOS_Y_OUT	RCMOS_YUV_RGB equals 1, RCMOS_Y_OUT equals 1, which means the DVP will output an image	R/W

		of the Y component, RCMOS_Y_OUT equal to 0 means that DVP converts YUV format to RGB format output.	
9	RCMOS_YUV_RGB	1 means the input image is in YUV format and 0 means the input image is in RGB format.	R/W
8	RBURST_SIZE	1 for 4 beats burst. 0 for 1 beats burst.	R/W
7:5	RSRV	Reserved	—
4	RDVP_AUTO	1 indicates that the transmission of the next frame will start automatically after 1 frame image transmission has been completed and the DMA address of the next frame needs to be configured in the time interval between two frames transmission.	R/W
3	RDVP_DISPLAY_FORMAT_EN	1 indicates that the output image is sent to the display memory address in the display format	R/W
2	RDVP_AI_FORMAT_EN	1 means that the output image will be sent to AI memory address in AI format. DVP supports simultaneous output of images in both display format and AI format to display and AI memory address.	R/W

1	FRAME_START_INT_EN	1 indicates pull-up interruption when frame start signal is detected and CPU should write DVP_FRAME_START to clear interruptions	R/W
0	FRAME_FINISH_INT_EN	1 indicates a pull-up interruption after a frame transfer has been completed and the CPU should write DVP_FRAME_FINISH to clear interruptions	R/W

3.12.4.2 DVP_R_ADDR(0x04)

Reset value：0x00

Bits	Name	Description	Memory Access
31:0	RDMA_ADDR_R	Base DMA address for R scale or Y scale in single Y output mode.	R/W

3.12.4.3 DVP_G_ADDR(0x08)

Reset value：0x00

Bits	Name	Description	Memory Access
31:0	RDMA_ADDR_G	Base DMA address for G scale.	R/W

3.12.4.4 DVP_B_ADDR(0x00C)

Reset value：0x00

Bits	Name	Description	Memory Access
31:0	RDMA_ADDR_B	Base DMA address for B scale.	R/W

3.12.4.5 DVP_CMOS(0x010)

Reset value：0x00100002

Bits	Name	Description	Memory Access
31:25	RSRV	Reserved	—
24	CMOS_PWND	1 for power down.	R/W
23:17	RSRV	Reserved	—
16	CMOS_RST_	0 for reset active	R/W
15:9	RSRV	Reserved	—
8	CMOS_XCLK_EN	1: Enablke output XCLK for OV.	R/W
7:0	CMOS_XCLK_DIV	Divider factor from PCLK to XCLK. 0 for divide 2, 1 for divide 4 …	R/W

3.12.4.6 SCCB_CFG(0x014)

Reset value：0x00383802

Bits	Name	Description	Memory Access
31:24	SCCB_RDATA	Read data for read transaction.	RO
23:16	SCCB_HIGH	SCCB high period time in PCLK.	R/W
15:8	SCCB_LOW	SCCB low period time in PCLK.	R/W
7:4	RSRV	Reserved	—
1:0	SCCB_BYTE_NUM	Write byte number. 1 for write 2 byte, 2 for 3 byte, 3 for 4 byte.	R/W

3.12.4.7 SCCB_W(0x018)

Reset value：0x7856AA2B

Bits	Name	Description	Memory Access
31:24	SCCB_WDATA_B0	write transaction byte 3 for data byte in 2 byte address camera.	R/W
23:16	SCCB_WDATA_B1	write transaction byte 2 for register address byte 1 in 2 byte address camera or data byte in 1 byte address camera.	R/W
15:8	SCCB_WDATA_B2	write transaction byte 1 for register address byte 0.	R/W
7:1	SCCB_WDATA_B3	write transaction byte 0 for ID address.	R/W
0	SCCB_WR	1 for write; 0 for read.	R/W

3.12.4.8 DVP_AXI(0x01C)

Reset value：0x03

Bits	Name	Description	Memory Access
31:15	RSRV	Reserved	—
14:12	GM_MPROT	WPROT of AXI interface is the protection type, Reset value is recommended.	R/W
11:8	GM_MCACHE	WCACHE of AXI interface is cache type, Reset value is recommended.	R/W
7:0	GM_MLEN	AXI Burst length - 1. 0 for burst 1, 3 for burst 4.	R/W

3.12.4.9 DVP_STS(0x020)

Reset value：0x00

Bits	Name	Description	Memory Access
31:26	RSRV	Reserved	—
25	SCCB_STS_WE	SCCB_Write enable bit for STS and SCCB_SCCB_at STSSTS_WE should also write 1.	W

24	SCCB_STS	Write 1 opens the transmission of SCCB, keeps 1 during the transmission process, and clears automatically after the transmission is completed	R/W
23:18	RSRV	Reserved	—
17	DVP_EN_WE	DVP_EN Write Enable Bit, DVP_DVP_EN_WE should also write 1	W
16	DVP_EN	Enable for DVP capture CMOS data. Write1 to assert this bit. HW clear this bit after one frame transfer done.	R/W
15:10	RSRV	Reserved	—
9	DVP_FRAM_FINISH_WE	Write enable for DVP_FRAM_FINISH	W
8	DVP_FRAM_FINISH	DVP_after one frame transfer is completeFRAME_FINISH sets 1 and the CPU writes 1 on this bit to zero it	R/W
7:2	RSRV	Reserved	—
1	DVP_FRAM_START_WE	DVP_FRAME_Write enable bit for START, write DVP_FRAME_DVP_at STARTFRAME_START_WE should also write 1	W
0	DVP_FRAM_START	DVP_after detecting the start of a frameFRAME_START is set to 1, and the CPU writes 1 to zero this bit.	R/W

3.12.4.10 DVP_DISPLAY_ADDR(0x028)

Reset value：0x80100000

Bits	Name	Description	Memory Access
31：0	RDMA_ADDR_DISPLAY	Display base address of memory	R/W

3.13 快速傅里叶变换加速器 (FFT Accelerater)

3.13.1 概述

FFT模块是用硬件的方式来实现FFT的基2时分运算加速。

- 支持多种运算长度，即支持64点、128点、256点以及512点运算；
- 支持两种运算模式，即FFT以及IFFT运算；
- 支持可配的输入数据位宽，即支持32bit及64bit输入；
- 支持可配的输入数据排列方式，即支持虚部、实部交替，纯实部以及实部、虚部分离三种数据排列方式；
- 支持可配的数据搬运方式，即CPU搬数和DMA搬数；

3.13.2 功能描述

目前该模块可以支持64点、128点、256点以及512点的FFT以及IFFT，具体工作模式可以通过总线进行配置。在FFT内部有两块大小为512*32bit的SRAM，在配置完成后FFT会向DMA发送TX请求，将DMA送来的送据放到其中的一块SRAM中去，直到满足当前FFT运算所需要的数据量并开始FFT运算，蝶形运算单元从包含待运算数据的SRAM中读出数据，运算结束后将数据写到另外一块SRAM中去，下次蝶形运算再从刚写入的SRAM中读出数据，运算结束后并写入另外一块SRAM，如此反复交替进行直到完成整个FFT运算，在运算结束后向DMA发送RX请求，将包含最终运算结果的数据从SRAM中送至DMA，完成一次完整的FFT运算，如果当前配置不需要改变，FFT可以直接向DMA发送TX请求，开始下一次的FFT运算。

配置的一般流程：

1. 初始化模块，配置FFT点数、选择模式（FFT/IFFT）、数据输入格式、使能模块；
2. 写入需要计算的数据；
3. 等待计算完成；
4. 读出结果；

3.13.3 寄存器列表

Base Address：FFT_BASE_ADDR(0x42000000)

Name	Description	Offset address
FFT_INPUT_FIFO	input data fifo	0x00
FFT_CTRL	fft ctrl reg	0x08
FFT_FIFO_CTRL	fifo ctrl	0x10
FFT_INTR_MASK	interrupt mask	0x18
FFT_INTR_CLEAR	interrupt clear	0x20
FFT_STATUS	fft status reg	0x28
FFT_STATUS_RAW	fft_status_raw	0x30
FFT_OUTPUT_FIFO	fft_output_fifo	0x38

3.13.4 寄存器设置

3.13.4.1 FFT_INPUT_FIFO(0x00)

Reset value: 0x00

Bits	Name	Description	Memory Access
63:0	FFT_INPUT_FIFO	input data fifo	R/W

3.13.4.2 FFT_CTRL(0x08)

Reset value: 0x00

Bits	Name	Description	Memory Access
63:18	RSRV		R/W
17	FFT_DATA_MODE	0:64bit input data 1:32bit input data	R/W
16:15	FFT_INPUT_MODE	0: RIRI, 2: only real part exist, RRRR, 3: RRRR...IIII	R/W
14	DMA_SEND	DMA Enable 0x1:Enable	R/W
13	FFT_ENABLE	FFT Enable 0x1:Enable	R/W
12:4	FFT_SHIFT	9-Layer Butterfly Operation Shift Operation 0x0:Non-displacement 0x1:displacement	R/W

3	FFT_MODE	1: fft; 0: ifft	R/W
2:0	FFT_POINT	0:512 point; 1:256 point; 2:128 point; 3: 64 point	R/W

3.13.4.3 FFT_FIFO_CTRL(0x10)

Reset value: 0x07

Bits	Name	Description	Memory Access
63:3	RSRV		R/W
2	FFT_GS_FIFO_FLUSH_N	default value is 1, set to 0 if you want to reset this fifo	R/W
1	FFT_CMD_FIFO_FLUSH_N	default value is 1, set to 0 if you want to reset this fifo	R/W
0	FFT_RESP_FIFO_FLUSH_N	default value is 1, set to 0 if you want to reset this fifo	R/W

3.13.4.4 FFT_INTR_MASK(0x18)

Reset value: 0x00

Bits	Name	Description	Memory Access
63:1	RSRV	Resvered	R/W
0	FFT_DONE_MASK	Status Return Settings 0x0:FFT Completed Return Status 0x1:FFT Completed Not Return Status	R/W

3.13.4.5 FFT_INTR_CLEAR(0x20)

Reset value: 0x00

Bits	Name	Description	Memory Access

63:1	RSRV	Reserved	R/W
0	FFT_DONE_CLEAR	Interrupt status clears 0x1: clears the current interrupt request from the FFT	R/W

3.13.4.6 FFT_STATUS(0x28)

Reset value: 0x00

Bits	Name	Description	Memory Access
63:1	RSRV	Reserved	R
0	FFT_DONE_STATUS	FFT operation status 0x0:FFT operation has not been completed 0x1:FFT operation has been completed	R

3.13.4.7 FFT_STATUS_RAW(0x30)

Reset value: 0x00

Bits	Name	Description	Memory Access
63:2	RSRV	Reserved	R
1	FFT_WORK	FFT is calculating 0x1: is calculating	R
0	FFT_DONE	FFT operation ends 0x0: not ends 0x1: ends	R

3.13.4.8 FFT_OUTPUT_FIFO(0x38)

Reset value: 0x00

Bits	Name	Description	Memory Access
63:0	FFT_OUTPUT_FIFO	Output of operational data via this register	R

3.14 通用异步收发传输器 (UART)

3.14.1 概述

嵌入式应用通常要求一个简单的并且占用系统资源少的方法来传输数据。通用异步收发传输器(UART)即可以满足这些要求，它能够灵活地与外部设备进行全双工数据交换。芯片中有3个UART控制器可供使用，并且兼容不同的 UART 设备。另外，UART还可以用作红外数据交换(IrDA)或RS-485调制解调器。

主要功能如下：

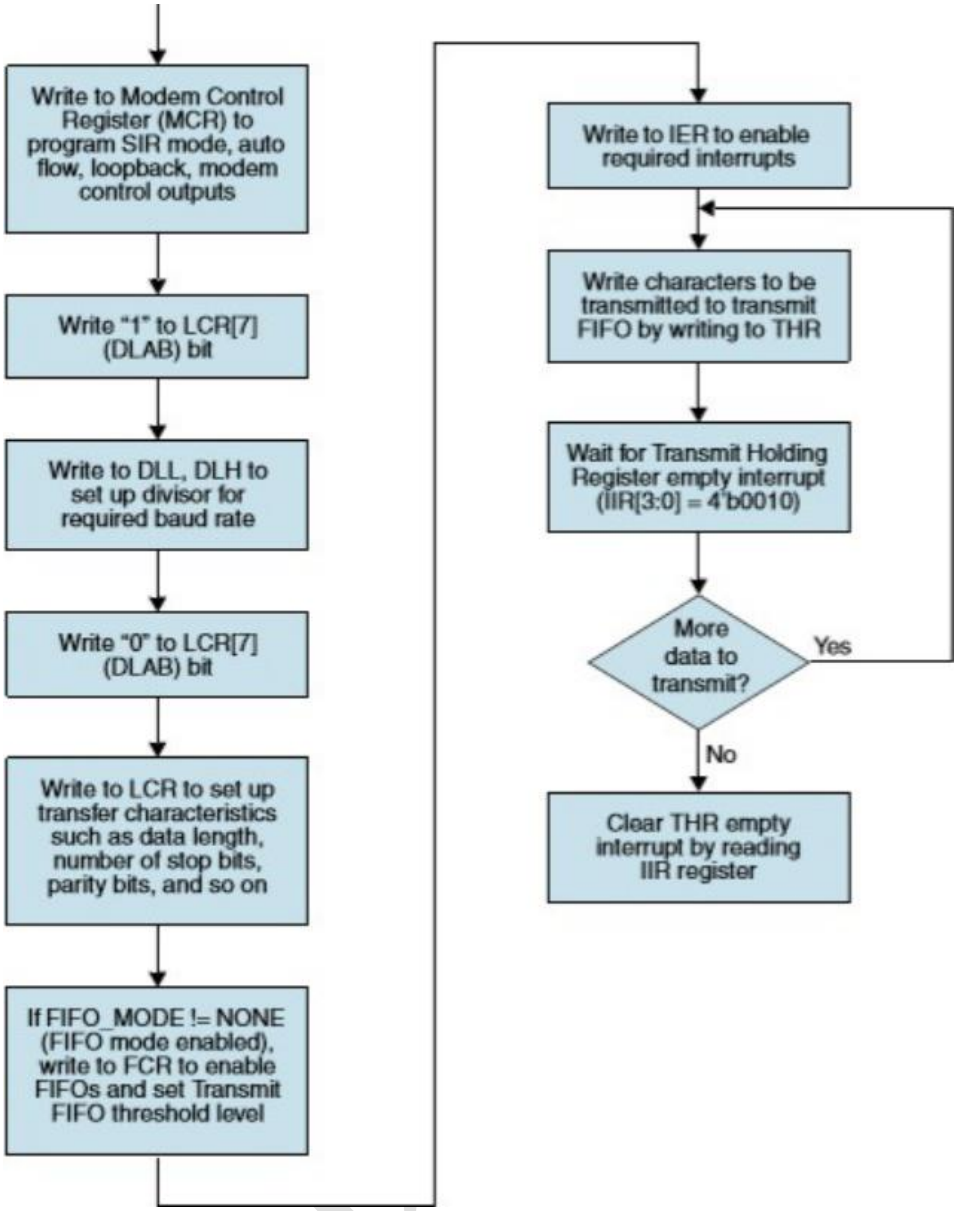
- 可编程收发波特率；
- 3个UART的发送FIFO以及接收FIFO共享1024 x 8-bit RAM全双工异步通信；
- 支持输入信号波特率自检功能；
- 支持 5/6/7/8 位数据长度；
- 支持 1/1.5/2/3/4 个停止位；
- 支持奇偶校验位；
- 支持 RS485 协议；
- 支持 IrDA 协议；
- 支持 DMA 高速数据通信；
- 支持 UART 唤醒模式；
- 支持软件流控和硬件流控；

3.14.2 功能描述

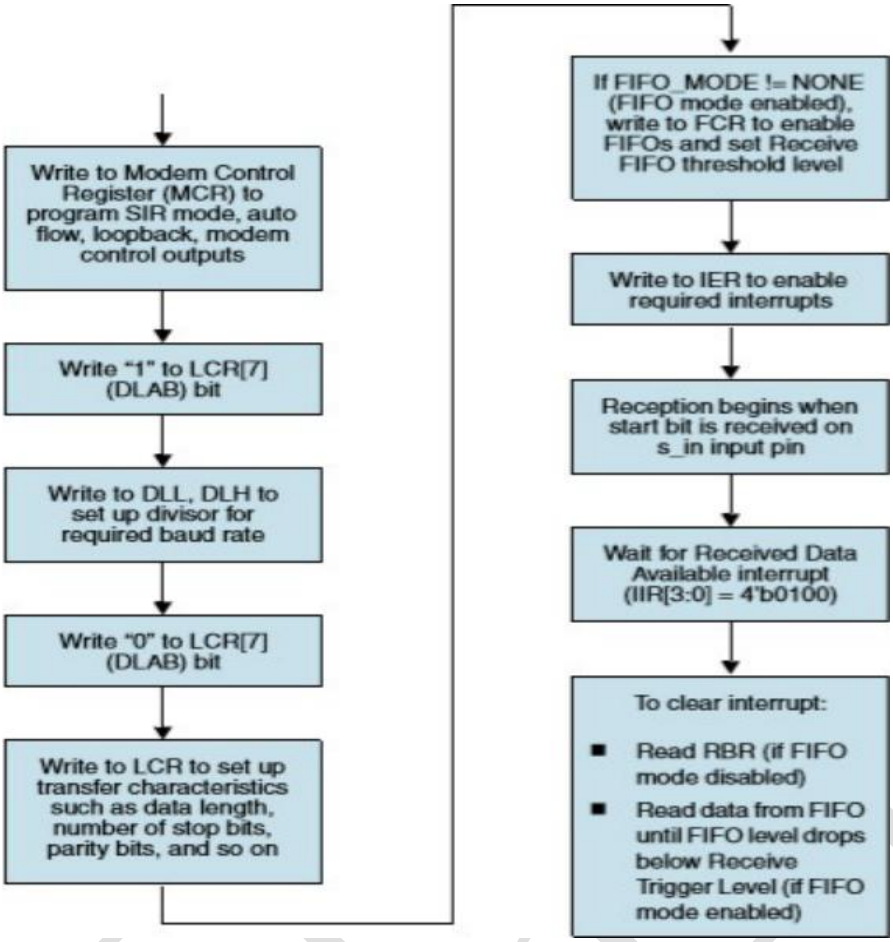
UART是一种以字符为单位的通用数据链，可以实现设备间的通信。异步传输的意思是不需要在发送数据上添加时钟信息。这也要求发送端和接收端的速率、停止位、奇偶校验位等都要相同，通信才能成功。一个典型的 UART 帧开始于一个起始位，紧接着是有效数据，然后是奇偶校验位（可有可无），最后是停止位。UART 控制器支持多种字符长度和停止位。另外，控制器还支持软硬件流控和DMA，可以实现数据的高速无缝传输。开发者可以使用多个UART端口，同时又能保证很少的软件开销。

软件流程参考

发送流程：



接收流程：



3.14.3 寄存器列表

Base Address：UART1_BASE_ADDR (0x50210000)

Base Address：UART2_BASE_ADDR (0x50220000)

Base Address：UART3_BASE_ADDR (0x50230000)

Name	Description	Offset address
RBR	Receive buffer register	0x0
DLL	Divisor lock (low)	0x0
THR	Transmit holding register	0x0
DLH	Divisor lock (high)	0x4
IER	Interrupt enable register	0x4
FCR	FIFO control register	0x8
IIR	Interrupt identification register	0x8

LCR	Wire control register	0xc
MCR	Modem control register	0x10
LSR	Line status register	0x14
MSR	Modem status register	0x18
SCR	Scratchpad register	0x1c
LPDLL	Low power divisor latch low register.	0x20
LPDLH	Low power divisor latch high register.	0x24
FAR	FIFO access register	0x70
TFR	Send FIFO read	0x74
RFW	Receive FIFO write	0x78
USR	UART status register	0x7c
TFL	Transmit FIFO depth	0x80
RFL	Receive FIFO depth	0x84
SRR	Software reset register	0x88
SRTS	Shadow request sending	0x8c
SBCR	Shadow control register	0x90
SDMAM	Shadow DMA mode register	0x94
SFE	Shadow FIFO enable register	0x98
SRT	Shadow receive trigger register	0x9c
STET	Shadow TX null trigger register	0xa0
HTX	Stop TX	0xa4
DMASA	DMA software confirmation register	0xa8
TCR	Transceiver control register	0xac

Name	Description	Offset address
DE_EN	Drive output enable register	0xb0
RE_EN	Drive input enable register	0xb4
DET	Driver output enable timing register	0xb8
TAT	TurnAround timing register	0xbc
DLF	Divisor lock fraction register	0xc0
RAR	Receive address register	0xc4
TAR	Send address register	0xc8
LCR_EXT	Line extended control register	0xcc
CPR	Component parameter register	0xf4
UCV	UART Component Version	0xf8
CTR	Component Type Register	0xfc

3.14.4 寄存器设置

3.14.4.1 RBR（0x0）

Reset value: 0x00

Bits	Name	Description	Memory Access
31：8	RSVD_RBR		R
7 : 0	RBR	Receive Buffer Register.	R

3.14.4.2 DLL（0x0）

Reset value: 0x00

Bits	Name	Description	Memory Access
31：8	RSVD_DLL_31to8	Reserved	R
7 : 0	DLL	Divisor Latch (Low). baud rate = (serial clock freq) / (16 * divisor).	R

3.14.4.3 THR（0x0）

Reset value: 0x00

Bits	Name	Description	Memory Access
31：8	RSVD_THR_31to8	Reserved	R
7 : 0	THR	Transmit Holding Register.	R

3.14.4.4 DLH（0x0）

Reset value: 0x00

Bits	Name	Description	Memory Access
31:8	RSVD_DLH	Reserved	R
7 : 0	DLH	Upper 8-bits of a 16-bit, read/write, Divisor Latch register that contains the baud rate divisor for the UART.	R

3.14.4.5 IER（0x4）

Reset value: 0x00

Bits	Name	Description	Memory Access
31:4	RSVD_IER_31to4	Reserved	R
3	EDSSI	This is used to enable / disable the generation of Modem Status Interrupt. This is the fourth highest priority interrupt. Values： 0: Disable Modem Status Interrupt 1: Enable Modem Status Interrupt	R/W
2	ELSI	This is used to enable/disable the generation of Receiver Line Status Interrupt. This is the highest priority interrupt. Values： 0:Disable Receiver Line Status Interrupt 1:Enable Receiver Line Status Interrupt	R/W

Bits	Name	Description	Memory Access
1	ETBEI	This is used to enable/disable the generation of Transmitter Holding Register Empty Interrupt. This is the third highest priority interrupt. Values： 0:Disable Transmit empty interrupt 1:Enable Transmit empty interrupt	R/W
0	ERBFI	This is used to enable/disable the generation of Received Data Available Interrupt and the Character Timeout Interrupt (if in FIFO mode and FIFO s enabled). These are the second highest priority interrupts. Values： 0:Disable Receive data Interrupt 1:Enable Receive data Interrupt	R/W

3.14.4.6 LCR（0xc)

Reset value: 0x00

Bits	Name	Description	Memory Access
31:8	RSVD_LCR_31to8	Reserved	R
7	DLAB	If UART_16550_COMPATIBLE == NO then, writeable only when UART is not busy (USR[0] is zero), otherwise always writable and always readable. This bit is used to enable reading and writing of the Divisor Latch register (DLL and DLH/LPDLL and LPDLH) to set the baud rate of the UART. This bit must be cleared after initial baud rate setup in order to access other registers. Values： 0:Divisor Latch register is writable only when UART Not BUSY 1:Divisor Latch register is always readable and writable	R/W

Bits	Name	Description	Memory Access

6	BC	This is used to cause a break condition to be transmitted to the receiving device. If set to one the serial output is forced to the spacing (logic 0) state. When not in Loopback Mode, as determined by MCR[4], the sout line is forced low until the Break bit is cleared. If SIR_MODE == Enabled and active (MCR[6] set to one) the sir_out_n line is continuously pulsed. When in Loopback Mode, the break condition is internally looped back to the receiver and the sir_out_n line is forced low. Values： 0:Serial output is released for data transmission 1:Serial output is forced to spacing state	R/W
5	SP	If UART_16550_COMPATIBLE = NO, then writeable only when UART is not busy (USR[0] is 0); otherwise always writable and always readable. This bit is used to force parity value. When PEN, EPS and Stick Parity are set to 1, the parity bit is transmitted and checked as logic 0. If PEN and Stick Parity are set to 1 and EPS is a logic 0, then parity bit is transmitted and checked as a logic 1. If this bit is set to 0, Stick Parity is disabled. Values： 0:Stick parity disabled 1:Stick parity enabled	R/W
4	EPS	Values： 0:an odd parity is transmitted or checked 1:an even parity is transmitted or checked	R/W
3	PEN	Values： 0:disable parity 1:enable parity	R/W

2	STOP	Values： 0:1 stop bit 1:1.5 stop bits when DLS (LCR[1:0]) is zero, else 2 stop bit	R/W

Bits	Name	Description	Memory Access
1：0	DLS	When DLS_E in LCR_EXT is set to 0, this register is used to select the number of data bits per character that the peripheral will transmit and receive. Values： 0:5 data bits per character 1:6 data bits per character 2:7 data bits per character 3:8 data bits per character	R/W

3.14.4.7 MCR（0x10）

Reset value: 0x00

Bits	Name	Description	Memory Access
31:2	RSVD_MCR_31to2	Reserved	R
1	RTS	Request to Send. This is used to directly control the Request to Send (rts_n) output. The Request To Send (rts_n) output is used to inform the modem or data set that the UART is ready to exchange data. Values： 0:Request to Send rts_n de-asserted (logic 1) 1:Request to Send rts_n asserted (logic 0)	R/W

Bits	Name	Description	Memory Access
0	DTR	Data Terminal Ready. This is used to directly control the Data Terminal Ready(dtr_n) output. The value written to this location is inverted and driven out on dtr_n. Values： 0:dtr_n de-asserted (logic1) 1:dtr_n asserted (logic 0)	R/W

3.14.4.8 LSR（0x14）

Reset value: 0x00

Bits	Name	Description	Memory Access
31：8	RSVD_LSR_31to8	Reserved	R
7	RFE	Receiver FIFO Error bit. Values： 0:No error in RX FIFO 1:Error in RX FIFO	R
6	TEMT	Transmitter Empty bit. Values： 0:Transmitter not empty 1:Transmitter empty	R
2	PE	Parity Error bit. Values： 0:no parity error 1:parity error	R
1	OE	Overrun error bit. Values： 0:no overrun error 1:overrun error	R
0	DR	Data Ready bit. Values： 0:data not ready 1:data ready	R

3.14.4.9 TCR（0xac）

Reset value: 0x00

Bits	Name	Description	Memory Access
31:5	RSVD_TCR_31to5	Reserved	R

4:3	RFE	Transfer Mode. 0: In this mode, transmit and receive can happen simultaneously. The user can enable DE_EN, RE_EN at any point of time. Turn around timing as programmed in the TAT register is not applicable in this mode. 1: In this mode, DE and RE are mutually exclusive. Either DE or RE only one of them is expected to be enabled through programming. Hardware will consider the Turn Around timings which are programmed in the TAT register while switching from RE to DE or DE to RE. For transmission Hardware will wait if it is in middle of receiving any transfer, before it starts transmitting. 2: In this mode, DE and RE are mutually exclusive. Once DE_EN/RE_EN is programed - by default re will be enabled and apb_uart controller will be ready to receive. If the user programs the TX FIFO with the data then apb_uart, after ensuring no receive is in progress, disable re and enable de signal. Once the TX FIFO becomes empty, re signal gets enabled and de signal will be disabled. In this mode of operation hardware will consider the Turn Around timings which are programmed in the TAT register while switching from RE to DE or DE to RE. In this mode, de and re signals are strictly complementary to each other.	R/W
2	DE_POL	Driver Enable Polarity. 1: DE signal is active high 0: DE signal is active low	R/W
1	RE_POL	Receiver Enable Polarity. 1: RE signal is active high 0: RE signal is active low	R/W

0	RS485_EN	RS485 Transfer Enable. : In this mode, the transfers are still in the RS232 mode.All other fields in this register are reserved and register DE_EN/RE_EN/TAT are also reserved. : In this mode, the transfers will happen in RS485 mode.All other fields of this register are applicable.	R/W

3.14.4.10 DE_EN（0xb0）

Reset value: 0x00

Bits	Name	Description	Memory Access
31:1	RSVD_DE_EN_31to1	Reserved	R
0	DE_Enable	DE Enable control. The DE Enable register bit is used to control assertion and de-assertion of de signal. 0: De-assert de signal 1: Assert de signal	R/W

3.14.4.11 RE_EN（0xb4）

Reset value: 0x00

Bits	Name	Description	Memory Access
31:1	RSVD_RE_EN_31to1	Reserved	R
0	RE_Enable	RE Enable control. The RE Enable register bit is used to control assertion and de-assertion of re signal. 0: De-assert re signal 1: Assert re signal	R/W

3.15 高速通用异步收发传输器 (UARTHS)

3.15.1 概述

该高速通用异步收发传输器集成在内核模块，所以它的特点就是速度快，主要特征如下：

- 通讯速率可达5Mbps
- 8字节收发和接收FIFO
- 可编程中断模式
- 不支持硬件流控或者其它调制解调控制信号，或同步串行数据转换器

3.15.2 UARTHS寄存器列表

UARTHS_BASE_ADDR (0x38000000)

Name	Description	Offset address
txdata	send	0x00
rxdata	receive	0x04
txctrl	send control	0x08
rxctrl	receive control	0x0c
ie	Interrupt Control Register	0x10
ip	Interrupt Status Register	0x14
div	Baud Rate Frequency Register	0x18

3.15.3 UARTHS寄存器设置

3.15.3.1 txdata（0x0）

Bits	Name	Description	Memory Access
31	full	1：Data not finished	R
30:8	zero	Reserved	R
7:0	data	send data	W

3.15.3.2 rxdata（0x4）

Bits	Name	Description	Memory Access
31	empty	1：empty	R
30:8	zero	Reserved	R
7:0	data	receive data	R

3.15.3.3 txctrl（0x8）

Bits	Name	Description	Memory Access
31 ： 19	resv1	Reserved	R
18:16	txcnt	Interruption Trigger Threshold	W
15:2	resv0	Reserved	R
1	nstop	0 for one stop bit and 1 for two stop bits	W
0	txen	Controls if the TX channel is active.	W

3.15.3.4 rxctrl（0xc）

Bits	Name	Description	Memory Access
31:19	resv1	Reserved	R
18:16	rxcnt	Interruption Trigger Threshold	W
15:1	resv0	Reserved	R
0	rxen	Control whether the RX channel is active.	W

3.15.3.5 ie（0x10）

Bits	Name	Description	Memory Access
31:2	resv1	Reserved	R
1	rxwm	Trigger interruption when receiving over rxcnt, 0: close interruption, 1: open interruption	W

0	txwm	Triggers interruption when sending data less than txcnt, 0: Close interruption, 1: Open interruption	W

3.15.3.6 ip (0x14)

Bits	Name	Description	Memory Access
31:2	resv1	Reserved	R
1	rxwm	Trigger interruption when receiving over rxcnt, 0: no interruption; 1: interruption	R
0	txwm	Triggers interruption when sending data less than txcnt, 0: no interruption; 1: interruption	R

3.15.3.7 div (0x14)

Bits	Name	Description	Memory Access
31:16	resv1	Reserved	R
15:0	div	Fractional frequency coefficient of baud rate，div = freq / baud - 1;	R

3.16 通用输入/输出接口 (GPIO)

3.16.1 概述

芯片有8个通用GPIO。每个IO可以分配到FPIOA上48个管脚之一。

通用GPIO共8个，具有如下特点:

- 8个IO使用一个中断源;
- 可配置输入输出信号;
- 可配置触发IO总中断，边沿触发和电平触发;
- 每个IO可以分配到FPIOA上48个管脚之一;
- 可配置上拉电阻，下拉电阻、或者高阻;

3.16.2 功能描述

- **简单GPIO输入输出**

通用通过配置寄存器GPIO_SWPORTA_DDR来设置输入输出方向，通过寄存器GPIO_SWPORTA_DR来设置输出值或者读取输入值。(8:0)位分别代表一个IO。

- **配置的一般流程**

1. 通过FPIOA配置GPIO的映射IO管脚;
2. 配置GPIO的基本属性（输入输出、上下拉等）;
3. 读取或者输入GPIO状态;

3.16.3 寄存器列表

Base Address：GPIO_BASE_ADDR (0x50200000)

Name	Description	Offset address
GPIO_SWPORTA_DR	Port A data register	0x0
GPIO_SWPORTA_DDR	Port A Data Direction Register	0x4
GPIO_INTEN	Name: Interrupt enable register Note: This register is available only if Port A is configured...	0x30
GPIO_INTMASK	Interrupt mask register Note: This register is available only if Port A is configured to generate...	0x34
GPIO_INTTYPE_LEVEL	Interrupt level Note: This register is available only if Port A is configured to generate interrupts...	0x38
GPIO_INT_POLARITY	Interrupt polarity Note: This register is available only if Port A is configured to generate interrupts...	0x3c
GPIO_INTSTATUS	Interrupt status Note: This register is available only if Port A is configured to generate interrupts...	0x40
GPIO_RAW_INTSTATUS	Raw interrupt status Note: This register is available only if Port A is configured to generate...	0x44
GPIO_DEBOUNCE	Debounce enable Note: This register is available only if Port A is configured to generate interrupts...	0x48
GPIO_PORTA_EOI	Port A clear interrupt register Note: This register is available only if Port A is configured to...	0x4c
GPIO_EXT_PORTA	External port A register	0x50
GPIO_LS_SYNC	Synchronization level	0x60
GPIO_ID_CODE	GPIO ID code	0x64
GPIO_INT_BOTHEDGE	Interrupt Both Edge type Note: This register is available only if PORT A is configured to generate...	0x68
GPIO_VER_ID_CODE	GPIO Component Version	0x6c
GPIO_CONFIG_REG2	GPIO Configuration Register 2 This register is a read- only register that is present when the configuration...	0x70
GPIO_CONFIG_REG1	GPIO Configuration Register 1 This register is present when the configuration parameter GPIO_ADD_ENCODED_PARAMS...	0x74

3.16.4 寄存器设置

3.16.4.1 GPIO_SWPORTA_DR(0x0)

Bits	Name	Description	Memory Access
31:0	GPIO_SWPORTA_DR	Values written to this register are output on the I/O signals for Port A if the corresponding data direction bits for Port A are set to Output mode and the corresponding control bit for Port A is set to Software mode. The value read back is equal to the last value written to this register.Reset Value: GPIO_SWPORTA_RESET	R/W

3.16.4.2 GPIO_SWPORTA_DDR(0x4)

Bits	Name	Description	Memory Access
31:0	GPIO_SWPORTA_DDR	Values written to this register independently control the direction of the corresponding data bit in Port A. The default direction can be configured as input or output after system reset through the GPIO_DFLT_DIR_A parameter.Values:0x0 (IN): Input Direction0x1 (OUT): Output DirectionValue After Reset: 0x0	R/W

3.16.4.3 GPIO_INTEN(0x30)

Bits	Name	Description	Memory Access
31:0	GPIO_INTEN	Allows each bit of Port A to be configured for interrupts. By default the generation of interrupts is disabled. Whenever a 1 is written to a bit of this register, it configures the corresponding bit on Port A to become an interrupt; otherwise, Port A operates as a normal GPIO signal. Interrupts are disabled on the corresponding bits of Port A if the corresponding data direction register is set to Output or if Port A mode is set to Hardware.Reset Value: 0x0Values:0x0 (DISABLED): Interrupt is disabled0x1 (ENABLED): Interrupt is enabled	R/W

3.16.4.4 GPIO_INTMASK(0x34)

Bits	Name	Description	Memory Access

31:0	GPIO_INTMASK	Controls whether an interrupt on Port A can create an interrupt for the interrupt controller by not masking it. By default, all interrupts bits are unmasked. Whenever a 1 is written to a bit in this register, it masks the interrupt generation capability for this signal; otherwise interrupts are allowed through. The unmasked status can be read as well as the resultant status after masking.Reset Value: 0x0Values:0x0 (DISABLED): Interrupt bits are unmasked0x1 (ENABLED): Mask interrupt	R/W

3.16.4.5 GPIO_INTTYPE_LEVEL(0x38)

Bits	Name	Description	Memory Access
31:0	GPIO_INTTYPE_LEVEL	Controls the type of interrupt that can occur on Port A. Whenever a 0 is written to a bit of this register, it configures the interrupt type to be level-sensitive; otherwise, it is edge-sensitive.Reset Value: 0x0Values:0x0 (LEVEL_SENSITIVE): Interrupt is level sensitive0x1 (EDGE_SENSITIVE): Interrupt is edge sensitive	R/W

3.16.4.6 GPIO_INT_POLARITY(0x3c)

Bits	Name	Description	Memory Access
31:0	GPIO INT POLARITY	Controls the polarity of edge or level sensitivity that can occur on input of Port A. Whenever a 0 is written to a bit of this register, it configures the interrupt type to falling-edge or active-low sensitive; otherwise, it is rising-edge or active-high sensitive. Reset Value: 0x0Values:0x0 (ACTIVE_LOW): Active Low polarity0x1 (ACTIVE_HIGH): Active High polarity.	R/W

3.16.4.7 GPIO_INTSTATUS(0x40)

Bits	Name	Description	Memory Access
31:0	GPIO_INTSTATUS	Interrupt status of Port A.Reset Value: 0x0Values:0x0 (INACTIVE): Inactive0x1 (ACTIVE): ActiveVolatile: true	R

3.16.4.8 GPIO_RAW_INTSTATUS(0x44)

Bits	Name	Description	Memory Access
31:0	GPIO_RAW_INTSTATUS	Raw interrupt of status of Port A (premasking bits) Reset Value: 0x0 Values:0x0 (INACTIVE): Inactive0x1 (ACTIVE): ActiveVolatile: true	R

3.16.4.9 GPIO_DEBOUNCE(0x48)

Bits	Name	Description	Memory Access
31:0	GPIO_DEBOUNCE	Controls whether an external signal that is the source of an interrupt needs to be debounced to remove any spurious glitches. Writing a 1 to a bit in this register enables the debouncing circuitry. A signal must be valid for two periods of an external clock before it is internally processed.Reset Value: 0x0Values:0x0 (DISABLED): No debounce0x1 (ENABLED): Enable debounce	R/W

3.16.4.10 GPIO_PORTA_EOI(0x4c)

Bits	Name	Description	Memory Access

31:0	GPIO_PORTA_EOI	Controls the clearing of edge type interrupts from Port A. When a 1 is written into a corresponding bit of this register, the interrupt is cleared. All interrupts are cleared when Port A is not configured for interrupts.Reset Value: 0x0Values:0x0 (DISABLED): No interrupt clear0x1 (ENABLED): Clear Interrupt	W

3.16.4.11 GPIO_EXT_PORTA(0x50)

Bits	Name	Description	Memory Access
31:0	GPIO_EXT_PORTA	This register always reflects the signals value on the External Port A.Reset Value: 0x0Volatile: true	R

3.16.4.12 GPIO_LS_SYNC(0x60)

Bits	Name	Description	Memory Access
31:1	RSVD_GPIO_LS_SYNC	RSVD_GPIO_LS_SYNC Reserved bits - read as zero	R
0	GPIO_LS_SYNC	Writing a 1 to this register results in all level-sensitive interrupts being synchronized to pclk_intr.Reset Value: 0x0Values:0x0 (DISABLED): No synchronization to pclk_int (default)r0x1 (ENABLED): Synchronize to pclk_intr	R/W

3.16.4.13 GPIO_ID_CODE(0x64)

Bits	Name	Description	Memory Access
31:0	GPIO_ID_CODE	This is a user-specified code that a system can read. It can be used for chip identification, and so on.Reset Value: GPIO_ID_NUM	R

3.16.4.14 GPIO_INT_BOTHEDGE(0x68)

Bits	Name	Description	Memory Access
31:0	GPIO_INT_BOTHEDGE	Controls the edge type of interrupt that can occur on Port A.-Whenever a particular bit is programmed to 1, it enables the generation of interrupts on both the rising edge and the falling edge of an external input signal corresponding to that bit on port A.- The values programmed in the registers gpio_intype_level and gpio_int_polarity for this particular bit are not considered when the corresponding bit of this register is set to 1. - Whenever a particular bit is programmed to 0, the interrupt type depends on the value of the corresponding bits in the gpio inttype level and gpio_int_polarity or this particular bit are not considered when the corresponding bit of this register is set to 1. - Whenever a particular bit is programmed to 0, the interrupt type depends on the value of the corresponding bits in the gpio_inttype_level and gpio_int_polarity registers.Reset Value: 0x0**Values:**0x0 (DISABLED): single edge sensitive0x1 (ENABLED): both edge sensitive	R/W

3.16.4.15 GPIO_VER_ID_CODE(0x6c)

Bits	Name	Description	Memory Access
31:0	GPIO_VER_ID_CODE	ASCII value for each number in the version, followed by . For example 32_31_32_2A represents the version 2.12.Reset Value: See the releases table in the Release Notes	R

3.16.4.16 GPIO_CONFIG_REG2(0x70)

Bits	Name	Description	Memory Access
31:20	RSVD_GPIO_CONFIG_REG2	RSVD_GPIO_CONFIG_REG2 Reserved bits - read as zero	R
19:15	ENCODED_ID_PWIDTH_D	The value of this register is derived from the GPIO_PWIDTH_D configuration parameter. 0x0 = 8 bits0x1 = 16 bits0x2 = 32 bits0x3 = Reserved	R
14:10	ENCODED_ID_PWIDTH_C	The value of this register is derived from the GPIO_PWIDTH_C configuration parameter.0x0 = 8 bits0x1 = 16 bits0x2 = 32 bits0x3 = Reserved	R
9:5	ENCODED_ID_PWIDTH_B	The value of this register is derived from the GPIO_PWIDTH_B confi guration parameter.0x0 = 8 bits0x1 = 16 bits0x2 = 32 bits0x3 = Reserved	R
4:0	ENCODED_ID_PWIDTH_A	The value of this register is derived from the GPIO_PWIDTH_A confi guration parameter.0x0 = 8 bits0x1 = 16 bits0x2 = 32 bits0x3 = Reserved	R

3.16.4.17 GPIO_CONFIG_REG1(0x74)

Bits	Name	Description	Memory Access
31:22	RSVD_GPIO_CONFIG_REG1	RSVD_GPIO_CONFIG_REG1 Reserved bits - read as zero	R
21	INTERRUPT_BOTH_EDGE_TYPE	The value of this register is derived from the GPIO_INT_BOTH_EDGE configuration parameterValues:0x0 (DISABLED): Interrupt generation on rising or falling edge0x1 (ENABLED): Interrupt generation on both rising and falling edge	R
20:16	ENCODED_ID_WIDTH	The value of this register is derived from the GPIO_ID_WIDTH configuration parameter.	R
15	GPIO_ID	The value of this register is derived from the GPIO_ID configuration parameter.Values:0x0 (DISABLED): GPIO_ID not included0x1 (ENABLED): GPIO_ID is included	R
14	ADD_ENCODED_PARAMS	The value of this register is derived from the GPIO_ADD_ENCODED_PARAMS configuration parameter.Values:0x0 (DISABLED): Encoded parameters not added0x1 (ENABLED): Encoded parameters added	R
13	DEBOUNCE	The value of this register is derived from the GPIO_DEBOUNCE configuration parameter.Values:0x0 (DISABLED): Exclude debounce capability0x1 (ENABLED): Include debounce capability	R

12	PORTA_INTR	The value of this register is derived from the GPIO_PORTA_INTR configuration parameter.Values:0x0 (DISABLED): PORT A is not used as an interrupt source0x1 (ENABLED): PORT A is required to be used as an interruptsource	R
11	HW_PORTD	The value of this register is derived from the GPIO_HW_PORTD configuration parameter.Values:0x0 (DISABLED): Port D has external, auxiliary hardware signalsexcluded0x1 (ENABLED): Port D has external, auxiliary hardware signalsincluded	R
10	HW_PORTC	The value of this register is derived from the GPIO_HW_PORTC confi guration parameter.Values:0x0 (DISABLED): Port C has external, auxiliary hardware signalsexcluded0x1 (ENABLED): Port C has external, auxiliary hardware signalsincluded	R
9	HW_PORTB	The value of this register is derived from the GPIO_HW_PORTB confi guration parameter.Values:0x0 (DISABLED): Port B has external, auxiliary hardware signalsexcluded0x1 (ENABLED): Port B has external, auxiliary hardware signalsincluded	R
8	HW_PORTA	The value of this register is derived from the GPIO_HW_PORTA confi guration parameter.Values:0x0 (DISABLED): Port A has external, auxiliary hardware signalsexcluded0x1 (ENABLED): Port A has external, auxiliary hardware signalsincluded	R

7	PORTD_SINGLE_CTL	The value of this register is derived from the GPIO_PORTD_SINGLE_CTL configuration parameter.Values:0x0 (DISABLED): PORTD is not controlled from a single source0x1 (ENABLED): PORTD is controlled from a single source	R
6	PORTC_SINGLE_CTL	The value of this register is derived from the GPIO_PORTC_SINGLE_CTL configuration parameter.Values:0x0 (DISABLED): PORTC is not controlled from a single source0x1 (ENABLED): PORTC is controlled from a single source	R
5	PORTB_SINGLE_CTL	The value of this register is derived from the GPIO_PORTB_SINGLE_CTL configuration parameter.Values:0x0 (DISABLED): PORTB is not controlled from a single source0x1 (ENABLED): PORTB is controlled from a single source	R
4	PORTA_SINGLE_CTL	The value of this register is derived from the GPIO_PORTA_SINGLE_CTL configuration parameter.Values:0x0 (DISABLED): PORTA is not controlled from a single source0x1 (ENABLED): PORTA is controlled from a single source	R
3:2	NUM_PORTS	The value of this register is derived from the GPIO_NUM_PORT confi guration parameter.Values:0x0 (NUM_PORTS_1): Number of ports is 10x1 (NUM_PORTS_2): Number of ports is 20x2 (NUM_PORTS_3): Number of ports is 30x3 (NUM_PORTS_4): Number of ports is 4	R

1:0	APB_DATA_WIDTH	The value of this register is derived from the GPIO_APB_DATA_WIDTH configuration parameter.Note: 0x3 = ReservedValues:0x0 (APB_8BITS): APB DATA WIDTH is 8 bits0x1 (APB_16BITS): APB DATA WIDTH is 16 bits0x2 (APB_32BITS): APB DATA WIDTH is 32 bits	R

3.17 直接内存存取控制器 (DMAC)

3.17.1 概述

直接存储访问（Direct Memory Access，DMA）用于在外设与存储器之间以及存储器与存储器之间提供高速数据传输。可以在无需任何CPU操作的情况下通过DMA快速移动数据，从而提高了CPU的效率。

系统中具有DMA功能的外设分别是：SPI1、SPI2、SPI3、I2C0、I2C1、I2C2、UART1、UART2、UART3、AES、SHA、AI、FFT、I2S0、I2S1、I2S2、I2S0_BF。

DMA 控制器具有以下几个特点：

- AHB 总线架构；
- 支持半双工和全双工收发数据；
- 数据传输以字节为单位，传输数据量可软件编程；
- 支持4-beat burst传输328KB DMA地址空间；
- 通过DMA实现高速数据传输；

3.17.2 功能描述

所有需要进行高速数据传输的模块都具有DMA功能。DMA控制器与CPU的数据总线使用相同的地址空间访问内部 RAM。

根据各自模块的需求，各个模块的DMA控制器功能有所差别，但是DMA引擎(DMA_ENGINE)的结构相同。

编程参考：

- **多块传输编程流程**

1. 软件读取DMAC通道启用寄存器（DMAC_ChEnReg）以选择可用的（未使用的）通道；
2. 软件对CHx_CFG寄存器写入适当的值配置DMA的传输，SRC_MLTBLK_TYPE和DST_MLTBLK_TYPE位必须设置为二进制10；
3. 为第一个块使用适当的值配置CHx_SAR、CHx_DAR、CHx_BLOCK_TS和CHx_CTL

寄存器。dmac使用这些值加载相应的影子寄存器。CHx_CTL寄存器的ShadowReg_Or_LLI_Valid位必须是最后设置为1，表示阴影寄存器内容有效。如果从接口数据总线宽度或传输大小小于64位，必须最后更新CHx_CTL[63:56]；

4. 软件通过将1写入DMAC_ChEnReg中的适当位位置来启用通道；
5. dmac基于块传输的设置来启动DMA块传输操作：
6. 软件轮询CHx_CTL寄存器中的ShadowReg_Or_LLI有效位，直到它为0。
7. 软件为下一个块的传输配置CHx_SAR、CHx_DAR、CHx_BLOCK_TS和CHx_CTL寄存器。
8. dmac基于块传输的设置来启动DMA块传输操作。
9. 软件等待块传输完成中断或轮询块传输完成 CHx_IntStatusReg寄存器的指示位（块结束），直到它变为1。

- **基于链表的多块传输编程流程**

1. 软件读取DMAC通道启用寄存器（DMAC_ChEnReg）以选择可用的（未使用的 ）通道。
2. 软件使用DMA传输的适当值对CHx_CFG寄存器进行编程。SRC_MLTBLK_TYPE和DST_MLTBLK_TYPE位必须设置为2'b11。
3. 软件设置第一个链表项的基地址和链接列表项在CHx_LLP寄存器中可用。
4. 软件在系统内存中创建一个或多个链表项。软件可以预先创建整个链表项，或者使用LLI的CHx_CTL.ShadowReg_Or_LLI_Valid 和CHx_CTL.LLI_Last位。
5. 软件通过将1写入DMAC_ChEnReg中的适当位位置来启用通道。
6. axi_dmac基于块传输的设置来启动DMA块传输操作。块传输可以立即开始，也可以在硬件或软件握手请求之后开始，取决于CHx_CFG寄存器中TT_FC字段的设置。axi_dmac将链表内容复制到用于执行DMA块的寄存器传输（即CHx_SAR、CHx_DAR、CHx_BLOCK_TS和CHx_CTL寄存器）和启动DMA块传输。
7. 在链表获取阶段：

a) 如果检测CHx_CTL.ShadowReg_Or_LLI_Last 为1，则当前块是传输中的最后一个块并完成DMA结束传输操作。

b) 如果检测CHx_CTL.ShadowReg_Or_LLI_Last 为0，则有一个或多个块要传输，然后转到步骤6。

c) 如果检测CHx_CTL.ShadowReg_Or_LLI_Last 为0，axi_dmac可能会生成“ShadowReg_Or_LLI_Invalid_ERR”中断。在尝试另一个LLI读取操作之前，等待软件将（任何值）写入CHx_BLK_TFR_ResumeReqReg以指示有效的LLI可用性。

- **单块传输程序流程**

1. 软件读取DMAC通道启用寄存器（DMAC_ChEnReg）以选择空闲（未使用）通道；
2. 具有源和目标多块类型值的设置CHx_CFG寄存器外围设备为2b00；
3. 软件配置CHx_SAR、CHx_DAR、CHx_BLOCK_TS和CHx_CTL寄存器块适当的值；
4. 软件通过将1写入DMAC_ChEnReg中的适当位位置来启用通道；
5. 源和目标请求单个或突发DMA事务来传输数据块（假设是非内存外设），dmac在完成块中进行的事务（突发和单一）并执行块传输。
6. 软件等待块传输完成中断/轮询块传输完成，即CHx_IntStatusReg寄存器中的指示位（块结束），直到该位为1。

3.17.3 寄存器列表

Base Address：DMAC_BASE_ADDR(0x50000000)

Name	Description	Offset address
DMAC_IDREG	DMAC ID Register contains a 64-bit value that is hardwired and read back by a read to the axi_dmac ID Register.	0x0
DMAC_COMPVERREG	This register contains a 64-bit value that is hardwired and read back by a read to the axi_dmac Component Version Register.	0x8
DMAC_CFGREG	This register is used to enable the axi_dmac, which must be done before any channel activity can begin. This register also contains global interrupt enable bit.	0x10
DMAC_CHENREG	This is axi_dmac Channel Enable Register.	0x18
DMAC_INTSTATUSREG	DMAC Interrupt Status Register captures the combined channel interrupt for each channel and Combined common register block interrupt.	0x30
DMAC_COMMONREG_INTCLEARREG	Writing 1 to specific field enables the corresponding interrupt status generation in DMAC Common register Interrupt Status Register (DMAC_CommonReg_IntStatusReg).	0x38
DMAC_COMMONREG_INTSTATUS_ENABLEREG	Writing 1 to specific field enables the corresponding interrupt status generation in DMAC Common register Interrupt Status Register (DMAC_CommonReg_IntStatusReg).	0x40

DMAC_COMMONREG_INTSIGNAL_ENABLEREG	Writing 1 to specific field will propagate the corresponding interrupt status in DMAC Common register Interrupt Status Register (DMAC_CommonReg_IntStatusReg) to generate an port level interrupt.	0x48
DMAC_COMMONREG_INTSTATUSREG	This Register captures Slave interface access errors.	0x50
DMAC_RESETREG	This register is used to initiate the Software Reset to DMAC.	0x58
CHn_SAR	The starting source address is programmed by software before the DMA channel is enabled, or by an LLI update before the start of the DMA transfer. While the DMA transfer is in progress, this register is updated to reflect the source address of the current AXI transfer.	0x100*n
CHn_DAR	The starting destination address is programmed by the software before the DMA channel is enabled, or by an LLI update before the start of the DMA transfer. While the DMA transfer is in progress, this register is updated to reflect the destination address of the current AXI transfer.	0x100*n+8
CHn_BLOCK_TS	When axi_dmac is the flow controller, the DMAC uses this register before the channel is enabled for block-size.	0x100*n+10
CHn_CTL	This register contains fields that control the DMA transfer. This register should be programmed prior to enabling the channel except for LLI-based multi-block transfer.	0x100*n+18
CHn_CFG	This register contains fields that configure the DMA transfer. This register should be programmed prior	0x100*n+20

	to enabling the channel.	
CHn_LLP	This is the Linked List Pointer register.	0x100*n+28
CHn_STATUSREG	Channelx Status Register contains fields that indicate the status of DMA transfers for Channelx.	0x100*n+30
CHn_SWHSSRCREG	Channelx Software handshake Source Register.	0x100*n+38
CHn_SWHSDSTREG	Channelx Software handshake Destination Register.	0x100*n+40
CHn_BLK_TFR_RESUMEREQREG	Channelx Block Transfer Resume Request Register. This register is used during Linked List or Shadow Register based multi-block transfer.	0x100*n+48
CHn_AXI_IDREG	Channelx AXI ID Register. This register is allowed to be updated only when the channel is disabled, which means that it remains fixed for the entire DMA transfer.	0x100*n+50
CHn_AXI_QOSREG	Channelx AXI QOS Register. This register is allowed to be updated only when the channel is disabled, which means that it remains fixed for the entire DMA transfer.	0x100*n+58
CHn_INTSTATUS_ENABLEREG	Writing 1 to specific field enables the corresponding interrupt status generation in Channelx Interrupt Status Register(CH1_IntStatusReg).	0x100*n+80
CHn_INTSTATUS	Channelx Interrupt Status Register captures the Channelx specific interrupts	0x100*n+88
CHn_INTSIGNAL_ENABLEREG	This register contains fields that are used to enable the generation of port level interrupt at the channel level.	0x100*n+90
CHn_INTCLEARREG	Writing 1 to specific field will clear the corresponding field in Channelx	0x100*n+98

	Interrupt Status Register(CHx_IntStatusReg).	

