UM3506 用户手册

版本：V1.3

广芯微电子（广州）股份有限公司

http://www.unicmicro.com/

条款协议

本文档的所有部分，其著作产权归广芯微电子（广州）股份有限公司（以下简称广芯微电子）所有，未经广芯微电子授权许可，任何个人及组织不得复制、转载、仿制本文档的全部或部分组件。本文档没有任何形式的担保、立场表达或其他暗示，若有任何因本文档或其中提及的产品所有资讯所引起的直接或间接损失，广芯微电子及所属员工恕不为其担保任何责任。除此以外，本文档所提到的产品规格及资讯仅供参考，内容亦会随时更新，恕不另行通知。

1. 本文档中所记载的关于电路、软件和其他相关信息仅用于说明半导体产品的操作和应用实例。用户如在设备设计中应用本文档中的电路、软件和相关信息，请自行负责。对于用户或第三方因使用上述电路、软件或信息而遭受的任何损失，广芯微电子不承担任何责任。
2. 在准备本文档所记载的信息的过程中，广芯微电子已尽量做到合理注意，但是，广芯微电子并不保证这些信息都是准确无误的。用户因本文档中所记载的信息的错误或遗漏而遭受的任何损失，广芯微电子不承担任何责任。
3. 对于因使用本文档中的广芯微电子产品或技术信息而造成的侵权行为或因此而侵犯第三方的专利、版权或其他知识产权的行为，广芯微电子不承担任何责任。本文档所记载的内容不应视为对广芯微电子或其他人所有的专利、版权或其他知识产权作出任何明示、默示或其它方式的许可及授权。
4. 使用本文档中记载的广芯微电子产品时，应在广芯微电子指定的范围内，特别是在最大额定值、电源工作电压范围、热辐射特性、安装条件以及其他产品特性的范围内使用。对于在上述指定范围之外使用广芯微电子产品而产生的故障或损失，广芯微电子不承担任何责任。
5. 虽然广芯微电子一直致力于提高广芯微电子产品的质量和可靠性，但是，半导体产品有其自身的具体特性，如一定的故障发生率以及在某些使用条件下会发生故障等。此外，广芯微电子产品均未进行防辐射设计。所以请采取安全保护措施，以避免当广芯微电子产品在发生故障而造成火灾时导致人身事故、伤害或损害的事故。例如进行软硬件安全设计（包括但不限于冗余设计、防火控制以及故障预防等）、适当的老化处理或其他适当的措施等。

版本修订

版本	日期	描述
V1.0	2019.04.26	Initial version
V1.1	2019.05.07	Update GPIO registers and analog trim registers, clock structure diagram is also updated. New general DMA is described.
V1.2	2019.08.31	Update registers description.
V1.3	2022.07.22	更新文档格式

目录

1 Introduction

The device target at the latest USB interface solution that complies with USB Power Delivery Rev. 3.0 and Type-C Rev1.3 specifications. It provides a flexible programmable architecture that allows for continuous evolution of specification and wide range functionality extension beyond PD application.
The device internally structured in TCPM/TCPC layered architecture to achieve a complete PD/Type-C system, including operating as source, sink, or DRP, depending on the customer application. The SoC chip integrates one native TCPC-like front end which including essential digital logic and analog circuits for Type-C interface detection and control, dead-battery power-up, packet BMC Encoding/Decoding in PD PHY layer, and timing-critical functions in PD Protocol layer.
The device also integrates innovative RISC-V ISA based 32-bit MCU core as centralized universal TCPM processor. The optimized RISC-V core incorporate on-chip Flash/SRAM memories, enhanced peripherals and extensively system resource to implements the upper layer protocol of PD specification in proven software package. In additional, it enables flexibility to manage power policy, extension to customized functionalities on control and interactive in one PD plus differentiation system.
This documentation of technical reference manual document describes each functional block of the UM3506 device in detail and complements the datasheets of UM3506 Series, providing information required for application and in particular software development.

1.1 Features

- RISC-V CPU core.
- Flash-based architecture that allows dynamic SW configuration and upgrades.
- Compliant with Type-C 1.3 and Power Delivery 3.0 specification
- Supports the following USB-PD 3.0 optional features:
  - Programming power supply (PPS),
  - Fast Role Swap (FRS),
  - Extended messages with 260Bytes
- Optimized Open-source RISC-V ISA Based 32bit MCU running @ 33MHz
- On-chip Flash/SRAM
- High-integration to reduce BOM material
  - Integrates Type-C detect circuit with configuration CS/Rd
  - Standard-compliant Type-C Configuration Channel, 1.05~1.2V voltage level
  - Integrates USB PD PHY and protocol layers
  - Integrates ROSC, PLL, High voltage LDO, PRG, POR etc.
  - Integrates multiple channel ADC for CC detection etc.
- Integrates Low side CSA,
- Integrates Shunt regulator
- Extensive peripherals, 2*UART/2*I2C/SPI/6*PWM/GPIO
- AON for Low-power mode
- ICP, ISP flash program