3.17.4 寄存器设置

3.17.4.1 DMAC_IDREG(0x0)

Bits	Name	Description	Memory Access
63:0	DMAC_ID	DMAC ID Number.	R

3.17.4.2 DMAC_COMPVERREG(0x8)

Bits	Name	Description	Memory Access
63:32	RSVD_DMAC_COMPVERREG	DMAC_COMPVERREG Reserved bits.	R
31:0	DMAC_COMPVER	DMAC Component Version Number.	R

3.17.4.3 DMAC_CFGREG(0x10)

Bits	Name	Description	Memory Access
63:2	RSVD_DMAC_CFGREG	DMAC_CFGREG Reserved bits.	R
1	INT_EN	This bit is used to globally enable the interrupt generation. 0: DMAC Interrupts are disabled 1: DMAC Interrupt logic is enabled Values: 0x1 (ENABLED): DMAC Interrupts are enabled 0x0 (DISABLED): DMAC Interrupts are disabled	R/W

0	DMAC_EN	This bit is used to enable the DMAC. 0: DMAC disabled 1: DMAC enabled NOTE: If this bit DMAC_EN bit is cleared while any channel is still active, then this bit still returns 1 to indicate that there are channels still active until DMAC hardware has terminated all activity on all channels, at which point this bit returns zero (0). Values: 0x1 (ENABLED): DMAC is enabled 0x0 (DISABLED): DMAC is disabled	R/W

3.17.4.4 DMAC_CHENREG(0x18)

Bits	Name	Description	Memory Access
63:48	RSVD_DMAC_CHENREG	DMAC_CHENREG Reserved bits. Volatile: true	R
47:44	RSVD_DMAC_CHENREG_CH_ABORT_WE	CH_ABORT_WE Reserved bits. Volatile: true	R
43	CH4_ABORT_WE	This bit is used to write enable the Channel-4 Abort bit.The read back value of this register bit is always 0. Values: 0x1:(ENABLE_WR_CH4_ABORT): Enable Write to CH4_ABORT bit 0x0 :(DISABLE_WR_CH4_ABORT): Disable Write to CH4_ABORT bit Volatile: true	R

42	CH3_ABORT_WE	This bit is used to write enable the Channel-3 Abort bit.The read back value of this register bit is always 0. Values: 0x1:(ENABLE_WR_CH3_ABORT): Enable Write to CH3_ABORT bit 0x0:(DISABLE_WR_CH3_ABORT): Disable Write to CH3_ABORT bit Volatile: true	R
41	CH2_ABORT_WE	This bit is used to write enable the Channel-2 Abort bit.The read back value of this register bit is always 0. Values: 0x1:(ENABLE_WR_CH2_ABORT): Enable Write to CH2_ABORT bit 0x0:(DISABLE_WR_CH2_ABORT): Disable Write to CH2_ABORT bit Volatile: true	R
40	CH1_ABORT_WE	This bit is used to write enable the Channel-1 Abort bit.The read back value of this register bit is always 0. Values: 0x1: (ENABLE_WR_CH1_ABORT): Enable Write to CH1_ABORT bit 0x0:(DISABLE_WR_CH1_ABORT): Disable Write to CH1_ABORT bit Volatile: true	R
39:36	RSVD_DMAC_CHENREG_CH_ABORT	CH_ABORT Reserved bits. Volatile: true	R
		Channel-4 Abort Request. Software sets this bit to 1 to request channel abort. If this bit is set to 1, DMAC disables the channel immediately. Aborting the channel might result in AXI Protocol violation as DMAC does not make sure that all AXI transfers initiated on the master interface are completed. Aborting the channel is not	

35	CH4_ABORT	recommended and should be used only in situations where a particular channel hangs due to no response from the corresponding AXI slave interface and software wants to disable the channel without resetting the entire DMAC. It is recommended to try channel disabling first and then only opt for channel aborting. 0: No Channel Abort Request. 1: Request for Channel Abort.DMAC clears this bit to 0 once the channel is aborted (when it sets CH4_Status.CH_ABORTED bit to 1). Values: 0x1: (ENABLE_CH4_ABORT): Request for Channel-4 Abort 0x0: (DISABLE_CH4_ABORT): No Request for Channel-4 Abort Volatile: true	R
		Channel-3 Abort Request. Software sets this bit to 1 to request channel abort. If this bit is set to 1, DMAC disables the channel immediately. Aborting the channel might result in AXI Protocol violation as DMAC does not make sure that all AXI transfers initiated on the master interface are completed. Aborting the channel is not recommended and should be used only in situations where a particular channel hangs due to no response from the corresponding AXI slave interface and software wants to disable the channel without resetting the entire DMAC. It is recommended to try channel disabling first and then only opt for	

34	CH3_ABORT	channel aborting. 0: No Channel Abort Request. 1: Request for Channel Abort.DMAC clears this bit to 0 once the channel is aborted (when it sets CH3_Status.CH_ABORTED bit to 1). Values: 0x1: (ENABLE_CH3_ABORT): Request for Channel-3 Abort 0x0: (DISABLE_CH3_ABORT): No Request for Channel-3 Abort Volatile: true	R
33	CH2_ABORT	Channel-2 Abort Request. Software sets this bit to 1 to request channel abort. If this bit is set to 1, DMAC disables the channel immediately. Aborting the channel might result in AXI Protocol violation as DMAC does not make sure that all AXI transfers initiated on the master interface are completed. Aborting the channel is not recommended and should be used only in situations where a particular channel hangs due to no response from the corresponding AXI slave interface and software wants to disable the channel without resetting the entire DMAC. It is recommended to try channel disabling first and then only opt for channel aborting. 0: No Channel Abort Request. 1: Request for Channel Abort.DMAC clears this bit to 0 once the channel is aborted (when it sets CH2_Status.CH_ABORTED bit to 1). Values: 0x1: (ENABLE_CH2_ABORT): Request for Channel-2 Abort 0x0:	R

		(DISABLE_CH2_ABORT): No Request for Channel-2 Abort Volatile: true	
32	CH1_ABORT	Channel-1 Abort Request. Software sets this bit to 1 to request channel abort. If this bit is set to 1, DMAC disables the channel immediately. Aborting the channel might result in AXI Protocol violation as DMAC does not make sure that all AXI transfers initiated on the master interface are completed. Aborting the channel is not recommended and should be used only in situations where a particular channel hangs due to no response from the corresponding AXI slave interface and software wants to disable the channel without resetting the entire DMAC. It is recommended to try channel disabling first and then only opt for channel aborting. 0: No Channel Abort Request. 1: Request for Channel Abort.DMAC clears this bit to 0 once the channel is aborted (when it sets CH1_Status.CH_ABORTED bit to 1). Values: 0x1: (ENABLE_CH1_ABORT): Request for Channel-1 Abort 0x0: (DISABLE_CH1_ABORT): No Request for Channel-1 Abort Volatile: true	R
31:28	RSVD_DMAC_CHENREG_CH_SUSP_WE	CH_SUSP_WE Reserved bits. Volatile: true	R
		This bit is used as a write enable to the Channel-4 Suspend bit. The read back value of this register	

27	CH4_SUSP_WE	bit is always 0. Values: 0x1:(ENABLE_WR_CH4_SUSP): Enable Write to respective CH4_SUSP bit 0x0: (DISABLE_WR_CH4_SUSP): Disable Write to CH${ch_num}_SUSP bit Volatile: true	W
26	CH3_SUSP_WE	This bit is used as a write enable to the Channel-3 Suspend bit. The read back value of this register bit is always 0. Values: 0x1:(ENABLE_WR_CH3_SUSP): Enable Write to respective CH3_SUSP bit 0x0:(DISABLE_WR_CH3_SUSP): Disable Write to CH${ch_num}_SUSP bit Volatile: true	W
25	CH2_SUSP_WE	This bit is used as a write enable to the Channel-2 Suspend bit. The read back value of this register bit is always 0. Values: 0x1: (ENABLE_WR_CH2_SUSP): Enable Write to respective CH2_SUSP bit 0x0: (DISABLE_WR_CH2_SUSP): Disable Write to CH${ch_num}_SUSP bit Volatile: true	W
24	CH1_SUSP_WE	This bit is used as a write enable to the Channel-1 Suspend bit. The read back value of this register bit is always 0. Values: 0x1: (ENABLE_WR_CH1_SUSP): Enable Write to respective CH1_SUSP bit 0x0: (DISABLE_WR_CH1_SUSP): Disable Write to CH${ch_num}_SUSP	W

		bit Volatile: true	
23:20	RSVD_DMAC_CHENREG_CH_SUSP	CH_SUSP Reserved bits. Volatile: true	R
19	CH4_SUSP	Channel-4 Suspend Request. Software sets this bit to 1 to request channel suspend. If this bit is set to 1, DMAC suspends all DMA data transfers from the source gracefully until this bit is cleared. There is no guarantee that the current dma transaction will complete. This bit can also be used in conjunction with CH4_Status.CH_SUSPENDED to cleanly disable the channel without losing any data. In this case, software first sets CH4_SUSP bit to 1 and polls CH4_Status.CH_SUSPENDED till it is set to 1. Software can then clear CH4_EN bit to 0 to disable the channel. 0: No Channel Suspend Request. 1: Request for Channel Suspend. Software can clear CH4_SUSP bit to 0, after DMAC sets CH4_Status.CH_SUSPENDED bit to 1, to exit the channel suspend mode. NOTE: CH_SUSP is cleared when channel is disabled. Values: 0x1: (ENABLE_CH4_SUSP): Request to Suspended Channel-4 0x0: (DISABLE_CH4_SUSP): No Channel Suspend Request Volatile: true	R/W
		Channel-3 Suspend Request. Software sets this bit to 1 to request channel suspend. If this bit is set to 1,	

18	CH3_SUSP	DMAC suspends all DMA data transfers from the source gracefully until this bit is cleared. There is no guarantee that the current dma transaction will complete. This bit can also be used in conjunction with CH3_Status.CH_SUSPENDED to cleanly disable the channel without losing any data. In this case, software first sets CH3_SUSP bit to 1 and polls CH3_Status.CH_SUSPENDED till it is set to 1. Software can then clear CH3_EN bit to 0 to disable the channel. 0: No Channel Suspend Request. 1: Request for Channel Suspend. Software can clear CH3_SUSP bit to 0, after DMAC sets CH3_Status.CH_SUSPENDED bit to 1, to exit the channel suspend mode. NOTE: CH_SUSP is cleared when channel is disabled. Values: 0x1: (ENABLE_CH3_SUSP): Request to Suspended Channel-3 0x0: (DISABLE_CH3_SUSP): No Channel Suspend Request Volatile: true	R/W
		Channel-2 Suspend Request. Software sets this bit to 1 to request channel suspend. If this bit is set to 1, DMAC suspends all DMA data transfers from the source gracefully until this bit is cleared. There is no guarantee that the current dma transaction will complete. This bit can also be used in conjunction with CH2_Status.CH_SUSPENDED to cleanly disable the channel without	

17	CH2_SUSP	losing any data. In this case, software first sets CH2_SUSP bit to 1 and polls CH2_Status.CH_SUSPENDED till it is set to 1. Software can then clear CH2_EN bit to 0 to disable the channel. 0: No Channel Suspend Request. 1: Request for Channel Suspend.Software can clear CH2_SUSP bit to 0, after DMAC sets CH2_Status.CH_SUSPENDED bit to 1, to exit the channel suspend mode. NOTE: CH_SUSP is cleared when channel is disabled. Values: 0x1 (ENABLE_CH2_SUSP): Request to Suspended Channel-2 0x0 (DISABLE_CH2_SUSP): No Channel Suspend Request Volatile: true	R/W
		Channel-1 Suspend Request. Software sets this bit to 1 to request channel suspend. If this bit is set to 1, DMAC suspends all DMA data transfers from the source gracefully until this bit is cleared. There is no guarantee that the current dma transaction will complete. This bit can also be used in conjunction with CH1_Status.CH_SUSPENDED to cleanly disable the channel without losing any data. In this case, software first sets CH1_SUSP bit to 1 and polls CH1_Status.CH_SUSPENDED till it is set to 1. Software can then clear CH1_EN bit to 0 to disable the channel. 0: No Channel Suspend Request. 1:	

16	CH1_SUSP	Request for Channel Suspend.Software can clear CH1_SUSP bit to 0, after DMAC sets CH1_Status.CH_SUSPENDED bit to 1, to exit the channel suspend mode. NOTE: CH_SUSP is cleared when channel is disabled. Values: 0x1 (ENABLE_CH1_SUSP): Request to Suspended Channel-1 0x0 (DISABLE_CH1_SUSP): No Channel Suspend Request Volatile: true	R/W
15:12	RSVD_DMAC_CHENREG_CH_WE_EN	CH_WE_EN Reserved bits Volatile: true	R
11	CH4_EN_WE	DMAC Channel-4 Enable Write Enable bit.Read back value of this register bit is always '0'. Values: 0x1 (ENABLE_WR_CH4_EN): Enable Write to CH4_EN bit 0x0 (DISABLE_WR_CH4_EN): Disable Write to respective CH4_EN bit Volatile: true	W

Bits	Name	Description	Memory Access
10	CH3_EN_WE	DMAC Channel-3 Enable Write Enable bit.Read back value of this register bit is always '0'. Values: 0x1 (ENABLE_WR_CH3_EN): Enable Write to CH3_EN bit 0x0 (DISABLE_WR_CH3_EN): Disable Write to respective CH3_EN bit Volatile: true	W

9	CH2_EN_WE	DMAC Channel-2 Enable Write Enable bit.Read back value of this register bit is always '0'. Values: 0x1 (ENABLE_WR_CH2_EN): Enable Write to CH2_EN bit 0x0 (DISABLE_WR_CH2_EN): Disable Write to respective CH2_EN bit Volatile: true	W
8	CH1_EN_WE	DMAC Channel-1 Enable Write Enable bit.Read back value of this register bit is always '0'. Values: 0x1 (ENABLE_WR_CH1_EN): Enable Write to CH1_EN bit 0x0 (DISABLE_WR_CH1_EN): Disable Write to respective CH1_EN bit Volatile: true	W
7:4	RSVD_DMAC_CHENREG_CH_EN	CH_EN Reserved bits Volatile: true	R
3	CH4_EN	This bit is used to enable the DMAC Channel-4. 0: DMAC Channel-4 is disabled 1: DMAC Channel-4 is enabled The bit DMAC_ChEnReg.CH4_EN is automatically cleared by hardware to disable the channel after the last AMBA transfer of the DMA transfer to the destination has completed. Software can therefore poll this bit to determine when this channel is free for a new DMA transfer. Values: 0x1 (ENABLE_CH4): DMAC: Channel-4 is enabled 0x0 (DISABLE_CH4): DMAC: Channel-4 is disabled Volatile: true	R/W
		This bit is used to enable the DMAC Channel-3. 0: DMAC Channel-3 is disabled 1:	

2	CH3_EN	DMAC Channel-3 is enabledThe bit DMAC_ChEnReg.CH3_EN is automatically cleared by hardware to disable the channel after the last AMBA transfer of the DMA transfer to the destination has completed. Software can therefore poll this bit to determine when this channel is free for a new DMA transfer. Values: 0x1 (ENABLE_CH3): DMAC: Channel-3 is enabled 0x0 (DISABLE_CH3): DMAC: Channel-3 is disabled Volatile: true	R/W
1	CH2_EN	This bit is used to enable the DMAC Channel-2. 0: DMAC Channel-2 is disabled 1: DMAC Channel-2 is enabled The bit DMAC_ChEnReg.CH2_EN is automatically cleared by hardware to disable the channel after the last AMBA transfer of the DMA transfer to the destination has completed. Software can therefore poll this bit to determine when this channel is free for a new DMA transfer. Values: 0x1 (ENABLE_CH2): DMAC: Channel-2 is enabled 0x0 (DISABLE_CH2): DMAC: Channel-2 is disabled Volatile: true	R/W
		This bit is used to enable the DMAC Channel-1. 0: DMAC Channel-1 is disabled 1: DMAC Channel-1 is enabled The bit DMAC_ChEnReg.CH1_EN is automatically cleared by hardware to disable the channel after the last AMBA transfer of the DMA transfer to the destination has completed.	

0	CH1_EN	Software can therefore poll this bit to determine when this channel is free for a new DMA transfer. Values: 0x1 (ENABLE_CH1): DMAC: Channel-1 is enabled 0x0 (DISABLE_CH1): DMAC: Channel-1 is disabled Volatile: true	R/W

3.17.4.5 DMAC_INTSTATUSREG(0x30)

Bits	Name	Description	Memory Access
63:17	RSVD_DMAC_INTSTATUSREG_63to17	DMAC Interrupt Status Register (bits 63to17) Reserved bits Volatile: true	R
16	CommonReg_IntStat	Common Register Interrupt Status Bit. Values: 0x1 (ACTIVE): Common Register Interrupt is Active 0x0 (INACTIVE): Common Register Interrupt is Inactive Volatile: true	R
15:4	RSVD_DMAC_INTSTATUSREG	DMAC Interrupt Status Register (bits 15to8) Reserved bits Volatile: true	R
3	CH4_IntStat	Channel 4 Interrupt Status Bit. Values: 0x1 (ACTIVE): Channel 4 Interrupt is Active 0x0 (INACTIVE): Channel 4 Interrupt is Inactive Volatile: true	R
2	CH3_IntStat	Channel 3 Interrupt Status Bit. Values: 0x1 (ACTIVE): Channel 3 Interrupt is Active 0x0 (INACTIVE): Channel 3 Interrupt is Inactive Volatile: true	R
1	CH2_IntStat	Channel 2 Interrupt Status Bit. Values: 0x1 (ACTIVE): Channel 2 Interrupt is Active 0x0 (INACTIVE): Channel 2 Interrupt is Inactive Volatile: true	R

0	CH1_IntStat	Channel 1 Interrupt Status Bit. Values: 0x1 (ACTIVE): Channel 1 Interrupt is Active 0x0 (INACTIVE): Channel 1 Interrupt is Inactive Volatile: true	R

3.17.4.6 DMAC_COMMONREG_INTCLEARREG(0x38)

Bits	Name	Description	Memory Access
63:9	RSVD	DMAC Common Register Interrupt Clear Register (bits 63to9) Reserved bits	W
8	Clear_SLVIF_UndefinedReg_DEC_ERR_IntStat	Slave Interface Undefined register Decode Error Interrupt clear Bit. This bit is used to clear the corresponding channel interrupt status bit(SLVIF_UndefinedReg_DEC_ERR_IntStat in DMAC_CommonReg_IntStatusReg. Values: 0x1 (CLEAR_SLVIF_UndefinedReg_DEC_ERR): Clear the SLVIF_UndefinedReg_DEC_ERR interrupt in the interrupt register DMAC_CommonReg_IntStatusReg 0x0 (No_ACTION): Inactive signal. No action taken.	W
7:4	RSVD_DMAC_COMMONREG_INTCLEARREG_7to4	DMAC Common Register Interrupt Clear Register (bits 7to4) Reserved bits	W
3	Clear_SLVIF_CommonReg_WrOnHold_ERR_IntStat	Slave Interface Common Register Write On Hold Error Interrupt clear Bit. This bit is used to clear the corresponding channel interrupt status bit(SLVIF_CommonReg_WrOnHold_ERR_IntStat in DMAC_CommonReg_IntStatusReg. Values: 0x1 (CLEAR_SLVIF_CommonReg_WrOnHold_ERR): Clear the SLVIF_CommonReg_WrOnHold_ERR interrupt in the interrupt register	W

		DMAC_CommonReg_IntStatusReg 0x0 (No_ACTION): Inactive signal. No action taken.	
2	Clear_SLVIF_CommonReg_RD2WO_ERR_IntStat	Slave Interface Common Register Read to Write only Error Interrupt clear Bit. This bit is used to clear the corresponding channel interrupt status bit(SLVIF_CommonReg_RD2WO_ERR_IntStat in DMAC_CommonReg_IntStatusReg. Values: 0x1 (CLEAR_SLVIF_CommonReg_RD2WO_ERR): Clear the SLVIF_CommonReg_RD2WO_ERR interrupt in the interrupt register DMAC_CommonReg_IntStatusReg 0x0 (No_ACTION): Inactive signal. No action taken.	W
1	Clear_SLVIF_CommonReg_WR2RO_ERR_IntStat	Slave Interface Common Register Write to Read only Error Interrupt clear Bit. This bit is used to clear the corresponding channel interrupt status bit(SLVIF_CommonReg_WR2RO_ERR_IntStat in DMAC_CommonReg_IntStatusReg. Values: 0x1 (CLEAR_SLVIF_CommonReg_WR2RO_ERR): Clear the SLVIF_CommonReg_WR2RO_ERR	W

		interrupt in the interrupt register DMAC_CommonReg_IntStatusReg 0x0 (No_ACTION): Inactive signal. No action taken.	
0	Clear_SLVIF_CommonReg_DEC_ERR_IntStat	Slave Interface Common Register Decode Error Interrupt clear Bit. This bit is used to clear the corresponding channel interrupt status bit (SLVIF_CommonReg_DEC_ERR_IntStat in DMAC_CommonReg_IntStatusReg. Values: 0x1 (CLEAR_SLVIF_CommonReg_DEC_ERR): Clear the SLVIF_CommonReg_DEC_ERR interrupt in the interrupt register DMAC_CommonReg_IntStatusReg 0x0 (No_ACTION): Inactive signal. No action taken.	W

3.17.4.7 DMAC_COMMONREG_INTSTATUS_ENABLEREG(0x40)

Bits	Name	Description	Memory Access
63:9	RSVD_DMAC_COMMONREG_INTSTATUS_ENABLEREG_63to9	DMAC Common Register Interrupt Status Enable Register (bits 63to9) Reserved bits	R
8	Enable_SLVIF_UndefinedReg_DEC_ERR_IntStat	Slave Interface Undefined register Decode Error Interrupt Status enable Bit. This bit is used to enable the corresponding channel interrupt status bit (SLVIF_UndefinedReg_DEC_ERR_IntStat in DMAC_CommonReg_IntStatusReg. Values: 0x1 (ENABLE_SLVIF_UndefinedReg_DEC_ERR): SLVIF_UndefinedReg_DEC_ERR_IntStat bit in DMAC_CommonReg_IntStatusReg is Enabled 0x0 (DISABLE_SLVIF_UndefinedReg_DEC_ERR): SLVIF_UndefinedReg_DEC_ERR_IntStat bit in DMAC_CommonReg_IntStatusReg is Disabled	R/W
7:4	RSVD_DMAC_COMMONREG_INTSTATUS_ENABLEREG_7to4	DMAC Common Register Interrupt Status Enable Register (bits 7to4) Reserved bits	R

3	Enable_SLVIF_CommonReg_WrOnHold_ERR_IntStat	Slave Interface Common Register Write On Hold Error Interrupt Status Enable Bit. This bit is used to enable the corresponding channel interrupt status bit (SLVIF_CommonReg_WrOnHold_ERR_IntStat in DMAC_CommonReg_IntStatusReg. Values: 0x1 (ENABLE_SLVIF_CommonReg_WrOnHold_ERR): SLVIF_CommonReg_WrOnHold_ERR_IntStat bit in DMAC_CommonReg_IntStatusReg is Enabled 0x0 (DISABLE_SLVIF_CommonReg_WrOnHold_ERR): SLVIF_CommonReg_WrOnHold_ERR_IntStat bit in DMAC_CommonReg_IntStatusReg is Disabled	R/W
2	Enable_SLVIF_CommonReg_RD2WO_ERR_IntStat	Slave Interface Common Register Read to Write only Error Interrupt Status Enable Bit. This bit is used to enable the corresponding channel interrupt status bit (SLVIF_CommonReg_RD2WO_ERR_IntStat in DMAC_CommonReg_IntStatusReg. Values: 0x1 (ENABLE_SLVIF_CommonReg_RD2WO_ERR): SLVIF_CommonReg_RD2WO_ERR_IntStat bit in DMAC_CommonReg_IntStatusReg is Enabled 0x0	R/W

		(DISABLE_SLVIF_CommonReg_RD2WO_ERR): SLVIF_CommonReg_RD2WO_ERR_IntStat bit in DMAC_CommonReg_IntStatusReg is Disabled	
1	Enable_SLVIF_CommonReg_WR2RO_ERR_IntStat	Slave Interface Common Register Write to Read only Error Interrupt Status Enable Bit. This bit is used to enable the corresponding channel interrupt status bit (SLVIF_CommonReg_WR2RO_ERR_IntStat in DMAC_CommonReg_IntStatusReg. Values: 0x1 (ENABLE_SLVIF_CommonReg_WR2RO_ERR): SLVIF_CommonReg_WR2RO_ERR_IntStat bit in DMAC_CommonReg_IntStatusReg is Enabled 0x0 (DISABLE_SLVIF_CommonReg_WR2RO_ERR): SLVIF_CommonReg_WR2RO_ERR_IntStat bit in DMAC_CommonReg_IntStatusReg is Disabled	R/W
0	Enable_SLVIF_CommonReg_DEC_ERR_IntStat	Slave Interface Common Register Decode Error Interrupt Status Enable Bit. This bit is used to enable the corresponding channel interrupt status bit (SLVIF_CommonReg_DEC_ERR_IntStat in DMAC_CommonReg_IntStatusReg. Values: 0x1	R/W

		(ENABLE_SLVIF_CommonReg_DEC_ERR): SLVIF_CommonReg_DEC_ERR_IntStat bit in DMAC_CommonReg_IntStatusReg is Enabled 0x0 (DISABLE_SLVIF_CommonReg_DEC_ERR): SLVIF_CommonReg_DEC_ERR_IntStat bit in DMAC_CommonReg_IntStatusReg is Disabled	

3.17.4.8 DMAC_COMMONREG_INTSIGNAL_ENABLEREG(0x48)

Bits	Name	Description	Memory Access
63:9	RSVD	Reserved bits	R
8	Enable_SLVIF_UndefinedReg_DEC_ERR_IntSignal	Slave Interface Undefined register Decode Error Interrupt Signal Enable Bit. This bit is used to enable the propagation of corresponding channel interrupt status bit(SLVIF_UndefinedReg_DEC_ERR_IntStat in DMAC_CommonReg_IntStatusReg) to generate a port level interrupt. Values: 0x1: SLVIF_UndefinedReg_DEC_ERR_IntStat signal in DMAC_CommonReg_IntStatusReg is Enabled at port level 0x0: SLVIF_UndefinedReg_DEC_ERR_IntStat signal in DMAC_CommonReg_IntStatusReg	R/W

		is Disabled at port level	
7:4	RSVD	Reserved bits	R
3	Enable_SLVIF_CommonReg_WrOnHold_ERR_IntSignal	Slave Interface Common Register Write On Hold Error Interrupt Signal Enable Bit.This bit is used to enable the propagation of corresponding channel interrupt status bit(SLVIF_CommonReg_WrOnHold_ERR_IntStat in DMAC_CommonReg_IntStatusReg) to generatea port level interrupt. Values: 0x1 : SLVIF_CommonReg_WrOnHold_ERR_IntStat signal in DMAC_CommonReg_IntStatusReg is Enabled at port level 0x0 : SLVIF_CommonReg_WrOnHold_ERR_IntStat signal in DMAC_CommonReg_IntStatusReg is Disabled at port level	R/W
2	Enable_SLVIF_CommonReg_RD2WO_ERR_IntSignal	Slave Interface Common Register Read to Write only Error Interrupt Signal Enable Bit. This bit is used to enable the propagation of corresponding channel interrupt status bit (SLVIF_CommonReg_RD2WO_ERR_IntStat in DMAC_CommonReg_IntStatusReg) to generate a port level interrupt. Values: 0x1 : SLVIF_CommonReg_RD2WO_ERR_IntStat signal in DMAC_CommonReg_IntStatusReg is	R/W

		Enabled at port level 0x0 : SLVIF_CommonReg_RD2WO_ERR_IntStat signal in DMAC_CommonReg_IntStatusReg is Disabled at port level	
1	Enable_SLVIF_CommonReg_WR2RO_ERR_IntSignal	Slave Interface Common Register Write to Read only Error Interrupt Signal Enable Bit. This bit is used to enable the propagation of corresponding channel interrupt status bit (SLVIF_CommonReg_WR2RO_ERR_IntStat in DMAC_CommonReg_IntStatusReg) to generate a port level interrupt. Values: 0x1 : SLVIF_CommonReg_WR2RO_ERR_IntStat signal in DMAC_CommonReg_IntStatusReg is Enabled at port level 0x0 : SLVIF_CommonReg_WR2RO_ERR_IntStat signal in DMAC_CommonReg_IntStatusReg is Disabled at port level	R/W
0	Enable_SLVIF_CommonReg_DEC_ERR_IntSignal	Slave Interface Common Register Decode Error Interrupt Signal Enable Bit. This bit is used to enable the propagation of corresponding channel interrupt status bit (SLVIF_CommonReg_DEC_ERR_IntStat in DMAC_CommonReg_IntStatusReg) to generate a port level interrupt. Values:	R/W

		0x1 : SLVIF_CommonReg_DEC_ERR_IntStat signal in DMAC_CommonReg_IntStatusReg is Enabled at port level 0x0): SLVIF_CommonReg_DEC_ERR_IntStat signal in DMAC_CommonReg_IntStatusReg is Disabled at port level	

3.17.4.9 DMAC_COMMONREG_INTSTATUSREG(0x50)

Bits	Name	Description	Memory Access
63:9	RSVD	Reserved	R
8	SLVIF_UndefinedReg_DEC_ERR_IntStat	Slave Interface Undefined register Decode Error Interrupt Signal Enable Bit. Decode Error generated by DMAC during register access. This error occurs if the register access is to undefined address range (>0x8FF if 8 channels are configured, >0x4FF if 4 channels are configured etc.) resulting in error response by DMAC slave interface. 0: No Slave Interface Decode Errors. 1: Slave Interface Decode Error detected. Error Interrupt Status is generated if the corresponding Status Enable bit in DMAC_CommonReg_IntStatus_Enable register bit is set to 1. This bit is cleared to 0 on writing 1 to the corresponding channel interrupt clear bit in DMAC_COMMONREG_INTCLEARREG on enabling the channel (required when the interrupt is not enabled). Values: 0x1 (Active_UndefinedReg_DEC_ERR): Slave Interface Decode Error detected 0x0 (Inactive_UndefinedReg_DEC_ERR): No Slave Interface Decode Errors Volatile: true	R
7:4	RSVD_DMAC_COMMONREG_INTSTATUSREG_7to4	DMAC Common Register Interrupt Status Register (bits 7to4) Reserved bits Volatile: true	R

Bits	Name	Description	Memory Access
3	SLVIF_CommonReg _WrOnHold_ERR_IntStat	Slave Interface Common Register Write On Hold Error Interrupt Status Bit. This error occurs if an illegal write operation is performed on a common register; this happens if a write operation is performed on a common register except DMAC_RESETREG with DMAC_RST field set to 1 when DMAC is in Hold mode. 0: No Slave Interface Common Register Write On Hold Errors. 1: Slave Interface Common Register Write On Hold Error detected. Error Interrupt Status is generated if the corresponding Status Enable bit in DMAC_CommonReg_IntStatus_Enable register bit is set to 1. This bit is cleared to 0 on writing 1 to the corresponding channel interrupt clear bit in DMAC_COMMONREG_INTCLEARREG on enabling the channel (required when the interrupt is not enabled). Values: 0x1 (Active_CommonReg_WrOnHold_ERR): Slave Interface Common Register Write On Hold Error detected 0x0 (Inactive_CommonReg_WrOnHold_ERR): No Slave Interface Common Register Write On Hold Errors Volatile: true	R

2	SLVIF_CommonReg_RD2WO_ERR_IntStat	Slave Interface Common Register Read to Write only Error Interrupt Status bit. This error occurs if Read operation is performed to a Write Only register in the common register space (0x000 to 0x0FF).0: No Slave Interface Read to Write Only Errors.1: Slave Interface Read to Write Only Error detected.Error Interrupt status is generated if the corresponding Status Enable bit in DMAC_CommonReg_IntStatus_Enable register bit is set to 1. This bit is cleared to 0 on writing 1 to the corresponding channel interrupt clear bit in DMAC_COMMONREG_INTCLEARREG on enabling the channel (required when the interrupt is not enabled). Values: 0x1 (Active_CommonReg_RD2WO_ERR): Slave Interface Read to Write Only Error detected 0x0 (Inactive_CommonReg_RD2WO_ERR): No Slave Interface Read to Write Only Errors Volatile: true	R

1	SLVIF_CommonReg _WR2RO_ERR_IntStat	Slave Interface Common Register Write to Read Only Error Interrupt Status bit. This error occurs if write operation is performed to a Read Only register in the common register space (0x000 to 0x0FF).0: No Slave Interface Write to Read Only Errors.1: Slave Interface Write to Read Only Error detected.Error Interrupt status is generated if the corresponding Status Enable bit in DMAC_CommonReg_IntStatus_Enable register bit is set to 1. This bit is cleared to 0 on writing 1 to the corresponding channel interrupt clear bit in DMAC_COMMONREG_INTCLEARREG on enabling the channel (required when the interrupt is not enabled). Values: 0x1 (Active_CommonReg_WR2RO_ERR): No Slave Interface Write to Read Only Errors 0x0 (Inactive_CommonReg_WR2RO_ERR): Slave Interface Write to Read Only Error detected Volatile: true	R
0	SLVIF_CommonReg_DEC_E RR_ IntStat	Slave Interface Common Register Decode Error Interrupt Status Bit. Decode Error generated by DW_axi_dmac during register access. This error occurs if the register access is to an invalid address in the common register space (0x000 to 0x0FF) resulting in error response by DW_axi_dmac slave interface. ■ 0: No Slave Interface Decode Errors. ■ 1: Slave Interface Decode Error detected.	R

3.17.4.10 DMAC_RESETREG(0x58)

Bits	Name	Description	Memory Access
63:1	RSVD_DMAC_ResetReg_1to63	DMAC_ResetReg (bits 1to63) Reserved bits Volatile: true	R
0	DMAC_RST	DMAC Reset Request bitSoftware writes 1 to this bit to reset the DMAC and polls this bit to see it as 0. DMAC resets all the modules except the slave bus interface module and clears this bit to 0. NOTE: Software is not allowed to write 0 to this bit. Volatile: true	R/W

3.17.4.11 CHn_SAR(0x100*n)

Bits	Name	Description	Memory Access
63:0	SAR	Current Source Address of DMA transfer. Updated after each source transfer. The SINC fields in the CHx_CTL register determines whether the address increments or is left unchanged on every source transfer throughout the block transfer. Volatile: true	R/W

3.17.4.12 CHn_DAR(0x100*n+8)

Bits	Name	Description	Memory Access

63:0	DAR	Current Destination Address of DMA transfer. Updated after each destination transfer. The DINC fields in the CHx_CTL register determines whether the address increments or is left unchanged on every destination transfer throughout the block transfer. Volatile: true	R/W

3.17.4.13 CHn_BLOCK_TS(0x100*n+10)

Bits	Name	Description	Memory Access
63:22	RSVD_DMAC_CHx_BLOCK_TSREG_63to22	DMAC Channelx Block Transfer Size Register (bits 63to22) Reserved bits Volatile: true	R
21:0	BLOCK_TS	Block Transfer Size. The number programmed into BLOCK_TS field indicates the total number of data of width CHx_CTL.SRC_TR_WIDTH to be transferred in a DMA block transfer.Block Transfer Size = BLOCK_TS+1 When the transfer starts, the read-back value is the total number of data items already read from the source peripheral, regardless of who is the flow controller. When the source or destination peripheral is assigned as the flow controller, the value before the transfer starts saturates at DMAX_CHx_MAX_BLK_SIZE, but the actual block size can be greater. Volatile: true	R/W

3.17.4.14 CHn_CTL(0x100*n+18)

Bits	Name	Description	Memory Access

63	SHADOWREG_OR_LLI_VALID	Shadow Register content/Linked List Item valid. Indicates whether the content of shadow register or the linked list item fetched from the memory is valid. 0: Shadow Register content/LLI is invalid. 1: Last Shadow Register/LLI is valid. LLI based multi-block transfer: The CHx_CTL register is loaded from the LLI. Hence, the software is not allowed to directly update this register through the DMAC slave interface.This field can be used to dynamically extend the LLI by the software. On noticing this bit as 0, DMAC discards the LLI and generates the ShadowReg_Or_LII_Invalid_ERR Interrupt if the corresponding channel error interrupt mask bit is set to 0. In the case of LLI pre-fetching, the ShadowReg_Or_LLI_Invalid_ERR interrupt is not generated even if the ShadowReg_Or_LLI_Valid bit is seen to be 0 for the pre-fetched LLI. In this case, DMAC attempts the LLI fetch operation again after completing the current block transfer and generates the ShadowReg_Or_LII_Invalid_ERR interrupt only if ShadowReg_Or_LII_Valid bit is still seen to be 0. This error condition causes the DMAC to halt the corresponding channel gracefully. DMAC waits until software writes (any value) to CHx_BLK_TFR_ResumeReqReg to indicate valid LLI availability before attempting another LLI read operation. This bit is cleared to 0 and written back	R/W

to the corresponding LLI location after block transfer completion when LLI write-back option is enabled. Hence, for LLI-based multi-block transfers, the software might manipulate/redefine any descriptor with the ShadowReg_Or_LII_Valid bit set to 0 if LLI write-back option is enabled. Shadow Reg based multi-block transfer: On noticing this bit as 0 during shadow register fetch phase, DMAC discards the Shadow Register contents and generates ShadowReg_Or_LLI_Invlid_ERR Interrupt.
In this case, the software has to write (any value) to CHx_BLK_TFR_ResumeReqReg after updating the shadow registers and after setting ShadowReg_Or_LLI_Valid bit to 1 to indicate to DMAC that shadow register contents are valid and the next block transfer can be resumed.DMAC clears this bit to 0 after copying the shadow register contents. Software can reprogram the shadow registers only if ShadowReg_Or_LLI_Valid bit is 0. Software needs to read this register in block completion interrupt service routine (if interrupt is enabled)/continuously poll this register (if interrupt is not enabled) to make sure that this bit is 0 before updating the shadow registers.If shadow-register-based multi-block transfer is enabled and software attempts to write to the shadow register when ShadowReg_Or_LLI_Valid bit is 1, DMAC generates

		SLVIF_ShadowReg_WrOnValid_ERR interrupt. Values: 0x1 (VALID): Indicates shadowreg/LLI content is Valid 0x0 (INVALID): Indicates shadowreg/LLI content is Invalid Volatile: true	
62	SHADOWREG_OR_LLI_LAST	Last Shadow Register/Linked List Item. Indicates whether shadow register content or the linked list item fetched from the memory is the last one or not. 0: Not last Shadow Register/LLI 1: Last Shadow Register/LLI LLI based multi-block transfer: DMAC uses this bit to decide if another LLI fetch is needed in the current DMA transfer.If this bit is 0, DMAC fetches the next LLI from the address pointed out by LLP field in the current LLI.If this bit is 1, DMAC understands that current block is the final block in the dma transfer and ends the dma transfer once the AMBA transfer corresponding to the current block completes. Shadow Reg based multi-block transfer: DMAC uses this bit to decide if another Shadow Register fetch is needed in the current DMA transfer.If this bit is 0, DMAC understands that there are one or more blocks to be transferred in the current block and hence one or more shadow register set contents will be valid and needs to be fetched.If this bit is 1, DMAC understands that current block is the final block in the dma transfer and ends the dma transfer once the AMBA transfer corresponding to the current block	R/W

		completes. Values: 0x1 (LAST_ITEM): Indicates shadowreg/LLI content is the last one 0x0 (NOT_LAST_ITEM): Indicates shadowreg/LLI content is not the last one Volatile: true	
61:59	RSVD_DMAC_CHx_CTL_59to61	DMAC Channelx Control Transfer Register (bits 59to61) Reserved bits	R
58	IOC_BlkTfr	Interrupt On completion of Block Transfer. This bit is used to control the block transfer completion interrupt generation on a block by block basis for shadow register or linked list based multi-block transfers. Writing 1 to this register field enables CHx_IntStatusReg.BLOCK_TFR_DONE_IntStat field if this interrupt generation is enabled in CHx_IntStatus_EnableReg register and the external interrupt output is is asserted if this interrupt generation is enabled in CHx_IntSignal_EnableReg register. NOTE: If a linked-list or shadow-register-based multi-block transfer is not used for both source and destination (for instance if source and destination use contiguous address or auto-reload-based multi-block transfer), the value of this field cannot be modified per block. Additionally, the value programmed before the channel is enabled is used for all the blocks in the DMA transfer. Values: 0x1 (Enable_BLKTFR_INTR): Enables	R/W

		CHx_IntStatusReg.BLOCK_TFR_DONE_IntStat field 0x0 (DISABLE_BLKTFR_INTR): Disables CHx_IntStatusReg.BLOCK_TFR_DONE_IntStat field	
57	DST_STAT_EN	Destination Status Enable Enable the logic to fetch status from destination peripheral of channel x pointed to by the content of CHx_DSTATAR register and stores it in CHx_DSTAT register. This value is written back to the CHx_DSTAT location of linked list at end of each block transfer if DMAX_CHx_LLI_WB_EN is set to 1 and if linked list based multi-block transfer is used by either source or destination peripheral. Values: 0x1 (Enable_STAT_FETCH): Enables status fetch for Destination and store the value in CH1_DSTAT register 0x0 (NO_STAT_FETCH): No status fetch for Destination device	R
56	SRC_STAT_EN	Source Status Enable Enable the logic to fetch status from source peripheral of channel x pointed to by the content of CHx_SSTATAR register and stores it in CHx_SSTAT register. This value is written back to the CHx_SSTAT location of linked list at end of each block transfer if DMAX_CHx_LLI_WB_EN is set to 1 and if linked list based multi-block transfer is used by either source or destination peripheral. Values: 0x1 (Enable_STAT_FETCH): Enables	R

		status fetch for Source and store the value in CH1_SSTAT register 0x0 (NO_STAT_FETCH): No status fetch for Source device	
55:48	AWLEN	Destination Burst Length AXI Burst length used for destination data transfer. The specified burst length is used for destination data transfer till the extent possible; remaining transfers use maximum possible value that is less than or equal to DMAX_CHx_MAX_AMBA_BURST_LENGTH. The maximum value of AWLEN is limited by DMAX_CHx_MAX_AMBA_BURST_LENGTH. NOTE: The AWLEN setting may not be honored towards end-to-block transfers, the end of a transaction (only applicable to non-memory peripharals), and during 4K boundary crossings.	R/W
47	AWLEN_EN	Destination Burst Length Enable If this bit is set to 1, DMAC uses the value of CHx_CTL.AWLEN as AXI Burst length for destination data transfer till the extent possible; remaining transfers use maximum possible burst length. If this bit is set to 0, DMAC uses any possible value which is less than or equal to DMAX_CHx_MAX_AMBA_BURST_LENGTH as AXI Burst length for destination data transfer. Values: 0x1 (Enable): AXI Burst Length is CH1_CTL.AWLEN (till the extent possible) for Destination data transfers 0x0 (Disable): AXI Burst Length is any	R/W

		possible value <=DMAX_CH1_MAX_AMBA_BURST_LENGTH for Destination data transfers	
46:39	ARLEN	Source Burst Length AXI Burst length used for source data transfer. The specified burst length is used for source data transfer till the extent possible; remaining transfers use maximum possible value that is less than or equal to DMAX_CHx_MAX_AMBA_BURST_LENGTH. The maximum value of ARLEN is limited by DMAX_CHx_MAX_AMBA_BURST_LENGTH	R/W
38	ARLEN_EN	Source Burst Length Enable If this bit is set to 1, DMAC uses the value of CHx_CTL.ARLEN as AXI Burst length for source data transfer till the extent possible; remaining transfers use maximum possible burst length.If this bit is set to 0, DMAC uses any possible value that is less than or equal to DMAX_CHx_MAX_AMBA_BURST_LENGTH as AXI Burst length for source data transfer. Values: 0x1 (Enable): AXI Burst Length is CH1_CTL.ARLEN (till the extent possible) for Source data transfers 0x0 (Disable): AXI Burst Length is any possible value <= DMAX_CH1_MAX_AMBA_BURST_LENGTH for Source data transfers	R/W
37:35	AW_PROT	AXI aw_prot signal	R/W
34:32	AR_PROT	AXI ar_prot signal	R/W

31	RSVD_DMAC_CHx_CTL_31	DMAC Channelx Control Transfer Register bit31 Reserved bits	R
30	NonPosted_LastWrite_En	Non Posted Last Write Enable This bit decides whether posted writes can be used throughout the block transfer. 0: Posted writes may be used throughout the block transfer. 1: Posted writes may be used till the end of the block (inside a block) and the last write in the block must be non-posted. This is to synchronize block completion interrupt generation to the last write data reaching the end memory/peripheral. Values: 0x1 (Enable): Last write in the block must be non- posted 0x0 (Disable): Posted writes may be used throughout the block transfer	R/W
29:26	AW_CACHE	AXI aw_cache signal	R/W
25:22	AR_CACHE	AXI ar_cache signal	R/W

21:18	DST_MSIZE	Destination Burst Transaction Length. Number of data items, each of width CHx_CTL.DST_TR_WIDTH, to be written to the destination every time a destination burst transaction request is made from the corresponding hardware or software handshaking interface. NOTE: This Value is not related to the AXI awlen signal. Values: 0x0 (DATA_ITEM_1): 1 Data Item read from Destination in the burst transaction0x1 (DATA_ITEMS_4): 4 Data Item read from Destination in the burst transaction0x2 (DATA_ITEMS_8): 8 Data Item read from Destination in the burst transaction0x3 (DATA_ITEMS_16): 16 Data Item read from Destination in the burst transaction0x4 (DATA_ITEMS_32): 32 Data Item read from Destination in the burst transaction0x5 (DATA_ITEMS_64): 64 Data Item read from Destination in the burst transaction0x6 (DATA_ITEMS_128): 128 Data Item read from Destination in the burst transaction0x7 (DATA_ITEMS_256): 256 Data Item read from Destination in the burst transaction0x8 (DATA_ITEMS_512): 512 Data Item read from Destination in the burst transaction0x9 (DATA_ITEMS_1024): 1024 Data Item read from Destination in the burst transaction	R/W
		Source Burst Transaction Length. Number of data items, each of width CHx_CTL.SRC_TR_WIDTH, to be	

17:14	SRC_MSIZE	read from the source every time a source burst transaction request is made from the corresponding hardware or software handshaking interface. The maximum value of DST_MSIZE is limited by DMAX_CHx_MAX_MULT_SIZE. NOTE: This Value is not related to the AXI arlen signal. Values: 0x0 (DATA_ITEM_1): 1 Data Item read from Source in the burst transaction 0x1 (DATA_ITEMS_4): 4 Data Item read from Source in the burs transaction 0x2 (DATA_ITEMS_8): 8 Data Item read from Source in the burst transaction 0x3 (DATA_ITEMS_16): 16 Data Item read from Source in the burst transaction 0x4 (DATA_ITEMS_32): 32 Data Item read from Source in the burst transaction 0x5 (DATA_ITEMS_64): 64 Data Item read from Source in the burst transaction 0x6 (DATA_ITEMS_128): 128 Data Item read from Source in the burst transaction 0x7 (DATA_ITEMS_256): 256 Data Item read from Source in the burst transaction 0x8 (DATA_ITEMS_512): 512 Data Item read from Source in the burst transaction 0x9 (DATA_ITEMS_1024): 1024 Data Item read from Source in the burst transaction	R/W

13:11	DST_TR_WIDTH	Destination Transfer Width. Mapped to AXI bus awsize, this value must be less than or equal to DMAX_M_DATA_WIDTH. Values: 0x0 (BITS_8): Destination Transfer Width is 8 bits 0x1 (BITS_16): Destination Transfer Width is 16 bits 0x2 (BITS_32): Destination Transfer Width is 32 bits 0x3 (BITS_64): Destination Transfer Width is 64 bits 0x4 (BITS_128): Destination Transfer Width is 128 bits 0x5 (BITS_256): Destination Transfer Width is 256 bits 0x6 (BITS_512): Destination Transfer Width is 512 bits	R/W
10:8	SRC_TR_WIDTH	Source Transfer Width. Mapped to AXI bus arsize, this value must be less than or equal to DMAX_M_DATA_WIDTH. Values: 0x0 (BITS_8): Source Transfer Width is 8 bits 0x1 (BITS_16): Source Transfer Width is 16 bits 0x2 (BITS_32): Source Transfer Width is 32 bits 0x3 (BITS_64): Source Transfer Width is 64 bits 0x4 (BITS_128): Source Transfer Width is 128 bits 0x5 (BITS_256): Source Transfer Width is 256 bits 0x6 (BITS_512): Source Transfer Width is 512 bits	R/W
		Source Burst Transaction Length. Number of data items, each of width CHx_CTL.SRC_TR_WIDTH, to be read from the source every time a source burst transaction request is made from the corresponding hardware or software handshaking interface. The maximum value of DST_MSIZE is	

17:14	SRC_MSIZE	limited by DMAX_CHx_MAX_MULT_SIZE. NOTE: This Value is not related to the AXI arlen signal. Values: 0x0 (DATA_ITEM_1): 1 Data Item read from Source in the burst transaction 0x1 (DATA_ITEMS_4): 4 Data Item read from Source in the burs transaction 0x2 (DATA_ITEMS_8): 8 Data Item read from Source in the burst transaction 0x3 (DATA_ITEMS_16): 16 Data Item read from Source in the burst transaction 0x4 (DATA_ITEMS_32): 32 Data Item read from Source in the burst transaction 0x5 (DATA_ITEMS_64): 64 Data Item read from Source in the burst transaction 0x6 (DATA_ITEMS_128): 128 Data Item read from Source in the burst transaction 0x7 (DATA_ITEMS_256): 256 Data Item read from Source in the burst transaction 0x8 (DATA_ITEMS_512): 512 Data Item read from Source in the burst transaction 0x9 (DATA_ITEMS_1024): 1024 Data Item read from Source in the burst transaction	R/W

13:11	DST_TR_WIDTH	Destination Transfer Width. Mapped to AXI bus awsize, this value must be less than or equal to DMAX_M_DATA_WIDTH. Values: 0x0 (BITS_8): Destination Transfer Width is 8 bits 0x1 (BITS_16): Destination Transfer Width is 16 bits 0x2 (BITS_32): Destination Transfer Width is 32 bits 0x3 (BITS_64): Destination Transfer Width is 64 bits 0x4 (BITS_128): Destination Transfer Width is 128 bits 0x5 (BITS_256): Destination Transfer Width is 256 bits 0x6 (BITS_512): Destination Transfer Width is 512 bits	R/W
10:8	SRC_TR_WIDTH	Source Transfer Width. Mapped to AXI bus arsize, this value must be less than or equal to DMAX_M_DATA_WIDTH. Values: 0x0 (BITS_8): Source Transfer Width is 8 bits 0x1 (BITS_16): Source Transfer Width is 16 bits 0x2 (BITS_32): Source Transfer Width is 32 bits 0x3 (BITS_64): Source Transfer Width is 64 bits 0x4 (BITS_128): Source Transfer Width is 128 bits 0x5 (BITS_256): Source Transfer Width is 256 bits 0x6 (BITS_512): Source Transfer Width is 512 bits	R/W
7	RSVD_DMAC_CHx_CTL_7	DMAC Channelx Control Transfer Register bit7 Reserved bits	R

6	DINC	Destination Address Increment. Indicates whether to increment the destination address on every destination transfer. If the device is writing data from a source peripheral FIFO with a fixed address, then set this field to No change . 0: Increment 1: No Change NOTE: Increment aligns the address to the next CHx_CTL.DST_TR_WIDTH boundary. Values: 0x1 (INCREMENTAL): Destination address incremented on every source transfer 0x0 (FIXED): Destination address is fixed	R/W
5	RSVD_DMAC_CHx_CTL_5	DMAC Channelx Control Transfer Register bit5 Reserved bits	R
4	SINC	Source Address Increment. Indicates whether to increment the source address on every source transfer. If the device is fetching data from a source peripheral FIFO with a fixed address, then set this field to No change . 0: Increment 1: No Change NOTE: Increment aligns the address to the next CHx_CTL.SRC_TR_WIDTH boundary. Values: 0x1 (INCREMENTAL): Source address incremented on every source transfer 0x0 (FIXED): Source address is fixed	R/W
3	RSVD_DMAC_CHx_CTL_3	DMAC Channelx Control Transfer Register bit3 Reserved bits	R

2	DMS	Destination Master Select. Identifies the Master Interface layer from which the destination device (peripheral or memory) is accessed. 0: AXI master 1 1: AXI Master 2 Values: 0x1 (MASTER2_INTF): Destination device on Master- 2 interface layer 0x0 (MASTER1_INTF): Destination device on Master- 1 interface layer	R
1	RSVD_DMAC_CHx_CTL_1	DMAC Channelx Control Transfer Register bit1 Reserved bits	R
0	SMS	Source Master Select. Identifies the Master Interface layer from which the source device (peripheral or memory) is accessed. 0: AXI master 1 1: AXI Master 2 Values: 0x1 (MASTER2_INTF): Source device on Master-2 interface layer 0x0 (MASTER1_INTF): Source device on Master-1 interface layer	R

3.17.4.15 CHn_CFG(0x100*n+20)

Bits	Name	Description	Memory Access
63	RSVD_DMAC_CHx_CFG_63	DMAC Channelx Transfer Configuration Register (63bit) Reserved bit	R
62:59	DST_OSR_LMT	Destination Outstanding Request LimitMaximum outstanding request supported is 16.Source Outstanding Request Limit = DST_OSR_LMT + 1	R/W
58:55	SRC_OSR_LMT	Source Outstanding Request LimitMaximum outstanding request supported is 16.Source Outstanding Request Limit = SRC_OSR_LMT + 1	R/W
54:53	LOCK_CH_L	Channel Lock LevelThis bit indicates the duration over which CHx_CFG.LOCK_CH bit applies. 00: Over complete DMA transfer 01: Over DMA block transfer1x: Reserved This field does not exist if the configuration parameter DMAX_CHx_LOCK_EN is set to False; in that case, the read-back value is always 0. Values: 0x0 (DMA_transfer_CH_LOCK): Duration of the Channel locking is for the entire DMA transfer 0x1 (BLOCK_TRANFER_CH_LOCK): Duration of the Channel locking is for the current block transfer	R
		Channel Lock bit When the channel is granted control of the master bus interface and if the CHx_CFG.LOCK_CH bit is asserted, then no other channels are granted control of the master bus interface for the duration specified in CHx_CFG.LOCK_CH_L. Indicates to	