2 Architecture

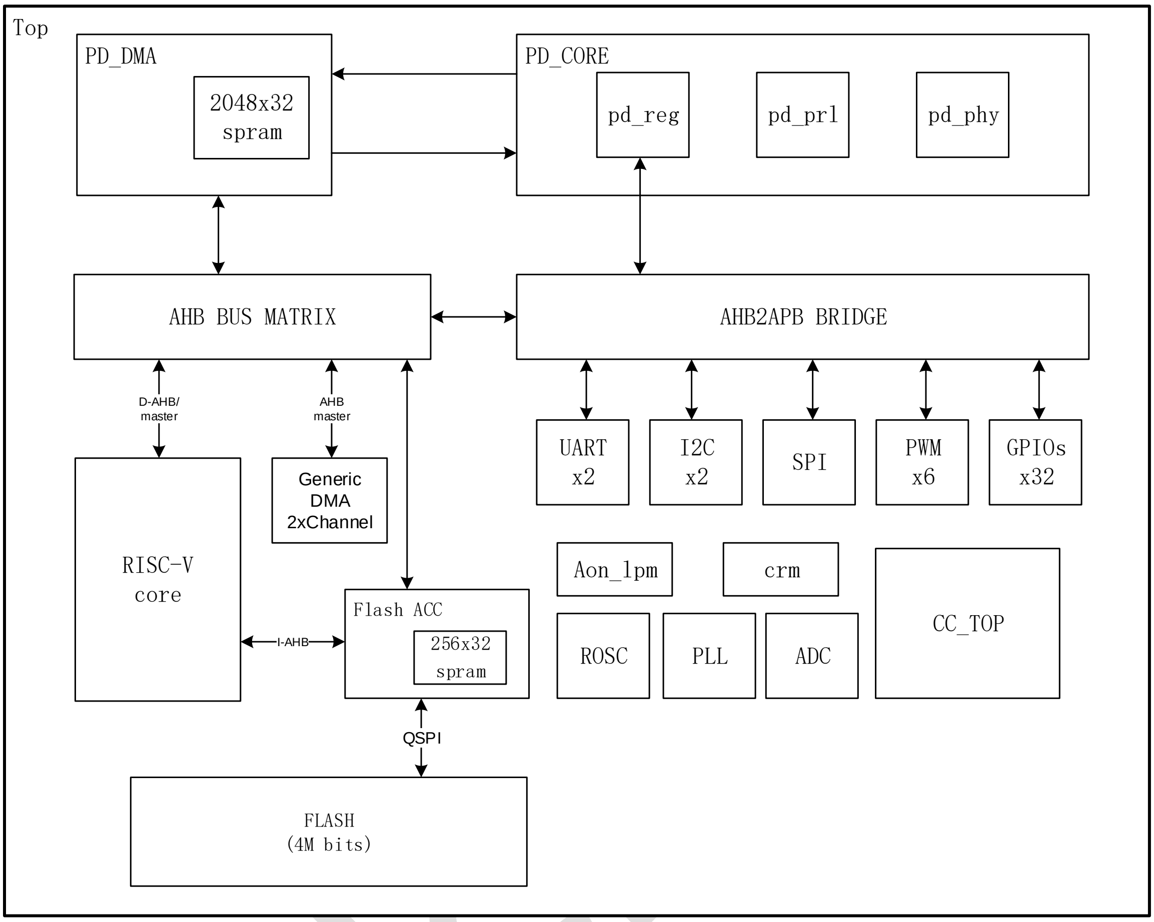


Figure 2-1 TOP block diagram

As above figure illustrates, Device top level block diagram include:

- PD_CORE

Includes USB PD PHY, protocol layer and related PD control registers which could be accessed by SW.

- PD_DMA, PD_CORE switches message with MCU through the PD_DMA module.
- RISC-V MCU CORE
- AHB Bus matrix.
- AHB2APB bridge
- FLASH hard macro, 4M bits.
- Flash access accelerator is implemented for XIP with QSPI FLASH.
- A serials of peripherals of MCU such as UART, I2C, SPI and multi-channel PWM.
- Clock system
- Power system

3 RISC-V Core

RISC-V is a free and open instruction set architecture (ISA) enabling a new era of processor innovation through open standard collaboration. It has been developed to provide a low-cost platform that meets the needs of embedded MCU implementation.
The device embeds an optimized RISC-V core for power and area as part of the 32-bit MCU subsystem, which execute RV32I base instruction set with M and C extensions.
The RISC-V core in device is load-store architecture, where only load and store instructions access memory and arithmetic instructions only operate on integer registers. The core provides a 32-bit user address space that is byte-addressed and little endian. The execution environment will define what portions of the address space are legal to access.

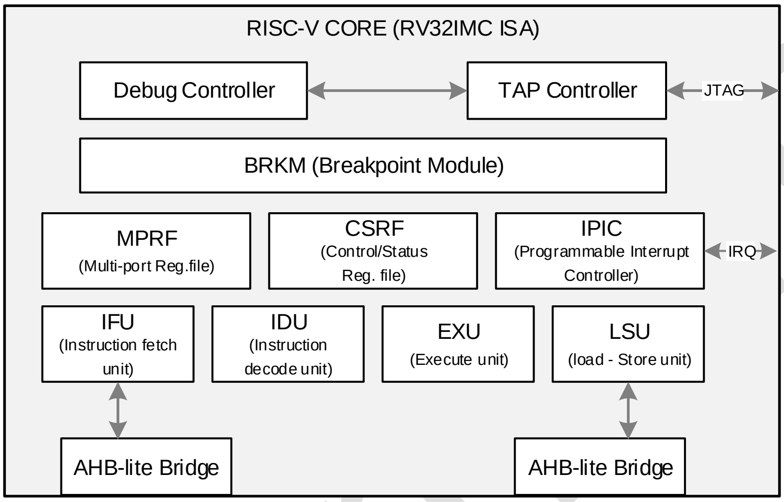


Figure 3-1 RISC-V Core

Summary of key features:

- Harvard architecture (separate instruction and data buses)
- 32-bit AHB-Lite external memory interface
- Machine privilege level
- 32 32-bit general purpose integer registers
- Instruction set is RV32I with M and C extensions
- 47 Integer (32-bit) instructions
- 27 Compact (16-bit) instructions
- 8 Multiply/Divide instructions
- Multiple stage pipeline implementation
- Integrated Programmable Interrupt Controller
- Low interrupt latency
- up to 8 IRQ lines
- HW Breakpoint, Debug Controller with JTAG interface
- 3 embedded 64bit performance counters
- Real time clock
- Cycle counter
- Instructions-retired counter
- Optimized for area and power consumption

For information on the RISC-V core, Pls. visit http://www.riscv.org..

4 Memory

4.1 Memory Map

Memory Space Address	Memory block Name
0x4000_0D08 0x4000_0D00	SPI Flash indirect access control registers
0x4000_0C2C 0x4000_0C00	GPIO Port1 registers
0x4000_0B2C 0x4000_0B00	GPIO Port0 registers
0x4000_0A0C 0x4000_0A00	SPI registers
0x4000_0920 0x4000_0900	PWM registers
0x4000_0818 0x4000_0800	Second I2C registers
0x4000_0718 0x4000_0700	I2C registers
0x4000_0620 0x4000_0600	Secondary UART registers
0x4000_0520 0x4000_0500	UART registers
0x4000_00FF 0x4000_0000	PD registers
0x6000_1FFF 0x6000_0000	8KB Memory for Data RAM and PD messages DMA buffers
0x00FF_FFFF 0x0000_0000	Maximum 16 MB Code Size

Table 4- 1 Memory Map Table

5 Interrupt

5.1 Overview

5.2 Description

RISC-V MCU core totally has 16 external interrupt sources. The device used the lowest 8 bits of them. Because MCU's external interrupts priority is fixed, to enhance the flexibility of external interrupt's priority mapping, we implemented 8 configure registers, see registers chapter 5.1.26, to map the following external interrupt source to MCU IRQ lines dynamically. Below table is the default mapping, SW could change the default mapping in according to different interrupt priority requirements.

CPU Interrupt vector	Interrupt Source Name
IRQ[15:8]	reserved
IRQ[7]	GPIO_INT
IRQ[6]	SPI_INT
IRQ[5]	SEC_UART_INT
IRQ[4]	UART_INT
IRQ[3]	SEC_I2C_INT
IRQ[2]	I2C_INT
IRQ[1]	DMA_INT
IRQ[0]	PD_INT

Table 5- 1 Interrupt Vector Table

5.3 Registers

5.3.1 MCU_INT_INDEX_REG

Address 0x4000_0090		MCU Interrupt Index Register		
Field Name	Field Bits	Reset Value	SW Access	Field description
INT7_INDEX	30:28	0x7	RW	MCU interrupt vector mapping index. 3'h0 : PD_INT is connected to MCU core corresponding INT_REQ bit; 3'h1 : DMA_INT is connected to MCU core corresponding INT_REQ bit; 3'h2 : I2C_INT is connected to MCU core corresponding INT_REQ bit; 3'h3 : SEC_I2C_INT is connected to MCU core corresponding INT_REQ bit; 3'h4 : UART0_INT is connected to MCU core corresponding INT_REQ bit; 3'h5 : UART1_INT is connected to MCU core corresponding INT_REQ bit; 3'h6 : SPI_INT is connected to MCU core corresponding INT_REQ bit; 3'h7 : GPIO_INT is connected to MCU core corresponding INT_REQ bit;
INT6_INDEX	26:24	0x6	RW	
INT5_INDEX	22:20	0x5	RW	
INT4_INDEX	18:16	0x4	RW	
INT3_INDEX	14:12	0x3	RW	
INT2_INDEX	10:8	0x2	RW	
INT1_INDEX	6:4	0x1	RW	

6 General DMA

6.1 Overview

The device support two channels of general DMA.
These two general DMA also support the data copy from system memory to CPU peripherals, such as UART, SPI. I2C DMA mode is not supported in the device.

6.2 Description

General DMA is used to do data copy between system memory and FLASH memory. There're configurable registers for source address, destination address and DMA copy data byte counter (see registers chapter 5.1.22 ~ 5.1.25). After these registers are configured, SW could write DMA_BUSY register to '1' to start the DMA data copy, after HW DMA data copy operation is done, DMA HW logic will generate DMA interrupt to indicate it and clear the DMA_BUSY register to '0' automatically by HW.
Here's the hardware program guide for CPU peripherals DMA mode by general DMA.
For UART or SPI's transmit:

1) Write the specific peripheral's TX data register address into the general DMA's destination start address register.
2) Write the data content, which is need to be transmitted, into the system memory address space.
3) Write above system memory's start address into general DMA's source start address register.
4) Set the byte number into the register "dma_byte_cnt" to tell the DMA how many bytes of data need copy to peripheral for transmitting.
5) After above steps, SW need to set the general DMA busy register to start the data copy in batch mode.
6) After all data bytes copy is done. DAM will generate the DMA done interrupt to inform CPU.

For UART or SPI's receive:

7) Write specific peripheral's RX data register address into general DAM's source start address register.
8) Write the DMA buffer space address of system memory into DMA's destination start address register.
9) Write the byte number register to general DMA's "dma_byte_cnt" register for the DMA buffer depth setting.
10) When the peripheral received one byte of data and generate the peripheral interrupt, it will trigger the DMA to copy this received byte of data to DMA buffer located in system memory space automatically.
11) After DMA finish the copy of each one received byte, DMA will generate interrupt to inform CPU/SW to handle the data.

6.3 General DMA Registers

6.3.1 DMA0_CTRL_REG 0

Address 0x4000_0070		DMA Control Register 0		
Field Name	Field Bits	Reset Value	SW Access	Field description
DMA_SRC_START_ADDR	31:0	0x0	RW	DMA source data start address.

6.3.2 DMA0_CTRL_REG 1

Address 0x4000_0074			DMA Control Register 1	
Field Name	Field Bits	Reset Value	SW Access	Field description
DMA_DES_START_ADDR	31:0	0x0	RW	DMA destination data start address.

6.3.3 DMA0_CTRL_REG 2

Address 0x4000_0078			DMA Control Register 1	
Field Name	Field Bits	Reset Value	SW Access	Field description
DMA_BUSY	31	0x0	RW	DMA busy bit, SW set this bit to start DMA data copy, HW clear it when DMA copy is done.
DMA_MODE	30:29	0x0	RW	2'b00: address increment mode. 2'b01: address decent mode 2'b10: source address hold constant mode. 2'b11: destination address hold constant mode.
DMA_BYTE_CNT	23：0	0x0	RW	DMA copy data byte counter, in unit of byte, which is used to indicate how many bytes of data need to be copied.

6.3.4 DMA1_CTRL_REG 0

Address 0x4000_0080			DMA Control Register 0	
Field Name	Field Bits	Reset Value	SW Access	Field description
DMA_SRC_START_ADDR	31:0	0x0	RW	DMA source data start address.

6.3.5 DMA1_CTRL_REG 1

Address 0x4000_0084			DMA Control Register 1	
Field Name	Field Bits	Reset Value	SW Access	Field description
DMA_DES_START_ADDR	31:0	0x0	RW	DMA destination data start address.

6.3.6 DMA1_CTRL_REG 2

Address 0x4000_0088			DMA Control Register 1	
Field Name	Field Bits	Reset Value	SW Access	Field description
DMA_BUSY	31	0x0	RW	DMA busy bit, SW set this bit to start DMA data copy, HW clear it when DMA copy is done.
DMA_MODE	30:29	0x0	RW	2'b00: address increment mode. 2'b01: address decent mode 2'b10: source address hold constant mode. 2'b11: destination address hold constant mode.
DMA_BYTE_CNT	23：0	0x0	RW	DMA copy data byte counter, in unit of byte, which is used to indicate how many bytes of data need to be copied.

6.3.7 DMA_INT_REG

Address 0x4000_008C			DMA Interrupt Register	
Field Name	Field Bits	Reset Value	SW Access	Field description
DMA1_INT_ENA	3	0x0	RW	DMA interrupt enable
DMA1_INT_STATUS	2	0x0	RC	DMA data copy is done interrupt status. SW read clear.

DMA0_INT_ENA	1	0x0	RW	DMA interrupt enable
DMA0_INT_STATUS	0	0x0	RC	DMA data copy is done interrupt status. SW read clear.

7 I2C

7.1 Overview

The I2C subcomponent is the I2C Bus Controller which provides an interface that meets the Philips I2C bus specification and supports all transfer modes from and to the I2C bus.
The I2C bus uses two wires to transfer information between devices connected to the bus: “scl” (serial clock line) and “sda” (serial data line).
The I2C logic handles bytes transfer autonomously. It also keeps track of serial transfers, and a status register (“i2csta”, 2.4.46) reflects the status of the I2C Bus Controller and the I2C bus.
Optionally, the SMBus extension can be implemented.