52	LOCK_CH	the master bus interface arbiter that this channel wants exclusive access to the master bus interface for the duration specified in CHx_CFG.LOCK_CH_L. This field does not exist if the configuration parameter DMAX_CHx_LOCK_EN is set to False; in this case, the read-back value is always 0.Locking the channel locks AXI Read Address, Write Address and Write Data channels on the corresponding master interface. NOTE: Channel locking feature is supported only for memory-to-memory transfer at Block Transfer and DMA Transfer levels. Hardware does not check for the validity of channel locking setting, hence the software must take care of enabling the channel locking only for memory-to- memory transfers at Block Transfer or DMA Transfer levels. Illegal programming of channel locking might result in unpredictable behavior. Values: 0x0 (NO_CHANNEL_LOCK): Channel is not locked during the transfers 0x0 (CHANNEL_LOCK): Channel is locked and granted exclusive access to the Master Bus Interface	R
51:49	CH_PRIOR	Channel PriorityA priority of 7 is the highest priority, and 0 is the lowest. This field must be programmed within the following range: 0: DMAX_NUM_CHANNELS-1A programmed value outside this range will cause erroneous behavior.	R/W

48	RSVD_DMAC_CHx_CFG_48	DMAC Channelx Transfer Configuration Register (48bit) Reserved bit	R
47	RSVD_DMAC_CHx_CFG_47_44	DMAC Channelx Transfer Configuration Register (bits (LOG2_DMAX_NUM_HS_IF+44) to 47) Reserved bits	R
46:44	DST_PER	Assigns a hardware handshaking interface (0 - DMAX_NUM_HS_IF-1) to the destination of Channelx if the CHx_CFG.HS_SEL_DST field is 0; otherwise, this field is ignored. The channel can then communicate with the destination peripheral connected to that interface through the assigned hardware handshaking interface.Reset Value = 1 NOTE: For correct DMAC operation, only one peripheral (source or destination) should be assigned to the same handshaking interface. This field does not exist if the configuration parameter DMAX_NUM_HS_IF is set to 0.x = 44 if DMAC_NUM_HS_IF is 1x = ceil(log2(DMAC_NUM_HS_IF)) + 43 if DMAC_NUM_HS_IF is greater than 1Bits 47: (x+1) do not exist and return 0 on a read.	R/W
43	RSVD_DMAC_CHx_CFG_43	DMAC Channelx Transfer Configuration Register (43bit) Reserved bit	R
42	RSVD_DMAC_CHx_CFG_42_39	DMAC Channelx Transfer Configuration Register (bits (LOG2_DMAX_NUM_HS_IF+39) to 42) Reserved bits	R

41:39	SRC_PER	Assigns a hardware handshaking interface (0 - DMAX_NUM_HS_IF-1) to the source of Channelx if the CHx_CFG.HS_SEL_SRC field is 0; otherwise, this field is ignored. The channel can then communicate with the source peripheral connected to that interface through the assigned hardware handshaking interface.Reset Value = 1 NOTE: For correct DMAC operation, only one peripheral (source or destination) should be assigned to the same handshaking interface. This field does not exist if the configuration parameter DMAX_NUM_HS_IF is set to 0.x = 39 if DMAC_NUM_HS_IF is 1x = ceil(log2(DMAC_NUM_HS_IF) + 38 if DMAC_NUM_HS_IF is greater than 1.Bits 42: (x+1) do not exist and return 0 on a read.	R/W
38	DST_HWHS_POL	Destination Hardware Handshaking Interface Polarity. 0: ACTIVE HIGH 1: ACTIVE LOW Values: 0x0 (ACTIVE_HIGH): Polarity of the Handshaking Interface used for the Destination peripheral is Active High 0x1 (ACTIVE_LOW): Polarity of the Handshaking Interface used for the Destination peripheral is Active Low	R

37	SRC_HWHS_POL	Source Hardware Handshaking Interface Polarity. 0: ACTIVE HIGH 1: ACTIVE LOW Values: 0x0 (ACTIVE_HIGH): Polarity of the Handshaking Interface used for the Source peripheral is Active High 0x1 (ACTIVE_LOW): Polarity of the Handshaking Interface used for the Source peripheral is Active Low	R
36	HS_SEL_DST	Destination Software or Hardware Handshaking Select. This register selects which of the handshaking interfaces (hardware or software) is active for destination requests on this channel. 0: Hardware handshaking interface. Software-initiated transaction requests are ignored. 1: Software handshaking interface. Hardware-initiated transaction requests are ignored.If the destination peripheral is memory, then this bit is ignored. Values: 0x0 (HARDWARE_HS): Hardware Handshaking Interface is used for the Destination peripheral 0x1 (SOFTWARE_HS): Software Handshaking Interface is used for the Destination peripheral	R/W

35	HS_SEL_SRC	Source Software or Hardware Handshaking Select. This register selects which of the handshaking interfaces (hardware or software) is active for source requests on this channel. 0: Hardware handshaking interface. Software-initiated transaction requests are ignored. 1: Software handshaking interface. Hardware-initiated transaction requests are ignored.If the source peripheral is memory, then this bit is ignored. Values: 0x0 (HARDWARE_HS): Hardware Handshaking Interface is used for the Source peripheral 0x1 (SOFTWARE_HS): Software Handshaking Interface is used for the Source peripheral	R/W
		Transfer Type and Flow Control. The following transfer types are supported.Memory to MemoryMemory to PeripheralPeripheral to MemoryPeripheral to PeripheralFlow Control can be assigned to the DMAC, the source peripheral, or hte destination peripheral. Values: 0x0 (MEM_TO_MEM_DMAC): Transfer Type is memory to memory and Flow Controller is DMAC 0x1 (MEM_TO_PER_DMAC): Transfer Type is memory to peripheral and Flow Controller is DMAC	

34:32	TT_FC	0x2 (PER_TO_MEM_DMAC): Transfer Type is peripheral to memory and Flow Controller is DMAC 0x3 (PER_TO_PER_DMAC): Transfer Type is peripheral to peripheral and Flow Controller is DMAC 0x4 (PER_TO_MEM_SRC): Transfer Type is peripheral to Memory and Flow Controller is Source peripheral 0x5 (PER_TO_PER_SRC): Transfer Type is peripheral to peripheral and Flow Controller is Source peripheral 0x6 (MEM_TO_PER_DST): Transfer Type is memory to peripheral and Flow Controller is Destination peripheral 0x7 (PER_TO_PER_DST): Transfer Type is peripheral to peripheral and Flow Controller is Destination peripheral	R/W
31:4	RSVD_DMAC_CHx_CFG_4to31	DMAC Channelx Transfer Configuration Register (bits 4 to 31) Reserved bits	R

3:2	DST_MULTBLK_TYPE	Destination Multi Block Transfer Type. These bits define the type of multi-block transfer used for destination peripheral. 00: Contiguous 01: Reload 10: Shadow Register 11: Linked List If the type selected is Contiguous, the CHx_DAR register is loaded with the value of the end source address of previous block + 1 at the end of every block for multi-block transfers. A new block transfer is then initiated.If the type selected is Reload, the CHx_DAR register is reloaded from the initial value of DAR at the end of every block for multi-block transfers. A new block transfer is then initiated.If the type selected is Shadow Register, the CHx_DAR register is loaded from the content of its shadow register if CHx_CTL.ShadowReg_Or_LLI_Valid bit is set to 1 at the end of every block for multi-block transfers. A new block transfer is then initiated. If the type selected is Linked List, the CHx_DAR register is loaded from the Linked List if CTL.ShadowReg_Or_LLI_Valid bit is set to 1 at the end of every block for multi- block transfers. A new block transfer is then initiated.CHx_CTL and CHx_BLOCK_TS registers are loaded from their initial values or from the contents of their shadow registers (if	R/W

		CHx_CTL.ShadowReg_Or_LLI_Valid bit is set to 1) or from the linked list (if CTL.ShadowReg_Or_LLI_Valid bit is set to 1) at the end of every block for multi- block transfers based on the multi-block transfer type programmed for source and destination peripherals.Contiguous transfer on both source and destination peripheral is not a valid multi-block transfer configuration. This field does not exist if the configuration parameter DMAX_CHx_MULTI_BLK_EN is not selected; in that case, the read-back value is always 0. Values: 0x0 (CONTINGUOUS): Contiguous Multiblock Type used for Destination Transfer 0x1 (RELOAD): Reload Multiblock Type used for Destination Transfer 0x2 (SHADOW_REGISTER): Shadow Register based Multiblock Type used for Destination Transfer 0x3 (LINKED_LIST): Linked List based Multiblock Type used for Destination Transfer	
1:0	SRC_MULTBLK_TYPE	Source Multi Block Transfer Type. These bits define the type of multi-block transfer used for source peripheral. 00: Contiguous 01: Reload 10: Shadow Register 11: Linked List If the type selected is Contiguous,	R/W

		the CHx_SAR register is loaded with the value of the end source address of previous block + 1 at the end of every block for multi-block transfers. A new block transfer is then initiated.If the type selected is Reload, the CHx_SAR register is reloaded from the initial value of SAR at the end of every block for multi-block transfers. A new block transfer is then initiated.If the type selected is Shadow Register, the CHx_SAR register is loaded from the content of its shadow register if CHx_CTL.ShadowReg_Or_LLI_Valid bit is set to 1 at the end of every block for multi-block transfers. A new block transfer is then initiated.If the type selected is Linked List, the CHx_SAR register is loaded from the Linked List if CTL.ShadowReg_Or_LLI_Valid bit is set to 1 at the end of every block for multi- block transfers. A new block transfer is then initiated.CHx_CTL and CHx_BLOCK_TS registers are loaded from their initial values or from the contents of their shadow registers (if CHx_CTL.ShadowReg_Or_LLI_Valid bit is set to 1) or from the linked list (if CTL.ShadowReg_Or_LLI_Valid bit is set to 1) at the end of every block for multi- block transfers based on the multi-block transfer type programmed for source and

		destination peripherals. Contiguous transfer on both source and destination peripheral is not a valid multi-block transfer configuration.This field does not exist if the configuration parameter DMAX_CHx_MULTI_BLK_EN is not selected; in that case, the read-back value is always 0. Values: 0x0 (CONTINGUOUS): Contiguous Multiblock Type used for Source Transfer 0x1 (RELOAD): Reload Multiblock Type used for Source Transfer 0x2 (SHADOW_REGISTER): Shadow Register based Multiblock Type used for Source Transfer 0x3 (LINKED_LIST): Linked List based Multiblock Type used for Source Transfer	

3.17.4.16 CHn_LLP(0x100*n+28)

Bits	Name	Description	Memory Access
63:6	LOC	Starting Address Memory of LLI blockStarting Address In Memory of next LLI if block chaining is enabled. The six LSBs of the starting address are not stored because the address is assumed to be aligned to a 64-byte boundary.LLI access always uses the burst size (arsize/awsize) that is same as the data bus width and cannot be changed or programmed to anything other than this. Burst length (awlen/arlen) is chosen based on the data bus width so that the access does not cross one complete LLI structure of 64 bytes. DMAC will fetch the entire LLI (40 bytes) in one AXI burst if the burst length is not limited by other settings. Volatile: true	R/W
5:1	RSVD_DMAC_CHx_LLP_1to5	DMAC Channelx Linked List Pointer Register (bits 1to5) Reserved bits Volatile: true	R

0	LMS	LLI master SelectThis bit identifies the AXI layer/interface where the memory device that stores the next linked list item resides. 0: AXI Master 1 1: AXI Master 2 This field does not exist if the configuration parameter DMAX_CHx_LMS is not set to NO_HARDCODE.In this case, the read- back value is always the hardcoded value. The maximum value of this field that can be read back is DMAX_NUM_MASTER_IF-1 . Values: 0x0 (MASTER1_INTF): next Linked List item resides on AXI Master1 interface 0x1 (MASTER2_INTF): next Linked List item resides on AXI Master2 interface Volatile: true	R

3.17.4.17 CHn_STATUSREG(0x100*n+30)

Bits	Name	Description	Memory Access
63:47	RSVD_DMAC_CHx_STATUSREG_47to63	DMAC Channelx Status Register (bits 47to63) Reserved bits Volatile: true	R
		Data Left in FIFO. This bit indicates the total number of data left in DMAC channel FIFO after completing the current block transfer.The width of the data in channel FIFO is equal to CHx_CTL.SRC_TR_WIDTH . For normal block transfer	

46:32	DATA_LEFT_IN_FIFO	completion without errors, Data_Left_In_FIFO = 0.If any error occurs during the dma transfer, the block transfer might be terminated early and in such a case, Data_Left_In_FIFO indicates the data remaining in channel FIFO which could not be transferred to destination peripheral. This field is cleared to zero on enabling the channel. NOTE: If CHx_CTL.DST_TR_WIDTH > CHx_CTL.SRC_TR_WIDTH , there may be residual data left in the FIFO which is not enough to form one CHx_CTL.SRC_TR_WIDTH of data and Data_Left_In_FIFO will return 0 in this case. Volatile: true	R
31:22	RSVD_DMAC_CHx_STATUSREG_22to31	DMAC Channelx Status Register (bits 22to31) Reserved bits Volatile: true	R
		Completed Block Transfer Size. This bit indicates the total number of data of width CHx_CTL.SRC_TR_WIDTH transferred for the previous block transfer.For normal block transfer completion without any errors, this value	

21:0	CMPLTD_BLK_TFR_SIZE	will be equal to the value programmed in BLOCK_TS field of CHx_BLOCK_TS register. If any error occurs during the dma transfer, the block transfer might be terminated early and in such a case, this value indicates the actual data transferred without error in the current block.This field is cleared to zero on enabling the channel. Volatile: true	R

3.17.4.18 CHn_SWHSSRCREG(0x100*n+38)

Bits	Name	Description	Memory Access
63:6	RSVD	Reserved bits	R
5	SWHS_LST_SRC_WE	Write Enable bit for Software Handshake Last Request for Channel Source. Values: 0x1 (ENABLE_SWHS_LAST_SRC): Enables write to the SWHS_LAST_SRC bit 0x0 (DISABLE_SWHS_LAST_SRC): Disables write to the SWHS_LAST_SRC bit Volatile: true	W
		Software Handshake Last Request for Channel Source. This bit is used to request LAST dma source data transfer if software handshaking method is selected for the source of the corresponding channel. This bit is ignored if software handshaking is not enabled for the source of the Channelx or if the source of Channelx is not the flow controller.CHx_SWHSSrcReg.SWHS_Req_Src bit	

4	SWHS_LST_SRC	must be set to 1 for DMAC to treat it as a valid software handshaking request. If CHx_SWHSSrcReg.SWHS_SglReq_Src is set to 1, the LAST request is for SINGLE dma transaction (AXI burst length = 1), else the request is treated as a BURST transaction request.DMAC clears this bit to 0 once software reads CHx_SWHSSrcReg.SWHS_Ack_Src bit and sees it as 1. Software can only set this bit to 1; it is not allowed to clear this bit to 0; only DMAC can clear this bit. NOTE: SWHS_Lst_Src bit is written only if the corresponding write enable bit, SWHS_Lst_Src_WE is asserted on the same register write operation and if the Channelx is enabled in the DMAC_ChEnReg register. This allows software to set a bit in the CHx_SWHSSrcReg register without performing a read-modified write operation. Values: 0x1 (ACTIVE_SWHS_LAST_SRC): Source peripheral indication to dmac that the current transfer is the last transfer 0x0 (INACTIVE_SWHS_LAST_SRC): Source peripheral indication that the curent transfer is not the last transfer Volatile: true	R/W
3	SWHS_SGLREQ_SRC_WE	Write Enable bit for Software Handshake Single Request for Channel Source. Values: 0x1 (ENABLE_SWHS_SGLREQ_SRC): Enables write to the SWHS_SGLREQ_SRC bit 0x0 (DISABLE_SWHS_SGLREQ_SRC): Disables	W

		write to the SWHS_SGLREQ_SRC bit Volatile: true	
2	SWHS_SGLREQ_SRC	Software Handshake Single Request for Channel Source. This bit is used to request SINGLE (AXI burst length = 1) dma source data transfer if software handshaking method is selected for the source of the corresponding channel. This bit is ignored if software handshaking is not enabled for the source of the Channelx. The functionality of this field depends on whether the peripheral is the flow controller.DMAC clears this bit to 0 once software reads CHx_SWHSSrcReg.SWHS_Ack_Src bit and sees it as 1. Software can only set this bit to 1; it is not allowed to clear this bit to 0; only DMAC can clear this bit. NOTE: SWHS_SglReq_Src bit is written only if the corresponding write enable bit, SWHS_SglReq_Src_WE is asserted on the same register write operation and if the Channelx is enabled in the DMAC_ChEnReg register. This allows software to set a bit in the CHx_SWHSSrcReg register without performing a read-modified write operation. Values: 0x1 (ACTIVE_SWHS_SGLREQ_SRC): Source peripheral request for a single dma transfer 0x0 (INACTIVE_SWHS_SGLREQ_SRC): Source	R/W

		peripheral is not requesting for a single transfer Volatile: true	
1	SWHS_REQ_SRC_WE	Write Enable bit for Software Handshake Request for Channel Source. NOTE: This bit always returns 0 on a read back. Values: 0x1 (ENABLE_SWHS_REQ_SRC): Enables write to the SWHS_REQ_SRC bit0x0 (DISABLE_SWHS_REQ_SRC): Disables write to the SWHS_REQ_SRC bit Volatile: true	W
0	SWHS_REQ_SRC	Software Handshake Request for Channel Source. This bit is used to request dma source data transfer if software handshaking method is selected for the source of the corresponding channel.This bit is ignored if software handshaking is not enabled for the source of the Channelx. The functionality of this field depends on whether the peripheral is the flow controller or not.DMAC clears this bit to 0 once software reads CHx_SWHSSrcReg.SWHS_Ack_Src bit and sees it as 1. Software can only set this bit to 1; it is not allowed to clear this bit to 0; only DMAC can clear this bit. NOTE: SWHS_Req_Src bit is written only if the corresponding write enable bit, SWHS_Req_Src_WE is asserted on the same register write operation and if the Channelx is enabled in the DMAC_ChEnReg register. This allows software to set a bit in the CHx_SWHSSrcReg register without performing a read-modified write operation. Values: 0x1 (ACTIVE_SWHS_REQ_SRC): Source peripheral request for a dma transfer0x0	R/W

		(INACTIVE_SWHS_REQ_SRC): Source peripheral is not requesting for a burst transfer Volatile: true	

3.17.4.19 CHn_SWHSDSTREG(0x100*n+40)

Bits	Name	Description	Memory Access
63:6	RSVD	Reserved bits	R
5	SWHS_LST_DST_WE	Write Enable bit for Software Handshake Last Request for Channel Destination. NOTE: This bit always returns 0 on a read back. Values: 0x1 (ENABLE_SWHS_LAST_DST): Enables write to the SWHS_LAST_DST bit 0x0 (DISABLE_SWHS_LAST_DST): Disables write to the SWHS_LAST_DST bit Volatile: true	W
		Software Handshake Last Request for Channel Destination. This bit is used to request LAST dma destination data transfer if software handshaking method is selected for the destination of the corresponding channel. This bit is ignored if software handshaking is not enabled for the destination of the Channelx or if the destination of Channelx is not the fl ow controller. CHx_SWHSDstReg.SWHS_Req_Dst bit must be set to 1 for DMAC to treat it as a valid software handshaking request. If CHx_SWHSDstReg.SWHS_SglReq_Dst is set to 1, the LAST request is for SINGLE dma transaction (AXI burst length = 1), else the request is treated as a BURST transaction request.DMAC clears this bit to 0 once software reads CHx_SWHSDstReg.SWHS_Ack_Dst bit	

4	SWHS_LST_DST	and sets it as 1. Software can only set this bit to 1; it is not allowed to clear this bit to 0; only DMAC can clear this bit. NOTE: SWHS_Lst_Src bit is written only if the corresponding write enable bit, SWHS_Lst_Src_WE is asserted on the same register write operation and if the Channelx is enabled in the DMAC_ChEnReg register. This allows software to set a bit in the CHx_SWHSDstReg register without performing a read-modi fi ed write operation. Values: 0x1 (ACTIVE_SWHS_LAST_DST): Destination peripheral indication to dmac that the current transfer is the last transfer 0x0 (INACTIVE_SWHS_LAST_DST): Destination peripheral indication that the current transfer is not the last transfer Volatile: true	R/W
3	SWHS_SGLREQ_DST_WE	Write Enable bit for Software Handshake Single Request for Channel Destination. NOTE: This bit always returns 0 on a read block. Values: 0x1 (ENABLE_SWHS_SGLREQ_DST): Enables write to the SWHS_SGLREQ_DST bit 0x0 (DISABLE_SWHS_SGLREQ_DST): Disables write to the SWHS_SGLREQ_DST bit Volatile: true	W

2	SWHS_SGLREQ_DST	Software Handshake Single Request for Channel Destination.This bit is used to request SINGLE (AXI burst length = 1) dma destination data transfer if software handshaking method is selected for the destination of the corresponding channel.This bit is ignored if software handshaking is not enabled for the destination of the Channelx. The functionality of this field depends on whether the peripheral is the flow controller.DMAC clears this bit to 0 once software reads CHx_SWHSDstReg.SWHS_Ack_Dst bit and sees it as 1. Software can only set this bit to 1; it is not allowed to clear this bit to 0; only DMAC can clear this bit. NOTE: SWHS_SglReq_Dst bit is written only if the corresponding write enable bit, SWHS_SglReq_Dst_WE is asserted on the same register write operation and if the Channelx is enabled in the DMAC_ChEnReg register. This allows software to set a bit in the CHx_SWHSDstReg register without performing a read-modified write operation. Values: 0x1 (ACTIVE_SWHS_SGLREQ_DST): Destination peripheral request for a single dma transfer 0x0 (INACTIVE_SWHS_SGLREQ_DST): Destination peripheral is not requesting for a single transfer Volatile: true	R/W

1	SWHS_REQ_DST_WE	Write Enable bit for Software Handshake Request for Channel Destination. NOTE: This bit always returns 0 on a read block. Values: 0x1 (ENABLE_SWHS_REQ_DST): Enables write to the SWHS_REQ_DST bit 0x0 (DISABLE_SWHS_REQ_DST): Disables write to the SWHS_REQ_DST bit Volatile: true	W
0	SWHS_REQ_DST	Software Handshake Request for Channel Destination. This bit is used to request dma destination data transfer if software handshaking method is selected for the destination of the corresponding channel.This bit is ignored if software handshaking is not enabled for the source of the Channelx. The functionality of this field depends on whether the peripheral is the flow controller.DMAC clears this bit to 0 once software reads CHx_SWHSDstReg.SWHS_Ack_Dst bit and sees it as 1. Software can only set this bit to 1; it is not allowed to clear this bit to 0; only DMAC can clear this bit. NOTE: SWHS_Req_Dst bit is written only if the corresponding write enable bit, SWHS_Req_Dst_WE is asserted on the same register write operation and if the Channelx is enabled in the DMAC_ChEnReg register. This allows software to set a bit in the CHx_SWHSDstReg register without performing a read-modified write operation. Values:	R/W

		0x1 (ACTIVE_SWHS_REQ_DST): Destination peripheral request for a dma transfer 0x0 (INACTIVE_SWHS_REQ_DST): Destination peripheral is not requesting for a burst transfer Volatile: true	

3.17.4.20 CHn_BLK_TFR_RESUMEREQREG(0x100*n+48)

Bits	Name	Description	Memory Access
63:1	RSVD	Reserved bits	W
0	BLK_TFR_RESUMEREQ	Block Transfer Resume Request during Linked-List or Shadow-Register-based multi-block transfer. Values: 0x1 (ACTIVE_BLK_TFR_RESUMEREQ): Request for resuming the block transfer0x1 (INACTIVE_BLK_TFR_RESUMEREQ): No request to resume the block transfer	W

3.17.4.21 CHn_AXI_IDREG(0x100*n+50)

Bits	Name	Description	Memory Access
63:32	RSVD_32to63	Reserved bits	R
31:17	RSVD_17to31	Reserved bits	R
16	AXI_WRITE_ID_SUFFIX	AXI Write ID Suffix. These bits form part of the AWID output of AXI3/AXI4 master interface.IDW = DMAX_M_ID_WIDTHL2NC = log2(DMAX_NUM_CHANNELS)The upper L2NC+1 bits of awidN is derived from the channel number which is currently accessing the master interface. This varies for LLI fetch and source data transfer.For source data transfer, awidN for channel1 4 b0000, awidN for channel8 4 b0111 and so on. For LLI fetch access, awidN for channel1 4 b1000, awidN for channel8 4 b1111 and so on.Lower bits are same as the value programmed in CHx_AXI_IDReg.AXI_Write_ID_Suffix filed.	R/W

15:1	RSVD_1to31	Reserved bits	R
0	AXI_READ_ID_SUFFIX	AXI Read ID SuffixThese bits form part of the ARID output of AXI3/AXI4 master interface.IDW = DMAX_M_ID_WIDTHL2NC = log2(DMAX_NUM_CHANNELS)The upper L2NC+1 bits of aridN is derived from the channel number which is currently accessing the master interface.This varies for LLI fetch and source data transfer. For source data transfer, aridN for channel1 4 b0000, aridN for channel8 4 b0111 and so on.For LLI fetch access, aridN for channel1 4 b1000, aridN for channel8 4 b1111 and so on. Lower bits are same as the value programmed in CHx_AXI_IDReg.AXI_Read_ID_Suffix filed.	R/W

3.17.4.22 CHn_AXI_QOSREG(0x100*n+58)

Bits	Name	Description	Memory Access
63:8	RSVD_DMAC_CHx_AXI_QOSREG_8to63	DMAC Channelx AXI QOS Register (bits 8to63) Reserved bits	R
7:4	AXI_ARQOS	AXI ARQOS.These bits form the arqos output of AXI4 master interface.	R
3:0	AXI_AWQOS	AXI AWQOS.These bits form the awqos output of AXI4 master interface.	R

3.17.4.23 CHn_INTSTATUS_ENABLEREG(0x100*n+80)

Bits	Name	Description	Memory Access
63:32	RSVD_32to63	Reserved bits	R
31	Enable_CH_ABORTED_IntStat	Channel Aborted Status Enable.0: Disable the generation of Channel Aborted Interrupt in CHx_INTSTATUSREG1: Enable the generation of Channel Aborted Interrupt in CHx_INTSTATUSREG Values: 0x1 (ENABLE_CH_ABORTED): Enable the generation of Channel Aborted Interrupt in CH1_INTSTATUSREG 0x0 (DISABLE_CH_ABORTED): Disable the generation of Channel Aborted Interrupt in CH1_INTSTATUSREG	R/W
30	Enable_CH_DISABLED_IntStat	Channel Disabled Status Enable.0: Disable the generation of Channel Disabled Interrupt in CHx_INTSTATUSREG1: Enable the generation of Channel Disabled Interrupt in CHx_INTSTATUSREG Values: 0x1 (ENABLE_CH_DISABLED): Enable the generation of Channel Disabled Interrupt in CH1_INTSTATUSREG 0x0 (DISABLE_CH_DISABLED): Disable the generation of Channel Disabled Interrupt in CH1_INTSTATUSREG	R/W
29	Enable_CH_SUSPENDED_IntStat	Channel Suspended Status Enable. 0: Disable the generation of Channel Suspended Interrupt in CHx_INTSTATUSREG 1: Enable the generation of Channel Suspended Interrupt in CHx_INTSTATUSREG Values: 0x1 (ENABLE_CH_SUSPENDED): Enable the generation of Channel Suspended Interrupt in CH1_INTSTATUSREG 0x0 (DISABLE_CH_SUSPENDED): Disable the	R/W

		generation of Channel Suspended Interrupt in CH1_INTSTATUSREG	
28	Enable_CH_SRC_SUSPENDED_IntStat	Channel Source Suspended Status Enable. 0: Disable the generation of Channel Source Suspended Interrupt in CHx_INTSTATUSREG 1: Enable the generation of Channel Source Suspended Interrupt in CHx_INTSTATUSREG Values: 0x1 (ENABLE_CH_SRC_SUSPENDED): Enable the generation of Channel Source Suspended Interrupt in CH1_INTSTATUSREG 0x0 (DISABLE_CH_SRC_SUSPENDED): Disable the generation of Channel Source Suspended Interrupt in CH1_INTSTATUSREG	R/W
27	Enable_CH_LOCK_CLEARED_IntStat	Channel Lock Cleared Status Enable. 0: Disable the generation of Channel LOCK CLEARED Interrupt in CHx_INTSTATUSREG 1: Enable the generation of Channel LOCK CLEARED Interrupt in CHx_INTSTATUSREG Values: 0x1 (ENABLE_CH_LOCK_CLEARED): Enable the generation of Channel LOCK CLEARED Interrupt in CH1_INTSTATUSREG 0x0 (DISABLE_CH_LOCK_CLEARED): Disable the generation of Channel LOCK CLEARED Interrupt in CH1_INTSTATUSREG	R/W
26:22	RSVD_22to26	Reserved bits	R
21	Enable_SLVIF_WRONHOLD_ERR_IntStat	Slave Interface Write On Hold Error Status Enable. 0: Disable the generation of Slave Interface Write On Hold Error Interrupt in CHx_INTSTATUSREG 1: Enable the generation of Slave Interface Write On Hold Error Interrupt in CHx_INTSTATUSREG Values: 0x1 (ENABLE_SLVIF_WRONHOLD_ERR): Enable the generation of Slave Interface Write On Hold Error Interrupt in	R/W

		CH1_INTSTATUSREG 0x0 (DISABLE_SLVIF_WRONHOLD_ERR): Disable the generation of Slave Interface Write On Hold Error Interrupt in CH1_INTSTATUSREG	
20	Enable_SLVIF_SHADOWREG_WRON_VALID_ERR_IntStat	Shadow Register Write On Valid Error Status Enable. 0: Disable the generation of Shadow Register Write On Valid Error Interrupt in CHx_INTSTATUSREG 1: Enable the generation of Shadow register Write On Valid Error Interrupt in CHx_INTSTATUSREG Values: 0x1 (ENABLE_SLVIF_SHADOWREG_WRON_VALID_ERR): Enable the generation of Shadow register Write On Valid Error Interrupt in CH1_INTSTATUSREG 0x0 (DISABLE_SLVIF_SHADOWREG_WRON_VALID_ERR): Disable the generation of Shadow Register Write On Valid Error Interrupt in CH1_INTSTATUSREG	R/W
19	Enable_SLVIF_WRONCHEN_ERR_IntStat	Slave Interface Write On Channel Enabled Error Status Enable. 0: Disable the generation of Slave Interface Write On Channel enabled Error Interrupt in CHx_INTSTATUSREG 1: Enable the generation of Slave Interface Write On Channel enabled Error Interrupt in CHx_INTSTATUSREG Values: 0x1 (ENABLE_SLVIF_WRONCHEN_ERR): Enable the generation of Slave Interface Write On Channel enabled Error Interrupt in CH1_INTSTATUSREG 0x0 (DISABLE_SLVIF_WRONCHEN_ERR): Disable the generation of Slave Interface Write On Channel enabled Error Interrupt in CH1_INTSTATUSREG	R/W

18	Enable_SLVIF_RD2RWO_ERR_IntStat	Slave Interface Read to write Only Error Status Enable. 0: Disable the generation of Slave Interface Read to Write only Error Interrupt in CHx_INTSTATUSREG 1: Enable the generation of Slave Interface Read to Write Only Error Interrupt in CHx_INTSTATUSREG Values: 0x1 (ENABLE_SLVIF_RD2RWO_ERR): Enable the generation of Slave Interface Read to Write Only Error Interrupt in CH1_INTSTATUSREG 0x0 (DISABLE_SLVIF_RD2RWO_ERR): Disable the generation of Slave Interface Read to Write only Error Interrupt in CH1_INTSTATUSREG	R/W
17	Enable_SLVIF_WR2RO_ERR_IntStat	Slave Interface Write to Read Only Error Status Enable. 0: Disable the generation of Slave Interface Write to Read only Error Interrupt in CHx_INTSTATUSREG 1: Enable the generation of Slave Interface Write to Read Only Error Interrupt in CHx_INTSTATUSREG Values: 0x1 (ENABLE_SLVIF_WR2RO_ERR): Enable the generation of Slave Interface Write to Read Only Error Interrupt in CH1_INTSTATUSREG 0x0 (DISABLE_SLVIF_WR2RO_ERR): Disable the generation of Slave Interface Write to Read only Error Interrupt in CH1_INTSTATUSREG	R/W
16	Enable_SLVIF_DEC_ERR_IntStat	Slave Interface Decode Error Status Enable. 0: Disable the generation of Slave Interface Decode Error Interrupt in CHx_INTSTATUSREG 1: Enable the generation of Slave Interface Decode Error Interrupt in CHx_INTSTATUSREG Values: 0x1 (ENABLE_SLVIF_DEC_ERR): Enable the generation of Slave Interface Decode Error Interrupt in CH1_INTSTATUSREG 0x0 (DISABLE_SLVIF_DEC_ERR): Disable the generation of Slave Interface Decode Error Interrupt in CH1_INTSTATUSREG	R/W

15	RSVD_15	Reserved bit	R
14	Enable_SLVIF_MULTIBLKTYPE_ERR_IntStat	Slave Interface Multi Block type Error Status Enable. 0: Disable the generation of Slave Interface Multi Block type Error Interrupt in CHx_INTSTATUSREG 1: Enable the generation of Slave Interface Multi Block type Error Interrupt in CHx_INTSTATUSREG Values: 0x1 (ENABLE_SLVIF_MULTIBLKTYPE_ERR): Enable the generation of Slave Interface Multi Block type Error Interrupt in CH1_INTSTATUSREG 0x0 (DISABLE_SLVIF_MULTIBLKTYPE_ERR): Disable the generation of Slave Interface Multi Block type Error Interrupt in CH1_INTSTATUSREG	R/W
13	Enable_SHADOWREG_OR_LLI_INVALID_ERR_IntStat	Shadow register or LLI Invalid Error Status Enable. 0: Disable the generation of Shadow Register or LLI Invalid Error Interrupt in CHx_INTSTATUSREG 1: Enable the generation of Shadow Register or LLI Invalid Error Interrupt in CHx_INTSTATUSREG Values: 0x1 (ENABLE_SHADOWREG_OR_LLI_INVALID_ERR): Disable the generation of Shadow Register or LLI Invalid Error Interrupt in CH1_INTSTATUSREG 0x0 (DISABLE_SHADOWREG_OR_LLI_INVALID_ERR): Enable the generation of Shadow Register or LLI Invalid Error Interrupt in CH1_INTSTATUSREG	R/W

12	Enable_LLI_WR_SLV_ERR_IntStat	LLI WRITE Slave Error Status Enable.0: Disable the generation of LLI WRITE Slave Error Interrupt in CHx_INTSTATUSREG1: Enable the generation of LLI WRITE Slave Error Interrupt in CHx_INTSTATUSREG Values: 0x1 (ENABLE_LLI_WR_SLV_ERR): Enable the generation of LLI WRITE Slave Error Interrupt in CH1_INTSTATUSREG0x0 (DISABLE_LLI_WR_SLV_ERR): Disable the generation of LLI WRITE Slave Error Interrupt in CH1_INTSTATUSREG	R/W
11	Enable_LLI_RD_SLV_ERR_IntStat	LLI Read Slave Error Status Enable. 0: Disable the generation of LLI Read Slave Error Interrupt in CHx_INTSTATUSREG 1: Enable the generation of LLI Read Slave Error Interrupt in CHx_INTSTATUSREG Values: 0x1 (ENABLE_LLI_RD_SLV_ERR): Enable the generation of LLI Read Slave Error Interrupt in CH1_INTSTATUSREG 0x0 (DISABLE_LLI_RD_SLV_ERR): Disable the generation of LLI Read Slave Error Interrupt in CH1_INTSTATUSREG	R/W
10	Enable_LLI_WR_DEC_ERR_IntStat	LLI WRITE Decode Error Status Enable. 0: Disable the generation of LLI WRITE Decode Error Interrupt in CHx_INTSTATUSREG 1: Enable the generation of LLI WRITE Decode Error Interrupt in CHx_INTSTATUSREG Values: 0x1 (ENABLE_LLI_WR_DEC_ERR): Disable the generation of LLI WRITE Decode Error Interrupt in CH1_INTSTATUSREG 0x0 (DISABLE_LLI_WR_DEC_ERR): Enable the generation of LLI WRITE Decode Error Interrupt in CH1_INTSTATUSREG	R/W

9	Enable_LLI_RD _DEC_ERR_IntStat	LLI Read Decode Error Status Enable. 0: Disable the generation of LLI Read Decode Error Interrupt in CHx_INTSTATUSREG 1: Enable the generation of LLI Read Decode Error Interrupt in CHx_INTSTATUSREG Values: 0x1 (ENABLE_LLI_RD_DEC_ERR): Enable the generation of LLI Read Decode Error Interrupt in CH1_INTSTATUSREG 0x0 (DISABLE_LLI_RD_DEC_ERR): Disable the generation of LLI Read Decode Error Interrupt in CH1_INTSTATUSREG	R/W
8	Enable_DST _SLV_ERR_IntStat	Destination Slave Error Status Enable. 0: Disable the generation of Destination Slave Error Interrupt in CHx_INTSTATUSREG 1: Enable the generation of Destination Slave Error Interrupt in CHx_INTSTATUSREG Values: 0x1 (ENABLE_DST_SLV_ERR): Enable the generation of Destination Slave Error Interrupt in CH1_INTSTATUSREG 0x0 (DISABLE_DST_SLV_ERR): Disable the generation of Destination Slave Error Interrupt in CH1_INTSTATUSREG	R/W
7	Enable_SRC _SLV_ERR_IntStat	Source Slave Error Status Enable. 0: Disable the generation of Source Slave Error Interrupt in CHx_INTSTATUSREG 1: Enable the generation of Source Slave Error Interrupt in CHx_INTSTATUSREG Values: 0x1 (ENABLE_SRC_SLV_ERR): Enable the generation of Source Slave Error Interrupt in CH1_INTSTATUSREG 0x0 (DISABLE_SRC_SLV_ERR): Disable the generation of Source Slave Error Interrupt in CH1_INTSTATUSREG	R/W

6	Enable_DST _DEC_ERR_IntStat	Destination Decode Error Status Enable. 0: Disable the generation of Destination Decode Error Interrupt in CHx_INTSTATUSREG 1: Enable the generation of Destination Decode Error Interrupt in CHx_INTSTATUSREG Values: 0x1 (ENABLE_DST_DEC_ERR): Enable the generation of Destination Decode Error Interrupt in CH1_INTSTATUSREG 0x0 (DISABLE_DST_DEC_ERR): Disable the generation of Destination Decode Error Interrupt in CH1_INTSTATUSREG	R/W
5	Enable_SRC _DEC_ERR_IntStat	Source Decode Error Status Enable. 0: Disable the generation of Source Decode Error Interrupt in CHx_INTSTATUSREG 1: Enable the generation of Source Decode Error Interrupt in CHx_INTSTATUSREG Values: 0x1 (ENABLE_SRC_DEC_ERR): Enable the generation of Source Decode Error Interrupt in CH1_INTSTATUSREG 0x0 (DISABLE_SRC_DEC_ERR): Disable the generation of Source Decode Error Interrupt in CH1_INTSTATUSREG	R/W
4	Enable_DST _TRANSCOMP_IntStat	Destination Transaction Completed Status Enable. 0: Disable the generation of Destination Transaction complete Interrupt in CHx_INTSTATUSREG 1: Enable the generation of Destination Transaction complete Interrupt in CHx_INTSTATUSREG Values: 0x1 (ENABLE_DST_TRANSCOMP): Enable the generation of Destination Transaction complete Interrupt in CH1_INTSTATUSREG 0x0 (DISABLE_DST_TRANSCOMP): Disable the generation of Destination Transaction complete Interrupt in CH1_INTSTATUSREG	R/W

3	Enable_SRC_TRANSCOMP_IntStat	Source Transaction Completed Status Enable. 0: Disable the generation of Source Transaction Complete Interrupt in CHx_INTSTATUSREG 1: Enable the generation of Source Transaction Complete Interrupt in CHx_INTSTATUSREG Values: 0x1 (ENABLE_SRC_TRANSCOMP): Enable the generation of Source Transaction Complete Interrupt in CH1_INTSTATUSREG 0x0 (DISABLE_SRC_TRANSCOMP): Disable the generation of Source Transaction Complete Interrupt in CH1_INTSTATUSREG	R/W
2	RSVD_2	Reserved bit	R
1	Enable_DMA_TFR_DONE_IntStat	DMA Transfer Done Interrupt Status Enable. 0: Disable the generation of DMA Transfer Done Interrupt in CHx_INTSTATUSREG 1: Enable the generation of DMA Transfer Done Interrupt in CHx_INTSTATUSREG Values: 0x1 (ENABLE_DMA_TFR_DONE): Enable the generation of DMA Transfer Done Interrupt in CH1_INTSTATUSREG 0x0 (DISABLE_DMA_TFR_DONE): Disable the generation of DMA Transfer Done Interrupt in CH1_INTSTATUSREG	R/W
0	Enable_BLOCK_TFR_DONE_IntStat	Block Transfer Done Interrupt Status Enable. 0: Disable the generation of Block Transfer Done Interrupt in CHx_INTSTATUSREG 1: Enable the generation of Block Transfer Done Interrupt in CHx_INTSTATUSREG Values: 0x1 (ENABLE_BLOCK_TFR_DONE): Enable the generation of Block Transfer Done Interrupt in CH1_INTSTATUSREG 0x0 (DISABLE_BLOCK_TFR_DONE): Disable the generation of Block Transfer Done Interrupt in CH1_INTSTATUSREG	R/W

3.17.4.24 CHn_INTSTATUS(0x100*n+88)

Bits	Name	Description	Memory Access
63:32	RSVD_32to63	Reserved bits	R
31	CH_ABORTED_IntStat	Channel Aborted. This indicates to the software that the corresponding channel in DMAC is aborted. 0: Channel is not aborted 1: Channel is aborted Error Interrupt is generated if the corresponding bit in CHx_INTSTATUS_ENABLEReg is enabled. This bit is cleared to 0 on writing 1 to the corresponding channel interrupt clear bit in CHx_IntClearReg register. Values: 0x1 (ACTIVE_CH_ABORTED): Channel is aborted 0x0 (INACTIVE_CH_ABORTED): Channel is not aborted Volatile: true	R
30	CH_DISABLED_IntStat	Channel Disabled. This indicates to the software that the corresponding channel in DMAC is disabled. 0: Channel is not disabled. 1: Channel is disabled. Error Interrupt is generated if the corresponding bit in CHx_INTSTATUS_ENABLEReg is enabled.This bit is cleared to 0 on writing 1 to the corresponding channel interrupt clear bit in CHx_IntClearReg register. Values: 0x1 (ACTIVE_CH_DISABLED): Channel is disabled 0x0 (INACTIVE_CH_DISABLED): Channel is not disabled Volatile: true	R

29	CH_SUSPENDED_IntStat	Channel Suspended. This indicates to the software that the corresponding channel in DMAC is suspended. 0: Channel is not suspended. 1: Channel is suspended. Error Interrupt is generated if the corresponding bit in CHx_INTSTATUS_ENABLEReg is enabled. This bit is cleared to 0 on writing 1 to the corresponding channel interrupt clear bit in CHx_IntClearReg register. Values: 0x1 (ACTIVE_CH_SUSPENDED): Channel is suspended 0x0 (INACTIVE_CH_SUSPENDED): Channel is not suspended Volatile: true	R
28	CH_SRC_SUSPENDED_IntStat	Channel Source Suspended. This indicates to the software that the corresponding channel source data transfer in DMAC is suspended. 0: Channel source is not suspended 1: Channel Source is suspended. Error Interrupt is generated if the corresponding bit in CHx_INTSTATUS_ENABLEReg is enabled. This bit is cleared to 0 on writing 1 to the corresponding channel interrupt clear bit in CHx_IntClearReg register. Values: 0x1 (ACTIVE_CH_SRC_SUSPENDED): Channel Source is suspended 0x0 (INACTIVE_CH_SRC_SUSPENDED): Channel source is not suspended Volatile: true	R

27	CH_LOCK_CLEARED_IntStat	Channel Lock Cleared. This indicates to the software that the locking of the corresponding channel in DMAC is cleared. 0: Channel locking is not cleared. 1: Channel locking is cleared. Channel locking is cleared by DMAC during the following situations:Channel locking is cleared and the channel locking settings in CHx_CFG register is reset if DMAC disables the channel upon request from software.Channel locking is cleared and the channel locking settings in CHx_CFG register is reset if DMAC disables the channel upon receiving error response on the master interface.This bit is cleared to 0 on enabling the channel. Values: 0x1 (ACTIVE_CH_LOCK_CLEARED): Channel Locking is cleared 0x0 (INACTIVE_CH_LOCK_CLEARED): Channel locking is not cleared, if present. Volatile: true	R
26:22	RSVD_22to26	Reserved bits	R
21	SLVIF_WRONHOLD_ERR_IntStat	Slave Interface Write On Hold Error. This error occurs if an illegal write operation is performed on a register; this happens if a write operation is performed on a channel register when DMAC is in Hold mode. 0: No Slave Interface Write On Hold Errors. 1: Slave Interface Write On Hold Error detected. Error Interrupt is generated if the corresponding bit in CHx_INTSTATUS_ENABLEReg is enabled. This bit is cleared to 0 on writing 1 to the corresponding channel interrupt clear bit in CHx_IntClearReg register. Values: 0x1 (ACTIVE_SLVIF_WRONHOLD_ERR): Slave Interface Write On Hold Error detected 0x0 (INACTIVE_SLVIF_WRONHOLD_ERR): No	R

		Slave Interface Write On Hold Errors Volatile: true	
20	SLVIF_SHADOWREG_WRON_VALID_ERR_IntStat	Shadow Register Write On Valid Error. This error occurs if shadow register based multi-block transfer is enabled and software tries to write to the shadow register when CHx_CTL.ShadowReg_Or_LLI_Valid bit is 1. 0: No Slave Interface Shadow Register Write On Valid Errors. 1: Slave Interface Shadow Register Write On Valid Error detected. Error Interrupt is generated if the corresponding bit in CHx_INTSTATUS_ENABLEReg is enabled. This bit is cleared to 0 on writing 1 to the corresponding channel interrupt clear bit in CHx_IntClearReg register. Values: 0x1 (ACTIVE_SLVIF_SHADOWREG_WRON_VALID_ERR): Slave Interface Shadow Register Write On Valid Error detected 0x0 (INACTIVE_SLVIF_SHADOWREG_WRON_VALID_ERR): No Slave Interface Shadow Register Write On Valid Errors Volatile: true	R

19	SLVIF_WRONCHEN_ERR_IntStat	Slave Interface Write On Channel Enabled Error. This error occurs if an illegal write operation is performed on a register; this happens if a write operation is performed on a register when the channel is enabled and if it is not allowed for the corresponding register as per the DMAC specification. 0: No Slave Interface Write On Channel Enabled Errors. 1: Slave Interface Write On Channel Enabled Error detected. Error Interrupt is generated if the corresponding bit in CHx_INTSTATUS_ENABLEReg is enabled. This bit is cleared to 0 on writing 1 to the corresponding channel interrupt clear bit in CHx_IntClearReg register. Values: 0x1 (ACTIVE_SLVIF_WRONCHEN_ERR): Slave Interface Write On Channel Enabled Error detected 0x0 (INACTIVE_SLVIF_WRONCHEN_ERR): No Slave Interface Write On Channel Enabled Errors Volatile: true	R
18	SLVIF_RD2RWO_ERR_IntStat	Slave Interface Read to write Only Error. This error occurs if read operation is performed to a Write Only register. 0: No Slave Interface Read to Write Only Errors. 1: Slave Interface Read to Write Only Error detected. Error Interrupt is generated if the corresponding bit in CHx_INTSTATUS_ENABLEReg is enabled. This bit is cleared to 0 on writing 1 to the corresponding channel interrupt clear bit in CHx_IntClearReg register. Values: 0x1 (ACTIVE_SLVIF_RD2RWO_ERR): Slave Interface Read to Write Only Error detected 0x0	R

		(INACTIVE_SLVIF_RD2RWO_ERR): No Slave Interface Read to Write Only Errors Volatile: true	
17	SLVIF_WR2RO_ERR_IntStat	Slave Interface Write to Read Only Error. This error occurs if write operation is performed to a Read Only register. 0: No Slave Interface Write to Read Only Errors. 1: Slave Interface Write to Read Only Error detected. Error Interrupt is generated if the corresponding bit in CHx_INTSTATUS_ENABLEReg is enabled. This bit is cleared to 0 on writing 1 to the corresponding channel interrupt clear bit in CHx_IntClearReg register. Values: 0x1 (ACTIVE_SLVIF_WR2RO_ERR): Slave Interface Write to Read Only Error detected 0x0 (INACTIVE_SLVIF_WR2RO_ERR): No Slave Interface Write to Read Only Errors Volatile: true	R
16	SLVIF_DEC_ERR_IntStat	Slave Interface Decode Error. Decode Error generated by DMAC during register access. This error occurs if the register access is to invalid address in Channelx register space resulting in error response by DMAC slave interface. 0: No Slave Interface Decode errors. 1: Slave Interface Decode Error detected. Error Interrupt is generated if the corresponding bit in CHxINTSTATUS_ENABLEReg is enabled. This bit is cleared to 0 on writing 1 to the corresponding channel interrupt clear bit in CHx_IntClearReg register. Values: 0x1 (ACTIVE_SLVIF_DEC_ERR): Slave Interface Decode Error detected 0x0 (INACTIVE_SLVIF_DEC_ERR):	R

		No Slave Interface Decode errors Volatile: true	
15	RSVD_15	Reserved bit	R
14	SLVIF_MULTIBLKTYPE_ERR_IntStat	Slave Interface Multi Block type Error. This error occurs if multi-block transfer type programmed in CHx_CFG register (SRC_MLTBLK_TYPE and DST_MLTBLK_TYPE) is invalid. This error condition causes the DMAC to halt the corresponding channel gracefully; Error Interrupt is generated if the corresponding channel error interrupt mask bit is set to 0 and the channel waits till software writes (any value) to CHx_BLK_TFR_ResumeReqReg to indicate valid multi- block transfer type availability. 0: No Multi-block transfer type Errors. 1: Multi-block transfer type Error detected. Error Interrupt is generated if the corresponding bit in CHx_INTSTATUS_ENABLEReg is enabled. This bit is cleared to 0 on writing 1 to the corresponding channel interrupt clear bit in CHx_IntClearReg register. Values: 0x1 (ACTIVE_SLVIF_MULTIBLKTYPE_ERR): Multi-block transfer type Error detected 0x0 (INACTIVE_SLVIF_MULTIBLKTYPE_ERR): No Multi-block transfer type Errors Volatile: true	R

13	SHADOWREG_OR_LLI_INVALID_ERR_IntStat	Shadow register or LLI Invalid Error. This error occurs if CHx_CTL.ShadowReg_Or_LLI_Valid bit is seen to be 0 during DMAC Shadow Register / LLI fetch phase. This error condition causes the DMAC to halt the corresponding channel gracefully; Error Interrupt is generated if the corresponding channel error interrupt mask bit is set to 0 and the channel waits till software writes (any value) to CHx_BLK_TFR_ResumeReqReg to indicate valid Shadow Register availability.In the case of LLI pre-fetching, ShadowReg_Or_LLI_Invalid_ERR Interrupt is not generated even if ShadowReg_Or_LLI_Valid bit is seen to be 0 for the pre- fetched LLI. In this case, DMAC re-attempts the LLI fetch operation after completing the current block transfer and generates ShadowReg_Or_LLI_Invalid_ERR Interrupt only if ShadowReg_Or_LLI_Valid bit is still seen to be 0. 0: No Shadow Register / LLI Invalid errors. 1: Shadow Register / LLI Invalid error detected. Error Interrupt is generated if the corresponding bit in CHx_INTSTATUS_ENABLEReg is enabled. This bit is cleared to 0 on writing 1 to the corresponding channel interrupt clear bit in CHx_IntClearReg register. Values: 0x1 (ACTIVE_SHADOWREG_OR_LLI_INVALID_ERR): Shadow Register / LLI Invalid error detected 0x0 (INACTIVE_SHADOWREG_OR_LLI_INVALID_ERR): No Shadow Register / LLI Invalid errors Volatile: true	R

12	LLI_WR_SLV_ERR_IntStat	LLI WRITE Slave Error. Slave Error detected by Master Interface during LLI write- back operation. This error occurs if the slave interface on which LLI resides issues a Slave Error. This error condition causes the DMAC to disable the corresponding channel gracefully; the DMAC_ChEnReg.CH_EN1 bit which received the error is set to 0. 0: No LLI write Slave Errors. 1: LLI Write SLAVE Error detected. Error Interrupt is generated if the corresponding bit in CHx_INTSTATUS_ENABLEReg is enabled. This bit is cleared to 0 on writing 1 to the corresponding channel interrupt clear bit in CHx_IntClearReg register. Values: 0x1 (ACTIVE_LLI_WR_SLV): LLI Write SLAVE Error detected 0x0 (INACTIVE_LLI_WR_SLV): No LLI write Slave Errors Volatile: true	R
11	LLI_RD_SLV_ERR_IntStat	LLI Read Slave Error. Slave Error detected by Master Interface during LLI read operation. This error occurs if the slave interface on which LLI resides issues a Slave Error. This error condition causes the DMAC to disable the corresponding channel gracefully; the DMAC_ChEnReg.CH_EN1 bit which received the error is set to 0. 0: No LLI Read Slave Errors. 1: LLI read Slave Error detected. Error Interrupt is generated if the corresponding bit in CHx_INTSTATUS_ENABLEReg is enabled. This bit is cleared to 0 on writing 1 to the corresponding channel interrupt clear bit in CHx_IntClearReg register. Values: 0x1 (ACTIVE_LLI_RD_SLV_ERR): LLI read Slave Error detected 0x0 (INACTIVE_LLI_RD_SLV_ERR): No LLI Read Slave Errors Volatile: true	R

10	LLI_WR_DEC_ERR_IntStat	LLI WRITE Decode Error. Decode Error detected by Master Interface during LLI write-back operation. This error occurs if the access is to invalid address and a Decode Error is returned from interconnect/slave. This error condition causes the DMAC to disable the corresponding channel gracefully; the DMAC_ChEnReg.CH_EN1 bit which received the error is set to 0. 0: NO LLI Write Decode Errors. 1: LLI write Decode Error detected. Error Interrupt is generated if the corresponding bit in CHx_INTSTATUS_ENABLEReg is enabled. This bit is cleared to 0 on writing 1 to the corresponding channel interrupt clear bit in CHx_IntClearReg register. Values: 0x1 (ACTIVE_LLI_WR_DEC_ERR): LLI write Decode Error detected 0x0 (INACTIVE_LLI_WR_DEC_ERR): NO LLI Write Decode Errors Volatile: true	R
9	LLI_RD_DEC_ERR_IntStat	LLI Read Decode Error. Decode Error detected by Master Interface during LLI read operation. This error occurs if the access is to invalid address and a Decode Error is returned from interconnect/slave. This error condition causes the DMAC to disable the corresponding channel gracefully; the DMAC_ChEnReg.CH_EN1 bit which received the error is set to 0. 0:NO LLI Read Decode Errors. 1: LLI Read Decode Error detected Error Interrupt is generated if the corresponding bit in CHx_INTSTATUS_ENABLEReg is enabled.This bit is cleared to 0 on writing 1 to the corresponding channel interrupt clear bit in CHx_IntClearReg register.	R

		Values: 0x1 (ACTIVELLI_RD_DEC_ERR_): LLI Read Decode Error detected 0x0 (INACTIVE_LLI_RD_DEC_ERR): NO LLI Read Decode Errors Volatile: true	
8	DST_SLV_ERR_IntStat	Destination Slave Error. Slave Error detected by Master Interface during destination data transfer. This error occurs if the slave interface to which the data is written issues a Slave Error. This error condition causes the DMAC to disable the corresponding channel gracefully; the DMAC_ChEnReg.CH_EN bit corresponding to the channel which received the error is set to 0. 0: No Destination Slave Errors 1: Destination Slave Errors Detected This bit is cleared to 0 on writing 1 to the corresponding channel interrupt clear bit in CHx_IntClearReg register. Values: 0x1 (ACTIVE_DST_SLV_ERR): Destination Slave Errors Detected 0x0 (INACTIVE_DST_SLV_ERR): No Destination Slave Errors Volatile: true	R
7	SRC_SLV_ERR_IntStat	Source Slave Error. Slave Error detected by Master Interface during source data transfer. This error occurs if the slave interface from which the data is read issues a Slave Error. This error condition causes the DMAC to disable the corresponding channel gracefully; the DMAC_ChEnReg.CH_EN bit corresponding to the channel which received the error is set to 0. 0:No Source Slave Errors 1: Source Slave Error Detected This bit is cleared to 0 on writing 1 to the corresponding channel interrupt clear bit in CHx_IntClearReg register. Values: 0x1 (ACTIVE_SRC_SLV_ERR): Source Slave Error Detected 0x0	R

		(INACTIVE_SRC_SLV_ERR): No Source Slave Errors Volatile: true	
6	DST_DEC_ERR_IntStat	Destination Decode Error. Decode Error detected by Master Interface during destination data transfer. This error occurs if the access is to invalid address and a Decode Error is returned from interconnect/slave. This error condition causes the DMAC to disable the corresponding channel gracefully; the DMAC_ChEnReg.CH_EN bit corresponding to the channel which received the error is set to 0. 0: No destination Decode Errors. 1: Destination Decode Error Detected This bit is cleared to 0 on writing 1 to the corresponding channel interrupt clear bit in CHx_IntClearReg register. Values: 0x1 (ACTIVE_DST_DEC_ERR): Destination Decode Error Detected 0x0 (INACTIVE_DST_DEC_ERR): No destination Decode Errors. Volatile: true	R
5	SRC_DEC_ERR_IntStat	Source Decode Error. Decode Error detected by Master Interface during source data transfer. This error occurs if the access is to invalid address and a Decode Error is returned from interconnect/slave. This error condition causes the DMAC to disable the corresponding channel gracefully; the DMAC_ChEnReg.CH_EN bit corresponding to the channel which received the error is set to 0. 0:No Source Decode Errors. 1: Source Decode Error detected. This bit is cleared to 0 on writing 1 to the corresponding channel interrupt clear bit in CHx_IntClearReg register. Values: 0x1 (ACTIVE_SRC_DEC_ERR): Source Decode Error detected 0x0 (INACTIVE_SRC_DEC_ERR): No Source Decode Errors Volatile: true	R

4	DST_TRANSCOMP_IntStat	Destination Transaction Completed. This bit is cleared to 0 on writing 1 to the corresponding channel interrupt clear bit in CHx_IntClearReg register or on enabling the channel (needed when interrupt is not enabled. Values: 0x1 (ACTIVE_DST_TRANSCOMP): Destination transaction is complete0x0 (INACTIVE_DST_TRANSCOMP): Destination transaction is not complete Volatile: true	R
3	SRC_TRANSCOMP_IntStat	Source Transaction Completed. This bit is cleared to 0 on writing 1 to the corresponding channel interrupt clear bit in CHx_IntClearReg register or on enabling the channel (needed when interrupt is not enabled. Values: 0x1 (ACTIVE_SRC_TRANSCOMP): Source transaction is complete0x0 (INACTIVE_SRC_TRANSCOMP): Source transation is not complete Volatile: true	R
2	RSVD_2	Reserved bit	R
1	DMA_TFR_DONE_IntStat	DMA Transfer Done. This indicates to the software that the DMAC has completed the requested DMA transfer.The DMAC sets this bit to 1 along with setting CHx_INTSTATUS.BLOCK_TFR_DONE bit to 1 when the last block transfer is completed. 0: DMA Transfer not completed. 1: DMA Transfer Completed This bit is cleared to 0 on writing 1 to the corresponding channel interrupt clear bit in CHx_IntClearReg register. Values: 0x1 (DMA_TFR_COMPLETED): DMA Transfer completed 0x0 (DMA_TFR_NOT_COMPLETE): DMA Transfer not complete Volatile: true	R

0	BLOCK_TFR_DONE_IntStat	Block Transfer Done. This indicates to the software that the DMAC has completed the requested block transfer.The DMAC sets this bit to 1 when the transfer is successfully completed. 0: Block Transfer not completed. 1: Block Transfer completed. This bit is cleared to 0 on writing 1 to the corresponding channel interrupt clear bit in CHx_IntClearReg register. Values: 0x1 (BLOCK_TFR_COMPLETED): Block Transfer completed 0x0 (BLOCK_TFR_NOT_COMPLETE): Block Transfer not complete Volatile: true	R

3.17.4.25 CHn_INTSIGNAL_ENABLEREG(0x100*n+90)

Bits	Name	Description	Memory Access
63:32	RSVD_32to63	Reserved bits	R
31	Enable_CH_ABORTED_IntSignal	Channel Aborted Signal Enable. 0: Disable the propagation of Channel Aborted Interrupt to generate a port level interrupt 1: Enable the propagation of Channel Aborted Interrupt to generate a port level interrupt Values: 0x1 (ENABLE_CH_ABORTED_IntSignal): Enable the propagation of Channel Aborted Interrupt to generate a port level interrupt 0x0 (DISABLE_CH_ABORTED_IntSignal): Disable the propagation of Channel Aborted Interrupt to generate a port level interrupt	R/W
30	Enable_CH_DISABLED_IntSignal	Channel Disabled Signal Enable. 0: Disable the propagation of Channel Disabled Interrupt to generate a port level interrupt 1: Enable the propagation of Channel Disabled Interrupt to generate a port level interrupt Values: 0x1 (ENABLE_CH_DISABLED_IntSignal): Enable the propagation of Channel Disabled Interrupt to generate a port level interrupt 0x0 (DISABLE_CH_DISABLED_IntSignal): Disable the propagation of Channel Disabled Interrupt to generate a port level interrupt	R/W
29	Enable_CH_SUSPENDED_IntSignal	Channel Suspended Signal Enable. 0: Disable the propagation of Channel Suspended Interrupt to generate a port level interrupt 1: Enable the propagation of Channel Suspended Interrupt to generate a port level interrupt Values: 0x1 (ENABLE_CH_SUSPENDED_IntSignal): Enable the propagation of Channel Suspended Interrupt to generate a port level interrupt 0x0 (DISABLE_CH_SUSPENDED_IntSignal): Disable the propagation of Channel Suspended Interrupt to	R/W

		generate a port level interrupt	
28	Enable_CH_SRC_SUSPENDED_IntSignal	Channel Source Suspended Signal Enable. 0: Disable the propagation of Channel Source Suspended Interrupt to generate a port level interrupt 1: Enable the propagation of Channel Source Suspended Interrupt to generate a port level interrupt Values: 0x1 (ENABLE_CH_SRC_SUSPENDED_IntSignal): Enable the propagation of Channel Source Suspended Interrupt to generate a port level interrupt 0x0 (DISABLE_CH_SRC_SUSPENDED_IntSignal): Disable the propagation of Channel Source Suspended Interrupt to generate a port level interrupt	R/W
27	Enable_CH_LOCK_CLEARED_IntSignal	Channel Lock Cleared Signal Enable. 0: Disable the propagation of Channel Lock Cleared Interrupt to generate a port level interrupt 1: Enable the propagation of Channel Lock Cleared Interrupt to generate a port level interrupt Values: 0x1 (ENABLE_CH_LOCK_CLEARED_IntSignal): Enable the propagation of Channel Lock Cleared Interrupt to generate a port level interrupt 0x0 (DISABLE_CH_LOCK_CLEARED_IntSignal): Disable the propagation of Channel Lock Cleared Interrupt to generate a port level interrupt	R/W
26:22	RSVD_22to26	Reserved bits	R

21	Enable_SLVIF_WRONHOLD_ERR_IntSignal	Slave Interface Write On Hold Error Signal Enable. 0: Disable the propagation of Slave Interface Write On Hold Error Interrupt to generate a port level interrupt 1: Enable the propagation of Slave Interface Write On Hold Error Interrupt to generate a port level interrupt Values: 0x1 (ENABLE_SLVIF_WRONHOLD_ERR_IntSignal): Enable the propagation of Slave Interface Write On Hold Error Interrupt to generate a port level interrupt 0x0 (DISABLE_SLVIF_WRONHOLD_ERR_IntSignal): Disable the propagation of Slave Interface Write On Hold Error Interrupt to generate a port level interrupt	R/W
20	Enable_SLVIF_SHADOWREG_WRON_VALID_ERR_IntSignal	Shadow Register Write On Valid Error Signal Enable. 0: Disable the propagation of Shadow Register Write On Valid Error Interrupt to generate a port level interrupt 1: Enable the propagation of Shadow register Write On Valid Error Interrupt to generate a port level interrupt Values: 0x1 (ENABLE_SLVIF_SHADOWREG_WRON_VALID_ERR_IntSignal): Enable the propagation of Shadow register Write On Valid Error Interrupt to generate a port level interrupt 0x0 (DISABLE_SLVIF_SHADOWREG_WRON_VALID_ERR_IntSignal): Disable the propagation of Shadow Register Write On Valid Error Interrupt to generate a port level interrupt	R/W

19	Enable_SLVIF_WRONCHEN_ERR_IntSignal	Slave Interface Write On Channel Enabled Error Signal Enable. 0: Disable the propagation of Slave Interface Write On Channel enabled Error Interrupt to generate a port level interrupt 1: Enable the propagation of Slave Interface Write On Channel enabled Error Interrupt to generate a port level interrupt Values: 0x1 (ENABLE_SLVIF_WRONCHEN_ERR_IntSignal): Enable the propagation of Slave Interface Write On Channel enabled Error Interrupt to generate a port level interrupt 0x0 (DISABLE_SLVIF_WRONCHEN_ERR_IntSignal): Disable the propagation of Slave Interface Write On Channel enabled Error Interrupt to generate a port level interrupt	R/W
18	Enable_SLVIF_RD2RWO_ERR_IntSignal	Slave Interface Read to write Only Error Signal Enable. 0: Disable the propagation of Slave Interface Read to Write only Error Interrupt to generate a port level interrupt 1: Enable the propagation of Slave Interface Read to Write Only Error Interrupt to generate a port level interrupt Values: 0x1 (ENABLE_SLVIF_RD2RWO_ERR_IntSignal): Enable the propagation of Slave Interface Read to Write Only Error Interrupt to generate a port level interrupt 0x0 (DISABLE_SLVIF_RD2RWO_ERR_IntSignal): Disable the propagation of Slave Interface Read to Write only Error Interrupt to generate a port level interrupt	R/W

17	Enable_SLVIF_WR2RO_ERR_IntSignal	Slave Interface Write to Read Only Error Signal Enable. 0: Disable the propagation of Slave Interface Write to Read only Error Interrupt to generate a port level interrupt 1: Enable the propagation of Slave Interface Write to Read Only Error Interrupt to generate a port level interrupt Values: 0x1 (ENABLE_SLVIF_WR2RO_ERR_IntSignal): Enable the propagation of Slave Interface Write to Read Only Error Interrupt to generate a port level interrupt 0x0 (DISABLE_SLVIF_WR2RO_ERR_IntSignal): Disable the propagation of Slave Interface Write to Read only Error Interrupt to generate a port level interrupt	R/W
16	Enable_SLVIF_DEC_ERR_IntSignal	Slave Interface Decode Error Signal Enable. 0: Disable the propagation of Slave Interface Decode Error Interrupt to generate a port level interrupt 1: Enable the propagation of Slave Interface Decode Error Interrupt to generate a port level interrupt Values: 0x1 (ENABLE_SLVIF_DEC_ERR_IntSignal): Enable the propagation of Slave Interface Decode Error Interrupt to generate a port level interrupt 0x0 (DISABLE_SLVIF_DEC_ERR_IntSignal): Disable the propagation of Slave Interface Decode Error Interrupt to generate a port level interrupt	R/W
15	RSVD_15	Reserved bit	R
14	Enable_SLVIF_MULTIBLKTYPE_ERR_IntSignal	Slave Interface Multi Block type Error Signal Enable. 0: Disable the propagation of Slave Interface Multi Block type Error Interrupt to generate a port level interrupt 1: Enable the propagation of Slave Interface Multi Block type Error Interrupt to generate a port level interrupt Values: 0x1 (ENABLE_SLVIF_MULTIBLKTYPE_ERR_IntSignal): Enable the propagation of Slave Interface Multi Block type Error Interrupt to generate a port level interrupt 0x0 (DISABLE_SLVIF_MULTIBLKTYPE_ERR_IntSignal): Disable the propagation of Slave Interface Multi Block	R/W

		type Error Interrupt to generate a port level interrupt	
13	Enable_SHADOWREG_OR_LLI_INVALID_ERR_IntSignal	Shadow register or LLI Invalid Error Signal Enable. 0: Disable the propagation of Shadow Register or LLI Invalid Error Interrupt to generate a port level interrupt 1: Enable the propagation of Shadow Register or LLI Invalid Error Interrupt to generate a port level interrupt Values: 0x1 (ENABLE_SHADOWREG_OR_LLI_INVALID_ERR_IntSignal): Enable the propagation of Shadow Register or LLI Invalid Error Interrupt to generate a port level interrupt 0x0 (DISABLE_SHADOWREG_OR_LLI_INVALID_ERR_IntSignal): Disable the propagation of Shadow Register or LLI Invalid Error Interrupt to generate a port level interrupt	R/W
12	Enable_LLI_WR_SLV_ERR_IntSignal	LLI WRITE Slave Error Signal Enable. 0: Disable the propagation of LLI WRITE Slave Error Interrupt to generate a port level interrupt 1: Enable the propagation of LLI WRITE Slave Error Interrupt to generate a port level interrupt Values: 0x1 (ENABLE_LLI_WR_SLV_ERR_IntSignal): Enable the propagation of LLI WRITE Slave Error Interrupt to generate a port level interrupt 0x0 (DISABLE_LLI_WR_SLV_ERR_IntSignal): Disable the propagation of LLI WRITE Slave Error Interrupt to generate a port level interrupt	R/W

11	Enable_LLI_RD_SLV_ERR_IntSignal	LLI Read Slave Error Signal Enable. 0: Disable the propagation of LLI Read Slave Error Interrupt to generate a port level interrupt 1: Enable the propagation of LLI Read Slave Error Interrupt to generate a port level interrupt Values: 0x1 (ENABLE_LLI_RD_SLV_ERR_IntSignal): Enable the propagation of LLI Read Slave Error Interrupt to generate a port level interrupt 0x0 (DISABLE_LLI_RD_SLV_ERR_IntSignal): Disable the propagation of LLI Read Slave Error Interrupt to generate a port level interrupt	R/W
10	Enable_LLI_WR_DEC_ERR_IntSignal	LLI WRITE Decode Error Signal Enable. 0: Disable the propagation of LLI WRITE Decode Error Interrupt to generate a port level interrupt 1: Enable the propagation of LLI WRITE Decode Error Interrupt to generate a port level interrupt Values: 0x1 (ENABLE_LLI_WR_DEC_ERR_IntSignal): Enable the propagation of LLI WRITE Decode Error Interrupt to generate a port level interrupt 0x0 (DISABLE_LLI_WR_DEC_ERR_IntSignal): Disable the propagation of LLI WRITE Decode Error Interrupt to generate a port level interrupt	R/W
9	Enable_LLI_RD_DEC_ERR_IntSignal	LLI Read Decode Error Signal Enable. 0: Disable the propagation of LLI Read Decode Error Interrupt to generate a port level interrupt 1: Enable the propagation of LLI Read Decode Error Interrupt to generate a port level interrupt Values: 0x1 (ENABLE_LLI_RD_DEC_ERR_IntSignal): Enable the propagation of LLI Read Decode Error Interrupt to generate a port level interrupt 0x0 (DISABLE_LLI_RD_DEC_ERR_IntSignal): Disable the propagation of LLI Read Decode	R/W

		Error Interrupt to generate a port level interrupt	
8	Enable_DST_SLV_ERR_IntSignal	Destination Slave Error Signal Enable. 0: Disable the propagation of Destination Slave Error Interrupt to generate a port level interrupt 1: Enable the propagation of Destination Slave Error Interrupt to generate a port level interrupt Values: 0x1 (ENABLE_DST_SLV_ERR_IntSignal): Enable the propagation of Destination Slave Error Interrupt to generate a port level interrupt 0x0 (DISABLE_DST_SLV_ERR_IntSignal): Disable the propagation of Destination Slave Error Interrupt to generate a port level interrupt	R/W
7	Enable_SRC_SLV_ERR_IntSignal	Source Slave Error Signal Enable. 0: Disable the propagation of Source Slave Error Interrupt to generate a port level interrupt 1: Enable the propagation of Source Slave Error Interrupt to generate a port level interrupt Values: 0x1 (ENABLE_SRC_SLV_ERR_IntSignal): Enable the propagation of Source Slave Error Interrupt to generate a port level interrupt 0x0 (DISABLE_SRC_SLV_ERR_IntSignal): Disable the propagation of Source Slave Error Interrupt to generate a port level interrupt	R/W
6	Enable_DST_DEC_ERR_IntSignal	Destination Decode Error Signal Enable. 0: Disable the propagation of Destination Decode Error Interrupt to generate a port level interrupt 1: Enable the propagation of Destination Decode Error Interrupt to generate a port level interrupt Values: 0x1 (ENABLE_DST_DEC_ERR_IntSignal): Enable the propagation of Destination Decode Error Interrupt to generate a port level interrupt 0x0 (DISABLE_DST_DEC_ERR_IntSignal): Disable the propagation of Destination Decode Error Interrupt to generate a port level interrupt	R/W

5	Enable_SRC_DEC_ERR_IntSignal	Source Decode Error Signal Enable. 0: Disable the propagation of Source Decode Error Interrupt to generate a port level interrupt 1: Enable the propagation of Source Decode Error Interrupt to generate a port level interrupt Values: 0x1 (ENABLE_SRC_DEC_ERR_IntSignal): Enable the propagation of Source Decode Error Interrupt to generate a port level interrupt 0x0 (DISABLE_SRC_DEC_ERR_IntSignal): Disable the propagation of Source Decode Error Interrupt to generate a port level interrupt	R/W
4	Enable_DST_TRANSCOMP_IntSignal	Destination Transaction Completed Signal Enable. 0: Disable the propagation of Destination Transaction complete Interrupt to generate a port level interrupt 1: Enable the propagation of Destination Transaction complete Interrupt to generate a port level interrupt Values: 0x1 (ENABLE_DST_TRANSCOMP_IntSignal): Enable the propagation of Destination Transaction complete Interrupt to generate a port level interrupt 0x0 (DISABLE_DST_TRANSCOMP_IntSignal): Disable the propagation of Destination Transaction complete Interrupt to generate a port level interrupt	R/W
3	Enable_SRC_TRANSCOMP_IntSignal	Source Transaction Completed Signal Enable. 0: Disable the propagation of Source Transaction Complete Interrupt to generate a port level interrupt 1: Enable the propagation of Source Transaction Complete Interrupt to generate a port level interrupt Values: 0x1 (ENABLE_SRC_TRANSCOMP_IntSignal): Enable the propagation of Source Transaction Complete Interrupt to generate a port level interrupt 0x0 (DISABLE_SRC_TRANSCOMP_IntSignal): Disable the propagation of Source Transaction Complete Interrupt to generate a port level interrupt	R/W
2	RSVD_2	Reserved bit	R

1	Enable_DMA_TFR_DONE_IntSignal	DMA Transfer Done Interrupt Signal Enable. 0: Disable the propagation of DMA Transfer Done Interrupt to generate a port level interrupt 1: Enable the propagation of DMA Transfer Done Interrupt to generate a port level interrupt Values: 0x1 (ENABLE_DMA_TFR_DONE_IntSignal): Enable the propagation of DMA Transfer Done Interrupt to generate a port level interrupt 0x0 (DISABLE_DMA_TFR_DONE_IntSignal): Disable the propagation of DMA Transfer Done Interrupt to generate a port level interrupt	R/W
0	Enable_BLOCK_TFR_DONE_IntSignal	Block Transfer Done Interrupt Signal Enable. 0: Disable the propagation of Block Transfer Done Interrupt to generate a port level interrupt 1: Enable the propagation of Block Transfer Done Interrupt to generate a port level interrupt Values: 0x1 (ENABLE_BLOCK_TFR_DONE_IntSignal): Enable the propagation of Block Transfer Done Interrupt to generate a port level interrupt 0x0 (DISABLE_BLOCK_TFR_DONE_IntSignal): Disable the propagation of Block Transfer Done Interrupt to generate a port level interrupt	R/W

3.17.4.26 CHn_INTCLEARREG(0x100*n+98)

Bits	Name	Description	Memory Access
63:32	RSVD_32to63	Reserved bit	W
31	Clear_CH _ABORTED_IntStat	Channel Aborted Interrupt Clear Bit. This bit is used to clear the corresponding channel interrupt status bit in CHx_INTSTATUSREG. Values: 0x1 (CLEAR_CH_ABORTED): Clear the CH_ABORTED interrupt in the Interrupt Status Register(CH1_IntStatusReg). 0x1 (NO_ACTION): Inactive signal. No action taken.	W
30	Clear_CH _DISABLED_IntStat	Channel Disabled Interrupt Clear Bit. This bit is used to clear the corresponding channel interrupt status bit in CHx_INTSTATUSREG. Values: 0x1 (CLEAR_CH_DISABLED): Clear the CH_DISABLED interrupt in the Interrupt Status Register(CH1_IntStatusReg). 0x1 (NO_ACTION): Inactive signal. No action taken.	W
29	Clear_CH _SUSPENDED_IntStat	Channel Suspended Interrupt Clear Bit. This bit is used to clear the corresponding channel interrupt status bit in CHx_INTSTATUSREG. Values: 0x1 (CLEAR_CH_SUSPENDED): Clear the CH_SUSPENDED interrupt in the Interrupt Status Register(CH1_IntStatusReg). 0x1 (NO_ACTION): Inactive signal. No action taken.	W

28	Clear_CH_SRC_SUSPENDED_IntStat	Channel Source Suspended Interrupt Clear Bit. This bit is used to clear the corresponding channel interrupt status bit in CHx_INTSTATUSREG. Values: 0x1 (CLEAR_CH_SRC_SUSPENDED): Clear the CH_SRC_SUSPENDED interrupt in the Interrupt Status Register(CH1_IntStatusReg). 0x1 (NO_ACTION): Inactive signal. No action taken.	W
27	Clear_CH_LOCK_CLEARED_IntStat	Channel Lock Cleared Interrupt Clear Bit. This bit is used to clear the corresponding channel interrupt status bit in CHx_INTSTATUSREG. Values: 0x1 (CLEAR_CH_LOCK_CLEARED): Clear the CH_LOCK_CLEARED interrupt in the Interrupt Status Register(CH1_IntStatusReg). 0x1 (NO_ACTION): Inactive signal. No action taken.	W
26:22	RSVD_22to26	DMAC Channelx Interrupt Clear Register (bits 22to26) Reserved bit	W
21	Clear_SLVIF_WRONHOLD_ERR_IntStat	Slave Interface Write On Hold Error Interrupt Clear Bit. This bit is used to clear the corresponding channel interrupt status bit in CHx_INTSTATUSREG. Values: 0x1 (CLEAR_SLVIF_WRONHOLD_ERR): Clear the SLVIF_WRONHOLD_ERR interrupt in the Interrupt Status Register(CH1_IntStatusReg). 0x1 (NO_ACTION): Inactive signal. No action taken.	W

20	Clear_SLVIF_SHADOWREG_WRON_VALID_ERR_IntStat	Shadow Register Write On Valid Error Interrupt Clear Bit. This bit is used to clear the corresponding channel interrupt status bit in CHx_INTSTATUSREG. Values: 0x1 (CLEAR_SLVIF_SHADOWREG_WRON_VALID_ERR): Clear the SLVIF_SHADOWREG_WRON_VALID_ERR interrupt in the Interrupt Status Register(CH1_IntStatusReg). 0x1 (NO_ACTION): Inactive signal. No action taken.	W
19	Clear_SLVIF_WRONCHEN_ERR_IntStat	Slave Interface Write On Channel Enabled Error Interrupt Clear Bit. This bit is used to clear the corresponding channel interrupt status bit in CHx_INTSTATUSREG. Values: 0x1 (CLEAR_SLVIF_WRONCHEN_ERR): Clear the SLVIF_WRONCHEN_ERR interrupt in the Interrupt Status Register(CH1_IntStatusReg). 0x1 (NO_ACTION): Inactive signal. No action taken.	W
18	Clear_SLVIF_RD2RWO_ERR_IntStat	Slave Interface Read to write Only Error Interrupt Clear Bit. This bit is used to clear the corresponding channel interrupt status bit in CHx_INTSTATUSREG. Values: 0x1 (CLEAR_SLVIF_RD2RWO_ERR): Clear the SLVIF_RD2RWO_ERR interrupt in the Interrupt Status Register(CH1_IntStatusReg). 0x1 (NO_ACTION): Inactive signal. No action taken.	W

17	Clear_SLVIF_WR2RO_ERR_IntStat	Slave Interface Write to Read Only Error Interrupt Clear Bit. This bit is used to clear the corresponding channel interrupt status bit in CHx_INTSTATUSREG. Values: 0x1 (CLEAR_SLVIF_WR2RO_ERR): Clear the SLVIF_WR2RO_ERR interrupt in the Interrupt Status Register(CH1_IntStatusReg). 0x1 (NO_ACTION): Inactive signal. No action taken.	W
16	Clear_SLVIF_DEC_ERR_IntStat	Slave Interface Decode Error Interrupt Clear Bit. This bit is used to clear the corresponding channel interrupt status bit in CHx_INTSTATUSREG. Values: 0x1 (CLEAR_SLVIF_DEC_ERR): Clear the SLVIF_DEC_ERR interrupt in the Interrupt Status Register(CH1_IntStatusReg). 0x1 (NO_ACTION): Inactive signal. No action taken.	W
15	RSVD_15	Reserved bit	W
14	Clear_SLVIF_MULTIBLKTYPE_ERR_IntStat	Slave Interface Multi Block type Error Interrupt Clear Bit. This bit is used to clear the corresponding channel interrupt status bit in CHx_INTSTATUSREG. Values: 0x1 (CLEAR_SLVIF_MULTIBLKTYPE_ERR): Clear the SLVIF_MULTIBLKTYPE_ERR interrupt in the Interrupt Status Register(CH1_IntStatusReg). 0x1 (NO_ACTION): Inactive signal. No action taken.	W

13	Clear_SHADOWREG_OR_LLI_INVALID_ERR_IntStat	Shadow register or LLI Invalid Error Interrupt Clear Bit. This bit is used to clear the corresponding channel interrupt status bit in CHx_INTSTATUSREG. Values: 0x1 (CLEAR_SHADOWREG_OR_LLI_INVALID_ERR): Clear the SHADOWREG_OR_LLI_INVALID_ERR interrupt in the Interrupt Status Register(CH1_IntStatusReg). 0x1 (NO_ACTION): Inactive signal. No action taken.	W
12	Clear_LLI_WR_SLV_ERR_IntStat	LLI WRITE Slave Error Interrupt Clear Bit. This bit is used to clear the corresponding channel interrupt status bit in CHx_INTSTATUSREG. Values: 0x1 (CLEAR_LLI_WR_SLV_ERR): Clear the LLI_WR_SLV_ERR interrupt in the Interrupt Status Register(CH1_IntStatusReg). 0x1 (NO_ACTION): Inactive signal. No action taken.	W
11	Clear_LLI_RD_SLV_ERR_IntStat	LLI Read Slave Error Interrupt Clear Bit. This bit is used to clear the corresponding channel interrupt status bit in CHx_INTSTATUSREG. Values: 0x1 (CLEAR_LLI_RD_SLV_ERR): Clear the LLI_RD_SLV_ERR interrupt in the Interrupt Status Register(CH1_IntStatusReg). 0x1 (NO_ACTION): Inactive signal. No action taken.	W
10	Clear_LLI_WR_DEC_ERR_IntStat	LLI WRITE Decode Error Interrupt Clear Bit. This bit is used to clear the corresponding channel interrupt status bit in CHx_INTSTATUSREG. Values: 0x1 (CLEAR_LLI_WR_DEC_ERR): Clear the LLI_WR_DEC_ERR interrupt in the Interrupt	W

		Status Register(CH1_IntStatusReg). 0x1 (NO_ACTION): Inactive signal. No action taken.	
9	Clear_LLI_RD _DEC_ERR_IntStat	LLI Read Decode Error Interrupt Clear Bit. This bit is used to clear the corresponding channel interrupt status bit in CHx_INTSTATUSREG. Values: 0x1 (CLEAR_LLI_RD_DEC_ERR): Clear the LLI_RD_DEC_ERR interrupt in the Interrupt Status Register(CH1_IntStatusReg). 0x1 (NO_ACTION): Inactive signal. No action taken.	W
8	Clear_DST _SLV_ERR_IntStat	Destination Slave Error Interrupt Clear Bit. This bit is used to clear the corresponding channel interrupt status bit in CHx_INTSTATUSREG. Values: 0x1 (CLEAR_DST_SLV_ERR): Clear the DST_SLV_ERR interrupt in the Interrupt Status Register(CH1_IntStatusReg). 0x1 (NO_ACTION): Inactive signal. No action taken.	W
7	Clear_SRC _SLV_ERR_IntStat	Source Slave Error Interrupt Clear Bit. This bit is used to clear the corresponding channel interrupt status bit in CHx_INTSTATUSREG. Values: 0x1 (CLEAR_SRC_SLV_ERR): Clear the SRC_SLV_ERR interrupt in the Interrupt Status Register(CH1_IntStatusReg). 0x1 (NO_ACTION): Inactive signal. No action taken.	W
6	Clear_DST _DEC_ERR_IntStat	Destination Decode Error Interrupt Clear Bit. This bit is used to clear the corresponding channel interrupt status bit in CHx_INTSTATUSREG. Values: 0x1 (CLEAR_DST_DEC_ERR): Clear the DST_DEC_ERR interrupt in the Interrupt	W

		Status Register(CH1_IntStatusReg). 0x1 (NO_ACTION): Inactive signal. No action taken.	
5	Clear_SRC_DEC_ERR_IntStat	Source Decode Error Interrupt Clear Bit. This bit is used to clear the corresponding channel interrupt status bit in CHx_INTSTATUSREG. Values: 0x1 (CLEAR_SRC_DEC_ERR): Clear the SRC_DEC_ERR interrupt in the Interrupt Status Register(CH1_IntStatusReg). 0x1 (NO_ACTION): Inactive signal. No action taken.	W
4	Clear_DST_TRANSCOMP_IntStat	Destination Transaction Completed Interrupt Clear Bit. This bit is used to clear the corresponding channel interrupt status bit in CHx_INTSTATUSREG. Values: 0x1 (CLEAR_DST_TRANSCOMP): Clear the DST_TRANSCOMP interrupt in the Interrupt Status Register(CH1_IntStatusReg). 0x1 (NO_ACTION): Inactive signal. No action taken.	W
3	Clear_SRC_TRANSCOMP_IntStat	Source Transaction Completed Interrupt Clear Bit. This bit is used to clear the corresponding channel interrupt status bit in CHx_INTSTATUSREG. Values: 0x1 (CLEAR_SRC_TRANSCOMP): Clear the SRC_TRANSCOMP interrupt in the Interrupt Status Register(CH1_IntStatusReg). 0x1 (NO_ACTION): Inactive signal. No action taken.	W
2	RSVD_DMAC_CHx_INTCLEARREG_2	DMAC Channelx Interrupt Clear Register (bit 2) Reserved bit	W

1	Clear_DMA_TFR_DONE_IntStat	DMA Transfer Done Interrupt Clear Bit. This bit is used to clear the corresponding channel interrupt status bit in CHx_INTSTATUSREG. Values: 0x1 (CLEAR_DMA_TFR_DONE): Clear the DMA_TFR_DONE interrupt in the Interrupt Status Register(CH1_IntStatusReg). 0x1 (NO_ACTION): Inactive signal. No action taken.	W
0	Clear_BLOCK_TFR_DONE_IntStat	Block Transfer Done Interrupt Clear Bit. This bit is used to clear the corresponding channel interrupt status bit in CH1_INTSTATUSREG Values: 0x1 (CLEAR_BLOCK_TFR_DONE): Clear the interrupt in the Interrupt Status Register(CHx_IntStatusReg). Writing a 1 to this register field clears the corresponding bit inthe CHx_IntStatusReg register. 0x1 (NO_ACTION): Inactive signal. No action taken.	W

3.18 集成电路内置总线 (I²C)

3.18.1 概述

I2C总线用于和多个外部设备进行通信。多个外部设备可以共用一个I2C总线。

I2C具有以下几个特点：

- 支持主机模式以及从机模式；
- 支持多主机多从机通信；
- 支持标准模式（100kbit/s）；
- 支持快速模式（400kbit/s）；
- 支持7-bit以及10-bit寻址；
- 支持关闭SCL时钟实现连续数据传输；
- 支持可编程数字噪声滤波功能；

3.18.2 功能描述

I2C是一个两线总线，由SDA线和SCL线构成。这些线设置为漏极开漏输出。因此，I2C总线上可以挂载多个外设，通常是和一个或多个主机以及一个或多个从机。主机通过总线访问从机。

主机发出开始信号，则通讯开始：在SCL为高电平时拉低SDA线，主机将通过SCL线发出9个时钟脉冲。前8个脉冲用于按位传输，该字节包括7bit地址和 1 个读写位。如果从机地址与该7bit地址一致，那么从机可以通过在第9个脉冲上拉低SDA线来应答。接下来，根据读/写标志位，主机和从机可以发送/接收更多的数据。

根据应答位的逻辑电平决定是否停止发送数据。在数据传输中，SDA线仅在SCL线为低电平时才发生变化。当主机完成通讯，回发送一个停止标志：在SCL为高电平时，拉高SDA线。

I2C控制器可以处理 I2C 协议，腾出处理器核用于其它任务。

操作步骤：

1） 通过将0写入IC_ENABLE寄存器的位0，禁用apb_i2c；

2) 写入IC_CON寄存器的bit4以设置支持的地址模式（7或者10bit），配置（bits 2:1）以设置支持的最大传输速度，bit6和bit0都设置为‘1’；
3) 向IC_TAR寄存器写入要寻址的I2C设备的地址（位9:0）。这个寄存器也指示将由I2C执行常规调用还是起始字节命令；
4) 仅适用于高速模式传输，向IC_HS_MADDR寄存器写入所需的apb_i2c的代码，主代码由程序员定义；
5) 通过将1写入IC_ENABLE寄存器的位0来启用apb_i2c；
6) 现在写入传输方向和要发送到IC_DATA_CMD寄存器的数据。如果在启用apb_i2c之前写入IC_DATA_CMD寄存器，数据和命令将会丢失，因为当apb_i2c被禁用时，缓冲区保持清除状态而丢失；

3.18.3 寄存器列表

Base Address: I2C0_BASE_ADDR (0x50280000)
Base Address: I2C1_BASE_ADDR (0x50290000)
Base Address: I2C2_BASE_ADDR (0x502A0000)

Name	Description	Offset address
IC_CON	I2C Control Register. This register can be written only when the apb_i2c is disabled, which corresponds...	0x0
IC_TAR	I2C Target Address Register If the configuration parameter I2C_DYNAMIC_TAR_UPDATE is set to No ...	0x4
IC_SAR	I2C Slave Address Register	0x8
IC_DATA_CMD	I2C Rx/Tx Data Buffer and Command Register; this is the register the CPU writes to when filling...	0x10
IC_SS_SCL_HCNT	Standard Speed I2C Clock SCL High Count Register	0x14
IC_SS_SCL_LCNT	Standard Speed I2C Clock SCL Low Count Register	0x18
IC_INTR_STAT	I2C Interrupt Status Register Each bit in this register has a corresponding mask bit in the IC_INTR_MASK...	0x2c
IC_INTR_MASK	I2C Interrupt Mask Register. These bits mask their corresponding interrupt status bits. This register...	0x30

IC_RAW_INTR_STAT	I2C Raw Interrupt Status Register Unlike the IC_INTR_STAT register, these bits are not masked so...	0x34
IC_RX_TL	I2C Receive FIFO Threshold Register	0x38
IC_TX_TL	I2C Transmit FIFO Threshold Register	0x3c
IC_CLR_INTR	Clear Combined and Individual Interrupt Register	0x40
IC_CLR_RX_UNDER	Clear RX_UNDER Interrupt Register	0x44
IC_CLR_RX_OVER	Clear RX_OVER Interrupt Register	0x48
IC_CLR_TX_OVER	Clear TX_OVER Interrupt Register	0x4c
IC_CLR_RD_REQ	Clear RD_REQ Interrupt Register	0x50
IC_CLR_TX_ABRT	Clear TX_ABRT Interrupt Register	0x54
IC_CLR_RX_DONE	Clear RX_DONE Interrupt Register	0x58
IC_CLR_ACTIVITY	Clear ACTIVITY Interrupt Register	0x5c
IC_CLR_STOP_DET	Clear STOP_DET Interrupt Register	0x60
IC_CLR_START_DET	Clear START_DET Interrupt Register	0x64
IC_CLR_GEN_CALL	Clear GEN_CALL Interrupt Register	0x68
IC_ENABLE	I2C Enable Register	0x6c
IC_STATUS	I2C Status Register This is a read-only register used to indicate the current transfer status...	0x70
IC_TXFLR	I2C Transmit FIFO Level Register This register contains the number of valid data entries in the...	0x74
IC_RXFLR	I2C Receive FIFO Level Register This register contains the number of valid data entries in the receive...	0x78
IC_SDA_HOLD	I2C SDA Hold Time Length Register The bits [15:0] of this register are used to control the hold...	0x7c
IC_TX_ABRT_SOURCE	I2C Transmit Abort Source Register This register has 32 bits that indicate the source of the TX_ABRT...	0x80
IC_DMA_CR	DMA Control Register This register is only valid when apb_i2c is configured with a set of DMA...	0x88

IC_DMA_TDLR	DMA Transmit Data Level Register This register is only valid when the apb_i2c is configured...	0x8c
IC_DMA_RDLR	I2C Receive Data Level Register This register is only valid when apb_i2c is configured with...	0x90
IC_SDA_SETUP	I2C SDA Setup Register This register controls the amount of time delay (in terms of number of ic_clk...	0x94
IC_ACK_GENERAL_CALL	I2C ACK General Call Register The register controls whether apb_i2c responds with a ACK or NACK...	0x98
IC_ENABLE_STATUS	I2C Enable Status Register The register is used to report the apb_i2c hardware status when the...	0x9c
IC_FS_SPKLEN	I2C SS, FS or FM+ spike suppression limit This register is used to store the duration, measured...	0xa0
IC_COMP_PARAM_1	Component Parameter Register 1 Note This is a constant read-only register that contains encoded...	0xf4
IC_COMP_VERSION	I2C Component Version Register	0xf8
IC_COMP_TYPE	I2C Component Type Register	0xfc

3.18.4 寄存器设置

3.18.4.1 IC_CON(0x0)

Bits	Name	Description	Memory Access
31:20	RSVD_IC_CON_2	IC_CON_2 Reserved bits - Read Only	R
19	RSVD_SMBUS_PERSISTENT_SLV_ADDR_EN	SMBUS_PERSISTENT_SLV_ADDR_EN Reserved bits - Read Only	R
18	RSVD_SMBUS_ARP_EN	SMBUS_ARP_EN Reserved bits - Read Only	R
17	RSVD_SMBUS_SLAVE_QUICK_EN	SMBUS_SLAVE_QUICK_EN Reserved bits - Read Only	R
16	RSVD_OPTIONAL_SAR_CTRL	OPTIONAL_SAR_CTRL Reserved bits - Read Only	R
15:12	RSVD_IC_CON_1	IC_CON_1 Reserved bits -	R

		Read Only	
11	RSVD_BUS_CLEAR_FEATURE_CTRL	BUS_CLEAR_FEATURE_CTRL Reserved bits - Read Only	R
10	STOP_DET_IF_MASTER_ACTIVE	In Master mode:1 b1: issues the STOP_DET interrupt only when master is active.1 b0: issues the STOP_DET irrespective of whether master is active or not.Reset value: 0x0.Values:0x1 (ENABLED): Master issues the STOP_DET interrupt only when master is active0x0 (DISABLED): Master issues the STOP_DET interrupt irrespective of whether master is active or not	R
9	RX_FIFO_FULL_HLD_CTRL	This bit controls whether apb_i2c should hold the bus when the Rx FIFO is physically full to its RX_BUFFER_DEPTH, as described in the IC_RX_FULL_HLD_BUS_EN parameter.Reset value: 0x0.Values:0x1 (ENABLED): Hold bus when RX_FIFO is full0x0 (DISABLED): Overflow when RX_FIFO is full	R
8	TX_EMPTY_CTRL	This bit controls the generation of the TX_EMPTY interrupt, as described in the IC_RAW_INTR_STAT register.Reset value: 0x0.Values:0x1 (ENABLED): Controlled generation of TX_EMPTY interrupt0x0 (DISABLED): Default behaviour of TX_EMPTY interrupt	R/W

7	STOP_DET_IFADDRESSED	In slave mode:1 b1: issues the STOP_DET interrrupt only when it is addressed.0 b0: issues the STOP_DET irrespective of whether it s addressed or not.Reset value: 0x0NOTE: During a general call address, this slave does not issue the STOP_DET interrupt if STOP_DET_IF_ADDRESSED = 1 b1, even if the slave responds to the general call address by generating ACK. The STOP_DET interrupt is generated only when the transmitted address matches the slave address (SAR).Values:0x1 (ENABLED): slave issues STOP_DET intr only if addressed0x0 (DISABLED): slave issues STOP_DET intr always	R/W
6	IC_SLAVE_DISABLE	This bit controls whether I2C has its slave disabled, which means once the presetn signal is applied, then this bit takes on the value of the configuration parameter IC_SLAVE_DISABLE. You have the choice of having the slave enabled or disabled after reset is applied, which means software does not have to configure the slave. By default, the slave is always enabled (in reset state as well). If you need to disable it after reset, set this bit to 1.If this bit is set (slave is disabled), apb_i2c functions	R/W

		only as a master and does not perform any action that requires a slave. Reset value: IC_SLAVE_DISABLE configuration parameterNOTE: Software should ensure that if this bit is written with 0, then bit 0 should also be written with a 0.Values:0x1 (SLAVE_DISABLED): Slave mode is disabled0x0 (SLAVE_ENABLED): Slave mode is enabled	
5	IC_RESTART_EN	Determines whether RESTART conditions may be sent when acting as a master. Some older slaves do not support handling RESTART conditions; however, RESTART conditions are used in several apb_i2c operations. When RESTART is disabled, the master is prohibited from performing the following functions:Sending a START BYTEPerforming any high-speed mode operationHigh-speed mode operationPerforming direction changes in combined format modePerforming a read operation with a 10-bit addressBy replacing RESTART condition followed by a STOP and a subsequent START condition, split operations are broken down into multiple apb_i2c transfers. If the above operations are performed, it will result in setting bit 6 (TX_ABRT)	R/W

		of the IC_RAW_INTR_STAT register.Reset value: IC_RESTART_EN configuration parameterValues:0x1 (ENABLED): Master restart enabled0x0 (DISABLED): Master restart disabled	
4	IC_10BITADDR_MASTER	If the I2C_DYNAMIC_TAR_UPDATE configuration parameter is set to No (0), this bit is named IC_10BITADDR_MASTER and controls whether the apb_i2c starts its transfers in 7- or 10-bit addressing mode when acting as a master. If I2C_DYNAMIC_TAR_UPDATE is set to Yes (1), the function of this bit is handled by bit 12 of IC_TAR register, and becomes a read-only copy called IC_10BITADDR_MASTER_rd_only.0: 7-bit addressing1: 10-bit addressingReset value: IC_10BITADDR_MASTER configuration parameterValues:0x1 (ADDR_10BITS): Master 10Bit addressing mode0x0 (ADDR_7BITS): Master 7Bit addressing mode	R/W
		When acting as a slave, this bit controls whether the apb_i2c responds to 7- or 10-bit addresses.0: 7-bit addressing. The apb_i2c	

3	IC_10BITADDR_SLAVE	ignores transactions that involve 10-bit addressing; for 7-bit addressing, only the lower 7 bits of the IC_SAR register are compared.1: 10-bit addressing. The apb_i2c responds to only 10-bit addressing transfers that match the full 10 bits of the IC_SAR register.Reset value: IC_10BITADDR_SLAVE configuration parameterValues:0x1 (ADDR_10BITS): Slave 10Bit addressing0x0 (ADDR_7BITS): Slave 7Bit addressing	R/W
2:1	SPEED	These bits control at which speed the apb_i2c operates; its setting is relevant only if one is operating the apb_i2c in master mode. Hardware protects against illegal values being programmed by software. These bits must be programmed appropriately for slave mode also, as it is used to capture correct value of spike filter as per the speed mode.This register should be programmed only with a value in the range of 1 to IC_MAX_SPEED_MODE; otherwise, hardware updates this register with the value of IC_MAX_SPEED_MODE.1: standard mode (100 kbit/s)2: fast mode (<=400 kbit/s) or	R/W

		fast mode plus (<=1000Kbit/s)3: high speed mode (3.4 Mbit/s)Note: This field is not applicable when IC_ULTRA_FAST_MODE=1Reset value: IC_MAX_SPEED_MODE configurationValues:0x1 (STANDARD): Standard Speed mode of operation0x2 (FAST): Fast or Fast Plus mode of operation0x3 (HIGH): High Speed mode of operation	
0	MASTER_MODE	This bit controls whether the apb_i2c master is enabled. Reset value: IC_MASTER_MODE configuration parameterNOTE: Software should ensure that if this bit is written with '1' then bit 6 should also be written with a '1'.Values:0x1 (ENABLED): Master mode is enabled0x0 (DISABLED): Master mode is disabled	R/W

3.18.4.2 IC_TAR(0x4)

Bits	Name	Description	Memory Access
31:17	RSVD_IC_TAR_2	IC_TAR_2 Reserved bits - Read Only	R
16	RSVD_SMBUS_QUICK_CMD	SMBUS_QUICK_CMD Reserved bits - Read Only	R
15:14	RSVD_IC_TAR_1	IC_TAR_1 Reserved bits - Read Only	R
13	RSVD_DEVICE_ID	DEVICE_ID Reserved bits - Read Only	R
12	RSVD_IC_10BITADDR_MASTER	IC_10BITADDR_MASTER Reserved bits - Read Only	R
11	SPECIAL	This bit indicates whether software performs a Device-ID or General Call or START BYTE command.0: ignore bit 10 GC_OR_START and use IC_TAR normally1: perform special I2C command as specified in Device_ID or GC_OR_START bitReset value: 0x0Values:0x1 (ENABLED): Enables programming of GENERAL_CALL or START_BYTE transmission0x0 (DISABLED): Disables programming of GENERAL_CALL or START_BYTE transmission	R/W

10	GC_OR_START	If bit 11 (SPECIAL) is set to 1 and bit 13(Device-ID) is set to 0, then this bit indicates whether a General Call or START byte command is to be performed by the apb_i2c.0: General Call Address - after issuing a General Call, only writes may be performed. Attempting to issue a read command results in setting bit 6 (TX_ABRT) of the IC_RAW_INTR_STAT register. The apb_i2c remains in General Call mode until the SPECIAL bit value (bit 11) is cleared.1: START BYTEReset value: 0x0Values:0x1 (START_BYTE): START byte transmission0x0 (GENERAL_CALL): GENERAL_CALL byte transmission	R/W
9:0	IC_TAR	This is the target address for any master transaction. When transmitting a General Call, these bits are ignored. To generate a START BYTE, the CPU needs to write only once into these bits.If the IC_TAR and IC_SAR are the same, loopback exists but the FIFOs are shared between master and slave, so full loopback is not feasible. Only one direction loopback mode is supported (simplex), not duplex. A master cannot transmit to itself; it can transmit to only a slave.Reset value: IC_DEFAULT_TAR_SLAVE_ADDR configuration parameter	R/W

3.18.4.3 IC_SAR(0x8)

Bits	Name	Description	Memory Access
31:10	RSVD_IC_SAR	IC_SAR Reserved bits - Read Only	R
9:0	IC_SAR	The IC_SAR holds the slave address when the I2C is operating as a slave. For 7-bit addressing, only IC_SAR[6:0] is used.This register can be written only when the I2C interface is disabled, which corresponds to the IC_ENABLE[0] register being set to 0. Writes at other times have no effect.Note: The default values cannot be any of the reserved address locations: that is, 0x00 to 0x07, or 0x78 to 0x7f. The correct operation of the device is not guaranteed if you program the IC_SAR or IC_TAR to a reserved value. Refer to Table I2C/SMBus Definition of Bits in First Byte for a complete list of these reserved values.Reset value: IC_DEFAULT_SLAVE_ADDR configuration parameter	R/W

3.18.4.4 IC_DATA_CMD(0x10)

Bits	Name	Description	Memory Access
31:12	RSVD_IC_DATA_CMD	IC_DATA_CMD Reserved bits - Read OnlyVolatile: true	R
11	RSVD_FIRST_DATA_BYTE	FIRST_DATA_BYTE Reserved bits - Read OnlyVolatile: true	R
10	RSVD_RESTART	RESTART Reserved bits - Read OnlyVolatile: true	R
9	RSVD_STOP	STOP Reserved bits - Read OnlyVolatile: true	R

8	CMD	This bit controls whether a read or a write is performed. This bit does not control the direction when the apb_i2c acts as a slave. It controls only the direction when it acts as a master.When a command is entered in the TX FIFO, this bit distinguishes the write and read commands. In slave- receiver mode, this bit is a don t care because writes to this register are not required. In slave-transmitter mode, a "0" indicates that the data in IC_DATA_CMD is to be transmitted.When programming this bit, you should remember the following: attempting to perform a read operation after a General Call command has been sent results in a TX_ABRT interrupt (bit 6 of the IC_RAW_INTR_STAT register), unless bit 11 (SPECIAL) in the IC_TAR register has been cleared. If a "1" is written to this bit after receiving a RD_REQ interrupt, then a TX_ABRT interrupt occurs.Reset value: 0x0Values:0x1 (READ): Master Read Command0x0 (WRITE): Master Write CommandVolatile: true	W

7:0	DAT	This register contains the data to be transmitted or received on the I2C bus. If you are writing to this register and want to perform a read, bits 7:0 (DAT) are ignored by the apb_i2c. However, when you read this register, these bits return the value of data received on the apb_i2c interface.Reset value: 0x0Volatile: true	R/W

3.18.4.5 IC_SS_SCL_HCNT(0x14)

Bits	Name	Description	Memory Access
31:16	RSVD_IC_SS_SCL_HIGH_COUNT	IC_SS_SCL_HCNT Reserved bits - Read Only	R
15:0	IC_SS_SCL_HCNT	This register must be set before any I2C bus transaction can take place to ensure proper I/O timing. This register sets the SCL clock high-period count for standard speed. For more information, refer to IC_CLK Frequency Configuration .This register can be written only when the I2C interface is disabled which corresponds to the IC_ENABLE[0] register being set to 0. Writes at other times have no effect.The minimum valid value is 6; hardware prevents values less than this being written, and if attempted results in 6 being set. For designs with APB_DATA_WIDTH = 8, the order of programming is important to ensure the correct operation of the apb_i2c. The lower byte must be programmed first. Then the upper byte is programmed.When the configuration parameter IC_HC_COUNT_VALUES is set to 1, this register is read only.NOTE:	R/W

		This register must not be programmed to a value higher than 65525, because apb_i2c uses a 16-bit counter to flag an I2C bus idle condition when this counter reaches a value of IC_SS_SCL_HCNT + 10.Reset value: IC_SS_SCL_HIGH_COUNT configuration parameter	

3.18.4.6 IC_SS_SCL_LCNT(0x18)

Bits	Name	Description	Memory Access
31:16	RSVD_IC_SS_SCL_LOW_COUNT	RSVD_IC_SS_SCL_LOW_COUNT Reserved bits - Read Only	R
15:0	IC_SS_SCL_LCNT	This register must be set before any I2C bus transaction can take place to ensure proper I/O timing. This register sets the SCL clock low period count for standard speed. For more information, refer to IC_CLK Frequency Configuration This register can be written only when the I2C interface is disabled which corresponds to the IC_ENABLE[0] register being set to 0. Writes at other times have no effect.The minimum valid value is 8; hardware prevents values less than this being written, and if attempted, results in 8 being set. For designs with APB_DATA_WIDTH = 8, the order of programming is important to ensure the correct operation of apb_i2c. The lower byte must be programmed first, and then the upper byte is programmed.When the configuration parameter IC_HC_COUNT_VALUES is set to 1, this register is read only.Reset value: IC_SS_SCL_LOW_COUNT configuration parameter	R/W