7.2 Description

The I2C bus uses two wires to transfer information between devices connected to the bus: “scl” (serial clock line) and “sda” (serial data line). The I2C component requires the use of external open-drain buffers since it has only unidirectional ports. The “sda” and “scl” lines referred further are the actual I2C bus signals, while the I2C component is connected to them with the use of open-drains (“sdao” as output and “sdai” as input, “sclo” as output and “scli” as input). Each device connected to the bus is software addressable by a unique address. The I2C is a true multi-master bus including collision detection and arbitration to prevent data corruption if two or more masters simultaneously initiate data transfer. The filtering logic rejects spikes on the bus data line to preserve data integrity.
a) Operating Modes
The I2C component performs 8-bit oriented, bi-directional data transfers up to 100 kbit/s in the standard mode or up to 400 kbit/s in the fast mode and may operate in the following four modes:
b) Master Transmitter Mode:
Serial data output through “sdao” while “sclo” outputs the serial clock.
c) Master Receiver Mode:
Serial data is received via “sdai” while “sclo” outputs the serial clock.
d) Slave Receiver Mode
Serial data and the serial clock are received through “sdai” and “scli”.
e) Slave Transmitter Mode
Serial data is transmitted via “sdao” while the serial clock is input through “scli”.
f) Arbitration and Synchronization Logic
In the master mode, the arbitration logic checks that every transmitted high state (‘1’) on “sdao” actually appears as high state (‘1’) on the I2C bus “sda”. If another device on the bus overrides high and pulls the “sda” line low (‘0’), arbitration is lost and the I2C immediately changes from master transmitter to slave receiver.
The synchronization logic synchronizes the serial clock generator with the clock pulses on the “scli” line from another device.
g) Serial Clock Generator
This programmable clock pulse generator provides the “sclo” clock pulses when the I2C is in the master mode. The clock generator is suppressed when the I2C is in the slave mode.
h) Input Filter
Input signals are synchronized with clock (“clk”), and spikes shorter than three clock periods are filtered out. Each filter consists of three flip-flops. The first one is used to latch the input directly, while the other two form a shift register which is loaded from the first one. When the state of the 2nd and 3rd flip-flop is either “11” or “00”, an internal filtered signal is set or reset, respectively.
i) Address Comparator
The received 7-bit slave address is compared with the I2C component own slave address. The own slave address can be programmed using the “i2caddr” register. Also the first received byte is compared with the general call address (00H), depending on the “gc” bit of “i2caddr” register. If equality is found, the “si” bit of the “i2ccon” register is set and an interrupt is requested.
j) Interrupt Generation
The “si“ flag of the “i2ccon” register is set by hardware when one of 25 out of 26 possible I2C states is

entered. The only state that does not set the “si” is state F8h, which indicates that no relevant state information is available. The “si” flag must be cleared by software. In order to clear the “si” bit, ‘0’ must be written to this bit. Writing a ‘1’ to si bit does not change value of the “si”.

k) Special Function Registers

The microprocessor interfaces to the I2C component via the following four special function registers: “i2ccon” (control register), “i2csta”, “i2cdat” and “i2cadr” (own slave address register).

The “i2cadr” register contains the Own Slave Address of the I2C component, and the “gc” flag which enables the recognition of a general call address.

The “i2ccon” register contains the global I2C enable bit “ens1”. It also provides flags to initiate sending START or STOP conditions to the I2C bus (“sta”, “sto” bits), a flag controlling the ACK bit in I2C transmission after receiving own slave address or general call address or after receiving data either in master or slave mode (“aa” – assert acknowledge flag). Finally the “i2ccon” provides the interrupt request flag “si” which is set by hardware when a change of the main controlling FSM is detected. See the “i2csta” register description for FSM details.

The “i2cdat” register contains a byte to be transmitted through I2C bus or a byte which has just been received through I2C bus. The “i2cdat” register is not shadowed or double buffered so the MCU should only read it when an I2C interrupt occurs.

The “i2csta” register reflects the state of the main FSM of the I2C component. The three least significant bits of this register are always zero. There are 26 possible status codes, which are presented in Table 119 … Table 123. When one of the 25 out of 26 possible I2C FSM states is entered, an interrupt is requested. The only state that does not generate an interrupt is the F8h state.

In the table below, referring to “SLA” means slave address, “R” means R/W bit=1 transferred together with the slave address, “W” means R/W bit=0 transferred together with the slave address.

Status Code	Status of I2C	Application SW Response					Next action taken by I2C Hardware
		To/From I2cdat	To I2con				
			sta	sto	si	aa	
60H	Own SLA+W has been received; ACK has been returned	No action or no action	X X	0 0	0 0	0 1	Data byte will be received and not ACK will be returned Data byte will be received and ACK will be returned
68H	Arbitration lost in SLA+R/W as master; own SLA+W has been received, ACK returned	No action or no action	X X	0 0	0 0	0 1	Data byte will be received and not ACK will be returned Data byte will be received and ACK will be returned
70H	General call address (00H) has been received; ACK has been returned	No action or no action	X X	0 0	0 0	0 1	Data byte will be received and not ACK will be returned Data byte will be received and ACK will be returned
78H	Arbitration lost in SLA+R/W as master; general call address has been received, ACK returned	No action or no action	X X	0 0	0 0	0 1	Data byte will be received and not ACK will be returned Data byte will be received and ACK will be returned
80H	Previously addressed with own SLV address; DATA has been received; ACK returned	Read data byte or read data byte	X X	0 0	0 0	0 1	Data byte will be received and not ACK will be returned Data byte will be received and ACK will be returned

88H	Previously addressed with own SLA; DATA byte has been received; not ACK returned	Read data byte or read data byte or read data byte or read data byte	0 0 1 1	0 0 0 0	0 0 0 0	0 1 0 1	Switched to not addressed SLV mode; no recognition of own SLA or general call address Switched to not addressed SLV mode; own SLA or general call address will be recognized Switched to not addressed SLV mode; no recognition of own SLA or general call address; START condition will be transmitted when the bus becomes free Switched to not addressed SLV mode; own SLA or general call address will be recognized; START condition will be transmitted when the bus becomes free
90H	Previously addressed with general call address; DATA has been received; ACK returned	Read data byte or read data byte	X X	0 0	0 0	0 1	Data byte will be received and not ACK will be returned Data byte will be received and ACK will be returned
98H	Previously addressed with general call address; DATA has been received; not ACK returned	Read data byte or read data byte or read data byte or read data byte	0 0 1 1	0 0 0 0	0 0 0 0	0 1 0 1	Switched to not addressed SLV mode; no recognition of own SLA or general call address Switched to not addressed SLV mode; own SLA or general call address will be recognized Switched to not addressed SLV mode; no recognition of own SLA or general call address; START condition will be transmitted when the bus becomes free Switched to not addressed SLV mode; own SLA or general call address will be recognized; START condition will be transmitted when the bus becomes free
A0H	A STOP condition or repeated START condition has been received while still addressed as SLV/REC or SLV/TRX	No action or no action or no action or no action	0 0 1 1	0 0 0 0	0 0 0 0	0 1 0 1	Switched to not addressed SLV mode; no recognition of own SLA or general call address Switched to not addressed SLV mode; own SLA or general call address will be recognized Switched to not addressed SLV mode; no recognition of own SLA or general call address; START condition will be transmitted when the bus becomes free Switched to not addressed SLV mode; own SLA or general call address will be recognized; START condition will be transmitted when the bus becomes free
A8H	Own SLA+R has been received; ACK has been returned	Load data byte or load data byte	X X	0 0	0 0	0 1	Last data byte will be transmitted and ACK will be received Data byte will be transmitted; ACK will be received
B0H	Arbitration lost in SLA+R/W as master; own SLA+R has been received; ACK has been returned	Load data byte or load data byte	X X	0 0	0 0	0 1	Last data byte will be transmitted and ACK will be received Data byte will be transmitted; ACK will be received
B8H	Data byte has been transmitted; ACK has been received	Load data byte or load data byte	X X	0 0	0 0	0 1	Last data byte will be transmitted and ACK will be received Data byte will be transmitted; ACK will be received
C0H	Data byte has been transmitted; not ACK has been received	No action or no action or no action or no action	0 0 1 1	0 0 0 0	0 0 0 0	0 1 0 1	Switched to not addressed SLV mode; no recognition of own SLA or general call address Switched to not addressed SLV mode; own SLA or general call address will be recognized Switched to not addressed SLV mode; no recognition of own SLA or general call address; START condition will be transmitted when the bus becomes free Switched to not addressed SLV mode; own SLA or general call address will be recognized; START condition will be transmitted when the bus becomes free

C8H	Last data byte has been transmitted; ACK has been received	No action or no action or no action or no action	0 0 1 1	0 0 0 0	0 0 0 0	0 1 0 1	Switched to not addressed SLV mode; no recognition of own SLA or general call address Switched to not addressed SLV mode; own SLA or general call address will be recognized Switched to not addressed SLV mode; no recognition of own SLA or general call address; START condition will be transmitted when the bus becomes free Switched to not addressed SLV mode; own SLA or general call address will be recognized; START condition will be transmitted when the bus becomes free
38H	Arbitration lost	No action or no action	0 1	0 0	0 0	X X	I2C bus will be released; A start condition will be transmitted when the bus becomes free (enter to a master mode)
F8H	No relevant state information available; si=0	No action	No action				Wait or proceed current transfer
00H	Bus error during MST or selected slave modes	No action	0	1	0	X	Only the internal hardware is affected in the MST or addressed SLV modes. In all cases, the bus is released and I2C is switched to the not addressed SLV mode. Sto flag is reset.

Table 7- 1 I2C Status Code Table

l) System Management Bus Extension
The optional System Management Bus (SMBus) protocol hardware implementation features:
- timeout detection (clock low timeout measurement)
- Tmext timeout detection (cumulative stretch clock cycles within one byte)
- Tsext timeout detection (cumulative stretch clock cycles between start and stop condition)

The SMBus requires two additional registers: "smb_sel" and "smb_dst". They both form a read/write access port to the timeout registers.
The "smb_sel" is used to enable the SMBus feature (bit 7), and for selecting the SMBus internal timeout register being accessed through the other register as a data port.
The "smb_dst" register is used to read or write internal SMBus timeout register, which is selected by the "smb_sel".
When the SMBus feature is enabled, the 3 least significant bits of the I2C status register hold the information about tiomeouts.

SMBus error status I2C2STA	Description
xxxxx000	No timeout errors.
Xxxxxxx1	Tout timeout error.
Xxxxxx1x	Tsext timeout error.
xxxxx1xx	Tmext timeout error.

7.3 I2C Registers

7.3.1 I2C_DATA_REG

Address 0x4000_0700		I2C_DATA_REG		
Field Name	Field Bits	Reset Value	SW Access	Field description
RESV	31:8	0x0	RO	Reserved
I2C_DAT	7:0	0x0	RW	I2C received or transmit data register

7.3.2 I2C_ADR_REG

Address 0x4000_0704		I2C_ADR_REG		
Field Name	Field Bits	Reset Value	SW Access	Field description
RESV	31:8	0x0	RO	Reserved

SLV_ADDR	7:1	0x0	RW	I2C slave address.
GC	0	0x0	RW	If set this bit, the general call address is recognized, otherwise it is ignored.

7.3.3 I2C_CTRL_REG

Address 0x4000_0708				I2C_CTRL_REG
Field Name	Field Bits	Reset Value	SW Access	Field description
RESV	31:7	0x0	RO	Reserved
I2C_ENA	6	0x0	RW	I2C enable bit
STA	5	0x0	RW	Start flag, when sta=1, I2C check the bus status, if free, the start signal will be generated.
STO	4	0x0	RW	Stop flag, when sto=1 and I2C is maste mode, the stop signal will be generated.
SI	3	0x0	RW	Serial interrupt flag, set by HW, clear by SW write 0 to this bit.
AA	2	0x0	RW	When SW set aa=1, an “acknowledge” will be returned, otherwise an “NAK” will be returned.
RESV	1:0	0x0	RO	Reserved

7.3.4 I2C_FREQ_REG

Address 0x4000_070C				I2C_FREQ_REG
Field Name	Field Bits	Reset Value	SW Access	Field description
RESV	31:8	0x0	RO	Reserved
CNT2_DIV	7:4	0x0	RW	Counter 2 divider.
CNT1_DIV	3:0	0x0	RW	Counter 1 divider.

7.3.5 I2C_STA_REG

Address 0x4000_0710				I2C_STA_REG
Field Name	Field Bits	Reset Value	SW Access	Field description
RESV	31:8	0x0	RO	Reserved
STATUS	7:3	0x1F	RW	I2C status code.
RESV	2:0	0x0	RO	Reserved

7.3.6 I2C_SMB_SEL_REG

Address 0x4000_0714				I2C_SMB_SEL_REG
Field Name	Field Bits	Reset Value	SW Access	Field description
RESV	31:8	0x0	RO	Reserved
SMB_EN	7	0x0	RW	SMBUS extension enable.
RESV	6:4	0x0	RO	Reserved
SMB_SEL	2:0	0x0	RW	select one of the timeout register for read/write access through the SMB_DST register

7.3.7 I2C_SMB_DST_REG

Address 0x4000_0718				I2C_SMB_DST_REG
Field Name	Field Bits	Reset Value	SW Access	Field description
RESV	31:8	0x0	RO	Reserved
SMB_DST				The optional System Management Bus (SMBus) protocol hardware implementation features:

	7:0	0x35	RW	- timeout detection (clock low timeout measurement) - Tmext timeout detection (cumulative stretch clock cycles within one byte) -Tsext timeout detection (cumulative stretch clock cycles between start and stop condition) The SMBus requires two additional registers: smb_sel and smb_dst. They both form a read/write access port to the timeout registers.