3.18.4.7 IC_INTR_STAT(0x2c)

Bits	Name	Description	Memory Access
31:15	RSVD_IC_INTR_STAT	IC_INTR_STAT Reserved bits - Read OnlyVolatile: true	R
14	RSVD_R_SCL_STUCK_AT_LOW	R_SCL_STUCK_AT_LOW Register field Reserved bits - Read OnlyReset Value: 0x0Values:0x1 (ACTIVE): R_SCL_STUCK_AT_LOW interrupt is active0x0 (INACTIVE): R_SCL_STUCK_AT_LOW interrupt is inactiveVolatile: true	R
13	R_MASTER_ON_HOLD	See IC_RAW_INTR_STAT for a detailed description of R_MASTER_ON_HOLD bit.Reset value: 0x0Values:0x1 (ACTIVE): R_MASTER_ON_HOLD interrupt is active0x0 (INACTIVE): R_MASTER_ON_HOLD interrupt is inactiveVolatile: true	R
12	R_RESTART_DET	See IC_RAW_INTR_STAT for a detailed description of R_RESTART_DET bit.Reset value: 0x0Values:0x1 (ACTIVE): R_RESTART_DET interrupt is active0x0 (INACTIVE): R_RESTART_DET interrupt is inactiveVolatile: true	R

Bits	Name	Description	Memory Access
11	R_GEN_CALL	See IC_RAW_INTR_STAT for a detailed description of R_GEN_CALL bit.Reset value: 0x0Values:0x1 (ACTIVE): R_GEN_CALL interrupt is active0x0 (INACTIVE): R_GEN_CALL interrupt is inactiveVolatile: true	R
10	R_START_DET	See IC_RAW_INTR_STAT for a detailed description of R_START_DET bit.Reset value: 0x0Values:0x1 (ACTIVE): R_START_DET interrupt is active0x0 (INACTIVE): R_START_DET interrupt is inactiveVolatile: true	R
9	R_STOP_DET	See IC_RAW_INTR_STAT for a detailed description of R_STOP_DET bit.Reset value: 0x0Values:0x1 (ACTIVE): R_STOP_DET interrupt is active0x0 (INACTIVE): R_STOP_DET interrupt is inactiveVolatile: true	R
8	R_ACTIVITY	See IC_RAW_INTR_STAT for a detailed description of R_ACTIVITY bit.Reset value: 0x0Values:0x1 (ACTIVE): R_ACTIVITY interrupt is active0x0 (INACTIVE): R_ACTIVITY interrupt is inactiveVolatile: true	R
7	R_RX_DONE	See IC_RAW_INTR_STAT for a detailed description of R_RX_DONE bit.Reset value: 0x0Values:0x1 (ACTIVE): R_RX_DONE interrupt is active0x0 (INACTIVE): R_RX_DONE	R

		interrupt is inactiveVolatile: true	
6	R_TX_ABRT	See IC_RAW_INTR_STAT for a detailed description of R_TX_ABRT bit.Reset value: 0x0Values:0x1 (ACTIVE): R_TX_ABRT interrupt is active0x0 (INACTIVE): R_TX_ABRT interrupt is inactiveVolatile: true	R

Bits	Name	Description	Memory Access
5	R_RD_REQ	See IC_RAW_INTR_STAT for a detailed description of R_RD_REQ bit.Reset value: 0x0Values:0x1 (ACTIVE): R_RD_REQ interrupt is active0x0 (INACTIVE): R_RD_REQ interrupt is inactiveVolatile: true	R
4	R_TX_EMPTY	See IC_RAW_INTR_STAT for a detailed description of R_TX_EMPTY bit.Reset value: 0x0Values:0x1 (ACTIVE): R_TX_EMPTY interrupt is active0x0 (INACTIVE): R_TX_EMPTY interrupt is inactiveVolatile: true	R
3	R_TX_OVER	See IC_RAW_INTR_STAT for a detailed description of R_TX_OVER bit.Reset value: 0x0Values:0x1 (ACTIVE): R_TX_OVER interrupt is active0x0 (INACTIVE): R_TX_OVER interrupt is inactiveVolatile: true	R

2	R_RX_FULL	See IC_RAW_INTR_STAT for a detailed description of R_RX_FULL bit.Reset value: 0x0Values:0x1 (ACTIVE): R_RX_FULL interrupt is active0x0 (INACTIVE): R_RX_FULL interrupt is inactiveVolatile: true	R
1	R_RX_OVER	See IC_RAW_INTR_STAT for a detailed description of R_RX_OVER bit.Reset value: 0x0Values:0x1 (ACTIVE): R_RX_OVER interrupt is active0x0 (INACTIVE): R_RX_OVER interrupt is inactiveVolatile: true	R
0	R_RX_UNDER	See IC_RAW_INTR_STAT for a detailed description of R_RX_UNDER bit.Reset value: 0x0Values:0x1 (ACTIVE): RX_UNDER interrupt is active0x0 (INACTIVE): RX_UNDER interrupt is inactiveVolatile: true	R

3.18.4.8 IC_INTR_MASK(0x30)

Bits	Name	Description	Memory Access
31:15	RSVD_IC_INTR_STAT	IC_INTR_STAT Reserved bits - Read Only	R
14	RSVD_M_SCL_STUCK_AT_LOW	M_SCL_STUCK_AT_LOW Register field Reserved bits - Read Only	R
13	M_MASTER_ON_HOLD_read_only	This M_MASTER_ON_HOLD_read_only bit masks the R_MASTER_ON_HOLD interrupt in IC_INTR_STAT register.Reset value: 0x0Values:0x1 (DISABLED): MASTER_ON_HOLD interrupt is unmasked0x0 (ENABLED): MASTER_ON_HOLD interrupt is masked	R
12	M_RESTART_DET_read_only	This M_RESTART_DET_read_only bit masks the R_RESTART_DET interrupt in IC_INTR_STAT register.Reset value: 0x0Values:0x1 (DISABLED): RESTART_DET interrupt is unmasked0x0 (ENABLED): RESTART_DET interrupt is masked	R
11	M_GEN_CALL	This bit masks the R_GEN_CALL interrupt in IC_INTR_STAT register.Reset value: 0x1Values:0x1 (DISABLED): GEN_CALL interrupt is unmasked0x0 (ENABLED): GEN_CALL interrupt is masked	R/W

10	M_START_DET	This bit masks the R_START_DET interrupt in IC_INTR_STAT register.Reset value: 0x0Values:0x1 (DISABLED): START_DET interrupt is unmasked0x0 (ENABLED): START_DET interrupt is masked	R/W
9	M_STOP_DET	This bit masks the R_STOP_DET interrupt in IC_INTR_STAT register.Reset value: 0x0Values:0x1 (DISABLED): STOP_DET interrupt is unmasked0x0 (ENABLED): STOP_DET interrupt is masked	R/W
8	M_ACTIVITY	This bit masks the R_ACTIVITY interrupt in IC_INTR_STAT register.Reset value: 0x0Values:0x1 (DISABLED): ACTIVITY interrupt is unmasked0x0 (ENABLED): ACTIVITY interrupt is masked	R/W

Bits	Name	Description	Memory Access
7	M_RX_DONE	This bit masks the R_RX_DONE interrupt in IC_INTR_STAT register.Reset value: 0x1Values:0x1 (DISABLED): RX_DONE interrupt is unmasked0x0 (ENABLED): RX_DONE interrupt is masked	R/W
6	M_TX_ABRT	This bit masks the R_TX_ABRT interrupt in IC_INTR_STAT register.Reset value: 0x1Values:0x1 (DISABLED): TX_ABORT interrupt is unmasked0x0 (ENABLED): TX_ABORT interrupt is masked	R/W

5	M_RD_REQ	This bit masks the R_RD_REQ interrupt in IC_INTR_STAT register.Reset value: 0x1Values:0x1 (DISABLED): RD_REQ interrupt is unmasked0x0 (ENABLED): RD_REQ interrupt is masked	R/W
4	M_TX_EMPTY	This bit masks the R_TX_EMPTY interrupt in IC_INTR_STAT register.Reset value: 0x1Values:0x1 (DISABLED): TX_EMPTY interrupt is unmasked0x0 (ENABLED): TX_EMPTY interrupt is masked	R/W
3	M_TX_OVER	This bit masks the R_TX_OVER interrupt in IC_INTR_STAT register.Reset value: 0x1Values:0x1 (DISABLED): TX_OVER interrupt is unmasked0x0 (ENABLED): TX_OVER interrupt is masked	R/W
2	M_RX_FULL	This bit masks the R_RX_FULL interrupt in IC_INTR_STAT register.Reset value: 0x1Values:0x1 (DISABLED): RX_FULL interrupt is unmasked0x0 (ENABLED): RX_FULL interrupt is masked	R/W
1	M_RX_OVER	This bit masks the R_RX_OVER interrupt in IC_INTR_STAT register.Reset value: 0x1Values:0x1 (DISABLED): RX_OVER interrupt is unmasked0x0 (ENABLED): RX_OVER interrupt is masked	R/W

0	M_RX_UNDER	This bit masks the R_RX_UNDER interrupt in IC_INTR_STAT register.Reset value: 0x1Values:0x1 (DISABLED): RX_UNDER interrupt is unmasked0x0 (ENABLED): RX_UNDER interrupt is masked	R/W

3.18.4.9 IC_RAW_INTR_STAT(0x34)

Bits	Name	Description	Memory Access
31:15	RSVD_IC_RAW_INTR_STAT	IC_RAW_INTR_STAT Reserved bits - Read OnlyVolatile: true	R
14	RSVD_SCL_STUCK_AT_LOW	SCL_STUCK_AT_LOW Register field Reserved bits - Read OnlyVolatile: true	R
13	MASTER_ON_HOLD	Indicates whether master is holding the bus and TX FIFO is empty. Enabled only when I2C_DYNAMIC_TAR_UPDATE=1 and IC_EMPTYFIFO_HOLD_MASTER_EN=1.Reset value: 0x0Values:0x1 (ACTIVE): MASTER_ON_HOLD interrupt is active0x0 (INACTIVE): MASTER_ON_HOLD interrupt is inactiveVolatile: true	R
12	RESTART_DET	Indicates whether a RESTART condition has occurred on the I2C interface when apb_i2c is operating in Slave mode and the slave is being addressed.Enabled only when IC_SLV_RESTART_DET_EN=1.Note: However, in high- speed mode or during a START BYTE transfer, the RESTART comes before the address field as per the I2C protocol. In this case, the slave is not the addressed slave when the RESTART is issued, therefore apb_i2c does not generate the RESTART_DET interrupt.Reset value: 0x0Values:0x1 (ACTIVE): RESTART_DET interrupt is active0x0 (INACTIVE): RESTART_DET interrupt is inactiveVolatile: true	R

11	GEN_CALL	Set only when a General Call address is received and it is acknowledged. It stays set until it is cleared either by disabling apb_i2c or when the CPU reads bit 0 of the IC_CLR_GEN_CALL register. apb_i2c stores the received data in the Rx buffer.Reset value: 0x0Values:0x1 (ACTIVE): GEN_CALL interrupt is active0x0 (INACTIVE): GEN_CALL interrupt is inactiveVolatile: true	R
10	START_DET	Indicates whether a START or RESTART condition has occurred on the I2C interface regardless of whether apb_i2c is operating in slave or master mode.Reset value: 0x0Values:0x1 (ACTIVE): START_DET interrupt is active0x0 (INACTIVE): START_DET interrupt is inactiveVolatile: true	R
9	STOP_DET	Indicates whether a STOP condition has occurred on the I2C interface regardless of whether apb_i2c is operating in slave or master mode.In Slave Mode:If IC_CON[7]=1 b1 (STOP_DET_IFADDRESSED), the STOP_DET interrupt will be issued only if slave is addressed.Note: During a general call address, this slave does not issue a STOP_DET interrupt if STOP_DET_IF_ADDRESSED=1 b1, even if the slave responds to the general call address by generating ACK. The STOP_DET interrupt is generated only when the transmitted address matches the slave address (SAR).If IC_CON[7]=1 b0 (STOP_DET_IFADDRESSED), the STOP_DET interrupt is issued irrespective of whether it is being addressed.In Master Mode:If IC_CON[10]=1 b1 (STOP_DET_IF_MASTER_ACTIVE),the	R

		STOP_DET interrupt will be issued only if Master is active.If IC_CON[10]=1 b0 (STOP_DET_IFADDRESSED),the STOP_DET interrupt will be issued irrespective of whether master is active or not.Reset value: 0x0Values:0x1 (ACTIVE): STOP_DET interrupt is active0x0 (INACTIVE): STOP_DET interrupt is inactiveVolatile: true	
8	ACTIVITY	This bit captures apb_i2c activity and stays set until it is cleared. There are four ways to clear it:Disabling the apb_i2cReading the IC_CLR_ACTIVITY registerReading the IC_CLR_INTR registerSystem resetOnce this bit is set, it stays set unless one of the four methods is used to clear it. Even if the apb_i2c module is idle, this bit remains set until cleared, indicating that there was activity on the bus.Reset value: 0x0Values:0x1 (ACTIVE): RAW_INTR_ACTIVITY interrupt is active0x0 (INACTIVE): RAW_INTR_ACTIVITY interrupt is inactiveVolatile: true	R
7	RX_DONE	When the apb_i2c is acting as a slave-transmitter, this bit is set to 1 if the master does not acknowledge a transmitted byte. This occurs on the last byte of the transmission, indicating that the transmission is done.Reset value: 0x0Values:0x1 (ACTIVE): RX_DONE interrupt is active0x0 (INACTIVE): RX_DONE interrupt is inactiveVolatile: true	R
		This bit indicates if apb_i2c, as an I2C transmitter, is unable to complete the intended actions on the contents of the transmit FIFO. This situation can occur both as an I2C master or an I2C slave, and is referred to as a transmit abort . When this	

6	TX_ABRT	bit is set to 1, the IC_TX_ABRT_SOURCE register indicates the reason why the transmit abort takes places.Note: The apb_i2c flushes/resets/empties only the TX_FIFO whenever there is a transmit abort caused by any of the events tracked by the IC_TX_ABRT_SOURCE register. The Tx FIFO remains in this flushed state until the register IC_CLR_TX_ABRT is read. Once this read is performed, the Tx FIFO is then ready to accept more data bytes from the APB interface. RX FIFO flush because of TX_ABRT is controlled by the coreConsultant parameter IC_AVOID_RX_FIFO_FLUSH_ON_TX_ABRT.Reset value: 0x0Values:0x1 (ACTIVE): TX_ABRT interrupt is active0x0 (INACTIVE): TX_ABRT interrupt is inactiveVolatile: true	R
5	RD_REQ	This bit is set to 1 when apb_i2c is acting as a slave and another I2C master is attempting to read data from apb_i2c. The apb_i2c holds the I2C bus in a wait state (SCL=0) until this interrupt is serviced, which means that the slave has been addressed by a remote master that is asking for data to be transferred. The processor must respond to this interrupt and then write the requested data to the IC_DATA_CMD register. This bit is set to 0 just after the processor reads the IC_CLR_RD_REQ register.Reset value: 0x0Values:0x1 (ACTIVE): RD_REQ interrupt is active0x0 (INACTIVE): RD_REQ interrupt is inactiveVolatile: true	R
		The behavior of the TX_EMPTY interrupt status differs based on the TX_EMPTY_CTRL selection in the IC_CON register.When TX_EMPTY_CTRL = 0:This bit is set to 1 when the transmit	

4	TX_EMPTY	buffer is at or below the threshold value set in the IC_TX_TL register.When TX_EMPTY_CTRL = 1:This bit is set to 1 when the transmit buffer is at or below the threshold value set in the IC_TX_TL register and the transmission of the address/data from the internal shift register for the most recently popped command is completed.It is automatically cleared by hardware when the buffer level goes above the threshold. When IC_ENABLE[0] is set to 0, the TX FIFO is flushed and held in reset. There the TX FIFO looks like it has no data within it, so this bit is set to 1, provided there is activity in the master or slave state machines. When there is no longer any activity, then with ic_en=0, this bit is set to 0.Reset value: 0x0.Values:0x1 (ACTIVE): TX_EMPTY interrupt is active0x0 (INACTIVE): TX_EMPTY interrupt is inactiveVolatile: true	R
3	TX_OVER	Set during transmit if the transmit buffer is filled to IC_TX_BUFFER_DEPTH and the processor attempts to issue another I2C command by writing to the IC_DATA_CMD register. When the module is disabled, this bit keeps its level until the master or slave state machines go into idle, and when ic_en goes to 0, this interrupt is cleared.Reset value: 0x0Values:0x1 (ACTIVE): TX_OVER interrupt is active0x0 (INACTIVE): TX_OVER interrupt is inactiveVolatile: true	R
2	RX_FULL	Set when the receive buffer reaches or goes above the RX_TL threshold in the IC_RX_TL register. It is automatically cleared by hardware when buffer level goes below the threshold. If the module is disabled (IC_ENABLE[0]=0), the RX FIFO is flushed and held in reset; therefore the	R

		RX FIFO is not full. So this bit is cleared once the IC_ENABLE bit 0 is programmed with a 0, regardless of the activity that continues.Reset value: 0x0Values:0x1 (ACTIVE): RX_FULL interrupt is active0x0 (INACTIVE): RX_FULL interrupt is inactiveVolatile: true	
1	RX_OVER	Set if the receive buffer is completely filled to IC_RX_BUFFER_DEPTH and an additional byte is received from an external I2C device. The apb_i2c acknowledges this, but any data bytes received after the FIFO is full are lost. If the module is disabled (IC_ENABLE[0]=0), this bit keeps its level until the master or slave state machines go into idle, and when ic_en goes to 0, this interrupt is cleared.Note: If the configuration parameter IC_RX_FULL_HLD_BUS_EN is enabled and bit 9 of the IC_CON register (RX_FIFO_FULL_HLD_CTRL) is programmed to HIGH, then the RX_OVER interrupt never occurs, because the Rx FIFO never overflows.Reset value: 0x0Values:0x1 (ACTIVE): RX_OVER interrupt is active0x0 (INACTIVE): RX_OVER interrupt is inactiveVolatile: true	R
0	RX_UNDER	Set if the processor attempts to read the receive buffer when it is empty by reading from the IC_DATA_CMD register. If the module is disabled (IC_ENABLE[0]=0), this bit keeps its level until the master or slave state machines go into idle, and when ic_en goes to 0, this interrupt is cleared.Reset value: 0x0Values:0x1 (ACTIVE): RX_UNDER interrupt is active0x0 (INACTIVE): RX_UNDER interrupt is inactiveVolatile: true	R

3.18.4.19 IC_CLR_ACTIVITY(0x5c)

Bits	Name	Description	Memory Access
31:1	RSVD_IC_CLR_ACTIVITY	IC_CLR_ACTIVITY Reserved bits - Read OnlyVolatile: true	R
0	CLR_ACTIVITY	Reading this register clears the ACTIVITY interrupt if the I2C is not active anymore. If the I2C module is still active on the bus, the ACTIVITY interrupt bit continues to be set. It is automatically cleared by hardware if the module is disabled and if there is no further activity on the bus. The value read from this register to get status of the ACTIVITY interrupt (bit 8) of the IC_RAW_INTR_STAT register.Reset value: 0x0Volatile: true	R

3.18.4.20 IC_CLR_STOP_DET(0x60)

Bits	Name	Description	Memory Access
31:1	RSVD_IC_CLR_STOP_DET	IC_CLR_STOP_DET Reserved bits - Read OnlyVolatile: true	R
0	CLR_STOP_DET	Read this register to clear the STOP_DET interrupt (bit 9) of the IC_RAW_INTR_STAT register.Reset value: 0x0Volatile: true	R

3.18.4.21 IC_CLR_START_DET(0x64)

Bits	Name	Description	Memory Access
31:1	RSVD_IC_CLR_START_DET	IC_CLR_START_DET Reserved bits - Read OnlyVolatile: true	R
0	CLR_START_DET	Read this register to clear the START_DET interrupt (bit 10) of the IC_RAW_INTR_STAT register.Reset value: 0x0Volatile: true	R

3.18.4.22 IC_CLR_GEN_CALL(0x68)

Bits	Name	Description	Memory Access
31:1	RSVD_IC_CLR_GEN_CALL	IC_CLR_GEN_CALL Reserved bits - Read OnlyVolatile: true	R
0	CLR_GEN_CALL	Read this register to clear the GEN_CALL interrupt (bit 11) of IC_RAW_INTR_STAT register.Reset value: 0x0Volatile: true	R

3.18.4.23 IC_ENABLE(0x6c)

Bits	Name	Description	Memory Access
31:19	RSVD_IC_ENABLE_2	IC_ENABLE Reserved bits - Read Only	R
18	RSVD_SMBUS_ALERT_EN	SMBUS_ALERT_EN Register field Reserved bits - Read Only	R
17	RSVD_SMBUS_SUSPEND_EN	SMBUS_SUSPEND_EN Register field Reserved bits - Read Only	R
16	RSVD_SMBUS_CLK_RESET	SMBUS_CLK_RESET Register field Reserved bits - Read Only	R
15:4	RSVD_IC_ENABLE_1	RSVD_IC_ENABLE_1 Reserved bits - Read Only	R
3	RSVD	Reserved bits - Read Only	R
2	TX_CMD_BLOCK	In Master mode:1 b1: Blocks the transmission of data on I2C bus even if Tx FIFO has data to transmit.1 b0: The transmission of data starts on I2C bus automatically, as soon as the first data is available in the Tx FIFO.Note: To block the execution of Master commands, set the TX_CMD_BLOCK bit only when Tx FIFO is empty (IC_STATUS[2]==1) and Master is in Idle state (IC_STATUS[5] == 0). Any further commands put in the Tx FIFO are not executed until TX_CMD_BLOCK bit is unset. Reset	R/W

		value:IC_TX_CMD_BLOCK_DEFAULTValues:0x1 (BLOCKED): Tx Command execution blocked0x0 (NOT_BLOCKED): Tx Command execution not blocked	
1	ABORT	When set, the controller initiates the transfer abort.0: ABORT not initiated or ABORT done1: ABORT operation in progressThe software can abort the I2C transfer in master mode by setting this bit. The software can set this bit only when ENABLE is already set; otherwise, the controller ignores any write to ABORT bit. The software cannot clear the ABORT bit once set. In response to an ABORT, the controller issues a STOP and flushes the Tx FIFO after completing the current transfer, then sets the TX_ABORT interrupt after the abort operation. The ABORT bit is cleared automatically after the abort operation.For a detailed description on how to abort I2C transfers, refer to Aborting I2C Transfers .Reset value: 0x0Values:0x1 (ENABLED): ABORT operation in progress0x0 (DISABLE): ABORT operation not in progress	R/W

0	ENABLE	Controls whether the apb_i2c is enabled.0: Disables apb_i2c (TX and RX FIFOs are held in an erased state)1: Enables apb_i2cSoftware can disable apb_i2c while it is active. However, it is important that care be taken to ensure that apb_i2c is disabled properly. A recommended procedure is described in Disabling apb_i2c .When apb_i2c is disabled, the following occurs:The TX FIFO and RX FIFO get flushed.Status bits in the IC_INTR_STAT register are still active until apb_i2c goes into IDLE state.If the module is transmitting, it stops as well as deletes the contents of the transmit buffer after the current transfer is complete. If the module is receiving, the apb_i2c stops the current transfer at the end of the current byte and does not acknowledge the transfer.In systems with asynchronous pclk and ic_clk when IC_CLK_TYPE parameter set to asynchronous (1), there is a two ic_clk delay when enabling or disabling the apb_i2c. For a detailed description on how to disable apb_i2c, refer to Disabling apb_i2c Reset value: 0x0Values:0x1 (ENABLED): I2C is enabled0x0 (DISABLED): I2C is disabled	R/W

3.18.4.24 IC_STATUS(0x70)

Bits	Name	Description	Memory Access
31:21	RSVD_IC_STATUS_2	IC_STATUS Reserved bits - Read OnlyVolatile: true	R
20	RSVD_SMBUS_ALERT_STATUS	SMBUS_ALERT_STATUS Register field Reserved bits - Read OnlyVolatile: true	R
19	RSVD_SMBUS_SUSPEND_STATUS	SMBUS_SUSPEND_STATUS Register field Reserved bits - Read OnlyVolatile: true	R
18	RSVD_SMBUS_SLAVE_ADDR_RESOLVED	SMBUS_SLAVE_ADDR_RESOLVED Register field Reserved bits - Read OnlyVolatile: true	R
17	RSVD_SMBUS_SLAVE_ADDR_VALID	SMBUS_SLAVE_ADDR_VALID Register field Reserved bits - Read OnlyVolatile: true	R
16	RSVD_SMBUS_QUICK_CMD_BIT	SMBUS_QUICK_CMD_BIT Register field Reserved bits - Read OnlyVolatile: true	R
15:12	RSVD_IC_STATUS_1	RSVD_IC_STATUS_1 Reserved bits - Read OnlyVolatile: true	R
11	RSVD_SDA_STUCK_NOT_RECOVERED	SDA_STUCK_NOT_RECOVERED Register field Reserved bits - Read OnlyVolatile: true	R
10	RSVD_SLV_HOLD_RX_FIFO_FULL	SLV_HOLD_RX_FIFO_FULL Register field Reserved bits - Read OnlyVolatile: true	R
9	RSVD_SLV_HOLD_TX_FIFO_EMPTY	SLV_HOLD_TX_FIFO_EMPTY Regsiter field Reserved bits - Read OnlyVolatile: true	R
8	RSVD_MST_HOLD_RX_FIFO_FULL	MST_HOLD_RX_FIFO_FULL Register field Reserved bits - Read OnlyVolatile: true	R

7	RSVD_MST_HOLD_TX_FIFO_EMPTY	MST_HOLD_TX_FIFO_EMPTY Register field Reserved bits - Read OnlyVolatile: true	R
6	SLV_ACTIVITY	Slave FSM Activity Status. When the Slave Finite State Machine (FSM) is not in the IDLE state, this bit is set.0: Slave FSM is in IDLE state so the Slave part of apb_i2c is not Active1: Slave FSM is not in IDLE state so the Slave part of apb_i2c is ActiveReset value: 0x0Values:0x1 (ACTIVE): Slave not idle0x0 (IDLE): Slave is idleVolatile: true	R

Bits	Name	Description	Memory Access
5	MST_ACTIVITY	Master FSM Activity Status. When the Master Finite State Machine (FSM) is not in the IDLE state, this bit is set.0: Master FSM is in IDLE state so the Master part of apb_i2c is not Active1: Master FSM is not in IDLE state so the Master part of apb_i2c is ActiveNote: IC_STATUS[0]-that is, ACTIVITY bit-is the OR of SLV_ACTIVITY and MST_ACTIVITY bits.Reset value: 0x0Values:0x1 (ACTIVE): Master not idle0x0 (IDLE): Master is idleVolatile: true	R

4	RFF	Receive FIFO Completely Full. When the receive FIFO is completely full, this bit is set. When the receive FIFO contains one or more empty location, this bit is cleared.0: Receive FIFO is not full1: Receive FIFO is fullReset value: 0x0Values:0x1 (FULL): Rx FIFO is full0x0 (NOT_FULL): Rx FIFO not fullVolatile: true	R
3	RFNE	Receive FIFO Not Empty. This bit is set when the receive FIFO contains one or more entries; it is cleared when the receive FIFO is empty.0: Receive FIFO is empty1: Receive FIFO is not emptyReset value: 0x0Values:0x1 (NOT_EMPTY): Rx FIFO not empty0x0 (EMPTY): Rx FIFO is emptyVolatile: true	R
2	TFE	Transmit FIFO Completely Empty. When the transmit FIFO is completely empty, this bit is set. When it contains one or more valid entries, this bit is cleared. This bit field does not request an interrupt.0: Transmit FIFO is not empty1: Transmit FIFO is emptyReset value: 0x1Values:0x1 (EMPTY): Tx FIFO is empty0x0 (NON_EMPTY): Tx FIFO not emptyVolatile: true	R

1	TFNF	Transmit FIFO Not Full. Set when the transmit FIFO contains one or more empty locations, and is cleared when the FIFO is full.0: Transmit FIFO is full1: Transmit FIFO is not fullReset value: 0x1Values:0x1 (NOT_FULL): Tx FIFO not full0x0 (FULL): Tx FIFO is fullVolatile: true	R
0	ACTIVITY	I2C Activity Status. Reset value: 0x0Values:0x1 (ACTIVE): I2C is active0x0 (INACTIVE): I2C is idleVolatile: true	R

3.18.4.25 IC_TXFLR(0x74)

Bits	Name	Description	Memory Access
31:4	RSVD_TXFLR	TXFLR Register field Reserved bits - Read OnlyVolatile: true	R
3:0	TXFLR	Transmit FIFO Level. Contains the number of valid data entries in the transmit FIFO.Reset value: 0x0Volatile: true	R

3.18.4.26 IC_RXFLR(0x78)

Bits	Name	Description	Memory Access
31:4	RSVD_RXFLR	RXFLR Reserved bits - Read OnlyVolatile: true	R
3:0	RXFLR	Receive FIFO Level. Contains the number of valid data entries in the receive FIFO.Reset value: 0x0Volatile: true	R

3.18.4.27 IC_SDA_HOLD(0x7c)

Bits	Name	Description	Memory Access
31:24	RSVD_IC_SDA_HOLD	IC_SDA_HOLD Reserved bits - Read Only	R
23:16	IC_SDA_RX_HOLD	Sets the required SDA hold time in units of ic_clk period, when apb_i2c acts as a receiver.Reset value: IC_DEFAULT_SDA_HOLD[23:16].	R/W
15:0	IC_SDA_TX_HOLD	Sets the required SDA hold time in units of ic_clk period, when apb_i2c acts as a transmitter.Reset value: IC_DEFAULT_SDA_HOLD[15:0].	R/W

3.18.4.28 IC_TX_ABRT_SOURCE(0x80)

Bits	Name	Description	Memory Access
31:23	TX_FLUSH_CNT	This field indicates the number of Tx FIFO Data Commands which are fl ushed due to TX_ABRT interrupt. It is cleared whenever I2C is disabled.Reset value: 0x0Role of apb_i2c: Master-Transmitter or Slave- TransmitterVolatile: true	R
22:21	RSVD_IC_TX_ABRT_SOURCE	IC_TX_ABRT_SOURCE Reserved bits - Read OnlyVolatile: true	R
20:18	RSVD_ABRT_DEVICE_WRITE	ABRT_DEVICE_WRITE Register fi eld Reserved bits - Read OnlyVolatile: true	R
17	RSVD_ABRT_SDA_STUCK_AT_ LOW	ABRT_SDA_STUCK_AT_LOW Register field Reserved bits - Read OnlyVolatile: true	R
16	ABRT_USER_ABRT	This is a master-mode-only bit. Master has detected the transfer abort (IC_ENABLE[1])Reset value: 0x0Role of apb_i2c: Master-TransmitterValues:0x1 (ABRT_USER_ABRT_GENERATED): Transfer abort detected by master0x0 (ABRT_USER_ABRT_VOID): Transfer abort detected by master-scenario not presentVolatile: true	R
15	ABRT_SLVRD_INTX	1: When the processor side responds to a slave mode request for data to be transmitted to a remote master and user writes a 1 in CMD (bit 8) of IC_DATA_CMD register.Reset value: 0x0Role of apb_i2c: Slave-TransmitterValues:0x1 (ABRT_SLVRD_INTX_GENERATED): Slave trying to transmit to remote master in read mode0x0 (ABRT_SLVRD_INTX_VOID): Slave	R

		trying to transmit to remote master in read mode- scenario not presentVolatile: true	
14	ABRT_SLV_ARBLOST	This field indicates that a Slave has lost the bus while transmitting data to a remote master. IC_TX_ABRT_SOURCE[12] is set at the same time.Note: Even though the slave never owns the bus, something could go wrong on the bus. This is a fail safe check. For instance, during a data transmission at the low-to-high transition of SCL, if what is on the data bus is not what is supposed to be transmitted, then apb_i2c no longer own the bus.Reset value: 0x0Role of apb_i2c: Slave-TransmitterValues:0x1 (ABRT_SLV_ARBLOST_GENERATED): Slave lost arbitration to remote master0x0 (ABRT_SLV_ARBLOST_VOID): Slave lost arbitration to remote master- scenario not presentVolatile: true	R

Bits	Name	Description	Memory Access
13	ABRT_SLVFLUSH_TXFIFO	This field specifies that the Slave has received a read command and some data exists in the TX FIFO, so the slave issues a TX_ABRT interrupt to flush old data in TX FIFO.Reset value: 0x0Role of apb_i2c: Slave-TransmitterValues:0x1 (ABRT_SLVFLUSH_TXFIFO_GENERATED): Slave flushes existing data in TX-FIFO upon getting read command0x0 (ABRT_SLVFLUSH_TXFIFO_VOID): Slave flushes existing data in TX-FIFO upon getting read command-scenario not presentVolatile: true	R
12	ARB_LOST	This field specifies that the Master has lost arbitration, or if IC_TX_ABRT_SOURCE[14] is also set, then the slave transmitter has lost arbitration.Reset value: 0x0Role of apb_i2c: Master-Transmitter or Slave- TransmitterValues:0x1 (ABRT_LOST_GENERATED): Master or Slave- Transmitter lost arbitration0x0 (ABRT_LOST_VOID): Master or Slave- Transmitter lost arbitration- scenario not presentVolatile: true	R
11	ABRT_MASTER_DIS	This field indicates that the User tries to initiate a Master operation with the Master mode disabled.Reset value: 0x0Role of apb_i2c: Master-Transmitter or Master-ReceiverValues:0x1 (ABRT_MASTER_DIS_GENERATED): User intitating master operation when	R

		MASTER disabled0x0 (ABRT_MASTER_DIS_VOID): User initiating master operation when MASTER disabled- scenario not presentVolatile: true	
10	ABRT_10B_RD_NORSTRT	This field indicates that the restart is disabled (IC_RESTART_EN bit (IC_CON[5]) =0) and the master sends a read command in 10-bit addressing mode.Reset value: 0x0Role of apb_i2c: Master-ReceiverValues:0x1 (ABRT_10B_RD_GENERATED): Master trying to read in 10Bit addressing mode when RESTART disabled0x0 (ABRT_10B_RD_VOID): Master not trying to read in 10Bit addressing mode when RESTART disabledVolatile: true	R

Bits	Name	Description	Memory Access
9	ABRT_SBYTE_NORSTRT	To clear Bit 9, the source of the ABRT_SBYTE_NORSTRT must be fixed first; restart must be enabled (IC_CON[5]=1), the SPECIAL bit must be cleared (IC_TAR[11]), or the GC_OR_START bit must be cleared (IC_TAR[10]). Once the source of the ABRT_SBYTE_NORSTRT is fixed, then this bit can be cleared in the same manner as other bits in this register. If the source of the ABRT_SBYTE_NORSTRT is not fixed before attempting to clear this bit, bit 9 clears for one cycle and then gets reasserted. When this field is set to 1, the restart is disabled (IC_RESTART_EN bit (IC_CON[5]) =0) and the user is trying to send a START Byte.Reset value: 0x0Role of apb_i2c: MasterValues:0x1 (ABRT_SBYTE_NORSTRT_GENERATED): User trying to send START byte when RESTART disabled0x0 (ABRT_SBYTE_NORSTRT_VOID): User trying to send START byte when RESTART disabled- scenario not presentVolatile: true	R
8	ABRT_HS_NORSTRT	This field indicates that the restart is disabled (IC_RESTART_EN bit (IC_CON[5]) =0) and the user is trying to use the master to transfer data in High Speed mode.Reset value: 0x0Role of apb_i2c: Master-Transmitter or Master-ReceiverValues:0x1 (ABRT_HS_NORSTRT_GENERATED): User	R

		trying to switch Master to HS mode when RESTART disabled0x0 (ABRT_HS_NORSTRT_VOID): User trying to switch Master to HS mode when RESTART disabled- scenario not presentVolatile: true	
7	ABRT_SBYTE_ACKDET	This field indicates that the Master has sent a START Byte and the START Byte was acknowledged (wrong behavior).Reset value: 0x0Role of apb_i2c: MasterValues:0x1 (ABRT_SBYTE_ACKDET_GENERATED): ACK detected for START byte0x0 (ABRT_SBYTE_ACKDET_VOID): ACK detected for START byte- scenario not presentVolatile: true	R
6	ABRT_HS_ACKDET	This field indicates that the Master is in High Speed mode and the High Speed Master code was acknowledged (wrong behavior).Reset value: 0x0Role of apb_i2c: MasterValues:0x1 (ABRT_HS_ACK_GENERATED): HS Master code ACKed in HS Mode0x0 (ABRT_HS_ACK_VOID): HS Master code ACKed in HS Mode- scenario not presentVolatile: true	R

Bits	Name	Description	Memory Access
5	ABRT_GCALL_READ	This field indicates that apb_i2c in the master mode has sent a General Call but the user programmed the byte following the General Call to be a read from the bus (IC_DATA_CMD[9] is set to 1).Reset value: 0x0Role of apb_i2c: Master-TransmitterValues:0x1 (ABRT_GCALL_READ_GENERATED): GCALL is followed by read from bus0x0 (ABRT_GCALL_READ_VOID): GCALL is followed by read from bus-scenario not presentVolatile: true	R
4	ABRT_GCALL_NOACK	This field indicates that apb_i2c in master mode has sent a General Call and no slave on the bus acknowledged the General Call.Reset value: 0x0Role of apb_i2c: Master-TransmitterValues:0x1 (ABRT_GCALL_NOACK_GENERATED): GCALL not ACKed by any slave0x0 (ABRT_GCALL_NOACK_VOID): GCALL not ACKed by any slave-scenario not presentVolatile: true	R
3	ABRT_TXDATA_NOACK	This field indicates the master-mode only bit. When the master receives an acknowledgement for the address, but when it sends data byte(s) following the address, it did not receive an acknowledge from the remote slave(s).Reset value: 0x0Role of apb_i2c: Master-TransmitterValues:0x1 (ABRT_TXDATA_NOACK_GENERATED):	R

		Transmitted data not ACKed by addressed slave0x0 (ABRT_TXDATA_NOACK_VOID): Transmitted data non-ACKed by addressed slave-scenario not presentVolatile: true	
2	ABRT_10ADDR2_NOACK	This field indicates that the Master is in 10-bit address mode and that the second address byte of the 10-bit address was not acknowledged by any slave.Reset value: 0x0Role of apb_i2c: Master-Transmitter or Master-ReceiverValues:0x1 (ACTIVE): Byte 2 of 10Bit Address not ACKed by any slave0x0 (INACTIVE): This abort is not generatedVolatile: true	R
1	ABRT_10ADDR1_NOACK	This field indicates that the Master is in 10-bit address mode and the first 10-bit address byte was not acknowledged by any slave.Reset value: 0x0Role of apb_i2c: Master-Transmitter or Master-ReceiverValues:0x1 (ACTIVE): Byte 1 of 10Bit Address not ACKed by any slave0x0 (INACTIVE): This abort is not generatedVolatile: true	R
0	ABRT_7B_ADDR_NOACK	This field indicates that the Master is in 7-bit addressing mode and the address sent was not acknowledged by any slave.Reset value: 0x0Role of apb_i2c: Master-Transmitter or Master-ReceiverValues:0x1 (ACTIVE): This abort is generated because of NOACK for 7-bit address0x0 (INACTIVE): This abort is	R

		not generatedVolatile: true	

3.18.4.29 IC_DMA_CR(0x88)

Bits	Name	Description	Memory Access
31:2	RSVD_IC_DMA_CR_2_31	RSVD_IC_DMA_CR_2_31 Reserved bits - Read Only	R
1	TDMAE	Transmit DMA Enable. This bit enables/disables the transmit FIFO DMA channel. Reset value: 0x0Values:0x1 (ENABLED): Transmit FIFO DMA channel enabled0x0 (DISABLED): transmit FIFO DMA channel disabled	R/W
0	RDMAE	Receive DMA Enable. This bit enables/disables the receive FIFO DMA channel. Reset value: 0x0Values:0x1 (ENABLED): Receive FIFO DMA channel enabled0x0 (DISABLED): Receive FIFO DMA channel disabled	R/W

3.18.4.30 IC_DMA_TDLR(0x8c)

Bits	Name	Description	Memory Access
31:3	RSVD_DMA_TDLR	DMA_TDLR Reserved bits - Read Only	R
2:0	DMATDL	Transmit Data Level. This bit field controls the level at which a DMA request is made by the transmit logic. It is equal to the watermark level; that is, the dma_tx_req signal is generated when the number of valid data entries in the transmit FIFO is equal to or below this field value, and TDMAE = 1.Reset value: 0x0	R/W

3.18.4.31 IC_DMA_RDLR(0x90)

Bits	Name	Description	Memory Access
31:3	RSVD_DMA_RDLR	DMA_RDLR Reserved bits - Read Only	R
2:0	DMARDL	Receive Data Level. This bit field controls the level at which a DMA request is made by the receive logic. The watermark level = DMARDL+1; that is, dma_rx_req is generated when the number of valid data entries in the receive FIFO is equal to or more than this field value + 1, and RDMAE =1. For instance, when DMARDL is 0, then dma_rx_req is asserted when 1 or more data entries are present in the receive FIFO.Reset value: 0x0	R/W

3.18.4.32 IC_SDA_SETUP(0x94)

Bits	Name	Description	Memory Access
31:8	RSVD_IC_SDA_SETUP	IC_SDA_SETUP Reserved bits - Read Only	R
7:0	SDA_SETUP	SDA Setup. It is recommended that if the required delay is 1000ns, then for an ic_clk frequency of 10 MHz, IC_SDA_SETUP should be programmed to a value of 11. IC_SDA_SETUP must be programmed with a minimum value of 2.Reset value: 0x64, but can be hardcoded by setting the IC_DEFAULT_SDA_SETUP configuration parameter.	R/W

3.18.4.33 IC_ACK_GENERAL_CALL(0x98)

Bits	Name	Description	Memory Access
31:1	RSVD_IC_ACK_GEN_1_31	RSVD_IC_ACK_GEN_1_31 Reserved bits - Read Only	R

0	ACK_GEN_CALL	ACK General Call. When set to 1, apb_i2c responds with a ACK (by asserting ic_data_oe) when it receives a General Call. Otherwise, apb_i2c responds with a NACK (by negating ic_data_oe).Reset value: 0x1, but can be hardcoded by setting the IC_DEFAULT_ACK_GENERAL_CALL configuration parameter.Values:0x1 (ENABLED): Generate ACK for a General Call0x0 (DISABLED): Generate NACK for General Call	R/W

3.18.4.34 IC_ENABLE_STATUS(0x9c)

Bits	Name	Description	Memory Access
31:3	RSVD_IC_ENABLE_STATUS	IC_ENABLE_STATUS Reserved bits - Read OnlyVolatile: true	R
2	SLV_RX_DATA_LOST	Slave Received Data Lost. This bit indicates if a Slave-Receiver operation has been aborted with at least one data byte received from an I2C transfer due to the setting bit 0 of IC_ENABLE from 1 to 0. When read as 1, apb_i2c is deemed to have been actively engaged in an aborted I2C transfer (with matching address) and the data phase of the I2C transfer has been entered, even though a data byte has been responded with a NACK.Note: If the remote I2C master terminates the transfer with a STOP condition before the apb_i2c has a chance to NACK a transfer, and IC_ENABLE[0] has been set to 0, then this bit is also set to 1.When read as 0, apb_i2c is deemed to have been disabled without being	R

		actively involved in the data phase of a Slave- Receiver transfer.Note: The CPU can safely read this bit when IC_EN (bit 0) is read as 0.Reset value: 0x0Values:0x1 (ACTIVE): Slave RX Data is lost0x0 (INACTIVE): Slave RX Data is not lostVolatile: true	
1	SLV_DISABLED_WHILE_BUSY	Slave Disabled While Busy (Transmit, Receive). This bit indicates if a potential or active Slave operation has been aborted due to the setting bit 0 of the IC_ENABLE register from 1 to 0. This bit is set when the CPU writes a 0 to the IC_ENABLE register while:(a) apb_i2c is receiving the address byte of the Slave-Transmitter operation from a remote master;OR, (b) address and data bytes of the Slave-Receiver operation from a remote master.When read as 1, apb_i2c is deemed to have forced a NACK during any part of an I2C transfer, irrespective of whether the I2C address matches the slave address set in apb_i2c (IC_SAR register) OR if the transfer is completed before IC_ENABLE is set to 0 but has not taken effect.Note: If the remote I2C master terminates the transfer with a STOP condition before the apb_i2c has a chance to NACK a transfer, and IC_ENABLE[0] has been set to 0, then this bit will also be set to 1.When read as 0, apb_i2c is deemed to have been disabled	R

		when there is master activity, or when the I2C bus is idle.Note: The CPU can safely read this bit when IC_EN (bit 0) is read as 0.Reset value: 0x0Values:0x1 (ACTIVE): Slave is disabled when it is active0x0 (INACTIVE): Slave is disabled when it is idleVolatile: true	
0	IC_EN	ic_en Status. This bit always reflects the value driven on the output port ic_en.When read as 1, apb_i2c is deemed to be in an enabled state.When read as 0, apb_i2c is deemed completely inactive.Note: The CPU can safely read this bit anytime. When this bit is read as 0, the CPU can safely read SLV_RX_DATA_LOST (bit 2) and SLV_DISABLED_WHILE_BUSY (bit 1).Reset value: 0x0Values:0x1 (ENABLED): I2C enabled0x0 (DISABLED): I2C disabledVolatile: true	R

3.18.4.35 IC_FS_SPKLEN(0xa0)

Bits	Name	Description	Memory Access
31:8	RSVD_IC_FS_SPKLEN	IC_FS_SPKLEN Reserved bits - Read Only	R
7:0	IC_FS_SPKLEN	This register must be set before any I2C bus transaction can take place to ensure stable operation. This register sets the duration, measured in ic_clk cycles, of the longest spike in the SCL or SDA lines that will be filtered out by the spike suppression logic. This register can be written only when the I2C interface is disabled which corresponds to the IC_ENABLE[0] register being set to 0. Writes at other times have no effect. The minimum valid value is 1; hardware prevents values less than this being written, and if attempted results in 1 being set. or more information, refer to Spike Suppression .Reset value: IC_DEFAULT_FS_SPKLEN configuration parameter.	R/W

3.18.4.36 IC_COMP_PARAM_1(0xf4)

Bits	Name	Description	Memory Access
31:24	RSVD_IC_COMP_PARAM_1	IC_COMP_PARAM_1 Reserved bits - Read Only	R
23:16	TX_BUFFER_DEPTH	The value of this register is derived from the IC_TX_BUFFER_DEPTH coreConsultant parameter.0x00 = Reserved0x01 = 20x02 = 3...0xFF = 256	R

15:8	RX_BUFFER_DEPTH	The value of this register is derived from the IC_RX_BUFFER_DEPTH coreConsultant parameter.0x00: Reserved0x01: 20x02: 3...0xFF: 256	R
7	ADD_ENCODED_PARAMS	The value of this register is derived from the IC_ADD_ENCODED_PARAMS coreConsultant parameter. Reading 1 in this bit means that the capability of reading these encoded parameters via software has been included. Otherwise, the entire register is 0 regardless of the setting of any other parameters that are encoded in the bits.Values:0x1 (ENABLED): Enables capability of reading encoded parameters0x0 (DISBALED): Disables capability of reading encoded parameters	R
6	HAS_DMA	The value of this register is derived from the IC_HAS_DMA coreConsultant parameter.Values:0x1 (ENABLED): DMA handshaking signals are enabled0x0 (DISABLED): DMA handshaking signals are disabled	R
5	INTR_IO	The value of this register is derived from the IC_INTR_IO coreConsultant parameter.Values:0x1 (COMBINED): COMBINED Interrupt outputs0x0 (INDIVIDUAL): INDIVIDUAL Interrupt outputs	R
4	HC_COUNT_VALUES	The value of this register is derived from the IC_HC_COUNT VALUES coreConsultant parameter.Values:0x1 (ENABLED): Hard code the count values for each	R

		mode.0x0 (DISABLED): Programmable count values for each mode.	
3:2	MAX_SPEED_MODE	The value of this register is derived from the IC_MAX_SPEED_MODE coreConsultant parameter.0x0: Reserved0x1: Standard0x2: Fast0x3: HighValues:0x1 (STANDARD): IC MAX SPEED is STANDARD MODE0x2 (FAST): IC MAX SPEED is FAST MODE0x3 (HIGH): IC MAX SPEED is HIGH MODE	R
1:0	APB_DATA_WIDTH	The value of this register is derived from the APB_DATA_WIDTH coreConsultant parameter.Values:0x0 (APB_08BITS): APB data bus width is 08 bits0x1 (APB_16BITS): APB data bus width is 16 bits0x2 (APB_32BITS): APB data bus width is 32 bits0x3 (RESERVED): Reserved bits	R

3.18.4.37 IC_COMP_VERSION(0xf8)

Bits	Name	Description	Memory Access
31:0	IC_COMP_VERSION	Specific values for this register are described in the Releases Table in the apb_i2c Release Notes	R

3.18.4.38 IC_COMP_TYPE(0xfc)

Bits	Name	Description	Memory Access
31:0	IC_COMP_TYPE	Designware Component Type number = 0x44_57_01_40. This assigned unique hex value is constant and is derived from the two ASCII letters DW followed by a 16-bit unsigned number.	R

3.19 串行外设接口 (SPI)

3.19.1 概述

串行外设接口包括4组SPI接口，其中SPI0、SPI1、SPI3只能工作在MASTER模式，SPI2只能工作在SLAVE模式，主要功能包括：

串行外设接口包括4组SPI接口，其中SPI0、SPI1、SPI3只能工作在MASTER模式，SPI2只能工作在SLAVE模式，它们有如下特性：

- 支持1/2/4/8线全双工模式；
- SPI0、SPI1、SPI2可支持25MHz时钟（待测更新） SPI3最高可支持100MHz时钟；
- 支持32位宽、32BYTE深的FIFO；
- 独立屏蔽中断模块：包括主机冲突、发送FIFO溢出、发送FIFO空、接收FIFO满、接收FIFO下溢、接收FIFO溢出中断、都可以被独立屏蔽；
- 支持DMA功能；
- 支持双沿的DDR传输模式SPI3支持XIP。

3.19.2 功能描述

- GP-SPI 主机模式

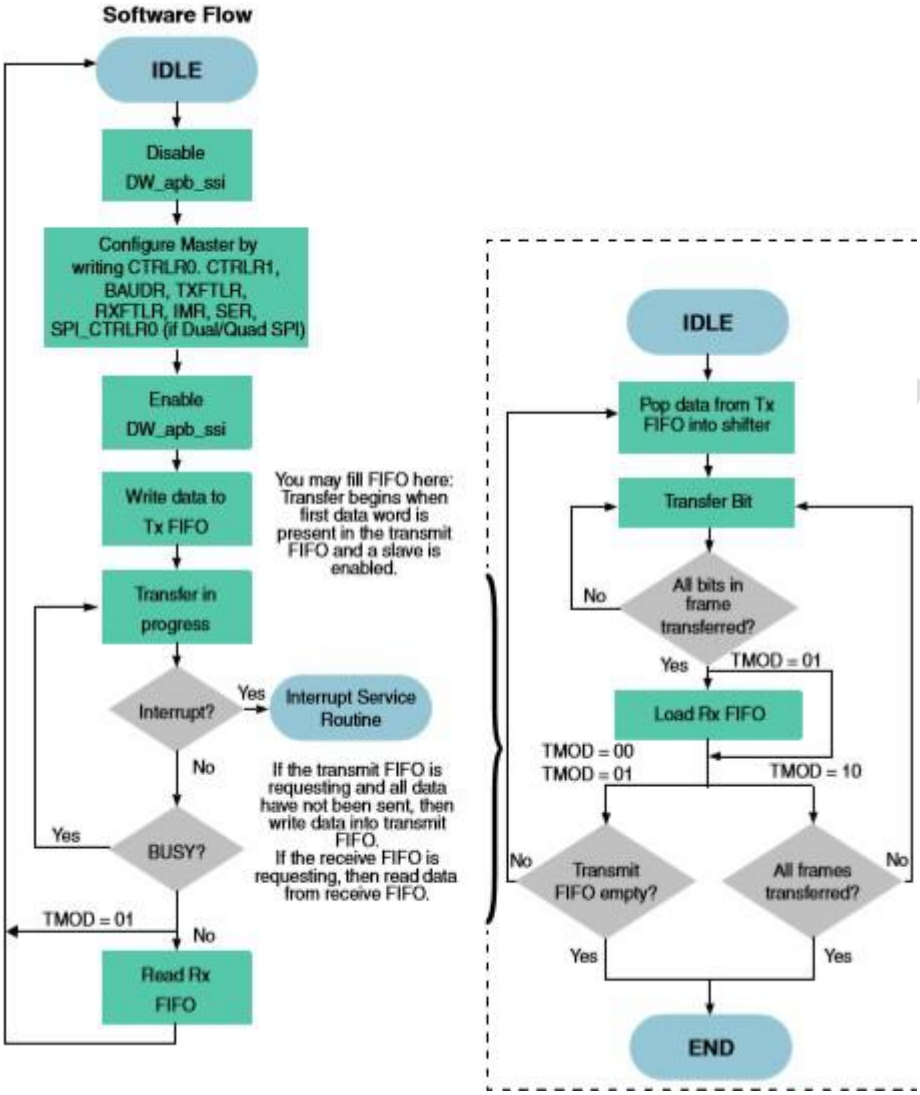
SPI主机支持四线全双工通信和三线半双工通信。SPI的四线全双工通信中，需要配置接收和发送数据的长度。SPI的三线半双工通信中，如果使用SPI作为从机，那么主机和从机的通信需要满足一定的通信格式。以SPI作为从机为

例，通信格式必须满足：命令 + 地址 + 收 / 发数据。要求主机地址和从机地址长度相等，地址的值为0。

- GP-SPI从机模式

SPI2只能作为从机，与其他主机进行通信。SPI 作为从机时，需要满足特定的通信协议。如果不使用DMA，一次最长可以接收 / 发送 64 byte 的数据。在一次读 / 写过程中，片选信号CS必须保持低电平。如果在发送过程中CS被拉高，从机内部状态将会复位。

- 操作步骤



3.19.3 寄存器列表

Base Address：SPI0_BASE_ADDR (0x52000000)

Base Address：SPI1_BASE_ADDR (0x53000000)

Base Address：SPI3_BASE_ADDR (0x54000000)

Name	Description	Offset address
CTRLR0	This register controls the serial data transfer.	0x0
CTRLR1	This register exists only when the SPI is configured as a master device.	0x4
SSIENR	This register enables and disables the SPI.	0x8
MWCR	This register controls the direction of the data word for the half-duplex-Microwire serial protocol.	0xc
SER	This register is valid only when the SPI is configured as a master device.	0x10
BAUDR	This register is valid only when the SPI is configured as a master device.	0x14
TXFTLR	This register controls the threshold value for the transmit FIFO memory.	0x18
RXFTLR	This register controls the threshold value for the receive FIFO memory.	0x1c
TXFLR	This register contains the number of valid data entries in the transmit FIFO memory.	0x20
RXFLR	This register contains the number of valid data entries in the receive FIFO memory. This register can be ready at any time.	0x24
SR	This is a read-only register used to indicate the current transfer status, FIFO status, and any transmission/reception errors that may have occurred.The status register may be read at any time.	0x28