7.4 Secondary I2C Registers

7.4.1 SECI2C_DATA_REG

Address 0x4000_0800			SECI2C_DATA_REG	
Field Name	Field Bits	Reset Value	SW Access	Field description
RESV	31:8	0x0	RO	Reserved
I2C_DAT	7:0	0x0	RW	I2C received or transmit data register

7.4.2 SECI2C_ADR_REG

Address 0x4000_0804			SECI2C_ADR_REG	
Field Name	Field Bits	Reset Value	SW Access	Field description
RESV	31:8	0x0	RO	Reserved
SLV_ADDR	7:1	0x0	RW	I2C slave address.
GC	0	0x0	RW	If set this bit, the general call address is recognized, otherwise it is ignored.

7.4.3 SECI2C_CTRL_REG

Address 0x4000_0808			SECI2C_CTRL_REG	
Field Name	Field Bits	Reset Value	SW Access	Field description
RESV	31:7	0x0	RO	Reserved
I2C_ENA	6	0x0	RW	I2C enable bit
STA	5	0x0	RW	Start flag, when sta=1, I2C check the bus status, if free, the start signal will be generated.
STO	4	0x0	RW	Stop flag, when sto=1 and I2C is maste mode, the stop signal will be generated.
SI	3	0x0	RW	Serial interrupt flag, set by HW, clear by SW write 0 to this bit.
AA	2	0x0	RW	When SW set aa=1, an "acknowledge" will be returned, otherwise an "NAK" will be returned.
RESV	1:0	0x0	RO	Reserved

7.4.4 SECI2C_FREQ_REG

Address 0x4000_080C			SECI2C_FREQ_REG	
Field Name	Field Bits	Reset Value	SW Access	Field description
RESV	31:8	0x0	RO	Reserved
CNT2_DIV	7:4	0x0	RW	Counter 2 divider.
CNT1_DIV	3:0	0x0	RW	Counter 1 divider.

7.4.5 SECI2C_STA_REG

Address 0x4000_0810				SECI2C_STA_REG
Field Name	Field Bits	Reset Value	SW Access	Field description
RESV	31:8	0x0	RO	Reserved
STATUS	7:3	0x1F	RW	I2C status code.
RESV	2:0	0x0	RO	Reserved

7.4.6 SECI2C_SMB_SEL_REG

Address 0x4000_0814				SECI2C_SMB_SEL_REG
Field Name	Field Bits	Reset Value	SW Access	Field description
RESV	31:8	0x0	RO	Reserved
SMB_EN	7	0x0	RW	SMBUS extension enable.
RESV	6:4	0x0	RO	Reserved
SMB_SEL	2:0	0x0	RW	select one of the timeout register for read/write access through the SMB_DST register

7.4.7 SECI2C_SMB_DST_REG

Address 0x4000_0818				SECI2C_SMB_DST_REG
Field Name	Field Bits	Reset Value	SW Access	Field description
RESV	31:8	0x0	RO	Reserved
SMB_DST	7:0	0x35	RW	The optional System Management Bus (SMBus) protocol hardware implementation features: - timeout detection (clock low timeout measurement) - Tmext timeout detection (cumulative stretch clock cycles within one byte) -Tsext timeout detection (cumulative stretch clock cycles between start and stop condition) The SMBus requires two additional registers: smb_sel and smb_dst. They both form a read/write access port to the timeout registers.

8 SPI_MS

8.1 Overview

The SPI_MS module allows full-duplex, synchronous, serial communication between the MCU and peripherals, including other MCUs. It is obvious that the MCU and any peripherals must include SPI module. The module may be programmed to work as master or as slave device.
The SPI_MS provides the following features:
• Full duplex mode
• Three wire synchronous transfers
• Master or Slave mode
• Configurable SPI Master baud rates
• Slave Clock rate up to Fclkper/8 (Fclkper/4 with single Flip-Flop of CDC synchronization)
• Serial clock with programmable polarity and phase
• Master Mode fault error flag with MCU interrupt capability
• Write collision flag protection
• 8-bit data transmitted Most Significant Bit (MSB) first, Least Significant Bit (LSB) last
• 4-bit Slave Select Output port to control external slave devices
• Special Function Registers interface to the host CPU
• No bi-directional ports; standard SPI pins to be externally connected to 3-state buffers

8.2 Description

The component communicates with host microprocessor through CSR interface and INT interface (i.e. “intspi”). Communication with other devices, which include the SPI module, is realized through TR interface (i.e. “mosi” group, “miso” group, “sck” group).
Functional blocks of SPI_MS module:
• INT – interrupt control block
• SFR – Special Functional Register block
• TR – Transmit block
The SFR block controls the write/read operations on SFR registers of SPI_MS module. It contains the following:
• address decoder,
• Special Function Registers – SPCON, SPSTA, SPDAT,
• output multiplexor.

The TR block controls the SPI transmission process. It is composed of the following:
• the Finite State Machine which plays a key role in operation of the SPI_MS module; it controls the Master or Slave functionality
• system clock counter/divider, which is used to generate the SPI Master clock “scko”; the Master clock is selected from one of seven clock rates i.e. the system clock divided by 2, 4, 8, 16, 32, 64 or 128
• rising and falling edge detector on “scki” input pin; it is used only in Slave mode
• transmission end detector
• level and falling edge detector on “ssn” input pin
• data shift register.

The INT block generates interrupt request upon “spif” and “modf” flags.

8.3 SPI Registers

8.3.1 SPSTA_REG

Address 0x4000_0A00	SPSTA_REG

Field Name	Field Bits	Reset Value	SW Access	Field description
RESV	30:8	0x0	RO	Reserved
SPIF	7	0x0	RO	Serial data transfer flag, set by hardware upon data transfer completion. Cleared by hardware when data transfer is in progress and Also could be cleared by reading SPSTA_REG.
WCOL	6	0x0	RO	Write collision Flag, set by hardware upon write collision to "SPDAT_REG". Cleared by hardware upon data Transfer completion when no collision has occurred. Can also be cleared by an access to "SPSTA_REG" And an access to "SPSTA_REG".
SSER	5	0x0	RO	Synchronous Serial Slave Error flag. Set by hardware when "ssn" input is deasserted before the end of receive sequence. Cleared by disabling the SPI module (clearing "spen" bit in "SPCON_REG".
MODF	4	0x0	RO	Mode fault flag. Set by hardware when the "ssn" pin level is in conflict with actual mode of SPI controller (configured as Master while externally selected as slave). Cleared by hardware when "ssn" pin is at appropriate level or Cleared by software by reading the SPSTA_REG.
RESV	3:0	0x0	RO	Reserved

8.3.2 SPCON_REG

Address 0x4000_0A04		SPCON_REG		
Field Name	Field Bits	Reset Value	SW Access	Field description
RESV	30:12	0x0	RO	Reserved
SPEN	11:8	0x0	RW	SPI interface enable
SPFRM	7	0x0	RW	1'b1 : 16 bit SPI frame mode; 1'b0 : 8 bit SPI frame mode.
SPR2	6	0x0	RW	together with SPR1, SPR0 defines the clock rate in master mode
SSDIS	5	0x0	RW	when cleared enables the "ssn" input in both master and slave modes. When set disables the "ssn" input In both master and slave modes. In slave mode, this bit has no effect if "cpha=0". When "ssdis" is set, no "modf" interrupt request is generated.
MSTR	4	0x1	RW	1'b0 : SPI slave mode; 1'b1 : SPI master mode
CPOL	3	0x0	RW	clock polarity, when cleared the "sck" is set to 0 in idle state. When set the "sck" is set to 1 in idle state.
CPHA	2	0x1	RW	clock phase: when cleared, data is sampled when the "sck" leaves the idle state (see CPOL). When set, data is sampled when the "sck" returns to idle state (see CPOL).
SPR1	1	0x0	RW	SPR2,SPR1,SPR0 together defines the sck rate in master mode : 3'b000 : Fclk/2 3'b001 : Fclk/4 3'b010 : Fclk/8 3'b011 : Fclk/16 3'b100 : Fclk/32 3'b101 : Fclk/64 3'b110 : Fclk/128 3'b111 : the master clock is not generated.
SPR0	0	0x0	RW	

8.3.3 SPDAT_REG

Address 0x4000_0A08		SPDAT_REG		
Field Name	Field Bits	Reset Value	SW Access	Field description
RESV	30:16	0x0	RO	Reserved
SPDAT	15:0	0x0	RW	When writing to the SPDAT, TX data is placed directly into the shift register. When reading the SPDAT returns the RX data value located in the receive buffer, not the shift register.

8.3.4 SPSSN_REG

Address 0x4000_0A0C		SPSSN_REG		
Field Name	Field Bits	Reset Value	SW Access	Field description
RESV	30:4	0x0	RO	Reserved
SPSSN	3:0	0xF	RW	The SPSSN is a read/write register used to control the "spssn[3:0]" output bus of SPI.

9 UART

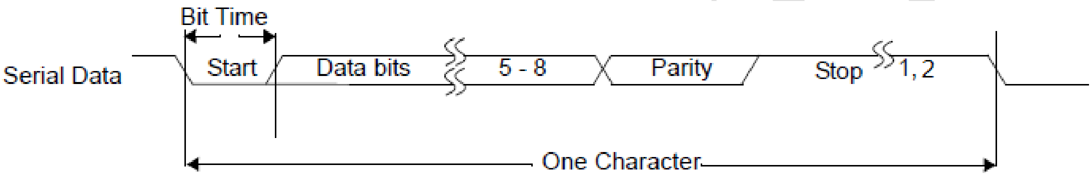
9.1 Overview

UART is a general serial full-duplex communication data bus.
Features：

- APB bus accessed registers；
- Full duplex independent transmit and receive channels；
- Software detected channel status；
- Configurable parity check interrupt, overwrite valid data interrupt, frame error interrupt generation;
- configurable baud rate；
- The maximum transfer rate is 1/16 of the system clock.

9.2 Description

The UART transmits data through the serial transmission bus TXD, and the RXD receives the data. The data stream format is as shown below:

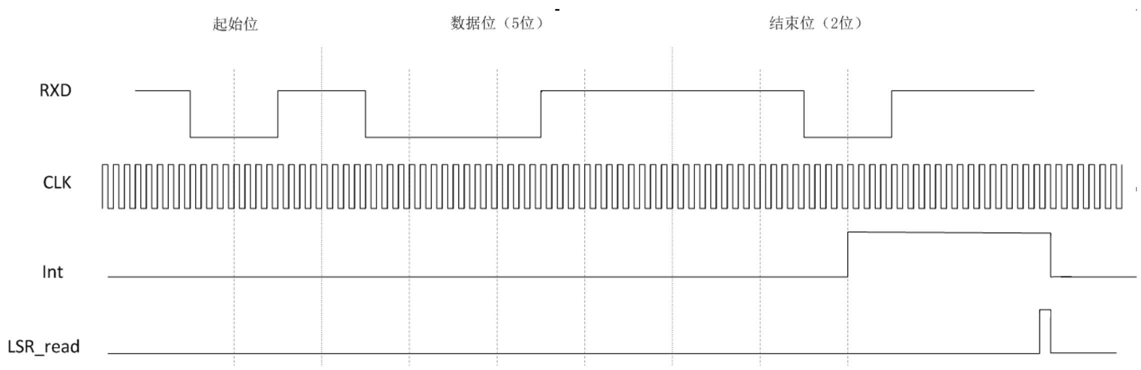


Each frame of data has a start bit, 5 to 8 bits of data bits, and optionally a parity bit and a 1 to 2 bit stop bit, where the start bit is low and the stop bit is high.

The data transmission is synchronized by the baud rate. The software first determines the baud rate. The UART automatically generates the baud rate clock.
The UART sets the corresponding baud rate clock by the software setting DLL and DLH (divided clock register). The calculation formula is: system clock / (baud rate x4). The DLL register stores the status value of the divided clock register. The DLH stores the high-order value of the divided clock. Before setting the DLL and DLH values, the DLAB bit of the LCR transfer controller needs to be set.

UART interrupt timing:
The UART supports transmission status interrupts that cover valid data, parity errors, and framing errors. It also supports accepting data valid interrupts and transmit holding register empty interrupts, as well as supporting UART busy interrupts. Software programmers can enable these interrupts by writing values to the IER register. The specific interrupt descriptions and generations are shown in the register description table below.

The following picture shows the occurrence of a frame error interrupt, as shown in the figure: the length of the data bit is 5 bits, the length of the stop bit is 2 bits, and the low level is detected at the second stop bit, so a frame error interrupt is generated. This interrupt is cleared after the software reads the LSR register.



It should be noted that all received interrupts, that is, the receive status interrupt and the receive data valid interrupt are generated after the complete reception of one frame of data, and the transmit hold register empty interrupt is generated immediately when the transfer holding register is empty.