Name	Description	Offset address
IMR	This read/write reigster masks or enables all interrupts generated by the SPI.	0x2c
ISR	This register reports the status of the SPI interrupts after they have been masked.	0x30
RISR	This read-only register reports the status of the SPI interrupts prior to masking.	0x34
TXOICR	Transmit FIFO Overflow Interrupt Clear Register.	0x38

RXOICR	Receive FIFO Overflow Interrupt Clear Register.	0x3c
RXUICR	Receive FIFO Underflow Interrupt Clear Register.	0x40
MSTICR	Multi-Master Interrupt Clear Register.	0x44
ICR	Interrupt Clear Register.	0x48
DMACR	This register is only valid when SPI is configured with a set of DMA Controller interface signals.	0x4c
DMATDLR	This register is only valid when the apb_ssi is configured with a set of DMA interface signals (SSI_HAS_DMA = 1). When apb_ssi is not configured for DMA operation, this register will not exist and writing to its address will have no effect; reading from its address will return zero.	0x50
DMARDLR	This register is only valid when apb_ssi is configured with a set of DMA interface signals (SSI_HAS_DMA = 1). When apb_ssi is not configured for DMA operation, this register will not exist and writing to its address will have no effect; reading from its address will return zero.	0x54
IDR	This register contains the peripherals identification code, which is written into the register at configuration time using coreConsultant.	0x58
SSI_VERSION_ID	This read-only register stores the specific apb_ssi component version.	0x5c
DR0	The data register is a 16/32-bit read/write buffer for the transmit/receive FIFOs.	0x60
DR1	The data register is a 16/32-bit read/write buffer for the transmit/receive FIFOs.	0x64
DR2	The data register is a 16/32-bit read/write buffer for the transmit/receive FIFOs.	0x68
DR3	The data register is a 16/32-bit read/write buffer for the transmit/receive FIFOs.	0x6c

Name	Description	Offset address
DR4	The data register is a 16/32-bit read/write buffer for the transmit/receive FIFOs.	0x70

DR5	The data register is a 16/32-bit read/write buffer for the transmit/receive FIFOs.	0x74
DR6	The data register is a 16/32-bit read/write buffer for the transmit/receive FIFOs.	0x78
DR7	The data register is a 16/32-bit read/write buffer for the transmit/receive FIFOs.	0x7c
DR8	The data register is a 16/32-bit read/write buffer for the transmit/receive FIFOs.	0x80
DR9	The data register is a 16/32-bit read/write buffer for the transmit/receive FIFOs.	0x84
DR10	The data register is a 16/32-bit read/write buffer for the transmit/receive FIFOs.	0x88
DR11	The data register is a 16/32-bit read/write buffer for the transmit/receive FIFOs.	0x8c
DR12	The data register is a 16/32-bit read/write buffer for the transmit/receive FIFOs.	0x90
DR13	The data register is a 16/32-bit read/write buffer for the transmit/receive FIFOs.	0x94
DR14	The data register is a 16/32-bit read/write buffer for the transmit/receive FIFOs.	0x98
DR15	The data register is a 16/32-bit read/write buffer for the transmit/receive FIFOs.	0x9c
DR16	The data register is a 16/32-bit read/write buffer for the transmit/receive FIFOs.	0xa0
DR17	The data register is a 16/32-bit read/write buffer for the transmit/receive FIFOs.	0xa4
DR18	The data register is a 16/32-bit read/write buffer for the transmit/receive FIFOs.	0xa8
DR19	The data register is a 16/32-bit read/write buffer for the transmit/receive FIFOs.	0xac
DR20	The data register is a 16/32-bit read/write buffer for the transmit/receive FIFOs.	0xb0
DR21	The data register is a 16/32-bit read/write buffer for the transmit/receive FIFOs.	0xb4

Name	Description	Offset address
DR22	The data register is a 16/32-bit read/write buffer for the transmit/receive FIFOs.	0xb8
DR23	The data register is a 16/32-bit read/write buffer for the transmit/receive FIFOs.	0xbc
DR24	The data register is a 16/32-bit read/write buffer for the transmit/receive FIFOs.	0xc0
DR25	The data register is a 16/32-bit read/write buffer for the transmit/receive FIFOs.	0xc4
DR26	The data register is a 16/32-bit read/write buffer for the transmit/receive FIFOs.	0xc8
DR27	The data register is a 16/32-bit read/write buffer for the transmit/receive FIFOs.	0xcc
DR28	The data register is a 16/32-bit read/write buffer for the transmit/receive FIFOs.	0xd0
DR29	The data register is a 16/32-bit read/write buffer for the transmit/receive FIFOs.	0xd4
DR30	The data register is a 16/32-bit read/write buffer for the transmit/receive FIFOs.	0xd8
DR31	The data register is a 16/32-bit read/write buffer for the transmit/receive FIFOs.	0xdc
DR32	The data register is a 16/32-bit read/write buffer for the transmit/receive FIFOs.	0xe0
DR33	The data register is a 16/32-bit read/write buffer for the transmit/receive FIFOs.	0xe4
DR34	The data register is a 16/32-bit read/write buffer for the transmit/receive FIFOs.	0xe8
DR35	The data register is a 16/32-bit read/write buffer for the transmit/receive FIFOs.	0xec
RX_SAMPLE_DLY	This register is only valid when the SPI is configured with rxd sample delay logic.	0xf0

SPI_CTRLR0	This register is valid only when SSI_SPI_MODE is either set to Dual or Quad or Octal mode.	0xf4
RSVD	RSVD - Reserved address location.	0xfc

3.19.4 寄存器设置

3.19.4.1 CTRLR0(0x0)

Bits	Name	Description	Memory Access
31:25	RSVD_CTRLR0	SSTE Reserved bits	R
24	SSTE	Slave Select Toggle Enable. When operating in SPI mode with clock phase (SCPH) set to 0, this register controls the behavior of the slave select line (ss__n) between data frames. If this register field is set to 1 the ss__n line will toggle between consecutive data frames, with the serial clock (sclk) being held to its default value while ss__n is high; if this register field is set to 0 the ss__n will stay low and sclk will run continuously for the duration of the transfer. Note: This register is only valid when SSI_SCPH0_SSTOGGLE is set to 1.	R/W
23	RSVD_CTRLR0_23	CTRLR0_23 Reserved bits	R
22:21	SPI_FRF	SPI Frame Format: Selects data frame format for Transmitting/Receiving the dataBits only valid when SSI_SPI_MODE is either set to Dual or Quad or Octal mode.When SSI_SPI_MODE is configured for Dual Mode , 10/11 combination is reserved.When SSI_SPI_MODE is configured for Quad Mode , 11 combination is reserved. Values: 0x0 (STD_SPI_FRF): Standard SPI Frame Format 0x1 (DUAL_SPI_FRF): Dual SPI Frame Format 0x2 (QUAD_SPI_FRF): Quad SPI Frame	R/W

		Format 0x3 (OCTAL_SPI_FRF): Octal SPI Frame Format	
20:16	DFS_32	Data Frame Size in 32-bit transfer size mode. Used to select the data frame size in 32-bit transfer mode. These bits are only valid when SSI_MAX_XFER_SIZE is configured to 32. When the data frame size is programmed to be less than 32 bits, the receive data are automatically right-justified by the receive logic, with the upper bits of the receive FIFO zero-padded. You are responsible for making sure that transmit data is right-justified before writing into the transmit FIFO. The transmit logic ignores the upper unused bits when transmitting the data. Note: When SSI_SPI_MODE is either set to Dual or Quad or Octal mode and SPI_FRF is not set to 2 b00.- DFS value should be multiple of 2 if SPI_FRF = 0x01,- DFS value should be multiple of 4 if SPI_FRF = 0x10,- DFS value should be multiple of 8 if SPI_FRF = 0x11. Values: 0x3 (FRAME_04BITS): 4-bit serial data transfe r0x4 (FRAME_05BITS): 5-bit serial data transfer 0x5 (FRAME_06BITS): 6-bit serial data transfer 0x6 (FRAME_07BITS): 7-bit serial data transfer 0x7 (FRAME_08BITS): 8-bit serial data transfer 0x8 (FRAME_09BITS): 9-bit serial data transfe r0x9 (FRAME_10BITS): 10-bit serial data transfer 0xa (FRAME_11BITS): 11-bit serial data transfer 0xb (FRAME_12BITS): 12-bit serial data transfer 0xc (FRAME_13BITS): 13-bit serial data transfer 0xd (FRAME_14BITS): 14-bit serial data transfer 0xe (FRAME_15BITS): 15-bit serial data transfer 0xf (FRAME_16BITS): 16-bit serial data transfer 0x10 (FRAME_17BITS): 17-bit serial data transfer 0x11 (FRAME_18BITS): 18-bit serial data transfer 0x12 (FRAME_19BITS): 19-bit serial data transfer 0x13 (FRAME_20BITS): 20-bit serial data transfer 0x14	R/W

		(FRAME_21BITS): 21-bit serial data transfer 0x15 (FRAME_22BITS): 22-bit serial data transfer 0x16 (FRAME_23BITS): 23-bit serial data transfer 0x17 (FRAME_24BITS): 24-bit serial data transfer 0x18 (FRAME_25BITS): 25-bit serial data transfer 0x19 (FRAME_26BITS): 26-bit serial data transfer 0x1a (FRAME_27BITS): 27-bit serial data transfer 0x1b (FRAME_28BITS): 28-bit serial data transfer 0x1c (FRAME_29BITS): 29-bit serial data transfer 0x1d (FRAME_30BITS): 30-bit serial data transfer 0x1e (FRAME_31BITS): 31-bit serial data transfer 0x1f (FRAME_32BITS): 32-bit serial data transfer	

Bits	Name	Description	Memory Access
15:12	CFS	Control Frame Size. Selects the length of the control word for the Microwire frame format. Values: 0x0 (SIZE_01_BIT): 1-bit Control Word 0x1 (SIZE_02_BIT): 2-bit Control Word 0x2 (SIZE_03_BIT): 3-bit Control Word 0x3 (SIZE_04_BIT): 4-bit Control Word 0x4 (SIZE_05_BIT): 5-bit Control Word 0x5 (SIZE_06_BIT): 6-bit Control Word 0x6 (SIZE_07_BIT): 7-bit Control Word 0x7 (SIZE_08_BIT): 8-bit Control Word 0x8 (SIZE_09_BIT): 9-bit Control Word 0x9 (SIZE_10_BIT): 10-bit Control Word 0xa (SIZE_11_BIT): 11-bit Control Word 0xb (SIZE_12_BIT): 12-bit Control Word 0xc (SIZE_13_BIT): 13-bit Control Word 0xd (SIZE_14_BIT): 14-bit Control Word 0xe (SIZE_15_BIT): 15-bit Control Word 0xf	R/W

		(SIZE_16_BIT): 16-bit Control Word	
11	SRL	Shift Register Loop. Used for testing purposes only. When internally active, connects the transmit shift register output to the receive shift register input.Can be used in both serial-slave and serial-master modes.When the ssi is configured as a slave in loopback mode, the ss_in_n and ssi_clk signals must be provided by an external source. In this mode, the slave cannot generate these signals because there is nothing to which to loop back Values: 0x1 (TESTING_MODE): Test mode: Tx & Rx shift reg connected 0x0 (NORMAL_MODE): Normal mode operation	R/W
10	RSVD_SLV_OE	SLV_OE Reserved field - Read-only	R
		Transfer Mode. Selects the mode of transfer for serial communication. This field does not affect the transfer duplicity. Only indicates whether the receive or transmit data are valid. In transmit-only mode, data received from the external device is not valid and is not stored in the receive FIFO memory; it is overwritten on the next transfer. In receive-only mode, transmitted data are not valid. After the first write to the transmit FIFO, the same word is retransmitted for the duration of the transfer. In transmit-and-receive mode, both transmit and receive data are valid. The transfer continues until the transmit FIFO is empty. Data received from the external device are stored into the receive FIFO memory, where it can be accessed by the host processor. In eeprom-read mode, receive data is not valid while control data is being transmitted. When all	

9:8	TMOD	control data is sent to the EEPROM, receive data becomes valid and transmit data becomes invalid. All data in the transmit FIFO is considered control data in this mode. This transfer mode is only valid when the ssi is configured as master device. - Transmit & Receive - Transmit Only 10 - Receive Only 11 - EEPROM Read When SSI_SPI_MODE is either set to Dual or Quad or Octal mode and SPI_FRF is not set to 2 b00. There are only two valid combinations: 10 - Read 01 - Write Values: 0x0 (TX_AND_RX): Transmit & receive 0x1 (TX_ONLY): Transmit only mode or Write (SPI_FRF != 2 b00) 0x2 (RX_ONLY): Receive only mode or Read (SPI_FRF != 2 b00) 0x3 (EEPROM_READ): EEPROM Read mode	R/W
7	SCPOL	Serial Clock Polarity. Valid when the frame format (FRF) is set to Motorola SPI. Used to select the polarity of the inactive serial clock, which is held inactive when the ssi master is not actively transferring data on the serial bus. Values: 0x0 (SCLK_LOW): Inactive state of serial clock is low 0x1 (SCLK_HIGH): Inactive state of serial clock is high	R/W
6	SCPH	Serial Clock Phase. Valid when the frame format (FRF) is set to Motorola SPI. The serial clock phase selects the relationship of the serial clock with the slave select signal.When SCPH = 0, data are captured on the first edge of the serial clock. When SCPH = 1, the serial clock starts toggling one cycle after the slave select line is activated, and data are captured on the second edge of the	R/W

		serial clock. Values: 0x0 (SCPH_MIDDLE): Serial clock toggles in middle of first data bit 0x1 (SCPH_START): Serial clock toggles at start of first data bit	
5:4	FRF	Frame Format. Selects which serial protocol transfers the data. Values: 0x0 (MOTOROLA_SPI): Motorolla SPI Frame Format 0x1 (TEXAS_SSP): Texas Instruments SSP Frame Format 0x2 (NS_MICROWIRE): National Microwire Frame Format 0x3 (RESERVED): Reserved value	R/W
3:0	DFS	Data Frame Size. This register field is only valid when SSI_MAX_XFER_SIZE is configured to 16. If SSI_MAX_XFER_SIZE is configured to 32, then writing to this field will not have any effect.Selects the data frame length. When the data frame size is programmed to be less than 16 bits, the receive data are automatically right-justified by the receive logic, with the upper bits of the receive FIFO zero-padded.You must right-justify transmit data before writing into the transmit FIFO. The transmit logic ignores the upper unused bits when transmitting the data Note: When SSI_SPI_MODE is either set to Dual or Quad or Octal mode and SPI_FRF is not set to 2 b00.- DFS value should be multiple of 2 if SPI_FRF = 01,- DFS value should be multiple of 4 if SPI_FRF = 10,- DFS value should be multiple of 8 if SPI_FRF = 11. Values: 0x3 (FRAME_04BITS): 4-bit serial data transfer 0x4 (FRAME_05BITS): 5-bit serial data transfer 0x5 (FRAME_06BITS): 6-bit serial data transfer 0x6 (FRAME_07BITS): 7-bit serial data transfer	R

		0x7 (FRAME_08BITS): 8-bit serial data transfer 0x8 (FRAME_09BITS): 9-bit serial data transfer 0x9 (FRAME_10BITS): 10-bit serial data transfer 0xa (FRAME_11BITS): 11-bit serial data transfer 0xb (FRAME_12BITS): 12-bit serial data transfer 0xc (FRAME_13BITS): 13-bit serial data transfer 0xd (FRAME_14BITS): 14-bit serial data transfer 0xe (FRAME_15BITS): 15-bit serial data transfer 0xf (FRAME_16BITS): 16-bit serial data transfer	

3.19.4.2 CTRLR1(0x4)

Bits	Name	Description	Memory Access
31:16	RSVD_CTRLR1	CTRLR1 Reserved bits	R
15:0	NDF	Number of Data Frames. When TMOD = 10 or TMOD = 11 , this register field sets the number of data frames to be continuously received by the ssi. The ssi continues to receive serial data until the number of data frames received is equal to this register value plus 1, which enables you to receive up to 64 KB of data in a continuous transfer.When the ssi is configured as a serial slave, the transfer continues for as long as the slave is selected. Therefore, this register serves no purpose and is not present when the ssi is configured as a serial slave.	R/W

3.19.4.3 SSIENR(0x8)

Bits	Name	Description	Memory Access
31:1	RSVD_SSIENR	SSIENR Reserved bits	R

0	SSI_EN	SSI Enable. Enables and disables all ssi operations. When disabled, all serial transfers are halted immediately. Transmit and receive FIFO buffers are cleared when the device is disabled. It is impossible to program some of the ssi control registers when enabled. When disabled, the ssi_sleep output is set (after delay) to inform the system that it is safe to remove the ssi_clk, thus saving power consumption in the system. Values: 0x0 (DISABLE): Disables Serial Transfer 0x1 (ENABLED): Enables Serial Transfer	R/W

3.19.4.4 MWCR(0xc)

Bits	Name	Description	Memory Access
31:3	RSVD_MWCR	MWCR Reserved bits	R
2	MHS	Microwire Handshaking. Relevant only when the ssi is configured as a serial-master device. When configured as a serial slave, this bit field has no functionality. Used to enable and disable the busy/ready handshaking interface for the Microwire protocol. When enabled, the ssi checks for a ready status from the target slave, after the transfer of the last data/control bit, before clearing the BUSY status in the SR register. Values: 0x0 (DISABLE): Handshaking interface is disabled 0x1 (ENABLED): Handshaking interface is enabled	R/W
1	MDD	Microwire Control. Defines the direction of the data word when the Microwire serial protocol is used. When this bit is set to 0, the data word is received by the ssi MacroCell from the external serial device. When this bit is set to 1, the data word is transmitted from the ssi MacroCell	R/W

		to the external serial device. Values: 0x0 (RECEIVE): SSI receives data 0x1 (TRANSMIT): SSI transmits data	
0	MWMOD	Microwire Transfer Mode. Defines whether the Microwire transfer is sequential or non-sequential. When sequential mode is used, only one control word is needed to transmit or receive a block of data words. When non-sequential mode is used, there must be a control word for each data word that is transmitted or received. Values: 0x0 (NON_SEQUENTIAL): Non-Sequential Microwire Transfer 0x1 (SEQUENTIAL): Sequential Microwire Transfer	R/W

3.19.4.5 SER(0x10)

Bits	Name	Description	Memory Access
31:1	RSVD_SER	SER Reserved bits	R
0	SER	Slave Select Enable Flag. Each bit in this register corresponds to a slave select line (ss_x_n) from the ssi master. When a bit in this register is set (1), the corresponding slave select line from the master is activated when a serial transfer begins. It should be noted that setting or clearing bits in this register have no effect on the corresponding slave select outputs until a transfer is started. Before beginning a transfer, you should enable the bit in this register that corresponds to the slave device with which the master wants to communicate. When not operating in broadcast mode, only one bit in this field should be set. Values: 0x0 (NOT_SELECTED): No slave selected 0x1 (SELECTED): Slave is selected	R/W

3.19.4.6 BAUDR(0x14)

Bits	Name	Description	Memory Access
31:16	RSVD_BAUDR	BAUDR Reserved bits	R
15:0	SCKDV	SSI Clock Divider. The LSB for this field is always set to 0 and is unaffected by a write operation, which ensures an even value is held in this register. If the value is 0, the serial output clock (sclk_out) is disabled. The frequency of the sclk_out is derived from the following equation:Fsclk_out = Fssi_clk/SCKDVwhere SCKDV is any even value between 2 and 65534. For example:for Fssi_clk = 3.6864MHz and SCKDV =2 Fsclk_out = 3.6864/2 = 1.8432MHz	R/W

3.19.4.7 TXFTLR(0x18)

Bits	Name	Description	Memory Access
31:3	RSVD_TXFTLR	TXFTLR Reserved bits	R
2:0	TFT	Transmit FIFO Threshold. Controls the level of entries (or below) at which the transmit FIFO controller triggers an interrupt. The FIFO depth is configurable in the range 2-256; this register is sized to the number of address bits needed to access the FIFO. If you attempt to set this value greater than or equal to the depth of the FIFO, this field is not written and retains its current value. When the number of transmit FIFO entries is less than or equal to this value, the transmit FIFO empty interrupt is triggered. For information on the Transmit FIFO Threshold values, see the Master SPI and SSP Serial Transfers in the ssi Databook.ssi_txe_intr is asserted when TFT or less data entries are present in transmit FIFO	R/W

3.19.4.8 RXFTLR(0x1c)

Bits	Name	Description	Memory Access
31:3	RSVD_RXFTLR	RXFTLR Reserved bits	R
2:0	RFT	Receive FIFO Threshold. Controls the level of entries (or above) at which the receive FIFO controller triggers an interrupt. The FIFO depth is configurable in the range 2-256. This register is sized to the number of address bits needed to access the FIFO. If you attempt to set this value greater than the depth of the FIFO, this field is not written and retains its current value. When the number of receive FIFO entries is greater than or equal to this value + 1, the receive FIFO full interrupt is triggered. For information on the Receive FIFO Threshold values, see the Master SPI and SSP Serial Transfers in the ssi Databook.ssi_rxf_intr is asserted when RFT or more data entries are present in receive FIFO.	R/W

3.19.4.9 TXFLR(0x20)

Bits	Name	Description	Memory Access
31:4	RSVD_TXFLR	TXFLR Reserved bits Volatile: true	R
3:0	TXTFL	Transmit FIFO Level. Contains the number of valid data entries in the transmit FIFO. Volatile: true	R

3.19.4.10 RXFLR(0x24)

Bits	Name	Description	Memory Access
31:4	RSVD_RXFLR	RXFLR Reserved bits Volatile: true	R

3:0	RXTFL	Receive FIFO Level. Contains the number of valid data entries in the receive FIFO. Volatile: true	R

3.19.4.11 SR(0x28)

Bits	Name	Description	Memory Access
31:7	RSVD_SR	SR Reserved bits Volatile: true	R
6	DCOL	Data Collision Error. Relevant only when the ssi is configured as a master device. This bit will be set if ss_in_n input is asserted by other master, when the ssi master is in the middle of the transfer. This informs the processor that the last data transfer was halted before completion. This bit is cleared when read. Values: 0x0 (NO_ERROR_CONDITION): No Error 0x1 (TX_COLLISION_ERROR): Transmit Data Collision Error Volatile: true	R
5	RSVD_TXE	TXE Reserved field Volatile: true	R
4	RFF	Receive FIFO Full. When the receive FIFO is completely full, this bit is set. When the receive FIFO contains one or more empty location, this bit is cleared. Values: 0x0 (NOT_FULL): Receive FIFO is not full0x1 (FULL): Receive FIFO is full Volatile: true	R

3	RFNE	Receive FIFO Not Empty. Set when the receive FIFO contains one or more entries and is cleared when the receive FIFO is empty. This bit can be polled by software to completely empty the receive FIFO. Values: 0x0 (EMPTY): Receive FIFO is empty 0x1 (NOT_EMPTY): Receive FIFO is not empty Volatile: true	R
2	TFE	Transmit FIFO Empty. When the transmit FIFO is completely empty, this bit is set. When the transmit FIFO contains one or more valid entries, this bit is cleared. This bit field does not request an interrupt. Values: 0x0 (NOT_EMPTY): Transmit FIFO is not empty 0x1 (EMPTY): Transmit FIFO is empty Volatile: true	R
1	TFNF	Transmit FIFO Not Full. Set when the transmit FIFO contains one or more empty locations, and is cleared when the FIFO is full. Values: 0x0 (FULL): Transmit FIFO is full 0x1 (NOT_FULL): Transmit FIFO is not Full Volatile: true	R
0	BUSY	SSI Busy Flag. When set, indicates that a serial transfer is in progress; when cleared indicates that the ssi is idle or disabled. Values: 0x0 (INACTIVE): ssi is idle or disabled 0x1 (ACTIVE): ssi is actively transferring data Volatile: true	R

3.19.4.12 IMR(0x2c)

Bits	Name	Description	Memory Access
31:6	RSVD_IMR	IMR Reserved bits	R

5	MSTIM	Multi-Master Contention Interrupt Mask. This bit field is not present if the ssi is configured as a serial-slave device. Values: 0x0 (MASKED): ssi_mst_intr interrupt is masked 0x1 (UNMASKED): ssi_mst_intr interrupt is not masked	R/W
4	RXFIM	Receive FIFO Full Interrupt Mask Values: 0x0 (MASKED): ssi_rxf_intr interrupt is masked 0x1 (UNMASKED): ssi_rxf_intr interrupt is not masked	R/W
3	RXOIM	Receive FIFO Overflow Interrupt Mask Values: 0x0 (MASKED): ssi_rxo_intr interrupt is masked 0x1 (UNMASKED): ssi_rxo_intr interrupt is not masked	R/W
2	RXUIM	Receive FIFO Underflow Interrupt Mask Values: 0x0 (MASKED): ssi_rxu_intr interrupt is masked 0x1 (UNMASKED): ssi_rxu_intr interrupt is not masked	R/W
1	TXOIM	Transmit FIFO Overflow Interrupt Mask Values: 0x0 (MASKED): ssi_txo_intr interrupt is masked 0x1 (UNMASKED): ssi_txo_intr interrupt is not masked	R/W
0	TXEIM	Transmit FIFO Empty Interrupt Mask Values: 0x0 (MASKED): ssi_txe_intr interrupt is masked 0x1 (UNMASKED): ssi_txe_intr interrupt is not masked	R/W

3.19.4.13 ISR(0x30)

Bits	Name	Description	Memory Access
31:6	RSVD_ISR	ISR Reserved bits Volatile: true	R
5	MSTIS	Multi-Master Contention Interrupt Status. This bit field is not present if the ssi is configured as a serial-slave device. Values: 0x0 (INACTIVE): ssi_mst_intr interrupt not active after masking 0x1 (ACTIVE): ssi_mst_intr interrupt is active after masking Volatile: true	R

4	RXFIS	Receive FIFO Full Interrupt Status Values: 0x0 (INACTIVE): ssi_rxf_intr interrupt is not active after masking 0x1 (ACTIVE): ssi_rxf_intr interrupt is full after masking Volatile: true	R
3	RXOIS	Receive FIFO Overflow Interrupt Status Values: 0x0 (INACTIVE): ssi_rxo_intr interrupt is not active after masking 0x1 (ACTIVE): ssi_rxo_intr interrupt is active after masking Volatile: true	R
2	RXUIS	Receive FIFO Underflow Interrupt Status Values: 0x0 (INACTIVE): ssi_rxu_intr interrupt is not active after masking 0x1 (ACTIVE): ssi_rxu_intr interrupt is active after masking Volatile: true	R
1	TXOIS	Transmit FIFO Overflow Interrupt Status Values: 0x0 (INACTIVE): ssi_txo_intr interrupt is not active after masking 0x1 (ACTIVE): ssi_txo_intr interrupt is active after masking Volatile: true	R
0	TXEIS	Transmit FIFO Empty Interrupt Status Values: 0x0 (INACTIVE): ssi_txe_intr interrupt is not active after masking 0x1 (ACTIVE): ssi_txe_intr interrupt is active after masking Volatile: true	R

3.19.4.14 RISR(0x34)

Bits	Name	Description	Memory Access
31:6	RSVD_RISR	RISR Reserved bits Volatile: true	R

5	MSTIR	Multi-Master Contention Raw Interrupt Status. This bit field is not present if the ssi is configured as a serial-slave device. Values: 0x0 (INACTIVE): ssi_mst_intr interrupt is not active prior to masking 0x1 (ACTIVE): ssi_mst_intr interrupt is active prior masking Volatile: true	R
4	RXFIR	Receive FIFO Full Raw Interrupt Status Values: 0x0 (INACTIVE): ssi_rxf_intr interrupt is not active prior to masking 0x1 (ACTIVE): ssi_rxf_intr interrupt is active prior to masking Volatile: true	R
3	RXOIR	Receive FIFO Overflow Raw Interrupt Status Values: 0x1 (ACTIVE): ssi_rxo_intr interrupt is not active prior to masking 0x0 (INACTIVE): ssi_rxo_intr interrupt is active prior masking Volatile: true	R
2	RXUIR	Receive FIFO Underflow Raw Interrupt Status Values: 0x0 (INACTIVE): ssi_rxu_intr interrupt is not active prior to masking 0x1 (ACTIVE): ssi_rxu_intr interrupt is active prior to masking Volatile: true	R
1	TXOIR	Transmit FIFO Overflow Raw Interrupt Status Values: 0x0 (INACTIVE): ssi_txo_intr interrupt is not active prior to masking 0x1 (ACTIVE): ssi_txo_intr interrupt is active prior masking Volatile: true	R

0	TXEIR	Transmit FIFO Empty Raw Interrupt Status Values: 0x0 (INACTIVE): ssi_txe_intr interrupt is not active prior to masking 0x1 (ACTIVE): ssi_txe_intr interrupt is active prior masking Volatile: true	R

3.19.4.15 TXOICR(0x38)

Bits	Name	Description	Memory Access
31:1	RSVD_TXOICR	TXOICR Reserved bits Volatile: true	R
0	TXOICR	Clear Transmit FIFO Overflow Interrupt. This register reflects the status of the interrupt. A read from this register clears the ssi_txo_intr interrupt; writing has no effect. Volatile: true	R

3.19.4.16 RXOICR(0x3c)

Bits	Name	Description	Memory Access
31:1	RSVD_RXOICR	RXOICR Reserved bits Volatile: true	R
0	RXOICR	Clear Receive FIFO Overflow Interrupt. This register reflects the status of the interrupt. A read from this register clears the ssi_rxo_intr interrupt; writing has no effect. Volatile: true	R

3.19.4.17 RXUICR(0x40)

Bits	Name	Description	Memory Access
31:1	RSVD_RXUICR	RXUICR Reserved bits Volatile: true	R

0	RXUICR	Clear Receive FIFO Underflow Interrupt. This register reflects the status of the interrupt. A read from this register clears the ssi_rxu_intr interrupt; writing has no effect. Volatile: true	R

3.19.4.18 MSTICR(0x44)

Bits	Name	Description	Memory Access
31:1	RSVD_MSTICR	MSTICR Reserved bits Volatile: true	R
0	MSTICR	Clear Multi-Master Contention Interrupt. This register reflects the status of the interrupt. A read from this register clears the ssi_mst_intr interrupt; writing has no effect. Volatile: true	R

3.19.4.19 ICR(0x48)

Bits	Name	Description	Memory Access
31:1	RSVD_ICR	ICR Reserved bits Volatile: true	R
0	ICR	Clear Interrupts. This register is set if any of the interrupts below are active. A read clears the ssi_txo_intr, ssi_rxu_intr, ssi_rxo_intr, and the ssi_mst_intr interrupts. Writing to this register has no effect. Volatile: true	R

3.19.4.20 DMACR(0x4c)

Bits	Name	Description	Memory Access
31:2	RSVD_DMACR	DMACR Reserved bits	R

1	TDMAE	Transmit DMA Enable. This bit enables/disables the transmit FIFO DMA channel. Values: 0x0 (DISABLE): Transmit DMA disabled 0x1 (ENABLED): Transmit DMA enabled	R/W
0	RDMAE	Receive DMA Enable. This bit enables/disables the receive FIFO DMA channel Values: 0x0 (DISABLE): Receive DMA disabled 0x1 (ENABLED): Receive DMA enabled	R/W

3.19.4.21 DMATDLR(0x50)

Bits	Name	Description	Memory Access
31:3	RSVD_DMATDLR	DMATDLR Reserved bits	R
2:0	DMATDL	Transmit Data Level. This bit field controls the level at which a DMA request is made by the transmit logic. It is equal to the watermark level; that is, the dma_tx_req signal is generated when the number of valid data entries in the transmit FIFO is equal to or below this field value, and TDMAE = 1. For information on the DMATDL decode values, see the Slave SPI and SSP Serial Transfers section in the ssi Databook.dma_tx_req is asserted when DMATDL or less data entries are present in the transmit FIFO	R/W

3.19.4.22 DMARDLR(0x54)

Bits	Name	Description	Memory Access
31:3	RSVD_DMARDLR	DMARDLR Reserved bits	R

2:0	DMARDL	Receive Data Level. This bit field controls the level at which a DMA request is made by the receive logic. The watermark level = DMARDL+1; that is, dma_rx_req is generated when the number of valid data entries in the receive FIFO is equal to or above this field value + 1, and RDMAE=1. For information on the DMARDL decode values, see the Slave SPI and SSP Serial Transfers section in the ssi Databook.dma_rx_req is asserted when DMARDL or more valid data entries are present in the receive FIFO.	R/W

3.19.4.23 IDR(0x58)

Bits	Name	Description	Memory Access
31:0	IDCODE	Identification code. The register contains the peripheral s identification code, which is written into the register at configuration time using CoreConsultant.	R

3.19.4.24 SSI_VERSION_ID(0x5c)

Bits	Name	Description	Memory Access
31:0	SSI_COMP_VERSION	Contains the hex representation of the Synopsys component version.	R

3.19.4.25 DR0(0x60)

Bits	Name	Description	Memory Access

31:0	DR	Data Register. When writing to this register, you must right-justify the data. Read data are automatically right-justified. If SSI_MAX_XFER_SIZE configuration parameter is set to 32, all 32 bits are valid. Otherwise, only 16 bits ([15:0]) of the register are valid. Read = Receive FIFO buffer Write = Transmit FIFO buffer. Volatile: true	R/W

3.19.4.26 DR1(0x64)

Bits	Name	Description	Memory Access
31:0	DR	Data Register. When writing to this register, you must right-justify the data. Read data are automatically right-justified. If SSI_MAX_XFER_SIZE configuration parameter is set to 32, all 32 bits are valid. Otherwise, only 16 bits ([15:0]) of the register are valid. Read = Receive FIFO buffer Write = Transmit FIFO buffer. Volatile: true	R/W

3.19.4.27 DR2(0x68)

Bits	Name	Description	Memory Access
31:0	DR	Data Register. When writing to this register, you must right-justify the data. Read data are automatically right-justified. If SSI_MAX_XFER_SIZE configuration parameter is set to 32, all 32 bits are valid. Otherwise, only 16 bits ([15:0]) of the register are valid. Read = Receive FIFO buffer Write = Transmit FIFO buffer. Volatile: true	R/W

3.19.4.28 DR3(0x6c)

Bits	Name	Description	Memory Access

31:0	DR	Data Register. When writing to this register, you must right-justify the data. Read data are automatically right-justified. If SSI_MAX_XFER_SIZE configuration parameter is set to 32, all 32 bits are valid. Otherwise, only 16 bits ([15:0]) of the register are valid. Read = Receive FIFO buffer Write = Transmit FIFO buffer. Volatile: true	R/W

3.19.4.29 DR4(0x70)

Bits	Name	Description	Memory Access
31:0	DR	Data Register. When writing to this register, you must right-justify the data. Read data are automatically right-justified. If SSI_MAX_XFER_SIZE configuration parameter is set to 32, all 32 bits are valid. Otherwise, only 16 bits ([15:0]) of the register are valid. Read = Receive FIFO buffer Write = Transmit FIFO buffer. Volatile: true	R/W

3.19.4.30 DR5(0x74)

Bits	Name	Description	Memory Access
31:0	DR	Data Register. When writing to this register, you must right-justify the data. Read data are automatically right-justified. If SSI_MAX_XFER_SIZE configuration parameter is set to 32, all 32 bits are valid. Otherwise, only 16 bits ([15:0]) of the register are valid. Read = Receive FIFO buffer Write = Transmit FIFO buffer. Volatile: true	R/W

3.19.4.31 DR6(0x78)

Bits	Name	Description	Memory Access

31:0	DR	Data Register. When writing to this register, you must right-justify the data. Read data are automatically right-justified. If SSI_MAX_XFER_SIZE configuration parameter is set to 32, all 32 bits are valid. Otherwise, only 16 bits ([15:0]) of the register are valid. Read = Receive FIFO buffer Write = Transmit FIFO buffer. Volatile: true	R/W

3.19.4.32 DR7(0x7c)

Bits	Name	Description	Memory Access
31:0	DR	Data Register. When writing to this register, you must right-justify the data. Read data are automatically right-justified. If SSI_MAX_XFER_SIZE configuration parameter is set to 32, all 32 bits are valid. Otherwise, only 16 bits ([15:0]) of the register are valid. Read = Receive FIFO buffer Write = Transmit FIFO buffer. Volatile: true	R/W

3.19.4.33 DR8(0x80)

Bits	Name	Description	Memory Access
31:0	DR	Data Register. When writing to this register, you must right-justify the data. Read data are automatically right-justified. If SSI_MAX_XFER_SIZE configuration parameter is set to 32, all 32 bits are valid. Otherwise, only 16 bits ([15:0]) of the register are valid. Read = Receive FIFO buffer Write = Transmit FIFO buffer. Volatile: true	R/W

3.19.4.34 DR9(0x84)

Bits	Name	Description	Memory Access

31:0	DR	Data Register. When writing to this register, you must right-justify the data. Read data are automatically right-justified. If SSI_MAX_XFER_SIZE configuration parameter is set to 32, all 32 bits are valid. Otherwise, only 16 bits ([15:0]) of the register are valid. Read = Receive FIFO buffer Write = Transmit FIFO buffer. Volatile: true	R/W

3.19.4.35 DR10(0x88)

Bits	Name	Description	Memory Access
31:0	DR	Data Register. When writing to this register, you must right-justify the data. Read data are automatically right-justified. If SSI_MAX_XFER_SIZE configuration parameter is set to 32, all 32 bits are valid. Otherwise, only 16 bits ([15:0]) of the register are valid. Read = Receive FIFO buffer Write = Transmit FIFO buffer. Volatile: true	R/W

3.19.4.36 DR11(0x8c)

Bits	Name	Description	Memory Access
31:0	DR	Data Register. When writing to this register, you must right-justify the data. Read data are automatically right-justified. If SSI_MAX_XFER_SIZE configuration parameter is set to 32, all 32 bits are valid. Otherwise, only 16 bits ([15:0]) of the register are valid. Read = Receive FIFO buffer Write = Transmit FIFO buffer. Volatile: true	R/W

3.19.4.37 DR12(0x90)

Bits	Name	Description	Memory Access

31:0	DR	Data Register. When writing to this register, you must right-justify the data. Read data are automatically right-justified. If SSI_MAX_XFER_SIZE configuration parameter is set to 32, all 32 bits are valid. Otherwise, only 16 bits ([15:0]) of the register are valid. Read = Receive FIFO buffer Write = Transmit FIFO buffer. Volatile: true	R/W

3.19.4.38 DR13(0x94)

Bits	Name	Description	Memory Access
31:0	DR	Data Register. When writing to this register, you must right-justify the data. Read data are automatically right-justified. If SSI_MAX_XFER_SIZE configuration parameter is set to 32, all 32 bits are valid. Otherwise, only 16 bits ([15:0]) of the register are valid. Read = Receive FIFO buffer Write = Transmit FIFO buffer. Volatile: true	R/W

3.19.4.39 DR14(0x98)

Bits	Name	Description	Memory Access
31:0	DR	Data Register. When writing to this register, you must right-justify the data. Read data are automatically right-justified. If SSI_MAX_XFER_SIZE configuration parameter is set to 32, all 32 bits are valid. Otherwise, only 16 bits ([15:0]) of the register are valid. Read = Receive FIFO buffer Write = Transmit FIFO buffer. Volatile: true	R/W

3.19.4.40 DR15(0x9c)

Bits	Name	Description	Memory Access

31:0	DR	Data Register. When writing to this register, you must right-justify the data. Read data are automatically right-justified. If SSI_MAX_XFER_SIZE configuration parameter is set to 32, all 32 bits are valid. Otherwise, only 16 bits ([15:0]) of the register are valid. Read = Receive FIFO buffer Write = Transmit FIFO buffer. Volatile: true	R/W

3.19.4.41 DR16(0xa0)

Bits	Name	Description	Memory Access
31:0	DR	Data Register. When writing to this register, you must right-justify the data. Read data are automatically right-justified. If SSI_MAX_XFER_SIZE configuration parameter is set to 32, all 32 bits are valid. Otherwise, only 16 bits ([15:0]) of the register are valid. Read = Receive FIFO buffer Write = Transmit FIFO buffer. Volatile: true	R/W

3.19.4.42 DR17(0xa4)

Bits	Name	Description	Memory Access
31:0	DR	Data Register. When writing to this register, you must right-justify the data. Read data are automatically right-justified. If SSI_MAX_XFER_SIZE configuration parameter is set to 32, all 32 bits are valid. Otherwise, only 16 bits ([15:0]) of the register are valid. Read = Receive FIFO buffer Write = Transmit FIFO buffer. Volatile: true	R/W

3.19.4.43 DR18(0xa8)

Bits	Name	Description	Memory Access

31:0	DR	Data Register. When writing to this register, you must right-justify the data. Read data are automatically right-justified. If SSI_MAX_XFER_SIZE configuration parameter is set to 32, all 32 bits are valid. Otherwise, only 16 bits ([15:0]) of the register are valid. Read = Receive FIFO buffer Write = Transmit FIFO buffer. Volatile: true	R/W

3.19.4.44 DR19(0xac)

Bits	Name	Description	Memory Access
31:0	DR	Data Register. When writing to this register, you must right-justify the data. Read data are automatically right-justified. If SSI_MAX_XFER_SIZE configuration parameter is set to 32, all 32 bits are valid. Otherwise, only 16 bits ([15:0]) of the register are valid. Read = Receive FIFO buffer Write = Transmit FIFO buffer. Volatile: true	R/W

3.19.4.45 DR20(0xb0)

Bits	Name	Description	Memory Access
31:0	DR	Data Register. When writing to this register, you must right-justify the data. Read data are automatically right-justified. If SSI_MAX_XFER_SIZE configuration parameter is set to 32, all 32 bits are valid. Otherwise, only 16 bits ([15:0]) of the register are valid. Read = Receive FIFO buffer Write = Transmit FIFO buffer. Volatile: true	R/W

3.19.4.46 DR21(0xb4)

Bits	Name	Description	Memory Access

31:0	DR	Data Register. When writing to this register, you must right-justify the data. Read data are automatically right-justified. If SSI_MAX_XFER_SIZE configuration parameter is set to 32, all 32 bits are valid. Otherwise, only 16 bits ([15:0]) of the register are valid. Read = Receive FIFO buffer Write = Transmit FIFO buffer. Volatile: true	R/W

3.19.4.47 DR22(0xb8)

Bits	Name	Description	Memory Access
31:0	DR	Data Register. When writing to this register, you must right-justify the data. Read data are automatically right-justified. If SSI_MAX_XFER_SIZE configuration parameter is set to 32, all 32 bits are valid. Otherwise, only 16 bits ([15:0]) of the register are valid. Read = Receive FIFO buffer Write = Transmit FIFO buffer. Volatile: true	R/W

3.19.4.48 DR23(0xbc)

Bits	Name	Description	Memory Access
31:0	DR	Data Register. When writing to this register, you must right-justify the data. Read data are automatically right-justified. If SSI_MAX_XFER_SIZE configuration parameter is set to 32, all 32 bits are valid. Otherwise, only 16 bits ([15:0]) of the register are valid. Read = Receive FIFO buffer Write = Transmit FIFO buffer. Volatile: true	R/W

3.19.4.49 DR24(0xc0)

Bits	Name	Description	Memory Access
31:0	DR	Data Register. When writing to this register, you must right-justify the data. Read data are automatically right-justified. If SSI_MAX_XFER_SIZE configuration parameter is set to 32, all 32 bits are valid. Otherwise, only 16 bits ([15:0]) of the register are valid. Read = Receive FIFO buffer Write = Transmit FIFO buffer. Volatile: true	R/W

3.19.4.50 DR25(0xc4)

Bits	Name	Description	Memory Access
31:0	DR	Data Register. When writing to this register, you must right-justify the data. Read data are automatically right-justified. If SSI_MAX_XFER_SIZE configuration parameter is set to 32, all 32 bits are valid. Otherwise, only 16 bits ([15:0]) of the register are valid. Read = Receive FIFO buffer Write = Transmit FIFO buffer. Volatile: true	R/W

3.19.4.51 DR26(0xc8)

Bits	Name	Description	Memory Access
31:0	DR	Data Register. When writing to this register, you must right-justify the data. Read data are automatically right-justified. If SSI_MAX_XFER_SIZE configuration parameter is set to 32, all 32 bits are valid. Otherwise, only 16 bits ([15:0]) of the register are valid. Read = Receive FIFO buffer Write = Transmit FIFO buffer. Volatile: true	R/W

3.19.4.52 DR27(0xcc)

Bits	Name	Description	Memory Access
31:0	DR	Data Register. When writing to this register, you must right-justify the data. Read data are automatically right-justified. If SSI_MAX_XFER_SIZE configuration parameter is set to 32, all 32 bits are valid. Otherwise, only 16 bits ([15:0]) of the register are valid. Read = Receive FIFO buffer Write = Transmit FIFO buffer. Volatile: true	R/W

3.19.4.53 DR28(0xd0)

Bits	Name	Description	Memory Access
31:0	DR	Data Register. When writing to this register, you must right-justify the data. Read data are automatically right-justified. If SSI_MAX_XFER_SIZE configuration parameter is set to 32, all 32 bits are valid. Otherwise, only 16 bits ([15:0]) of the register are valid. Read = Receive FIFO buffer Write = Transmit FIFO buffer. Volatile: true	R/W

3.19.4.54 DR29(0xd4)

Bits	Name	Description	Memory Access
31:0	DR	Data Register. When writing to this register, you must right-justify the data. Read data are automatically right-justified. If SSI_MAX_XFER_SIZE configuration parameter is set to 32, all 32 bits are valid. Otherwise, only 16 bits ([15:0]) of the register are valid. Read = Receive FIFO buffer Write = Transmit FIFO buffer. Volatile: true	R/W

3.19.4.55 DR30(0xd8)

Bits	Name	Description	Memory Access
31:0	DR	Data Register. When writing to this register, you must right-justify the data. Read data are automatically right-justified. If SSI_MAX_XFER_SIZE configuration parameter is set to 32, all 32 bits are valid. Otherwise, only 16 bits ([15:0]) of the register are valid. Read = Receive FIFO buffer Write = Transmit FIFO buffer. Volatile: true	R/W

3.19.4.56 DR31(0xdc)

Bits	Name	Description	Memory Access
31:0	DR	Data Register. When writing to this register, you must right-justify the data. Read data are automatically right-justified. If SSI_MAX_XFER_SIZE configuration parameter is set to 32, all 32 bits are valid. Otherwise, only 16 bits ([15:0]) of the register are valid. Read = Receive FIFO buffer Write = Transmit FIFO buffer. Volatile: true	R/W

3.19.4.57 DR32(0xe0)

Bits	Name	Description	Memory Access
31:0	DR	Data Register. When writing to this register, you must right-justify the data. Read data are automatically right-justified. If SSI_MAX_XFER_SIZE configuration parameter is set to 32, all 32 bits are valid. Otherwise, only 16 bits ([15:0]) of the register are valid. Read = Receive FIFO buffer Write = Transmit FIFO buffer. Volatile: true	R/W

3.19.4.58 DR33(0xe4)

Bits	Name	Description	Memory Access
31:0	DR	Data Register. When writing to this register, you must right-justify the data. Read data are automatically right-justified. If SSI_MAX_XFER_SIZE configuration parameter is set to 32, all 32 bits are valid. Otherwise, only 16 bits ([15:0]) of the register are valid. Read = Receive FIFO buffer Write = Transmit FIFO buffer. Volatile: true	R/W

3.19.4.59 DR34(0xe8)

Bits	Name	Description	Memory Access
31:0	DR	Data Register. When writing to this register, you must right-justify the data. Read data are automatically right-justified. If SSI_MAX_XFER_SIZE configuration parameter is set to 32, all 32 bits are valid. Otherwise, only 16 bits ([15:0]) of the register are valid. Read = Receive FIFO buffer Write = Transmit FIFO buffer. Volatile: true	R/W

3.19.4.60 DR35(0xec)

Bits	Name	Description	Memory Access
31:0	DR	Data Register. When writing to this register, you must right-justify the data. Read data are automatically right-justified. If SSI_MAX_XFER_SIZE configuration parameter is set to 32, all 32 bits are valid. Otherwise, only 16 bits ([15:0]) of the register are valid. Read = Receive FIFO buffer Write = Transmit FIFO buffer. Volatile: true	R/W

3.19.4.61 RX_SAMPLE_DLY(0xf0)

Bits	Name	Description	Memory Access
31:8	RSVD_RX_SAMPLE_DLY	SAMPLE_DLY Reserved bits	R
7:0	RSD	Rxd Sample Delay. This register is used to delay the sample of the rxd input port. Each value represents a single ssi_clk delay on the sample of rxd. Note: If this register is programmed with a value that exceeds the depth of the internal shift registers (SSI_RX_DLY_SR_DEPTH) zero delay will be applied to the rxd sample.	R/W

3.19.4.62 SPI_CTRLR0(0xf4)

Bits	Name	Description	Memory Access
31:19	RSVD_SPI_CTRLR0	SPI_CTRLR0 Reserved bits	R
18	SPI_RXDS_EN	Read data strobe enable bit. Once this bit is set to 1 ssi will use Read data strobe (rxds) to capture read data in DDR mode.	R
17	INST_DDR_EN	Instruction DDR Enable bit. This will enable Dual-data rate transfer for Instruction phase.	R
16	SPI_DDR_EN	SPI DDR Enable bit. This will enable Dual-data rate transfers in Dual/Quad/Octal frame formats of SPI.	R
15:11	WAIT_CYCLES	Wait cycles Number of wait cycles in Dual/Quad/Octal mode between control frames transmit and data reception. This value is specified as number of SPI clock cycles. For information on the WAIT_CYCLES decode value, see Read Operation in Enhanced SPI Modes section in the ssi Databook.	R/W
10	RSVD_SPI_CTRLR0_10	CTRLR0_10 Reserved bits - Read Only	R

Bits	Name	Description	Memory Access
9:8	INST_L	Instruction Length Dual/Quad/Octal mode instruction length in bits. Values: 0x0 (INST_L_0): 0-bit (No Instruction) 0x1 (INST_L_1): 4-bit Instruction 0x2 (INST_L_2): 8-bit Instruction 0x3 (INST_L_3): 16-bit Instruction	R/W
7:6	RSVD_SPI_CTRLR0_6_7	CTRLR0_6_7 Reserved bits - Read Only	R
5:2	ADDR_L	Address Length. This bit defines Length of Address to be transmitted. Only after this much bits are programmed in to the FIFO the transfer can begin. For information on the ADDR_Ldecode value, see Read Operation in Enhanced SPI Modes section in the ssi Databook. Values: 0x0 (ADDR_L_0): 0-bit Address Width 0x1 (ADDR_L_1): 4-bit Address Width 0x2 (ADDR_L_2): 8-bit Address Width 0x3 (ADDR_L_3): 12-bit Address Width 0x4 (ADDR_L_4): 16-bit Address Width 0x5 (ADDR_L_5): 20-bit Address Width 0x6 (ADDR_L_6): 24-bit Address Width 0x7 (ADDR_L_7): 28-bit Address Width 0x8 (ADDR_L_8): 32-bit Address Width 0x9 (ADDR_L_9): 36-bit Address Width 0xa (ADDR_L_10): 40-bit Address Width 0xb (ADDR_L_11): 44-bit Address Width 0xc (ADDR_L_12): 48-bit Address Width 0xd (ADDR_L_13): 52-bit Address Width 0xe (ADDR_L_14): 56-bit Address Width 0xf (ADDR_L_15): 60-bit Address Width	R/W

1:0	TRANS_TYPE	Address and instruction transfer format. Selects whether ssi will transmit instruction/address either in Standard SPI mode or the SPI mode selected in CTRLR0.SPI_FRF field. - Instruction and Address will be sent in Standard SPI Mode. - Instruction will be sent in Standard SPI Mode and Address will be sent in the mode specified by CTRLR0.SPI_FRF. 10 - Both Instruction and Address will be sent in the mode specified by SPI_FRF. 11 - Reserved.	R/W

3.19.4.63 RSVD(0xfc)

Bits	Name	Description	Memory Access
31:0	RSVD	RSVD 31 to 0 Reserved address location Volatile: true	R

3.20 集成电路内置音频总线 (I²S)

3.20.1 概述

I2S标准总线定义了三种信号：时钟信号BCK、声道选择信号WS和串行数据信号SD。一个基本的I2S数据总线有一个主机和一个从机。主机和从机的角色在通信过程中保持不变。I2S模块包含独立的发送和接收声道，能够保证优良的通信性能。系统拥有3个I2S标准接口，都是master模式。其中I2S0支持可配置连接语音处理模块，实现语音增强和声源定向的功能，有以下特性：

- 总线宽度可配置为8，16，和32位；
- 每个接口最多支持4个立体声通道；
- 由于发送器和接收器的独立性，所以支持全双工通讯；
- APB总线和I2S sclk的异步时钟；
- 音频频数据分辨率为12,16,20,24和32位；

- 可配置的FIFO深度为2,4,8和16个字；
- 可配置可编程的DMA寄存器；
- 可编程FIFO阈值。

3.20.2 功能描述

- 支持的音频标准

I2S 模式下，BCK为串行时钟；WS为通道选择信号，用于表示左右声道的切换；SD为串行数据信号， 传输音频数据。WS信号和SD信号在BCK的下降沿发生变化，并在BCK的上升沿采样SD信号。支持的标准：Philips 标准、MSB 对齐标准、PCM 标准。

- 发送数据

I2S发送数据分为三个阶段：

1） 第一阶段从内存中读出数据并写入FIFO；
2） 第二阶段将待发送数据从FIFO中读出；
3） 第三阶段，在I2S模式下，将待发送数据转换为串行数据流输出；在LCD模式下，将待发送数据转换为位宽固定的并行数据流输出。

- 接收数据

I2S接收数据也分为三个阶段：

1） 第一阶段，在I2S模式下，输入的串行比特流数据会被以声道属性转换成宽度为64位的并行数据流；在LCD模式下，输入的位宽固定的并行数据流会被扩展成宽度为64位的并行数据流；
2） 第二阶段将待接收的数据写入FIFO；
3） 第三阶段将待接收的数据从FIFO中读出，并写入内存。

3.20.3 寄存器列表

Base Address：I2S0_BASE_ADDR (0x50250000)

Base Address：I2S1_BASE_ADDR (0x50260000)

Base Address：I2S2_BASE_ADDR (0x50270000)

Name	Description	Offset address

IER	This register acts as a global enable/disable for apb_i2s.	0x0
IRER	This register acts as an enable/disable for the apb_i2s Receiver block.	0x4
ITER	This register acts as an enable/disable for the apb_i2s Transmitter block.	0x8
CER	This register acts as an enable/disable for the apb_i2s Clock Generation block, which is only...	0xc
CCR	This register configures the ws_out and sclk_gate signals when apb_i2s is a master.	0x10
RXFFR	This register specifies the Receiver Block FIFO Reset Register.	0x14
TXFFR	This register specifies the Transmitter Block FIFO Reset Register.	0x18
LRBR0	This specifies the Left Receive Buffer Register.	0x20
LTHR0	This specifies the Left Transmit Holding Register.	0x20
RRBR0	This specifies the Right Receive Buffer Register.	0x24
RTHR0	This specifies the Right Transmit Holding Register.	0x24
RER0	This specifies the Receive Enable Register.	0x28
TER0	This specifies the Transmit Enable Register.	0x2c
RCR0	This specifies the Receive Configuration Register.	0x30
TCR0	This specifies the Transmit Configuration Register.	0x34
ISR0	This specifies the Interrupt Status Register.	0x38
IMR0	This specifies the Interrupt Mask Register.	0x3c
ROR0	This specifies the Receive Overrun Register.	0x40
TOR0	This specifies the Transmit Overrun Register.	0x44
RFCR0	This specifies the Receive FIFO Configuration Register.	0x48
TFCR0	This specifies the Transmit FIFO Configuration Register.	0x4c
RFF0	This specifies the Receive FIFO Flush Register.	0x50
TFF0	This specifies the Transmit FIFO Flush Register.	0x54

LRBR1	Left Receive Buffer Register 1	0x60
LTHR1	Left Transmit Holding Register 1	0x60

Name	Description	Offset address
RRBR1	Right Receive Buffer Register 1.	0x64
RTHR1	Right Transmit Holding Register 1	0x64
RER1	Receive Enable Register 1	0x68
TER1	Transmit Enable Register 1	0x6c
RCR1	Receive Configuration Register 1	0x70
TCR1	Transmit Configuration Register 1	0x74
ISR1	Interrupt Status Register 1	0x78
IMR1	Interrupt Mask Register 1.	0x7c
ROR1	Receive Overrun Register 1	0x80
TOR1	Transmit Overrun Register 1	0x84
RFCR1	Receive FIFO Configuration Register 1	0x88
TFCR1	Transmit FIFO Configuration Register 1	0x8c
RFF1	Receive FIFO Flush Register 1	0x90
TFF1	Transmit FIFO Flush Register 1	0x94
LRBR2	Left Receive Buffer Register 2	0xa0
LTHR2	Left Transmit Holding Register 2	0xa0
RRBR2	Right Receive Buffer Register 2	0xa4
RTHR2	Right Transmit Holding Register 2	0xa4
RER2	Receive Enable Register 2	0xa8
TER2	Transmit Enable Register 2	0xac
RCR2	Receive Configuration Register 2	0xb0
TCR2	Transmit Configuration Register 2	0xb4
ISR2	Interrupt Status Register 2	0xb8

IMR2	Interrupt Mask Register 2	0xbc
ROR2	Receive Overrun Register 2	0xc0
TOR2	Transmit Overrun Register 2	0xc4
RFCR2	Receive FIFO Configuration Register 2	0xc8
TFCR2	Transmit FIFO Configuration Register 2	0xcc
RFF2	Receive FIFO Flush Register 2	0xd0
TFF2	Transmit FIFO Flush Register 2	0xd4

Name	Description	Offset address
LRBR3	Left Receive Buffer Register 3	0xe0
LTHR3	Left Transmit Holding Register 3	0xe0
RRBR3	Right Receive Buffer Register 3	0xe4
RTHR3	Right Transmit Holding Register 3	0xe4
RER3	Receive Enable Register 3	0xe8
TER3	Transmit Enable Register 3	0xec
RCR3	Receive Configuration Register 3	0xf0
TCR3	Transmit Configuration Register 3	0xf4
ISR3	Interrupt Status Register 3	0xf8
IMR3	Interrupt Mask Register 3	0xfc
ROR3	Receive Overrun Register 3	0x100
TOR3	Transmit Overrun Register 3	0x104
RFCR3	Receive FIFO Configuration Register 3	0x108
TFCR3	Transmit FIFO Configuration Register 3	0x10c
RFF3	Receive FIFO Flush Register 3	0x110
TFF3	Transmit FIFO Flush Register 3	0x114
RXDMA	The RXDMA register allows access to all enabled Receive channels via a single point rather than...	0x1c0

RRXDMA	The RXDMA can be reset to the lowest enabled Channel via the RRXDMA register. The RRXDMA register can...	0x1c4
TXDMA	The TXDMA register functions similar to the RXDMA register and allows write accesses to all of...	0x1c8
RTXDMA	This register provides the same functionality as the RRXDMA register but targets TXDMA...	0x1cc
I2S_COMP_PARAM_2	This specifies bits for Component Parameter Register 2. Note: This is a constant read-only register...	0x1f0
I2S_COMP_PARAM_1	This specifies bits for Component Parameter Register 1. Note: This is a constant read-only register...	0x1f4
I2S_COMP_VERSION	This register specifies the I2S Component Version.	0x1f8
I2S_COMP_TYPE	This register specifies the I2S Component Type.	0x1fc

3.20.4 寄存器设置

3.20.4.1 IER(0x0)

Bits	Name	Description	Memory Access
31:1	RSVD_IER	RSVD_IER Reserved bits - Read Only	R
0	IEN	I2S enable.This bit enables or disables I2S. A disable on this bit overrides any other block or channel enables and flushes all FIFOs. For more information about how this register affects the other I2S blocks, refer to I2S Enable .Values:0x0 (DISABLED): I2S disabled.0x1 (ENABLED): I2S enabledValue After Reset: 0x0	R/W

3.20.4.2 IRER(0x4)

Bits	Name	Description	Memory Access
31:1	RSVD_IRER	RSVD_IRER Reserved bits - Read Only	R
0	RXEN	Receiver block enable.This bit enables or disables the receiver. A disable on this bit overrides any individual receive channel enables. For more information about the receiver block, refer to I2S as Receiver .Values:0x0 (DISABLED): Receiver disabled0x1 (ENABLED):	R/W

		Receiver enabled	

3.20.4.3 ITER(0x8)

Bits	Name	Description	Memory Access
31:1	RSVD_ITER	RSVD_ITER Reserved bits - Read Only	R
0	TXEN	Transmitter block enable.This bit enables or disables the transmitter. A disable on this bit overrides any individual transmit channel enables. For more information about the transmitter block, refer to I2S as Transmitter .Values:0x0 (DISABLED): Transmitter disabled0x1 (ENABLED): Transmitter enabled	R/W

3.20.4.4 CER(0xc)

Bits	Name	Description	Memory Access
31:1	RSVD_CER	RSVD_CER Reserved bits - Read Only	R
0	CLKEN	Clock generation enable/disable.This bit enables/disables the clock generation signals when I2S is a master: sclk_en, ws_out, and sclk_gate. For more information about clock generation, refer to “Clock Generation (Master Mode)”.Values:0x0 (DISABLED): Clock generation disabled 0x1 (ENABLED): Clock generation enabled	R/W

3.20.4.5 CCR(0x10)

Bits	Name	Description	Memory Access
31:5	RSVD_CCR	RSVD_CCR Reserved bits - Read Only	R

4:3	WSS	These bits are used to program the number of sclk cycles for which the word select line (ws_out) stays in the left or right sample mode. The I2S Clock Generation block must be disabled (CER[0] = 0) prior to any changes in this value.Values:0x0 (CLOCK_CYCLES_16): 16 sclk cycles0x1 (CLOCK_CYCLES_24): 24 sclk cycles0x2 (CLOCK_CYCLES_32): 32 sclk cycles	R/W
2:0	SCLKG	These bits are used to program the gating of sclk. The programmed gating value must be less than or equal to the largest configured/programmed audio resolution to prevent the truncating of RX/TX data. The I2S Clock Generation block must be disabled (CER[0] = 0) before making any changes in this value.Values:0x0 (NO_CLOCK_GATING): Clock gating is disabled0x1 (CLOCK_CYCLES_12): Gating after 12 sclk cycles0x2 (CLOCK_CYCLES_16): Gating after 16 sclk cycles0x3 (CLOCK_CYCLES_20): Gating after 20 sclk cycles0x4 (CLOCK_CYCLES_24): Gating after 24 sclk cycles	R/W

3.20.4.6 RXFFR(0x14)

Bits	Name	Description	Memory Access
31:1	RSVD_RXFFR	RSVD_RXFFR Reserved bits - Write OnlyVolatile: true	W
0	RXFFR	Receiver FIFO Reset.Writing a 1 to this register flushes all the RX FIFOs (this is a self clearing bit). The Receiver Block must be disabled before writing to this bit.Values:0x0 (NO_FLUSH): Does not flush the RX FIFO0x1 (FLUSH): Flushes the RX FIFOVolatile: true	W

3.20.4.7 TXFFR(0x18)

Bits	Name	Description	Memory Access
31:1	RSVD_TXFFR	RSVD_TXFFR Reserved bits - Write OnlyVolatile: true	W

0	TXFFR	Transmitter FIFO Reset.Writing a 1 to this register flushes all the TX FIFOs (this is a self clearing bit). The Transmitter Block must be disabled prior to writing this bit.Values:0x0 (NO_FLUSH): Does not flush the TX FIFO0x1 (FLUSH): Flushes the TX FIFOVolatile: true	W

3.20.4.8 LRBR0(0x20)

Bits	Name	Description	Memory Access
31:16	RSVD_LRBx	RSVD_LRBRx Reserved bits - Read OnlyVolatile: true	R
15:0	LRBRx	The left stereo data received serially from the receive channel input (sdix). If the RX FIFO is full and the two-stage read operation (for instance, a read from LRBRx followed by a read from RRBRx) is not performed before the start of the next stereo pair, then the new data is lost and an overrun interrupt occurs. (data already in the RX FIFO is preserved.)Note: Before reading this register again, the right stereo data must be read from RRBRx or the status/interrupts will not be valid.Volatile: true	R

3.20.4.9 LTHR0(0x20)

Bits	Name	Description	Memory Access
31:16	RSVD_LTHRx	RSVD_LTHRx Reserved bits - Write OnlyVolatile: true	W
15:0	LTHRx	The left stereo data to be transmitted serially through the transmit channel output (sdox) is written through this register.Writing is a two-stage process:1. A write to this register passes the left stereo sample to the transmitter.2. This MUST be followed by writing the right stereo sample to the RTHRx register.Data must only be written to the FIFO when it is not full. Any attempt to write to a full FIFO results in that data being lost and an overrun interrupt being generated.Volatile: true	W

3.20.4.10 RRBR0(0x24)

Bits	Name	Description	Memory Access
31:16	RSVD_RRBRx	RSVD_RRBRx Reserved bits - Read OnlyVolatile: true	R
15:0	RRBRx	The right stereo data received serially from the receive channel input (sdix) is read through this register. If the RX FIFO is full and the two-stage read operation (for instance, read from LRBRx followed by a read from RRBRx) is not performed before the start of the next stereo pair, then the new data is lost and an overrun interrupt occurs. (Data already in the RX FIFO is preserved.)Note: Prior to reading this register, the left stereo data MUST be read from LRBRx, or the status/interrupts will not be valid.Volatile: true	R

3.20.4.11 RTHR0(0x24)

Bits	Name	Description	Memory Access
31:16	RSVD_RRBRx	RSVD_RRBRx Reserved bits - Read OnlyVolatile: true	R
15:0	RRBRx	The right stereo data received serially from the receive channel input (sdix) is read through this register. If the RX FIFO is full and the two-stage read operation (for instance, read from LRBRx followed by a read from RRBRx) is not performed before the start of the next stereo pair, then the new data is lost and an overrun interrupt occurs. (Data already in the RX FIFO is preserved.)Note: Prior to reading this register, the left stereo data MUST be read from LRBRx, or the status/interrupts will not be valid.Volatile: true	R

3.20.4.12 RER0(0x28)

Bits	Name	Description	Memory Access
31:1	RSVD_RERx	RSVD_RERx Reserved bits - Read Only	R
0	RXCHENx	Receive channel enable. This bit enables/disables a receive channel, independently of all other channels.On enable, the channel begins receiving on the next left stereo cycle.A global disable of I2S (IER[0] = 0) or the Receiver block (IRER[0] = 0) overrides this value.Values:0x0 (DISABLED): Receive Channel Disable0x1 (ENABLED): Receive Channel EnableValue After Reset: 0x1	R/W

3.20.4.13 TER0(0x2c)

Bits	Name	Description	Memory Access
31:1	RSVD_TERx	RSVD_TERx Reserved bits - Read Only	R
0	TXCHENx	Transmit channel enable. This bit enables/disables a transmit channel, independently of all other channels.On enable, the channel begins transmitting on the next left stereo cycle.A global disable of I2S (IER[0] = 0) or Transmitter block (ITER[0] = 0) overrides this value.Values:0x0 (DISABLED): Transmit Channel Disable0x1 (ENABLED): Transmit Channel EnableValue After Reset: 0x1	R/W

3.20.4.14 RCR0(0x30)

Bits	Name	Description	Memory Access
31:3	RSVD_RCRx	RSVD_RCRx Reserved bits - Read Only	R

2:0	WLEN	These bits are used to program the desired data resolution of the receiver and enables the LSB of the incoming left (or right) word to be placed in the LSB of the LRBRx (or RRBRx) register.Programmed data resolution must be less than or equal to I2S_RX_WORDSIZE_x. If the selected resolution is greater than the I2S_RX_WORDSIZE_x, the receive channel defaults back to I2S_RX_WORDSIZE_RESET_VALUE_x.The channel must be disabled prior to any changes in this value (RER0[0] = 0).Values:0x0 (IGNORE_WORD_LENGTH): Ignore the word length0x1 (RESOLUTION_12_BIT): 12-bit data resolution of the receiver.0x2 (RESOLUTION_16_BIT): 16-bit data resolution of the receiver.0x3 (RESOLUTION_20_BIT): 20-bit data resolution of the receiver.0x4 (RESOLUTION_24_BIT): 24-bit data resolution of the receiver.0x5 (RESOLUTION_32_BIT): 32-bit data resolution of the receiver.	R/W

3.20.4.15 TCR0(0x34)

Bits	Name	Description	Memory Access
31:3	RSVD_TCRx	RSVD_TCRx Reserved bits - Read Only	R
2:0	WLEN	These bits are used to program the data resolution of the transmitter and ensures the MSB of the data is transmitted first.Programmed resolution must be less than or equal to I2S_TX_WORDSIZE_x. If the selected resolution is greater than I2S_TX_WORDSIZE_x, the transmit channel defaults back to I2S_TX_WORDSIZE_RESET_VALUE_x value.The channel must be disabled prior to any changes in this value (TER0[0] = 0).Values:0x0 (IGNORE_WORD_LENGTH): Ignore the word length0x1 (RESOLUTION_12_BIT): 12-bit data resolution of the transmitter.0x2 (RESOLUTION_16_BIT): 16-bit data resolution of the	R/W

		transmitter.0x3 (RESOLUTION_20_BIT): 20-bit data resolution of the transmitter.0x4 (RESOLUTION_24_BIT): 24-bit data resolution of the transmitter.0x5 (RESOLUTION_32_BIT): 32-bit data resolution of the transmitter.	

3.20.4.16 ISR0(0x38)

Bits	Name	Description	Memory Access
31:6	RSVD31_6	RSVD31_6 Reserved bits - Read OnlyVolatile: true	R
5	TXFO	Status of Data Overrun interrupt for the TX channel.This bit specifies whether the TX FIFO write is valid or an overrun. Attempt to write to full TX FIFO.Values:0x0 (WRITE_VALID): TX FIFO write valid0x1 (WRITE_OVERRUN): TX FIFO write overrunValue After Reset: 0x0Volatile: true	R
4	TXFE	Status of Transmit Empty Trigger interrupt.This bit specifies whether the TX FIFO trigger level has reached or not. TX FIFO is empty.Values:0x0 (REACHED_TRIGGER_LEVEL): TX FIFO trigger level is reached0x1 (NOT_REACHED): TX FIFO trigger level is not reachedValue After Reset: 0x1Volatile: true	R
3:2	RSVD3_2	RSVD3_2 Reserved bits - Read OnlyVolatile: true	R
1	RXFO	Status of Data Overrun interrupt for the RX channel. Incoming data lost due to a full RX FIFO.Values:0x0 (WRITE_VALID): RX FIFO write valid0x1 (WRITE_OVERRUN): RX FIFO write overrunValue After Reset: 0x0Volatile: true	R
0	RXDA	Status of Receive Data Available interrupt. This bit denotes the status of the RX FIFO trigger level.Values:0x1 (REACHED_TRIGGER_LEVEL): RX FIFO trigger level is reached0x0 (NOT_REACHED): RX FIFO trigger level is not reachedValue After Reset: 0x0Volatile: true	R

3.20.4.17 IMR0(0x3c)

Bits	Name	Description	Memory Access
31:6	RSVD_IMR0_6_31	RSVD_IMR0_6_31 Reserved bits - Read Only	R
5	TXFOM	Mask TX FIFO Overrun interrupt.This bit masks or unmasks a TX FIFO overrun interrupt.Values:0x1 (MASK_INTERRUPT): Masks TX FIFO Overrun interrupt0x0 (UNMASK_INTERRUPT): Unmasks TX FIFO Overrun interruptValue After Reset: 0x1	R/W
4	TXFEM	Mask TX FIFO Empty interrupt.This bit masks or unmasks a TX FIFO Empty interrupt.Values:0x1 (MASK_INTERRUPT): Masks TX FIFO Empty interrupt0x0 (UNMASK_INTERRUPT): Unmasks TX FIFO Empty interruptValue After Reset: 0x1	R/W
3:2	RSVD_IMR0_2_3	RSVD_IMRO_2_3 Reserved bits - Read Only	R
1	RXFOM	unmasks an RX FIFO Overrun interrupt.Values:0x1 (MASK_INTERRUPT): Masks RX FIFO Overrun interrupt0x0 (UNMASK_INTERRUPT): Unmasks RX FIFO Overrun interruptValue After Reset: 0x1	R/W
0	RXDAM	Mask RX FIFO Data Available interrupt.This bit masks or unmasks an RX FIFO Data Available interrupt.Values:0x1 (MASK_INTERRUPT): Masks RX FIFO data available interrupt0x0 (UNMASK_INTERRUPT): Unmasks RX FIFO data available interruptValue After Reset: 0x1	R/W

3.20.4.18 ROR0(0x40)

Bits	Name	Description	Memory Access
31:1	RSVD_RORx	RSVD_RORx Reserved bits - Read OnlyVolatile: true	R
0	RXCHO	Read this bit to clear the RX FIFO Data Overrun interrupt.Values:0x0 (WRITE_VALID): RX FIFO write valid0x1 (WRITE_OVERRUN): RX FIFO write overrunVolatile: true	R

3.20.4.19 TOR0(0x44)

Bits	Name	Description	Memory Access
31:1	RSVD_TORx	RSVD_TORx Reserved bits - Read OnlyVolatile: true	R
0	TXCHO	Read this bit to clear the TX FIFO Data Overrun interrupt.Values:0x0 (WRITE_VALID): TX FIFO write valid0x1 (WRITE_OVERRUN): TX FIFO write overrunVolatile: true	R

3.20.4.20 RFCR0(0x48)

Bits	Name	Description	Memory Access
31:4	RSVD_RFCRx	RSVD_RFCRx Reserved bits - Read Only	R
3:0	RXCHDT	These bits program the trigger level in the RX FIFO at which the Received Data Available interrupt is generated.Trigger Level = Programmed Value + 1(for example, 1 to I2S_RX_FIFO_DEPTH_0)Valid RXCHDT values: 0 to (I2S_RX_FIFO_0 - 1)If an illegal value is programmed, these bits saturate to (I2S_RX_FIFO_0 - 1).The channel must be disabled prior to any changes in this value (that is, RER0[0] = 0).Values:0x0 (TRIGGER_LEVEL_1): Interrupt trigger when FIFO level is 1.0x1 (TRIGGER_LEVEL_2): Interrupt trigger when FIFO level is 2.0x2 (TRIGGER_LEVEL_3): Interrupt trigger when FIFO level is 3.0x3 (TRIGGER_LEVEL_4): Interrupt trigger when FIFO level is 4.0x4 (TRIGGER_LEVEL_5): Interrupt trigger when FIFO level is 5.0x5 (TRIGGER_LEVEL_6): Interrupt trigger when FIFO level is 6.0x6 (TRIGGER_LEVEL_7): Interrupt trigger when FIFO level is 7.0x7 (TRIGGER_LEVEL_8): Interrupt trigger when FIFO level is 8.0x8 (TRIGGER_LEVEL_9): Interrupt trigger when FIFO level is 9.0x9 (TRIGGER_LEVEL_10): Interrupt trigger when FIFO level is 10.0xa (TRIGGER_LEVEL_11): Interrupt trigger when FIFO level is 11.0xb (TRIGGER_LEVEL_12): Interrupt trigger when FIFO level is 12.0xc (TRIGGER_LEVEL_13): Interrupt	R/W

		trigger when FIFO level is 13.0xd (TRIGGER_LEVEL_14): Interrupt trigger when FIFO level is 14.0xe (TRIGGER_LEVEL_15): Interrupt trigger when FIFO level is 15.0xf (TRIGGER_LEVEL_16): Interrupt trigger when FIFO level is 16.	

3.20.4.21 TFCR0(0x4c)

Bits	Name	Description	Memory Access
31:4	RSVD_TFCRx	RSVD_TFCRx Reserved bits - Read Only	R
3:0	TXCHET	These bits program the trigger level in the TX FIFO at which the Empty Threshold Reached Interrupt is generated.Trigger Level = TXCHETTXCHET values: 0 to (I2S_TX_FIFO_0 - 1)If an illegal value is programmed, these bits saturate to (I2S_TX_FIFO_0 - 1). The channel must be disabled prior to any changes in this value (that is, TER0[0] = 0).Values:0x0 (TRIGGER_LEVEL_1): Interrupt trigger when FIFO level is 1.0x1 (TRIGGER_LEVEL_2): Interrupt trigger when FIFO level is 2.0x2 (TRIGGER_LEVEL_3): Interrupt trigger when FIFO level is 3.0x3 (TRIGGER_LEVEL_4): Interrupt trigger when FIFO level is 4.0x4 (TRIGGER_LEVEL_5): Interrupt trigger when FIFO level is 5.0x5 (TRIGGER_LEVEL_6): Interrupt trigger when FIFO level is 6.0x6 (TRIGGER_LEVEL_7): Interrupt trigger when FIFO level is 7.0x7 (TRIGGER_LEVEL_8): Interrupt trigger when FIFO level is 8.0x8 (TRIGGER_LEVEL_9): Interrupt trigger when FIFO level is 9.0x9 (TRIGGER_LEVEL_10): Interrupt trigger when FIFO level is 10.0xa (TRIGGER_LEVEL_11): Interrupt trigger when FIFO level is 11.0xb (TRIGGER_LEVEL_12): Interrupt trigger when FIFO level is 12.0xc (TRIGGER_LEVEL_13): Interrupt trigger when FIFO level is 13.0xd (TRIGGER_LEVEL_14): Interrupt trigger when FIFO level is 14.0xe (TRIGGER_LEVEL_15): Interrupt	R/W

		trigger when FIFO level is 15.0xf (TRIGGER_LEVEL_16): Interrupt trigger when FIFO level is 16.	

3.20.4.22 RFF0(0x50)

Bits	Name	Description	Memory Access
31:1	RSVD_RFFx	RSVD_RFFx Reserved bits - Write OnlyVolatile: true	W
0	RXCHFR	Receive Channel FIFO Reset.Writing a 1 to this register flushes an individual RX FIFO (This is a self clearing bit.). A Rx channel or block must be disabled prior to writing to this bit.Values:0x0 (NO_FLUSH): Does not flush an individual RX FIFO0x1 (FLUSH): Flushes an individual RX FIFOVolatile: true	W

3.20.4.23 TFF0(0x54)

Bits	Name	Description	Memory Access
31:1	RSVD_TFFx	RSVD_TFFx Reserved bits - Write OnlyVolatile: true	W
0	TXCHFR	Transmit Channel FIFO Reset.Writing a 1 to this register flushes channel s TX FIFO (This is a self clearing bit.). The TX channel or block must be disabled prior to writing to this bit.Values:0x0 (NO_FLUSH): Do not flushes channel s TX FIFO.0x1 (FLUSH): Flushes channel s TX FIFO.Volatile: true	W

3.20.4.24 LRBR1(0x60)

Bits	Name	Description	Memory Access
31:16	RSVD_LRBR1	RSVD_LRBR1 Reserved bits - Read OnlyVolatile: true	R

15:0	LRBR1	The left stereo data received serially from the receive channel input (sdi1) is read through this register. If the RX FIFO is full and the two-stage read operation (for instance, a read from LRBR1 followed by a read from RRBR1) is not performed before the start of the next stereo pair, then the new data is lost and an overrun interrupt occurs. (Data already in the RX FIFO is preserved.) Dependencies: I2S_RX_CHANNELS > 1 NOTE: Before reading this register again, the right stereo data MUST be read from RRBR1, or the status/interrupts will not be valid. Reset: 0x0Volatile: true	R

3.20.4.25 LTHR1(0x60)

Bits	Name	Description	Memory Access
31:16	RSVD_LTHR1	RSVD_LTHR1 Reserved bits - Write OnlyVolatile: true	W
15:0	LTHR1	The left stereo data to be transmitted serially through the transmit channel output (sdo1) is written through this register. Writing is a two-stage process: (1) A write to this register passes the left stereo sample to the transmitter. (2) This MUST be followed by writing the right stereo sample to the RTHR1 register. Data should only be written to the FIFO when it is not full. Any attempt to write to a full FIFO results in that data being lost and an overrun interrupt being generated. Dependencies: I2S_TX_CHANNELS > 1 Reset: 0x0Volatile: true	W

3.20.4.26 RRBR1(0x64)

Bits	Name	Description	Memory Access
31:16	RSVD_RRBR1	RSVD_RRBR1 Reserved bits - Read OnlyVolatile: true	R

15:0	RRBR1	The right stereo data received serially from the receive channel input (sdi1) is read through this register. If the RX FIFO is full and the two-stage read operation (for instance, read from LRBR1 followed by a read from RRBR1) is not performed before the start of the next stereo pair, then the new data is lost and an overrun interrupt occurs. (Data already in the RX FIFO is preserved.) Dependencies: I2S_RX_CHANNELS > 1 NOTE: Prior to reading this register, the left stereo data MUST be read from LRBR1, or the status/interrupts will not be valid. Reset: 0x0Volatile: true	R

3.20.4.27 RTHR1(0x64)

Bits	Name	Description	Memory Access
31:16	RSVD_RTHR1	RSVD_RTHR1 Reserved bits - Write OnlyVolatile: true	W
15:0	RTHR1	The right stereo data to be transmitted serially through the transmit channel output (sdo1) is written through this register. Writing is a two-stage process: (1) A left stereo sample MUST first be written to the LTHR1 register. (2) A write to this register passes the right stereo sample to the transmitter. Data should only be written to the FIFO when it is not full. Any attempt to write to a full FIFO results in that data being lost and an overrun interrupt being generated. Dependencies: I2S_TX_CHANNELS > 1 Reset: 0x0Volatile: true	W

3.20.4.28 RER1(0x68)

Bits	Name	Description	Memory Access
31:1	RSVD_RER1	RSVD_RER1 Reserved bits - Read Only	R

0	RXCHEN1	Receive channel enable. This bit enables/disables a receive channel, independently of all other channels. On enable, the channel begins receiving on the next left stereo cycle. A global disable of I2S (IER[0] = 0) or the Receiver block (IRER[0] = 0) overrides this value. 1: Enable 0: Disable Dependencies: I2S_RX_CHANNELS > 1 Reset: 1Values:0x0 (DISABLED): Receive Channel Disable.0x1 (ENABLED): Receive Channel Enable.	R/W

3.20.4.29 TER1(0x6c)

Bits	Name	Description	Memory Access
31:1	RSVD_TER1	RSVD_TER1 Reserved bits - Read Only	R
0	TXCHEN1	Transmit channel enable. This bit enables/disables a transmit channel, independently of all other channels. On enable, the channel begins transmitting on the next left stereo cycle. A global disable of I2S (IER[0] = 0) or Transmitter block (ITER[0] = 0) overrides this value. 0: Disable 1: Enable Dependencies: I2S_TX_CHANNELS > 1 Reset: 1Values:0x0 (DISABLED): Transmit Channel Disable.0x1 (ENABLED): Transmit Channel Enable.	R/W

3.20.4.30 RCR1(0x70)

Bits	Name	Description	Memory Access
31:3	RSVD_RCR1	RSVD_RCR1 Reserved bits - Read Only	R

2:0	WLEN	These bits are used to program the desired data resolution of the receiver and enables the LSB of the incoming left (or right) word to be placed in the LSB of the LRBR1 (or RRBR1) register. 000 = Ignore word length 001 = 12 bit resolution 010 = 16 bit resolution 011 = 20 bit resolution 100 = 24 bit resolution 101 = 32 bit resolution Programmed data resolution must be less than or equal to I2S_RX_WORDSIZE_1. If the selected resolution is greater than the I2S_RX_WORDSIZE_1, the receive channel defaults back to I2S_RX_WORDSIZE_RESET_VALUE_1. The channel must be disabled prior to any changes in this value (RER1[0] = 0). Dependencies: I2S_RX_CHANNELS > 1 Reset: I2S_RX_WORDSIZE_RESET_VALUE_1Values:0x0 (IGNORE_WORD_LENGTH): Ignore the word length0x1 (RESOLUTION_12_BIT): 12 bit data resolution of the receiver and enables the LSB of the incoming left (or right) word to be placed in the LSB of the LRBR1 (RRBR1) register0x2 (RESOLUTION_16_BIT): 16 bit data resolution of the receiver and enables the LSB of the incoming left (or right) word to be placed in the LSB of the LRBR1 (RRBR1) register0x3 (RESOLUTION_20_BIT): 20 bit data resolution of the receiver and enables the LSB of the incoming left (or right) word to be placed in the LSB of the LRBR1 (RRBR1) register0x4 (RESOLUTION_24_BIT): 24 bit data resolution of the receiver and enables the LSB of the incoming left (or right) word to be placed in the LSB of the LRBR1 (RRBR1) register0x5 (RESOLUTION_32_BIT): 32 bit data resolution of the receiver and enables the LSB of the incoming left (or right) word to be placed in the LSB of the LRBR1 (RRBR1) register	R/W

3.20.4.31 TCR1(0x74)

Bits	Name	Description	Memory Access
31:3	RSVD_TCR1	RSVD_TCR1 Reserved bits - Read Only	R

2:0	WLEN	These bits are used to program the data resolution of the transmitter and ensures the MSB of the data is transmitted first. 000 = Ignore word length 001 = 12 bit resolution 010 = 16 bit resolution 011 = 20 bit resolution 100 = 24 bit resolution 101 = 32 bit resolution Programmed resolution must be less than or equal to I2S_TX_WORDSIZE_1. If the selected resolution is greater than I2S_TX_WORDSIZE_1, the transmit channel defaults back to I2S_TX_WORDSIZE_RESET_VALUE_1 value. The channel must be disabled prior to any changes in this value (TER1[0] = 0). Dependencies: I2S_TX_CHANNELS > 1 Reset: I2S_TX_WORDSIZE_RESET_VALUE_1Values:0x0 (IGNORE_WORD_LENGTH): Ignoring the word length0x1 (RESOLUTION_12_BIT): 12 bit data resolution of the transmitter and ensures the MSB of the data is transmitted first0x2 (RESOLUTION_16_BIT): 16 bit data resolution of the transmitter and ensures the MSB of the data is transmitted first0x3 (RESOLUTION_20_BIT): 20 bit data resolution of the transmitter and ensures the MSB of the data is transmitted first0x4 (RESOLUTION_24_BIT): 24 bit data resolution of the transmitter and ensures the MSB of the data is transmitted first0x5 (RESOLUTION_32_BIT): 32 bit data resolution of the transmitter and ensures the MSB of the data is transmitted first	R/W

3.20.4.32 ISR1(0x78)

Bits	Name	Description	Memory Access
31:6	RSVD31_6	ISR1 Reserved field for RSVD31_6Volatile: true	R

5	TXFO	Status of Data Overrun interrupt for the TX channel. Attempt to write to full TX FIFO. 0: TX FIFO write valid 1: TX FIFO write overrun Dependencies: I2S_TX_CHANNELS > 1 Reset: 0Values:0x0 (WRITE_VALID): TX FIFO write valid0x1 (WRITE_OVERRUN): TX FIFO write overrunValue After Reset: 0x0Volatile: true	R
4	TXFE	Status of Transmit Empty Trigger interrupt. TX FIFO is empty. 1: trigger level reached 0: trigger level not reached Dependencies: I2S_TX_CHANNELS > 1 Reset: 1Values:0x0 (REACHED_TRIGGER_LEVEL): TX FIFO trigger level is reached0x1 (NOT_REACHED): TX FIFO trigger level is not reachedValue After Reset: 0x1Volatile: true	R
3:2	RSVD3_2	ISR1 Reserved field for RSVD3_2Volatile: true	R
1	RXFO	Status of Data Overrun interrupt for the RX channel. Incoming data lost due to a full RX FIFO. 0: RX FIFO write valid 1: RX FIFO write overrun Dependencies: I2S_RX_CHANNELS > 1 Reset: 0Values:0x0 (WRITE_VALID): RX FIFO write valid0x1 (WRITE_OVERRUN): RX FIFO write overrunValue After Reset: 0x0Volatile: true	R
0	RXDA	Status of Receive Data Available interrupt. RX FIFO data available. 1: trigger level reached 0: trigger level not reached Dependencies: I2S_RX_CHANNELS > 1 Reset: 0Values:0x1 (REACHED_TRIGGER_LEVEL): RX FIFO trigger level is reached0x0 (NOT_REACHED): RX FIFO trigger level is not reachedValue After Reset: 0x0Volatile: true	R

3.20.4.33 IMR1(0x7c)

Bits	Name	Description	Memory Access
31:6	RSVD_IMR1_6_31	RSVD_IMR1_6_31 Reserved bits - Read Only	R

5	TXFOM	Masks TX FIFO Overrun interrupt. 1: masks interrupt 0: unmasks interrupt Dependencies: I2S_TX_CHANNELS > 1 Reset: 0Values:0x1 (MASK_INTERRUPT): Masks TX FIFO Overrun interrupt.0x0 (UNMASK_INTERRUPT): Unmasks TX FIFO Overrun interrupt.Value After Reset: 0x1	R/W
4	TXFEM	Masks TX FIFO Empty interrupt. 1: masks interrupt 0: unmasks interrupt Dependencies: I2S_TX_CHANNELS > 1 Reset: 1Values:0x1 (MASK_INTERRUPT): Masks TX FIFO Empty interrupt.0x0 (UNMASK_INTERRUPT): Unmasks TX FIFO Empty interrupt.Value After Reset: 0x1	R/W
3:2	RSVD_IMR1_2_3	RSVD_IMR1_2_3 Reserved bits - Read Only	R
1	RXFOM	Masks RX FIFO Overrun interrupt. 1: masks interrupt 0: unmasks interrupt Dependencies: I2S_RX_CHANNELS > 1 Reset: 0Values:0x1 (MASK_INTERRUPT): Masks RX FIFO Overrun interrupt.0x0 (UNMASK_INTERRUPT): Unmasks RX FIFO Overrun interrupt.Value After Reset: 0x1	R/W
0	RXDAM	Masks RX FIFO Data Available interrupt. 1: masks interrupt 0: unmasks interrupt Dependencies: I2S_RX_CHANNELS > 1 Reset: 0Values:0x1 (MASK_INTERRUPT): Masks RX FIFO data available interrupt.0x0 (UNMASK_INTERRUPT): Unmasks RX FIFO data available interrupt.Value After Reset: 0x1	R/W

3.20.4.34 ROR1(0x80)

Bits	Name	Description	Memory Access
31:1	RSVD_ROR1	RSVD_ROR1 Reserved bits - Read OnlyVolatile: true	R
0	RXCHO	Read this bit to clear the RX FIFO Data Overrun interrupt. 0: RX FIFO write valid 1: RX FIFO write overrun Dependencies: I2S_RX_CHANNELS > 1 Reset: 0Values:0x0 (WRITE_VALID): RX FIFO write valid0x1 (WRITE_OVERRUN): RX FIFO write overrunVolatile: true	R

3.20.4.35 TOR1(0x84)

Bits	Name	Description	Memory Access
31:1	RSVD_TOR1	RSVD_TOR1 Reserved bits - Read OnlyVolatile: true	R
0	TXCHO	Read this bit to clear the TX FIFO Data Overrun interrupt. 0: TX FIFO write valid 1: TX FIFO write overrun Dependencies: I2S_TX_CHANNELS > 1 channel. Reset: 0Values:0x0 (WRITE_VALID): TX FIFO write valid0x1 (WRITE_OVERRUN): TX FIFO write overrunVolatile: true	R

3.20.4.36 RFCR1(0x88)

Bits	Name	Description	Memory Access
31:4	RSVD_RFCR1	RSVD_RFCR1 Reserved bits - Read Only	R
3:0	RXCHDT	These bits program the trigger level in the RX FIFO at which the Received Data Available interrupt is generated. Trigger Level = Programmed Value + 1 (for example, 1 to I2S_RX_FIFO_DEPTH_1) Valid RXCHDT values: 0 to (I2S_RX_FIFO_1 - 1) If an illegal value is programmed, these bits saturate to (I2S_RX_FIFO_1 - 1). The channel must be disabled prior to any changes in this value (that is, RER1[0] = 0). Dependencies: I2S_RX_CHANNELS > 1 Reset: I2S_RX_FIFO_THRE_1**Values:**0x0 (TRIGGER_LEVEL_1): Interrupt trigger when fifo level is 1.0x1 (TRIGGER_LEVEL_2): Interrupt trigger when fifo level is 2.0x2 (TRIGGER_LEVEL_3): Interrupt trigger when fifo level is 3.0x3 (TRIGGER_LEVEL_4): Interrupt trigger when fifo level is 4.0x4 (TRIGGER_LEVEL_5): Interrupt trigger when fifo level is 5.0x5 (TRIGGER_LEVEL_6): Interrupt trigger when fifo level is 6.0x6 (TRIGGER_LEVEL_7): Interrupt trigger when fifo level is 7.0x7 (TRIGGER_LEVEL_8): Interrupt trigger when fifo level is 8.0x8	R/W

		(TRIGGER_LEVEL_9): Interrupt trigger when fifo level is 9.0x9 (TRIGGER_LEVEL_10): Interrupt trigger when fifo level is 10.0xa (TRIGGER_LEVEL_11): Interrupt trigger when fifo level is 11.0xb (TRIGGER_LEVEL_12): Interrupt trigger when fifo level is 12.0xc (TRIGGER_LEVEL_13): Interrupt trigger when fifo level is 13.0xd (TRIGGER_LEVEL_14): Interrupt trigger when fifo level is 14.0xe (TRIGGER_LEVEL_15): Interrupt trigger when fifo level is 15.0xf (TRIGGER_LEVEL_16): Interrupt trigger when fifo level is 16.	

3.20.4.37 TFCR1(0x8c)

Bits	Name	Description	Memory Access
31:4	RSVD_TFCR1	RSVD_TFCR1 Reserved bits - Read Only	R
3:0	TXCHET	Transmit Channel Empty Trigger. These bits program the trigger level in the TX FIFO at which the Empty Threshold Reached Interrupt is generated. Trigger Level = TXCHET TXCHET values: 0 to (I2S_TX_FIFO_1 - 1) If an illegal value is programmed, these bits saturate to (I2S_TX_FIFO_1 - 1). The channel must be disabled prior to any changes in this value (that is, TER1[0] = 0). Dependencies: I2S_TX_CHANNELS > 1 Reset: I2S_TX_FIFO_THRE_1Values:0x0 (TRIGGER_LEVEL_1): Interrupt trigger when fifo level is 1.0x1 (TRIGGER_LEVEL_2): Interrupt trigger when fifo level is 2.0x2 (TRIGGER_LEVEL_3): Interrupt trigger when fifo level is 3.0x3 (TRIGGER_LEVEL_4): Interrupt trigger when fifo level is 4.0x4 (TRIGGER_LEVEL_5): Interrupt trigger when fifo level is 5.0x5 (TRIGGER_LEVEL_6): Interrupt trigger when fifo level is 6.0x6 (TRIGGER_LEVEL_7): Interrupt trigger when fifo level is 7.0x7 (TRIGGER_LEVEL_8): Interrupt trigger when fifo level is 8.0x8 (TRIGGER_LEVEL_9): Interrupt trigger when fifo level	R/W

		is 9.0x9 (TRIGGER_LEVEL_10): Interrupt trigger when fifo level is 10.0xa (TRIGGER_LEVEL_11): Interrupt trigger when fifo level is 11.0xb (TRIGGER_LEVEL_12): Interrupt trigger when fifo level is 12.0xc (TRIGGER_LEVEL_13): Interrupt trigger when fifo level is 13.0xd (TRIGGER_LEVEL_14): Interrupt trigger when fifo level is 14.0xe (TRIGGER_LEVEL_15): Interrupt trigger when fifo level is 15.0xf (TRIGGER_LEVEL_16): Interrupt trigger when fifo level is 16.	

3.20.4.38 RFF1(0x90)

Bits	Name	Description	Memory Access
31:1	RSVD_RFF1	RSVD_RFF1 Reserved bits - Write OnlyVolatile: true	W
0	RXCHFR	Receive Channel FIFO Reset. Writing a 1 to this register flushes an individual RX FIFO. (This is a self clearing bit.) RX channel or block must be disabled prior to writing to this bit. Dependencies: I2S_RX_CHANNELS > 1 Reset: 0Values:0x0 (NO_FLUSH): Do not flushes an individual RX FIFO.0x1 (FLUSH): Flushes the an inidividual RX FIFO.Volatile: true	W

3.20.4.39 TFF1(0x94)

Bits	Name	Description	Memory Access
31:1	RSVD_TFF1	RSVD_TFF1 Reserved bits - Write OnlyVolatile: true	W
0	TXCHFR	Transmit Channel FIFO Reset. Writing a 1 to this register flushes channel s TX FIFO. (This is a self clearing bit.) TX channel or block must be disabled prior to writing to this bit. Dependencies: I2S_TX_CHANNELS > 1 Reset: 0Values:0x0 (NO_FLUSH): Do not flushes channel s TX FIFO.0x1 (FLUSH): Flushes channel s TX FIFO.Volatile: true	W

3.20.4.40 LRBR2(0xa0)

Bits	Name	Description	Memory Access
31:16	RSVD_LRBR2	RSVD_LRBR2 Reserved bits - Read OnlyVolatile: true	R
15:0	LRBR2	The left stereo data received serially from the receive channel input (sdi2) is read through this register. If the RX FIFO is full and the two-stage read operation (for instance, a read from LRBR2 followed by a read from RRBR2) is not performed before the start of the next stereo pair, then the new data is lost and an overrun interrupt occurs. (Data already in the RX FIFO is preserved.) Dependencies: I2S_RX_CHANNELS > 2 NOTE: Before reading this register again, the right stereo data MUST be read from RRBR2, or the status/interrupts will not be valid. Reset: 0x0Volatile: true	R

3.20.4.41 LTHR2(0xa0)

Bits	Name	Description	Memory Access
31:16	RSVD_LTHR2	RSVD_LTHR2 Reserved bits - Write OnlyVolatile: true	W
15:0	LTHR2	The left stereo data to be transmitted serially through the transmit channel output (sdo2) is written through this register. Writing is a two-stage process: (1) A write to this register passes the left stereo sample to the transmitter. (2) This MUST be followed by writing the right stereo sample to the RTHR2 register. Data should only be written to the FIFO when it is not full. Any attempt to write to a full FIFO results in that data being lost and an overrun interrupt being generated. Dependencies: I2S TX CHANNELS > 2 Reset: 0x0Volatile: true	W

3.20.4.42 PRBR2(0xa4)

Bits	Name	Description	Memory Access
31:16	RSVD_RRBR2	RSVD_RRBR2 Reserved bits - Read OnlyVolatile: true	R
15:0	RRBR2	The right stereo data received serially from the receive channel input (sdi2) is read through this register. If the RX FIFO is full and the two-stage read operation (for instance, read from LRBR2 followed by a read from RRBR2) is not performed before the start of the next stereo pair, then the new data is lost and an overrun interrupt occurs. (Data already in the RX FIFO is preserved.) Dependencies: I2S_RX_CHANNELS > 2 NOTE: Prior to reading this register, the left stereo data MUST be read from LRBR2, or the status/interrupts will not be valid. Reset: 0x0Volatile: true	R

3.20.4.43 RTHR2(0xa4)

Bits	Name	Description	Memory Access
31:16	RSVD_RTHR2	RSVD_RTHR2 Reserved bits - Write OnlyVolatile: true	W
15:0	RTHR2	The right stereo data to be transmitted serially through the transmit channel output (sdo2) is written through this register. Writing is a two-stage process: (1) A left stereo sample MUST first be written to the LTHR2 register. (2) A write to this register passes the right stereo sample to the transmitter. Data should only be written to the FIFO when it is not full. Any attempt to write to a full FIFO results in that data being lost and an overrun interrupt being generated. Dependencies: I2S_TX_CHANNELS > 2 Reset: 0x0Volatile: true	W

3.20.4.44 RER2(0xa8)

Bits	Name	Description	Memory Access

31:1	RSVD_RER2	RSVD_RER2 Reserved bits - Read Only	R
0	RXCHEN2	Receive channel enable. This bit enables/disables a receive channel, independently of all other channels. On enable, the channel begins receiving on the next left stereo cycle. A global disable of I2S (IER[0] = 0) or the Receiver block (IRER[0] = 0) overrides this value. 1: Enable 0: Disable Dependencies: I2S_RX_CHANNELS > 2 Reset: 1Values:0x0 (DISABLED): Receive Channel Disable.0x1 (ENABLED): Receive Channel Enable.	R/W

3.20.4.45 TER2(0xac)

Bits	Name	Description	Memory Access
31:1	RSVD_TER2	RSVD_TER2 Reserved bits - Read Only	R
0	TXCHEN2	Transmit channel enable. This bit enables/disables a transmit channel, independently of all other channels. On enable, the channel begins transmitting on the next left stereo cycle. A global disable of I2S (IER[0] = 0) or Transmitter block (ITER[0] = 0) overrides this value. 0: Disable 1: Enable Dependencies: I2S_TX_CHANNELS > 2 Reset: 1Values:0x0 (DISABLED): Transmit Channel Disable.0x1 (ENABLED): Transmit Channel Enable.	R/W

3.20.4.46 RCR2(0xb0)

Bits	Name	Description	Memory Access
31:3	RSVD_RCR2	RSVD_RCR2 Reserved bits - Read Only	R
		These bits are used to program the desired data resolution of the receiver and enables the LSB of the incoming left (or right) word to be placed in the LSB of the LRBR2 (or RRBR2) register. 000 = Ignore word length 001 = 12 bit resolution 010 = 16 bit resolution 011 = 20 bit resolution 100 = 24 bit resolution 101 = 32 bit resolution Programmed data resolution must be less	

2:0	WLEN	than or equal to I2S_RX_WORDSIZE_2. If the selected resolution is greater than the I2S_RX_WORDSIZE_2, the receive channel defaults back to I2S_RX_WORDSIZE_RESET_VALUE_2. The channel must be disabled prior to any changes in this value (RER2[0] = 0). Dependencies: I2S_RX_CHANNELS > 2 Reset: I2S_RX_WORDSIZE_RESET_VALUE_2Values:0x0 (IGNORE_WORD_LENGTH): Ignore the word length0x1 (RESOLUTION_12_BIT): 12 bit data resolution of the receiver and enables the LSB of the incoming left (or right) word to be placed in the LSB of the LRBR2 (RRBR2) register0x2 (RESOLUTION_16_BIT): 16 bit data resolution of the receiver and enables the LSB of the incoming left (or right) word to be placed in the LSB of the LRBR2 (RRBR2) register0x3 (RESOLUTION_20_BIT): 20 bit data resolution of the receiver and enables the LSB of the incoming left (or right) word to be placed in the LSB of the LRBR2 (RRBR2) register0x4 (RESOLUTION_24_BIT): 24 bit data resolution of the receiver and enables the LSB of the incoming left (or right) word to be placed in the LSB of the LRBR2 (RRBR2) register0x5 (RESOLUTION_32_BIT): 32 bit data resolution of the receiver and enables the LSB of the incoming left (or right) word to be placed in the LSB of the LRBR2 (RRBR2) register	R/W

3.20.4.47 TCR2(0xb4)

Bits	Name	Description	Memory Access
31:3	RSVD_TCR2	RSVD_TCR2 Reserved bits - Read Only	R
		These bits are used to program the data resolution of the transmitter and ensures the MSB of the data is transmitted first. 000 = Ignore word length 001 = 12 bit resolution 010 = 16 bit resolution 011 = 20 bit resolution 100 = 24 bit resolution 101 = 32 bit resolution Programmed resolution must be less than or equal to	

2:0	WLEN	I2S_TX_WORDSIZE_2. If the selected resolution is greater than I2S_TX_WORDSIZE_2, the transmit channel defaults back to I2S_TX_WORDSIZE_RESET_VALUE_2 value. The channel must be disabled prior to any changes in this value (TER2[0] = 0). Dependencies: I2S_TX_CHANNELS > 2 Reset: I2S_TX_WORDSIZE_RESET_VALUE_2Values:0x0 (IGNORE_WORD_LENGTH): Ignore the word length0x1 (RESOLUTION_12_BIT): 12 bit data resolution of the transmitter and ensures the MSB of the data is transmitted first0x2 (RESOLUTION_16_BIT): 16 bit data resolution of the transmitter and ensures the MSB of the data is transmitted first0x3 (RESOLUTION_20_BIT): 20 bit data resolution of the transmitter and ensures the MSB of the data is transmitted first0x4 (RESOLUTION_24_BIT): 24 bit data resolution of the transmitter and ensures the MSB of the data is transmitted first0x5 (RESOLUTION_32_BIT): 32 bit data resolution of the transmitter and ensures the MSB of the data is transmitted first	R/W

3.20.4.48 ISR2(0xb8)

Bits	Name	Description	Memory Access
31:6	RSVD31_6	ISR2 Reserved field for RSVD31_6Volatile: true	R
5	TXFO	Status of Data Overrun interrupt for the TX channel. Attempt to write to full TX FIFO. 0: TX FIFO write valid 1: TX FIFO write overrun Dependencies: I2S_TX_CHANNELS > 2 Reset: 0Values:0x0 (WRITE_VALID): TX FIFO write valid0x1 (WRITE_OVERRUN): TX FIFO write overrunValue After Reset: 0x0Volatile: true	R

4	TXFE	Status of Transmit Empty Trigger interrupt. TX FIFO is empty. 1: trigger level reached 0: trigger level not reached Dependencies: I2S_TX_CHANNELS > 2 Reset: 1Values:0x0 (REACHED_TRIGGER_LEVEL): TX FIFO trigger level is reached0x1 (NOT_REACHED): TX FIFO trigger level is not reachedValue After Reset: 0x1Volatile: true	R
3:2	RSVD3_2	ISR2 Reserved field for RSVD3_2Volatile: true	R
1	RXFO	Status of Data Overrun interrupt for the RX channel. Incoming data lost due to a full RX FIFO. 0: RX FIFO write valid 1: RX FIFO write overrun Dependencies: I2S_RX_CHANNELS > 2 Reset: 0Values:0x0 (WRITE_VALID): RX FIFO write valid0x1 (WRITE_OVERRUN): RX FIFO write overrunValue After Reset: 0x0Volatile: true	R
0	RXDA	Status of Receive Data Available interrupt. RX FIFO data available. 1: trigger level reached 0: trigger level not reached Dependencies: I2S_RX_CHANNELS > 2 Reset: 0Values:0x1 (REACHED_TRIGGER_LEVEL): RX FIFO trigger level is reached0x0 (NOT_REACHED): RX FIFO trigger level is not reachedValue After Reset: 0x0Volatile: true	R

3.20.4.49 IMR2(0xbc)

Bits	Name	Description	Memory Access
31:6	RSVD_IMR2_6_31	RSVD_IMR2_6_31 Reserved bits - Read Only	R
5	TXFOM	Masks TX FIFO Overrun interrupt. 1: masks interrupt 0: unmasks interrupt Dependencies: I2S_TX_CHANNELS > 2 Reset: 0Values:0x1 (MASK_INTERRUPT): Masks TX FIFO Overrun interrupt.0x0 (UNMASK_INTERRUPT): Unmasks TX FIFO Overrun interrupt.Value After Reset: 0x1	R/W
4	TXFEM	Masks TX FIFO Empty interrupt. 1: masks interrupt 0: unmasks interrupt Dependencies: I2S_TX_CHANNELS > 2 Reset: 1Values:0x1 (MASK_INTERRUPT): Masks TX FIFO Empty interrupt.0x0 (UNMASK_INTERRUPT): Unmasks TX FIFO Empty interrupt.Value After Reset: 0x1	R/W

3:2	RSVD_IMR2_2_3	RSVD_IMR2_2_3 Reserved bits - Read Only	R
1	RXFOM	Masks RX FIFO Overrun interrupt. 1: masks interrupt 0: unmasks interrupt Dependencies: I2S_RX_CHANNELS > 2 Reset: 0Values:0x1 (MASK_INTERRUPT): Masks RX FIFO Overrun interrupt.0x0 (UNMASK_INTERRUPT): Unmasks RX FIFO Overrun interrupt.Value After Reset: 0x1	R/W
0	RXDAM	Masks RX FIFO Data Available interrupt. 1: masks interrupt 0: unmasks interrupt Dependencies: I2S_RX_CHANNELS > 2 Reset: 0Values:0x1 (MASK_INTERRUPT): Masks RX FIFO data available interrupt.0x0 (UNMASK_INTERRUPT): Unmasks RX FIFO data available interrupt.Value After Reset: 0x1	R/W

3.20.4.50 ROR2(0xc0)

Bits	Name	Description	Memory Access
31:1	RSVD_ROR2	RSVD_ROR2 Reserved bits - Read OnlyVolatile: true	R
0	RXCHO	Read this bit to clear the RX FIFO Data Overrun interrupt. 0: RX FIFO write valid 1: RX FIFO write overrun Dependencies: I2S_RX_CHANNELS > 2 Reset: 0Values:0x0 (WRITE_VALID): RX FIFO write valid0x1 (WRITE_OVERRUN): RX FIFO write overrunVolatile: true	R

3.20.4.51 TOR2(0xc4)

Bits	Name	Description	Memory Access
31:1	RSVD_TOR2	RSVD_TOR2 Reserved bits - Read OnlyVolatile: true	R

0	TXCHO	Read this bit to clear the TX FIFO Data Overrun interrupt. 0: TX FIFO write valid 1: TX FIFO write overrun Dependencies: I2S_TX_CHANNELS > 2 channel. Reset: 0Values:0x0 (WRITE_VALID): TX FIFO write valid0x1 (WRITE_OVERRUN): TX FIFO write overrunVolatile: true	R

3.20.4.52 RFCR2(0xc8)

Bits	Name	Description	Memory Access
31:4	RSVD_RFCR2	RSVD_RFCR2 Reserved bits - Read Only	R
3:0	RXCHDT	These bits program the trigger level in the RX FIFO at which the Received Data Available interrupt is generated. Trigger Level = Programmed Value + 1 (for example, 1 to I2S_RX_FIFO_DEPTH_2) Valid RXCHDT values: 0 to (I2S_RX_FIFO_2 - 1) If an illegal value is programmed, these bits saturate to (I2S_RX_FIFO_2 - 1). The channel must be disabled prior to any changes in this value (that is, RER2[0] = 0). Dependencies: I2S_RX_CHANNELS > 2 Reset: I2S_RX_FIFO_THRE_2Values:0x0 (TRIGGER_LEVEL_1): Interrupt trigger when fifo level is 1.0x1 (TRIGGER_LEVEL_2): Interrupt trigger when fifo level is 2.0x2 (TRIGGER_LEVEL_3): Interrupt trigger when fifo level is 3.0x3 (TRIGGER_LEVEL_4): Interrupt trigger when fifo level is 4.0x4 (TRIGGER_LEVEL_5): Interrupt trigger when fifo level is 5.0x5 (TRIGGER_LEVEL_6): Interrupt trigger when fifo level is 6.0x6 (TRIGGER_LEVEL_7): Interrupt trigger when fifo level is 7.0x7 (TRIGGER_LEVEL_8): Interrupt trigger when fifo level is 8.0x8 (TRIGGER_LEVEL_9): Interrupt trigger when fifo level is 9.0x9 (TRIGGER_LEVEL_10): Interrupt trigger when fifo level is 10.0xa (TRIGGER_LEVEL_11): Interrupt trigger when fifo level is 11.0xb (TRIGGER_LEVEL_12): Interrupt trigger when fifo	R/W

		level is 12.0xc (TRIGGER_LEVEL_13): Interrupt trigger when fifo level is 13.0xd (TRIGGER_LEVEL_14): Interrupt trigger when fifo level is 14.0xe (TRIGGER_LEVEL_15): Interrupt trigger when fifo level is 15.0xf (TRIGGER_LEVEL_16): Interrupt trigger when fifo level is 16.	

3.20.4.53 TFCR2(0xcc)

Bits	Name	Description	Memory Access
31:4	RSVD_TFCR2	RSVD_TFCR2 Reserved bits - Read Only	R
3:0	TXCHET	Transmit Channel Empty Trigger. These bits program the trigger level in the TX FIFO at which the Empty Threshold Reached Interrupt is generated. Trigger Level = TXCHET TXCHET values: 0 to (I2S_TX_FIFO_2 - 1) If an illegal value is programmed, these bits saturate to (I2S_TX_FIFO_2 - 1). The channel must be disabled prior to any changes in this value (that is, TER2[0] = 0). Dependencies: I2S_TX_CHANNELS > 2 Reset: I2S_TX_FIFO_THRE_2Values:0x0 (TRIGGER_LEVEL_1): Interrupt trigger when fifo level is 1.0x1 (TRIGGER_LEVEL_2): Interrupt trigger when fifo level is 2.0x2 (TRIGGER_LEVEL_3): Interrupt trigger when fifo level is 3.0x3 (TRIGGER_LEVEL_4): Interrupt trigger when fifo level is 4.0x4 (TRIGGER_LEVEL_5): Interrupt trigger when fifo level is 5.0x5 (TRIGGER_LEVEL_6): Interrupt trigger when fifo level is 6.0x6 (TRIGGER_LEVEL_7): Interrupt trigger when fifo level is 7.0x7 (TRIGGER_LEVEL_8): Interrupt trigger when fifo level is 8.0x8 (TRIGGER_LEVEL_9): Interrupt trigger when fifo level is 9.0x9 (TRIGGER_LEVEL_10): Interrupt trigger when fifo level is 10.0xa (TRIGGER_LEVEL_11): Interrupt	R/W

		trigger when fifo level is 11.0xb (TRIGGER_LEVEL_12): Interrupt trigger when fifo level is 12.0xc (TRIGGER_LEVEL_13): Interrupt trigger when fifo level is 13.0xd (TRIGGER_LEVEL_14): Interrupt trigger when fifo level is 14.0xe (TRIGGER_LEVEL_15): Interrupt trigger when fifo level is 15.0xf (TRIGGER_LEVEL_16): Interrupt trigger when fifo level is 16.	

3.20.4.54 RFF2(0xd0)

Bits	Name	Description	Memory Access
31:1	RSVD_RFF2	RSVD_RFF2 Reserved bits - Write OnlyVolatile: true	W
0	RXCHFR	Receive Channel FIFO Reset. Writing a 1 to this register flushes an individual RX FIFO. (This is a self clearing bit.) RX channel or block must be disabled prior to writing to this bit. Dependencies: I2S_RX_CHANNELS > 2 Reset: 0Values:0x0 (NO_FLUSH): Do not flushes an individual RX FIFO.0x1 (FLUSH): Flushes the an inidividual RX FIFO.Volatile: true	W

3.20.4.54 TFF2(0xd4)

Bits	Name	Description	Memory Access
31:1	RSVD_TFF2	RSVD_TFF2 Reserved bits - Write OnlyVolatile: true	W
0	TXCHFR	Transmit Channel FIFO Reset. Writing a 1 to this register flushes channel s TX FIFO. (This is a self clearing bit.) TX channel or block must be disabled prior to writing to this bit. Dependencies: I2S_TX_CHANNELS > 2 Reset: 0Values:0x0 (NO_FLUSH): Do not flushes channel s TX FIFO.0x1 (FLUSH): Flushes channel s TX FIFO.Volatile: true	W

3.20.4.55 LRBR3(0xe0)

Bits	Name	Description	Memory Access
31:16	RSVD_LRBR3	RSVD_LRBR3 Reserved bits - Read OnlyVolatile: true	R
15:0	LRBR3	The left stereo data received serially from the receive channel input (sdi3) is read through this register. If the RX FIFO is full and the two-stage read operation (for instance, a read from LRBR3 followed by a read from RRBR3) is not performed before the start of the next stereo pair, then the new data is lost and an overrun interrupt occurs. (Data already in the RX FIFO is preserved.) Dependencies: I2S_RX_CHANNELS > 3 NOTE: Before reading this register again, the right stereo data MUST be read from RRBR3, or the status/interrupts will not be valid. Reset: 0x0Volatile: true	R

3.20.4.56 LTHR3(0xe0)

Bits	Name	Description	Memory Access
31:16	RSVD_LTHR3	RSVD_LTHR3 Reserved bits - Write OnlyVolatile: true	W
15:0	LTHR3	The left stereo data to be transmitted serially through the transmit channel output (sdo3) is written through this register. Writing is a two-stage process: (1) A write to this register passes the left stereo sample to the transmitter. (2) This MUST be followed by writing the right stereo sample to the RTHR3 register. Data should only be written to the FIFO when it is not full. Any attempt to write to a full FIFO results in that data being lost and an overrun interrupt being generated. Dependencies: I2S_TX_CHANNELS > 3 Reset: 0x0Volatile: true	W

3.20.4.57 RRBR3(0xe4)

Bits	Name	Description	Memory Access

31:16	RSVD_RRBR3	RSVD_RRBR3 Reserved bits - Read OnlyVolatile: true	R
15:0	RRBR3	The right stereo data received serially from the receive channel input (sdi3) is read through this register. If the RX FIFO is full and the two-stage read operation (for instance, read from LRBR3 followed by a read from RRBR3) is not performed before the start of the next stereo pair, then the new data is lost and an overrun interrupt occurs. (Data already in the RX FIFO is preserved.) Dependencies: I2S_RX_CHANNELS > 3 NOTE: Prior to reading this register, the left stereo data MUST be read from LRBR3, or the status/interrupts will not be valid. Reset: 0x0Volatile: true	R

3.20.4.58 RTHR3(0xe4)

Bits	Name	Description	Memory Access
31:16	RSVD_RTHR3	RSVD_RTHR3 Reserved bits - Write OnlyVolatile: true	W
15:0	RTHR3	The right stereo data to be transmitted serially through the transmit channel output (sdo3) is written through this register. Writing is a two-stage process: (1) A left stereo sample MUST first be written to the LTHR3 register. (2) A write to this register passes the right stereo sample to the transmitter. Data should only be written to the FIFO when it is not full. Any attempt to write to a full FIFO results in that data being lost and an overrun interrupt being generated. Dependencies: I2S_TX_CHANNELS > 3 Reset: 0x0Volatile: true	W

3.20.4.59 RER3(0xe8)

Bits	Name	Description	Memory Access
31:1	RSVD_RER3	RSVD_RER3 Reserved bits - Read Only	R

0	RXCHEN3	Receive channel enable. This bit enables/disables a receive channel, independently of all other channels. On enable, the channel begins receiving on the next left stereo cycle. A global disable of I2S (IER[0] = 0) or the Receiver block (IRER[0] = 0) overrides this value. 1: Enable 0: Disable Dependencies: I2S_RX_CHANNELS > 3 Reset: 1Values:0x0 (DISABLED): Receive Channel Disable.0x1 (ENABLED): Receive Channel Enable.	R/W

3.20.4.60 TER3(0xec)

Bits	Name	Description	Memory Access
31:1	RSVD_TER3	RSVD_TER3 Reserved bits - Read Only	R
0	TXCHEN3	Transmit channel enable. This bit enables/disables a transmit channel, independently of all other channels. On enable, the channel begins transmitting on the next left stereo cycle. A global disable of I2S (IER[0] = 0) or Transmitter block (ITER[0] = 0) overrides this value. 0: Disable 1: Enable Dependencies: I2S_TX_CHANNELS > 3 Reset: 1Values:0x0 (DISABLED): Transmit Channel Disable.0x1 (ENABLED): Transmit Channel Enable.	R/W

3.20.4.61 RCR3(0xf0)

Bits	Name	Description	Memory Access
31:3	RSVD_RCR3	RSVD_RCR3 Reserved bits - Read Only	R
		These bits are used to program the desired data resolution of the receiver and enables the LSB of the incoming left (or right) word to be placed in the LSB of the LRBR3 (or RRBR3) register. 000 = Ignore word length 001 = 12 bit resolution 010 = 16 bit resolution 011 = 20 bit resolution 100 = 24 bit resolution 101 = 32 bit resolution Programmed data resolution must be less than or equal to I2S_RX_WORDSIZE_3. If the selected	

2:0	WLEN	resolution is greater than the I2S_RX_WORDSIZE_3, the receive channel defaults back to I2S_RX_WORDSIZE_RESET_VALUE_3. The channel must be disabled prior to any changes in this value (RER3[0] = 0). Dependencies: I2S_RX_CHANNELS > 3 Reset: I2S_RX_WORDSIZE_RESET_VALUE_3Values:0x0 (IGNORE_WORD_LENGTH): Ignore the word length0x1 (RESOLUTION_12_BIT): 12 bit data resolution of the receiver and enables the LSB of the incoming left (or right) word to be placed in the LSB of the LRBR3 (RRBR3) register0x2 (RESOLUTION_16_BIT): 16 bit data resolution of the receiver and enables the LSB of the incoming left (or right) word to be placed in the LSB of the LRBR3 (RRBR3) register0x3 (RESOLUTION_20_BIT): 20 bit data resolution of the receiver and enables the LSB of the incoming left (or right) word to be placed in the LSB of the LRBR3 (RRBR3) register0x4 (RESOLUTION_24_BIT): 24 bit data resolution of the receiver and enables the LSB of the incoming left (or right) word to be placed in the LSB of the LRBR3 (RRBR3) register0x5 (RESOLUTION_32_BIT): 32 bit data resolution of the receiver and enables the LSB of the incoming left (or right) word to be placed in the LSB of the LRBR3 (RRBR3) register	R/W

3.20.4.62 TCR3(0xf4)

Bits	Name	Description	Memory Access
31:3	RSVD_TCR3	RSVD_TCR3 Reserved bits - Read Only	R

2:0	WLEN	These bits are used to program the data resolution of the transmitter and ensures the MSB of the data is transmitted first. 000 = Ignore word length 001 = 12 bit resolution 010 = 16 bit resolution 011 = 20 bit resolution 100 = 24 bit resolution 101 = 32 bit resolution Programmed resolution must be less than or equal to I2S_TX_WORDSIZE_3. If the selected resolution is greater than I2S_TX_WORDSIZE_3, the transmit channel defaults back to I2S_TX_WORDSIZE_RESET_VALUE_3 value. The channel must be disabled prior to any changes in this value (TER3[0] = 0). Dependencies: I2S_TX_CHANNELS > 3 Reset: I2S_TX_WORDSIZE_RESET_VALUE_3Values:0x0 (IGNORE_WORD_LENGTH): Ignoring the word length0x1 (RESOLUTION_12_BIT): 12 bit data resolution of the transmitter and ensures the MSB of the data is transmitted first0x2 (RESOLUTION_16_BIT): 16 bit data resolution of the transmitter and ensures the MSB of the data is transmitted first0x3 (RESOLUTION_20_BIT): 20 bit data resolution of the transmitter and ensures the MSB of the data is transmitted first0x4 (RESOLUTION_24_BIT): 24 bit data resolution of the transmitter and ensures the MSB of the data is transmitted first0x5 (RESOLUTION_32_BIT): 32 bit data resolution of the transmitter and ensures the MSB of the data is transmitted first	R/W

3.20.4.63 ISR3(0xf8)

Bits	Name	Description	Memory Access
31:6	RSVD31_6	ISR3 Reserved field for RSVD31_6Volatile: true	R

5	TXFO	Status of Data Overrun interrupt for the TX channel. Attempt to write to full TX FIFO. 0: TX FIFO write valid 1: TX FIFO write overrun Dependencies: I2S_TX_CHANNELS > 3 Reset: 0Values:0x0 (WRITE_VALID): TX FIFO write valid0x1 (WRITE_OVERRUN): TX FIFO write overrunValue After Reset: 0x0Volatile: true	R
4	TXFE	Status of Transmit Empty Trigger interrupt. TX FIFO is empty. 1: trigger level reached 0: trigger level not reached Dependencies: I2S_TX_CHANNELS > 3 Reset: 1Values:0x0 (REACHED_TRIGGER_LEVEL): TX FIFO trigger level is reached0x1 (NOT_REACHED): TX FIFO trigger level is not reachedValue After Reset: 0x1Volatile: true	R
3:2	RSVD3_2	ISR3 Reserved field for RSVD3_2Volatile: true	R
1	RXFO	Status of Data Overrun interrupt for the RX channel. Incoming data lost due to a full RX FIFO. 0: RX FIFO write valid 1: RX FIFO write overrun Dependencies: I2S_RX_CHANNELS > 3 Reset: 0Values:0x0 (WRITE_VALID): RX FIFO write valid0x1 (WRITE_OVERRUN): RX FIFO write overrunValue After Reset: 0x0Volatile: true	R
0	RXDA	Status of Receive Data Available interrupt. RX FIFO data available. 1: trigger level reached 0: trigger level not reached Dependencies: I2S_RX_CHANNELS > 3 Reset: 0Values:0x1 (REACHED_TRIGGER_LEVEL): RX FIFO trigger level is reached0x0 (NOT_REACHED): RX FIFO trigger level is not reachedValue After Reset: 0x0Volatile: true	R

3.20.4.64 IMR3(0xfc)

Bits	Name	Description	Memory Access
31:6	RSVD_IMR3_6_31	RSVD_IMR3_6_31 Reserved bits - Read Only	R

5	TXFOM	Masks TX FIFO Overrun interrupt. 1: masks interrupt 0: unmasks interrupt Dependencies: I2S_TX_CHANNELS > 3 Reset: 0Values:0x1 (MASK_INTERRUPT): Masks TX FIFO Overrun interrupt.0x0 (UNMASK_INTERRUPT): Unmasks TX FIFO Overrun interrupt.Value After Reset: 0x1	R/W
4	TXFEM	Masks TX FIFO Empty interrupt. 1: masks interrupt 0: unmasks interrupt Dependencies: I2S_TX_CHANNELS > 3 Reset: 1Values:0x1 (MASK_INTERRUPT): Masks TX FIFO Empty interrupt.0x0 (UNMASK_INTERRUPT): Unmasks TX FIFO Empty interrupt.Value After Reset: 0x1	R/W
3:2	RSVD_IMR3_2_3	RSVD_IMR3_2_3 Reserved bits - Read Only	R
1	RXFOM	Masks RX FIFO Overrun interrupt. 1: masks interrupt 0: unmasks interrupt Dependencies: I2S_RX_CHANNELS > 3 Reset: 0Values:0x1 (MASK_INTERRUPT): Masks RX FIFO Overrun interrupt.0x0 (UNMASK_INTERRUPT): Unmasks RX FIFO Overrun interrupt.Value After Reset: 0x1	R/W
0	RXDAM	Masks RX FIFO Data Available interrupt. 1: masks interrupt 0: unmasks interrupt Dependencies: I2S_RX_CHANNELS > 3 Reset: 0Values:0x1 (MASK_INTERRUPT): Masks RX FIFO data available interrupt.0x0 (UNMASK_INTERRUPT): Unmasks RX FIFO data available interrupt.Value After Reset: 0x1	R/W

3.20.4.65 ROR3(0x100)

Bits	Name	Description	Memory Access
31:1	RSVD_ROR3	RSVD_ROR3 Reserved bits - Read OnlyVolatile: true	R

0	RXCHO	Read this bit to clear the RX FIFO Data Overrun interrupt. 0: RX FIFO write valid 1: RX FIFO write overrun Dependencies: I2S_RX_CHANNELS > 3 Reset: 0Values:0x0 (WRITE_VALID): RX FIFO write valid0x1 (WRITE_OVERRUN): RX FIFO write overrunVolatile: true	R

3.20.4.66 TOR3(0x104)

Bits	Name	Description	Memory Access
31:1	RSVD_TOR3	RSVD_TOR3 Reserved bits - Read OnlyVolatile: true	R
0	TXCHO	Read this bit to clear the TX FIFO Data Overrun interrupt. 0: TX FIFO write valid 1: TX FIFO write overrun Dependencies: I2S_TX_CHANNELS > 3 channel. Reset: 0Values:0x0 (WRITE_VALID): TX FIFO write valid0x1 (WRITE_OVERRUN): TX FIFO write overrunVolatile: true	R

3.20.4.67 RFCR3(0x108)

Bits	Name	Description	Memory Access
31:4	RSVD_RFCR3	RSVD_RFCR3 Reserved bits - Read Only	R
		These bits program the trigger level in the RX FIFO at which the Received Data Available interrupt is generated. Trigger Level = Programmed Value + 1 (for example, 1 to I2S_RX_FIFO_DEPTH_3) Valid RXCHDT values: 0 to (I2S_RX_FIFO_3 - 1) If an illegal value is programmed, these bits saturate to (I2S_RX_FIFO_3 - 1). The channel must be disabled prior to any changes in this value (that is, RER3[0] = 0). Dependencies: I2S_RX_CHANNELS > 3 Reset: I2S_RX_FIFO_THRE_3**Values:**0x0 (TRIGGER_LEVEL_1): Interrupt trigger when fifo level is 1.0x1 (TRIGGER_LEVEL_2): Interrupt trigger when fifo level is 2.0x2 (TRIGGER_LEVEL_3): Interrupt trigger when fifo level is 3.0x3 (TRIGGER_LEVEL_4):	

3:0	RXCHDT	Interrupt trigger when fifo level is 4.0x4 (TRIGGER_LEVEL_5): Interrupt trigger when fifo level is 5.0x5 (TRIGGER_LEVEL_6): Interrupt trigger when fifo level is 6.0x6 (TRIGGER_LEVEL_7): Interrupt trigger when fifo level is 7.0x7 (TRIGGER_LEVEL_8): Interrupt trigger when fifo level is 8.0x8 (TRIGGER_LEVEL_9): Interrupt trigger when fifo level is 9.0x9 (TRIGGER_LEVEL_10): Interrupt trigger when fifo level is 10.0xa (TRIGGER_LEVEL_11): Interrupt trigger when fifo level is 11.0xb (TRIGGER_LEVEL_12): Interrupt trigger when fifo level is 12.0xc (TRIGGER_LEVEL_13): Interrupt trigger when fifo level is 13.0xd (TRIGGER_LEVEL_14): Interrupt trigger when fifo level is 14.0xe (TRIGGER_LEVEL_15): Interrupt trigger when fifo level is 15.0xf (TRIGGER_LEVEL_16): Interrupt trigger when fifo level is 16.	R/W

3.20.4.68 TFCR3(0x10c)

Bits	Name	Description	Memory Access
31:4	RSVD_TFCR3	(RSVD_TFCR3) Reserved bits - Read Only	R
		Transmit Channel Empty Trigger. These bits program the trigger level in the TX FIFO at which the Empty Threshold Reached Interrupt is generated. Trigger Level = TXCHET TXCHET values: 0 to (I2S_TX_FIFO_3 - 1) If an illegal value is programmed, these bits saturate to (I2S_TX_FIFO_3 - 1). The channel must be disabled prior to any changes in this value (that is, TER3[0] = 0).Values:0x0 (TRIGGER_LEVEL_1): Interrupt trigger when fifo level is 1.0x1 (TRIGGER_LEVEL_2): Interrupt trigger when fifo level is 2.0x2 (TRIGGER_LEVEL_3): Interrupt trigger when fifo level is 3.0x3 (TRIGGER_LEVEL_4): Interrupt trigger when fifo level is 4.0x4 (TRIGGER_LEVEL_5): Interrupt	

3:0	TXCHET	trigger when fifo level is 5.0x5 (TRIGGER_LEVEL_6): Interrupt trigger when fifo level is 6.0x6 (TRIGGER_LEVEL_7): Interrupt trigger when fifo level is 7.0x7 (TRIGGER_LEVEL_8): Interrupt trigger when fifo level is 8.0x8 (TRIGGER_LEVEL_9): Interrupt trigger when fifo level is 9.0x9 (TRIGGER_LEVEL_10): Interrupt trigger when fifo level is 10.0xa (TRIGGER_LEVEL_11): Interrupt trigger when fifo level is 11.0xb (TRIGGER_LEVEL_12): Interrupt trigger when fifo level is 12.0xc (TRIGGER_LEVEL_13): Interrupt trigger when fifo level is 13.0xd (TRIGGER_LEVEL_14): Interrupt trigger when fifo level is 14.0xe (TRIGGER_LEVEL_15): Interrupt trigger when fifo level is 15.0xf (TRIGGER_LEVEL_16): Interrupt trigger when fifo level is 16.	R/W

3.20.4.69 RFF3(0x110)

Bits	Name	Description	Memory Access
31:1	RSVD_RFF3	(RSVD_RFF3) Reserved bits - Write OnlyVolatile: true	W
0	RXCHFR	Receive Channel FIFO ResetThis bit flushes an individual RX FIFO. (This is a self clearing bit.) The RX channel or block must be disabled prior to writing to this bit.Values:0x0 (NO_FLUSH): Do not flush an individual RX FIFO0x1 (FLUSH): Flushes an inidividual RX FIFOVolatile: true	W

3.20.4.70 TFF3(0x114)

Bits	Name	Description	Memory Access
31:1	RSVD_TFF3	(RSVD_TFF3) Reserved bits - Write OnlyVolatile: true	W

0	TXCHFR	Transmit Channel FIFO Reset. Writing a 1 to this register flushes channel s TX FIFO. (This is a self clearing bit.) TX channel or block must be disabled prior to writing to this bit.Values:0x0 (NO_FLUSH): Do not flushes channel s TX FIFO.0x1 (FLUSH): Flushes channel s TX FIFO.Volatile: true	W

3.20.4.71 RXDMA(0x1c0)

Bits	Name	Description	Memory Access
31:0	RXDMA	Receiver Block DMA Register.set_register_field_attribute $field DocRegisterResetValue {0x0} These bits are used to cycle repeatedly through the enabled receive channels (from lowest numbered to highest), reading stereo data pairs.Volatile: true	R

3.20.4.72 RRXDMA(0x1c4)

Bits	Name	Description	Memory Access
31:1	RSVD_RRXDMA	RSVD_RRXDMA Reserved bits - Write OnlyVolatile: true	W
0	RRXDMA	Reset Receiver Block DMA Register.Writing a 1 to this self-clearing register resets the RXDMA register mid-cycle to point to the lowest enabled Receive channel.Note: Writing to this register has no effect if the component is performing a stereo pair read (such as, when left stereo data has been read but not right stereo data).Values:0x1 (RESET): Reset Receiver Block DMA RegisterVolatile: true	W

3.20.4.73 TXDMA(0x1c8)

Bits	Name	Description	Memory Access

31:0	TXDMA	Transmitter Block DMA Register.The register bits can be used to cycle repeatedly through the enabled Transmit channels (from lowest numbered to highest) to allow writing of stereo data pairs.Volatile: true	W

3.20.4.74 RTXDMA(0x1cc)

Bits	Name	Description	Memory Access
31:1	RSVD_RTXDMA	RSVD_RTXDMA Reserved bits - Write OnlyVolatile: true	W
0	RTXDMA	Reset Transmitter Block DMA Register.Writing a 1 to this self-clearing register resets the TXDMA register mid-cycle to point to the lowest enabled Transmit channel.Note: This register has no effect in the middle of a stereo pair write (such as, when left stereo data has been written but not right stereo data).Values:0x1 (RESET): Reset Transmitter Block DMA RegisterVolatile: true	W

3.20.4.75 I2S_COMP_PARAM_2(0x1f0)

Bits	Name	Description	Memory Access
31:13	RSVD_31_13	RSVD_31_13 Reserved bits - Read Only	R
12:10	I2S_RX_WORDSIZE_3	These bits specify the RX resolution for WORDSIZE_3.Values:0x0 (RESOLUTION_12_BIT): 12-bit Resolution0x1 (RESOLUTION_16_BIT): 16-bit Resolution0x2 (RESOLUTION_20_BIT): 20-bit Resolution0x3 (RESOLUTION_24_BIT): 24-bit Resolution0x4 (RESOLUTION_32_BIT): 32-bit Resolution	R

Bits	Name	Description	Memory Access
9:7	I2S_RX_WORDSIZE_2	These bits specify the RX resolution for WORDSIZE_2.Values:0x0 (RESOLUTION_12_BIT): 12-bit Resolution0x1 (RESOLUTION_16_BIT): 16-bit Resolution0x2 (RESOLUTION_20_BIT): 20-bit Resolution0x3 (RESOLUTION_24_BIT): 24-bit Resolution0x4 (RESOLUTION_32_BIT): 32-bit Resolution	R
6	RSVD_I2S_COMP_PARAM_2_6	RSVD_I2S_COMP_PARAM_2_6 Reserved bits - Read Only	R
5:3	I2S_RX_WORDSIZE_1	These bits specify the RX resolution for WORDSIZE_1.Values:0x0 (RESOLUTION_12_BIT): 12-bit Resolution0x1 (RESOLUTION_16_BIT): 16-bit Resolution0x2 (RESOLUTION_20_BIT): 20-bit Resolution0x3 (RESOLUTION_24_BIT): 24-bit Resolution0x4 (RESOLUTION_32_BIT): 32-bit Resolution	R
2:0	I2S_RX_WORDSIZE_0	These bits specify the RX resolution for WORDSIZE_0.Values:0x0 (RESOLUTION_12_BIT): 12-bit Resolution0x1 (RESOLUTION_16_BIT): 16-bit Resolution0x2 (RESOLUTION_20_BIT): 20-bit Resolution0x3 (RESOLUTION_24_BIT): 24-bit Resolution0x4 (RESOLUTION_32_BIT): 32-bit	R

		Resolution	

3.20.4.76 I2S_COMP_PARAM_1(0x1f4)

Bits	Name	Description	Memory Access
31:28	RSVD_PARAM_1_28_31	RSVD_I2S_COMP_PARAM_1_28_31 Reserved bits - Read Only	R
27:25	I2S_TX_WORDSIZE_3	These bits specify the TX resolution for WORDSIZE_3.Values:0x0 (RESOLUTION_12_BIT): 12-bit Resolution0x1 (RESOLUTION_16_BIT): 16-bit Resolution0x2 (RESOLUTION_20_BIT): 20-bit Resolution0x3 (RESOLUTION_24_BIT): 24-bit Resolution0x4 (RESOLUTION_32_BIT): 32-bit Resolution	R
24:22	I2S_TX_WORDSIZE_2	These bits specify the TX resolution for WORDSIZE_2.Values:0x0 (RESOLUTION_12_BIT): 12-bit Resolution0x1 (RESOLUTION_16_BIT): 16-bit Resolution0x2 (RESOLUTION_20_BIT): 20-bit Resolution0x3 (RESOLUTION_24_BIT): 24-bit	R

		Resolution0x4 (RESOLUTION_32_BIT): 32-bit Resolution	
21:19	I2S_TX_WORDSIZE_1	These bits specify the TX resolution for WORDSIZE_1.Values:0x0 (RESOLUTION_12_BIT): 12-bit Resolution0x1 (RESOLUTION_16_BIT): 16-bit Resolution0x2 (RESOLUTION_20_BIT): 20-bit Resolution0x3 (RESOLUTION_24_BIT): 24-bit Resolution0x4 (RESOLUTION_32_BIT): 32-bit Resolution	R
18:16	I2S_TX_WORDSIZE_0	These bits specify the TX resolution for WORDSIZE_0.Values:0x0 (RESOLUTION_12_BIT): 12-bit Resolution0x1 (RESOLUTION_16_BIT): 16-bit Resolution0x2 (RESOLUTION_20_BIT): 20-bit Resolution0x3 (RESOLUTION_24_BIT): 24-bit Resolution0x4 (RESOLUTION_32_BIT): 32-bit Resolution	R
15:11	RSVD_PARAM_1_11_15	RSVD_I2S_COMP_PARAM_1_11_15 Reserved bits - Read Only	R

Bits	Name	Description	Memory Access

10:9	I2S_TX_CHANNELS	These bits specify the number of TX channels.Values:0x0 (TX_CHANNEL_1): 1 Transmit Channel0x1 (TX_CHANNEL_2): 2 Transmit Channels0x2 (TX_CHANNEL_3): 3 Transmit Channels0x3 (TX_CHANNEL_4): 4 Transmit Channels	R
8:7	I2S_RX_CHANNELS	These bits specify the number of RX channels.Values:0x0 (RX_CHANNEL_1): 1 Receive Channel0x1 (RX_CHANNEL_2): 2 Receive Channels0x2 (RX_CHANNEL_3): 3 Receive Channels0x3 (RX_CHANNEL_4): 4 Receive Channels	R
6	I2S_RECEIVER_BLOCK	This bit specifies whether the receiver block is enabled or not.Values:0x0 (FALSE): Receiver block is enabled0x1 (TRUE): Receiver block is disabled	R
5	I2S_TRANSMITTER_BLOCK	This bit specifies whether the transmitter block is enabled or not.Values:0x0 (FALSE): Transmitter block is enabled0x1 (TRUE): Transmitter block is disabled	R
4	I2S_MODE_EN	This bit specifies whether the master mode is enabled or not.Values:0x0 (FALSE): Master mode is enabled0x1 (TRUE): Master mode is disabled	R
3:2	I2S_FIFO_DEPTH_GLOBAL	These bits specify the FIFO depth for TX and RX channels.Values:0x0 (FIFO_DEPTH_2): FIFO depth is equals to 2 for TX and RX channels0x0 (FIFO_DEPTH_4): FIFO depth is equals to 4 for TX and RX channels0x0 (FIFO_DEPTH_8): FIFO depth is equals to 8 for TX and RX channels0x0 (FIFO_DEPTH_16): FIFO depth is equals to 16 for TX and RX channels	R

1:0	APB_DATA_WIDTH	These bits specify the APB data width.Values:0x0 (BITS_8): 8 bits APB data width0x0 (BITS_16): 16 bits APB data width0x0 (BITS_32): 32 bits APB data width	R

3.20.4.77 I2S_COMP_VERSION(0x1f8)

Bits	Name	Description	Memory Access
31:0	I2S_COMP_VERSION	These bits specify the I2S component version. The value for I2S_COMP_VERSION are described in the DesignWare Synthesizable Components for AMBA 2, AMBA 3 AXI, and AMBA 4 AXI Release Notes .	R

3.20.4.78 I2S_COMP_TYPE(0x1fc)

Bits	Name	Description	Memory Access
31:0	I2S_COMP_TYPE	DesignWare Component Type number = 0x445701a0.This unique hexadecimal value is constant and is derived from the two ASCII letters DW followed by a 16-bit unsigned number.	R

3.21 定时器 (TIMER)

3.21.1 概述

系统拥有3个TIMER模块，它们有如下特性：

- 时钟独立可配；

- 每个中断的可配置极性；
- 单个或组合中断输出标志可配置；
- 可配置启用定时器切换输出的可编程脉冲宽度调制；
- 可配置选项，包括定时器切换输出的脉冲宽度调制，0%和100%占空比。

3.21.2 功能描述

- 预分频器

  请参考sysctrl章节。

- 报警产生

  定时器可以触发报警，报警则会引发重新加载触发中断，如果报警寄存器TIMER1LOADCOUNT的值等于当前定时器的值，则报警触发。为解决寄存器设置过晚，计数器值超过报警值的问题，当前定时器值高于（适用于向上定时器）或低于（适用于向下定时器）当前报警值也会触发报警。在这种情况下，使能报警功能会马上触发报警。

- 中断

  TIMERNINTSTAT 该中断由定时器上的报警事件产生。

- 编程参考

  使用以下编程流程启用0%和100%占空比模式：

  1） 禁用TimerNControlReg寄存器中的定时器启用位；

  2） 用适当的值对TimerNLoadCount和TimerNLoadCount2寄存器进行编程；

  3） 启用0%和100%占空比模式位、脉冲宽度调制位并设置定时器模式,在TimerNControlReg寄存器中设置为用户定义的计数模式；

  4） 在TimerNControlReg寄存器中设置定时器启用位，使切换输出为100%（高）或0%（低）。当0%和100%占空比模式启用时，内部定时器被禁用。内部计时器可以通过切换到正常切换输出模式或脉冲宽度调制切换再次启用输出模式。

  用户可配置 PWM 定时器模块的以下功能：

  - 通过指定 PWM 定时器频率或周期来控制事件发生的频率；
  - 配置特定 PWM 定时器与其他 PWM 定时器或模块同步；
  - PWM 定时器与其他 PWM 定时器或模块同相；

- 设置定时器计数模式：递增，递减，或递增递减循环计数模式；
- 使用预分频器更改 PWM 定时器时钟（PT_clk）的速率。每个定时器都有自己的预分频器，通过寄存器PWM_TIMER0_CFG0_REG 的 PWM_TIMERx_PRESCALE 进行配置。PWM 定时器根据该寄存器的设置以较慢的速度递增或递减。

3.21.3 寄存器列表

Base Address：TIMER0_BASE_ADDR (0x502D0000)

Base Address：TIMER1_BASE_ADDR (0x502E0000)

Base Address：TIMER2_BASE_ADDR (0x502F0000)

Name	Description	Offset address
TIMER1LOADCOUNT	Timer N Load Count Register Value to be loaded into Timer N	0x0
TIMER1CURRENTVAL	Current value of Timer N	0x4
TIMER1CONTROLREG	Timer N Control Register Control Register for Timer N. This register controls enabling, operating mode (free-running or defined-count), and interrupt mask of Timer N.	0x8
TIMER1EOI	Timer N End-of-Interrupt Register Clears the interrupt from Timer N	0xc
TIMER1INTSTAT	Timer N Interrupt Status Register Contains the interrupt status for Timer N	0x10
TIMER2LOADCOUNT	Timer2 Load Count Register Name: Timer2 Load Count Register Size: 8-32 bits	0x14
TIMER2CURRENTVAL	Timer2 Current Value Name: Timer2 Current Value Size: 8-32 bits	0x18
TIMER2CONTROLREG	Timer2 Control Register	0x1c
TIMER2EOI	Timer2 End-of-Interrupt Register Name: Timer2 End-of-Interrupt Register Size: 1 bit	0x20
TIMER2INTSTAT	Timer2 Interrupt Status Register	0x24
TIMER3LOADCOUNT	Timer1 Load Count Register	0x28

TIMER3CURRENTVAL	Timer3 Current Value	0x2c
TIMER3CONTROLREG	Timer3 Control Register	0x30
TIMER3EOI	Timer3 End-of-Interrupt Register	0x34
TIMER3INTSTAT	Timer3 Interrupt Status Register	0x38
TIMER4LOADCOUNT	Timer4 Load Count Register Name: Timer4 Load Count Register Size: 8-32 bits Address 60 Read/Write Access: Read/Write	0x3c
TIMER4CURRENTVAL	Timer4 Current Value Register Name: Timer4 Current Value Size: 8-32 bits Address 64 Read/Write Access: Read	0x40

Name	Description	Offset address
TIMER4CONTROLREG	Timer4 Control Register Name: Timer4 Control Register Size: 4 bits Address 68 Read/Write Access: Read/Write This register controls enabling, operating mode (free-running or defined-count), and interrupt mask of Timer4. You can program each Timer4ControlReg to enable or disable a specific timer and to control its mode of operation.	0x44
TIMER4EOI	Timer4 End-of-Interrupt Register Name: Timer4 End-of-Interrupt Register Size: 1 bit Address 72 Read/Write Access: Read	0x48
TIMER4INTSTAT	Timer4 Interrupt Status Register Name: Timer4 Interrupt Status Register Size: 1 bit Address 76 Read/Write Access: Read	0x4c
TimersIntStatus	Timers Interrupt Status Register Contains the interrupt status of all timers in the component.	0xa0
TimersEOI	Timers End-of-Interrupt Register Returns all zeroes (0) and clears all active interrupts.	0xa4
TimersRawIntStatus	Timers Raw Interrupt Status Register Contains the unmasked interrupt status of all timers in the component.	0xa8

TIMERS_COMP_VERSION	Timers Component Version Current revision number of the apb_timers component.	0xac
TIMER1LOADCOUNT2	Timer N Load Count2 Register Value to be loaded into Timer N when toggle output changes from 0 to 1	0xb0
TIMER2LOADCOUNT2	Timer2 Load Count2 Register Name: Timer2 Load Count2 Register Size: 8-32 bits Address 180 Read/Write Access: Read/Write	0xb4
TIMER3LOADCOUNT2	Timer3 Load Count2 Register Name: Timer3 Load Count2 Register Size: 8-32 bits Address 184 Read/Write Access: Read/Write	0xb8
TIMER4LOADCOUNT2	Timer4 Load Count2 Register Name: Timer4 Load Count2 Register Size: 8-32 bits Address 188 Read/Write Access: Read/Write	0xbc

3.21.4 寄存器设置

3.21.4.1 TIMER1LOADCOUNT(0x0)

Bits	Name	Description	Memory Access
31:0	TimerNLoadCount	Value to be loaded into Timer N. This is the value from which counting commences. Any value written to this register is loaded into the associated timer.	R/W

3.21.4.2 TIMER1CURRENTVAL(0x4)

Bits	Name	Description	Memory Access

31:0	TimerNCurrentValue	Current Value of Timer N. When TIM_NEWMODE=0, This register is supported only when timer_N_clk is synchronous to pclk. Reading this register when using independent clocks results in an undefined value. When TIM_NEWMODE=1, no restrictions apply. Volatile: true	R

3.21.4.3 TIMER1CONTROLREG(0x8)

Bits	Name	Description	Memory Access
31:5	RSVD_TimerNControlReg	TimerNControlReg 31 to 5 Reserved field	R
4	RSVD_TIMER_0N100PWM_EN	TimerNControlReg 4 Reserved field	R
3	TIMER_PWM	Pulse Width Modulation of timer_N_toggle output. This field is only present when TIM_NEWMODE is enabled Values: 0x1 (ENABLED): PWM for timer_N_toggle o/p is enabled 0x0 (DISABLE): PWM for timer_N_toggle o/p is disabled	R/W
2	TIMER_INTERRUPT_MASK	Timer interrupt mask for Timer N. Values: 0x1 (MASKED): Timer N interrupt is masked 0x0 (UNMASKED): Timer N interrupt is unmasked	R/W
1	TIMER_MODE	Timer mode for Timer N. Note: You must set the Timer1LoadCount register to all 1s before enabling the timer in free-running mode. Values: 0x1 (USER_DEFINED): User-	R/W

		Defined mode of operation 0x0 (FREE_RUNNING): Free Running mode of operation	
0	TIMER_ENABLE	Timer enable bit for Timer N. Values: 0x1 (ENABLED): Timer N is enabled 0x0 (DISABLE): Timer N is disabled	R/W

3.21.4.4 TIMER1EOI(0xc)

Bits	Name	Description	Memory Access
31:1	RSVD_TimerNEOI	TimerNEOI 31to1 Reserved field	R
0	TimerNEOI	Reading from this register returns all zeroes (0) and clears the interrupt from Timer N.	R

3.21.4.5 TIMER1INTSTAT(0x10)

Bits	Name	Description	Memory Access
31:1	RSVD_TimerNIntStatus	TimerNIntStatus 31 to 1 Reserved field Volatile: true	R
0	TimerNIntStatus	Contains the interrupt status for Timer N. Values: 0x1 (ACTIVE): Timer N Interrupt is active 0x0 (INACTIVE): Timer N Interrupt is inactive Volatile: true	R

3.21.4.6 TIMER2LOADCOUNT(0x14)

Bits	Name	Description	Memory Access

31:0	TIMER2LOADCOUNT	Value to be loaded into Timer2. This is the value from which counting commences. Any value written to this register is loaded into the associated timer.	R/W

3.21.4.7 TIMER2CURRENTVAL(0x18)

Bits	Name	Description	Memory Access
31:0	TIMER2CURRENTVAL	Current Value of Timer2. This register is supported only when timer_2_clk is synchronous to pclk. Reading this register when using independent clocks results in an undefined value. Volatile: true	R

3.21.4.8 TIMER2CONTROLREG(0x1c)

Bits	Name	Description	Memory Access
31:5	RSVD_TIMER2CONTROLREG	TimerNControlReg 31 to 5 Reserved field	R
4	RSVD_TIMER_0N100PWM_EN	TimerNControlReg 4 Reserved field	R

Bits	Name	Description	Memory Access
3	TIMER_PWM	Pulse Width Modulation of timer_2_toggle output. 0: Disabled 1: Enabled This field is only present when TIM_NEWMODE is enabled Values: 0x1 (ENABLED): PWM for timer_2_toggle o/p is enabled 0x0 (DISABLE): PWM for timer_2_toggle o/p is disabled	R/W

2	TIMER_INTERRUPT_MASK	Timer interrupt mask for Timer2. 0: not masked 1: masked Values: 0x1 (MASKED): Timer2 interrupt is masked 0x0 (UNMASKED): Timer2 interrupt is unmasked	R/W
1	TIMER_MODE	Timer mode for Timer2. 0: free-running mode 1: user-defined count mode NOTE: You must set the Timer2LoadCount register to all 1s before enabling the timer in free-running mode. Values: 0x1 (USER_DEFINED): User-Defined mode of operation 0x0 (FREE_RUNNING): Free Running mode of operation	R/W
0	TIMER_ENABLE	Timer enable bit for Timer2. 0: disable 1: enable Values: 0x1 (ENABLED): Timer2 is enabled 0x0 (DISABLE): Timer2 is disabled	R/W

3.21.4.9 TIMER2EOI(0x20)

Bits	Name	Description	Memory Access
31:1	RSVD_TIMER2EOI	TimerNEOI 31 to 1 Reserved field	R
0	TIMER2EOI	Reading from this register returns all zeroes (0) and clears the interrupt from Timer2.	R

3.21.4.10 TIMER2INTSTAT(0x24)

Bits	Name	Description	Memory Access
31:1	RSVD_TIMER2INTSTAT	TimerNIntStatus 31 to 1 Reserved field Volatile: true	R

0	TIMER2INTSTAT	Contains the interrupt status for Timer2. Values: 0x1 (ACTIVE): Timer2 Interrupt is active 0x0 (INACTIVE): Timer2 Interrupt is inactive Volatile: true	R

3.21.4.11 TIMER3LOADCOUNT(0x28)

Bits	Name	Description	Memory Access
31:0	TIMER3LOADCOUNT	Value to be loaded into Timer3. This is the value from which counting commences. Any value written to this register is loaded into the associated timer.	R/W

3.21.4.12 TIMER3CURRENTVAL(0x2c)

Bits	Name	Description	Memory Access
31:0	TIMER3CURRENTVAL	Current Value of Timer3. This register is supported only when timer_3_clk is synchronous to pclk. Reading this register when using independent clocks results in an undefined value. Volatile: true	R

3.21.4.13 TIMER3CONTROLREG(0x30)

Bits	Name	Description	Memory Access
31:5	RSVD_TIMER3CONTROLREG	TimerNControlReg 31 to 5 Reserved field	R
4	RSVD_TIMER_0N100PWM_EN	TimerNControlReg 4 Reserved field	R

3	TIMER_PWM	Pulse Width Modulation of timer_3_toggle output. 0: Disabled 1: Enabled This field is only present when TIM_NEWMODE is enabled Values: 0x1 (ENABLED): PWM for timer_3_toggle o/p is enabled 0x0 (DISABLE): PWM for timer_3_toggle o/p is disabled	R/W
2	TIMER_INTERRUPT_MASK	Timer interrupt mask for Timer3. 0: not masked 1: masked Values: 0x1 (MASKED): Timer3 interrupt is masked 0x0 (UNMASKED): Timer3 interrupt is unmasked	R/W
1	TIMER_MODE	Timer mode for Timer3. 0: free-running mode 1: user-defined count mode NOTE: You must set the Timer3LoadCount register to all 1s before enabling the timer in free-running mode. Values: 0x1 (USER_DEFINED): User-Defined mode of operation 0x0 (FREE_RUNNING): Free Running mode of operation	R/W
0	TIMER_ENABLE	Timer enable bit for Timer3. 0: disable 1: enable Values: 0x1 (ENABLED): Timer3 is enabled 0x0 (DISABLE): Timer3 is disabled	R/W

3.21.4.14 TIMER3EOI(0x34)

Bits	Name	Description	Memory Access
31:1	RSVD_TIMER3EOI	TimerNEOI 31 to 1 Reserved field	R
0	TIMER3EOI	Reading from this register returns all zeroes (0) and clears the interrupt from Timer3.	R

3.21.4.15 TIMER3INTSTAT(0x38)

Bits	Name	Description	Memory Access
31:1	RSVD_TIMER3INTSTAT	TimerNIntStatus 31 to 1 Reserved field Volatile: true	R
0	TIMER3INTSTAT	Contains the interrupt status for Timer3. Values: 0x1 (ACTIVE): Timer3 Interrupt is active 0x0 (INACTIVE): Timer3 Interrupt is inactive Volatile: true	R

3.21.4.16 TIMER4LOADCOUNT(0x3c)

Bits	Name	Description	Memory Access
31:0	TIMER4LOADCOUNT	Value to be loaded into Timer4. This is the value from which counting commences. Any value written to this register is loaded into the associated timer.	R/W

3.21.4.17 TIMER4CURRENTVAL(0x40)

Bits	Name	Description	Memory Access

31:0	TIMER4CURRENTVAL	Current Value of Timer4. This register is supported only when timer_4_clk is synchronous to pclk. Reading this register when using independent clocks results in an undefined value. Volatile: true	R

3.21.4.18 TIMER4CONTROLREG(0x44)

Bits	Name	Description	Memory Access
31:5	RSVD_TIMER4CONTROLREG	TimerNControlReg 31 to 5 Reserved field	R
4	RSVD_TIMER_0N100PWM_EN	TimerNControlReg 4 Reserved field	R
3	TIMER_PWM	Pulse Width Modulation of timer_4_toggle output. 0: Disabled 1: Enabled This field is only present when TIM_NEWMODE is enabled Values: 0x1 (ENABLED): PWM for timer_4_toggle o/p is enabled 0x0 (DISABLE): PWM for timer_4_toggle o/p is disabled	R/W
2	TIMER_INTERRUPT_MASK	Timer interrupt mask for Timer4. 0: not masked 1: masked Values: 0x1 (MASKED): Timer4 interrupt is masked 0x0 (UNMASKED): Timer4 interrupt is unmasked	R/W
1	TIMER_MODE	Timer mode for Timer4. 0: free-running mode 1: user-defined count mode NOTE: You must set the Timer4LoadCount register to all 1s before enabling the timer in free-running mode. Values: 0x1 (USER_DEFINED): User-Defined mode of operation	R/W

		0x0 (FREE_RUNNING): Free Running mode of operation	
0	TIMER_ENABLE	Timer enable bit for Timer4. 0: disable 1: enable Values: 0x1 (ENABLED): Timer4 is enabled 0x0 (DISABLE): Timer4 is disabled	R/W

3.21.4.19 TIMER4EOI(0x48)

Bits	Name	Description	Memory Access
31:1	RSVD_TIMER4EOI	TimerNEOI 31 to 1 Reserved field	R
0	TIMER4EOI	Reading from this register returns all zeroes (0) and clears the interrupt from Timer4.	R

3.21.4.20 TIMER4INTSTAT(0x4c)

Bits	Name	Description	Memory Access
31:1	RSVD_TIMER4INTSTAT	TimerNIntStatus 31 to 1 Reserved field Volatile: true	R
0	TIMER4INTSTAT	Contains the interrupt status for Timer4. Values: 0x1 (ACTIVE): Timer4 Interrupt is active 0x0 (INACTIVE): Timer4 Interrupt is inactive Volatile: true	R

3.21.4.21 TimersIntStatus(0xa0)

Bits	Name	Description	Memory Access
31:4	RSVD_TimersIntStatus	TimersIntStatus 31 to NUM_TIMERS Reserved field Volatile: true	R
3:0	TimersIntStatus	Contains the interrupt status of all timers in the component. If a bit of this register is 0, then the corresponding timer interrupt is not active and the corresponding interrupt could be on either the timer_intr bus or the timer_intr_n bus, depending on the interrupt polarity you have chosen. Similarly, if a bit of this register is 1, then the corresponding interrupt bit has been set in the relevant interrupt bus. In both cases, the status reported is the status after the interrupt mask has been applied. Reading from this register does not clear any active interrupts. Values: 0x1 (ACTIVE): Timer_intr(_n) is active 0x0 (INACTIVE): Timer_intr(_n) is inactive Volatile: true	R

3.21.4.22 TimersEOI(0xa4)

Bits	Name	Description	Memory Access
31:4	RSVD_TIMERSEOI	TimersEOI 31toNUM_TIMERS Reserved field	R
3:0	TIMERSEOI	Reading this register returns all zeroes (0) and clears all active interrupts.	R

3.21.4.23 TimersRawIntStatus(0xa8)

Bits	Name	Description	Memory Access

31:4	RSVD_TIMERSRAWINTSTAT	TimersRawIntStatus 31 to NUM_TIMERS Reserved field Volatile: true	R
3:0	TIMERSRAWINTSTAT	The register contains the unmasked interrupt status of all timers in the component. Values: 0x1 (ACTIVE): Raw Timer_intr(_n) is active 0x0 (INACTIVE): Raw Timer_intr(_n) is Inactive Volatile:true	R

3.21.4.24 TIMERS_COMP_VERSION(0xac)

Bits	Name	Description	Memory Access
31:0	TIMERSCOMPVERSION	Current revision number of the apb_timers component. For the value, see the releases table in the AMBA 2 release notes	R

3.21.4.25 TIMER1LOADCOUNT2(0xb0)

Bits	Name	Description	Memory Access
31:0	TIMERNLOADCOUNT2	Value to be loaded into Timer N when timer_N_toggle output changes from 0 to 1. This value determines the width of the HIGH period of the timer_N_toggle output.	R/W

3.21.4.26 TIMER2LOADCOUNT2(0xb4)

Bits	Name	Description	Memory Access

31:0	TIMER2LOADCOUNT2	Value to be loaded into Timer2 when timer_2_toggle output changes from 0 to 1. This value determines the width of the HIGH period of the timer_2_toggle output.	R/W

3.21.4.27 TIMER3LOADCOUNT2(0xb8)

Bits	Name	Description	Memory Access
31:0	TIMER3LOADCOUNT2	Value to be loaded into Timer3 when timer_3_toggle output changes from 0 to 1. This value determines the width of the HIGH period of the timer_3_toggle output.	R/W

3.21.4.28 TIMER4LOADCOUNT2(0xbc)

Bits	Name	Description	Memory Access
31:0	TIMER4LOADCOUNT2	Value to be loaded into Timer4 when timer 4 toggle output changes from 0 to 1.This value determines the width of the HIGH period of the timer_4_toggle output.	

3.22 高速异步串行收发器 (GPIOHS)

3.22.1 概述

芯片有32个高速GPIO。每个IO可以分配到FPIOA上48个管脚之一。

高速GPIO：

高速GPIO为GPIOHS，共32个，具有如下特点：

- 可配置输入输出信号；
- 每个IO具有独立中断源；
- 中断支持边沿触发和电平触发；

- 每个IO可以分配到FPIOA上48个管脚之一；
- 可配置上拉电阻，下拉电阻、或者高阻；

3.22.2 功能描述

- **简单GPIO输入输出**

高速GPIO配置寄存器input_en相应的位为1，则相应IO为输入，配置寄存器output_en相应的位为1，则相应的IO为输出。寄存器input_val和output_val分别用来读取输入值和设置输出值。

- **配置的一般流程**

4. 通过FPIOA配置GPIO的映射IO管脚；
5. 配置GPIO的基本属性（输入输出、上下拉等）；
6. 读取或者输入GPIO状态；

3.22.3 寄存器列表

Base Address：GPIOHS_BASE_ADDR (**0x38001000**)

Name	Description	Offset Address
input_val	输入值	0x00
input_en	输入使能	0x04
output_en	输出使能	0x08
output_val	输出值	0x0c
pullup_en	内部上拉使能	0x10
drive	加强驱动力	0x14
rise_ie	上升沿中断使能	0x18
rise_ip	上升沿中断挂起	0x1c
fall_ie	下升沿中断使能	0x20
fall_ip	下升沿中断挂起	0x24
high_ie	高电平中断使能	0x28

high_ip	高电平中断挂起	0x2c
low_ie	低电平中断使能	0x30
low_ip	低电平中断挂起	0x34
iof_en	IO特殊功能使能	0x38
iof_sel	IO特殊功能启用	0x3c
output_xor	输出翻转	0x40

配置注意：高速GPIO共有32个，寄存器都是32位的，每一位代表相应的IO号，