9.3 UART Registers

9.3.1 UART_PBR_REG

Address 0x4000_0500			UART_PBR_REG	
Field Name	Field Bits	Reset Value	SW Access	Field description
RESV	31:8	0x0	RO	Reserved
RXD	7:0	0x0	RO	Receive Bufffer Data register, content is only valid when LSR[DR] = 1'b1 and access select by LCR[7] = 1'b0.

9.3.2 UART_THR_REG

Address 0x4000_0504			UART_THR_REG	
Field Name	Field Bits	Reset Value	SW Access	Field description
RESV	31:8	0x0	RO	Reserved
THR	7:0	0x0	WO	Transmit Data Register, content could be written only when LSR[THRE]=1'b1 and access select by LCR[7] = 1'b0.

9.3.3 UART_DLL_REG

Address 0x4000_0508			UART_DLL_REG	
Field Name	Field Bits	Reset Value	SW Access	Field description
RESV	31:8	0x0	RO	Reserved
DLL	7:0	0x0	RW	Frequency divider LSB register, access select by LCR[7] = 1'b1.

9.3.4 UART_DLH_REG

Address 0x4000_050C			UART_DLH_REG	
Field Name	Field Bits	Reset Value	SW Access	Field description
RESV	31:8	0x0	RO	Reserved
DLH	7:0	0x0	RW	Frequency divider MSB register, access select by LCR[7] = 1'b1.

9.3.5 UART_IER_REG

Address 0x4000_0510		UART_IER_REG		
Field Name	Field Bits	Reset Value	SW Access	Field description
RESV	31:3	0x0	RO	Reserved
ELSI	2	0x0	RW	Enable receive status interrupt, access select by LCR[7] = 1'b0.
ETBEI	1	0x0	RW	Enable transmit data register empty interrupt
ERBFI	0	0x0	RW	Enable receive data active interrupt

9.3.6 UART_IIR_REG

Address 0x4000_0514		UART_IIR_REG		
Field Name	Field Bits	Reset Value	SW Access	Field description
RESV	31:4	0x0	RO	Reserved
IID	3:0	0x1	RO	4'b0001 : No interrupt 4'b0010 : THR empty interrupt 4'b0100 : RXD active interrupt 4'b0110 : Receive Status interrupt 4'b0111 : UART is busy interrupt

9.3.7 UART_LCR_REG

Address 0x4000_0518		UART_LCR_REG		
Field Name	Field Bits	Reset Value	SW Access	Field description
RESV	31:8	0x0	RO	Reserved
DLAB	7	0x0	RW	Register access selection.
FCTL_EN	6	0x0	RW	UART with flow control signals "RTS/CTS" enable
RESV	5	0x0	RO	Reserved
EPS	4	0x0	RW	Parity check select, 1'b1 : even parity check; 1'b0 : odd parity check.
PEN	3	0x0	RW	Parity check enable
STOP	2	0x0	RW	Stop bit number, 1'b0 : one stop bit, 1'b1 : two stop bits.
DLS	1:0	0x0	RW	Data length select: 2'b00 : 5 bits 2'b01 : 6 bits 2'b10 : 7 bits 2'b11 : 8 bits

9.3.8 UART_LSR_REG

Address 0x4000_051C		UART_LSR_REG		
Field Name	Field Bits	Reset Value	SW Access	Field description
RESV	31:7	0x0	RO	Reserved
TEMT	6	0x0	RW	1'b1 : Both THR and transmit shift register is empty
THRE	5	0x0	RW	1'b1 : THR is empty
RESV	4	0x0	RO	Reserved
FE	3	0x0	RW	1'b1 : frame error
PE	2	0x0	RW	1'b1 : parity check error
OE	1	0x0	RW	1'b1 : active data is override error
DR	0	0x0	RW	1'b1 : receive data is active

9.3.9 UART_USR_REG

Address 0x4000_0520			UART_LSR_REG	
Field Name	Field Bits	Reset Value	SW Access	Field description
RESV	31:1	0x0	RO	Reserved
BUSY	0	0x0	RO	1'b1 : UART is busy

9.4 Secondary UART Registers

9.4.1 SEC_UART_PBR_REG

Address 0x4000_0600			SEC_UART_PBR_REG	
Field Name	Field Bits	Reset Value	SW Access	Field description
RESV	31:8	0x0	RO	Reserved
RXD	7:0	0x0	RO	Receive Bufffer Data register, content is only valid when LSR[DR] = 1'b1 and access select by LCR[7] = 1'b0.

9.4.2 SEC_UART_THR_REG

Address 0x4000_0604			SEC_UART_THR_REG	
Field Name	Field Bits	Reset Value	SW Access	Field description
RESV	31:8	0x0	RO	Reserved
THR	7:0	0x0	WO	Transmit Data Register, content could be written only when LSR[THRE]=1'b1 and access select by LCR[7] = 1'b0.

9.4.3 SEC_UART_DLL_REG

Address 0x4000_0608			SEC_UART_DLL_REG	
Field Name	Field Bits	Reset Value	SW Access	Field description
RESV	31:8	0x0	RO	Reserved
DLL	7:0	0x0	RW	Frequency divider LSB register, access select by LCR[7] = 1'b1.

9.4.4 SEC_UART_DLH_REG

Address 0x4000_060C			SEC_UART_DLH_REG	
Field Name	Field Bits	Reset Value	SW Access	Field description
RESV	31:8	0x0	RO	Reserved
DLH	7:0	0x0	RW	Frequency divider MSB register, access select by LCR[7] = 1'b1.

9.4.5 SEC_UART_IER_REG

Address 0x4000_0610			SEC_UART_IER_REG	
Field Name	Field Bits	Reset Value	SW Access	Field description
RESV	31:3	0x0	RO	Reserved
ELSI	2	0x0	RW	Enable receive status interrupt, access select by LCR[7] = 1'b0.
ETBEI	1	0x0	RW	Enable transmit data register empty interrupt
ERBFI	0	0x0	RW	Enable receive data active interrupt

9.4.6 SEC_UART_IIR_REG

Address 0x4000_0614		SEC_UART_IIR_REG		
Field Name	Field Bits	Reset Value	SW Access	Field description
RESV	31:4	0x0	RO	Reserved
IID	3:0	0x1	RO	4'b0001 : No interrupt 4'b0010 : THR empty interrupt 4'b0100 : RXD active interrupt 4'b0110 : Receive Status interrupt 4'b0111 : UART is busy interrupt

9.4.7 SEC_UART_LCR_REG

Address 0x4000_0618		SEC_UART_LCR_REG		
Field Name	Field Bits	Reset Value	SW Access	Field description
RESV	31:8	0x0	RO	Reserved
DLAB	7	0x0	RW	Register access selection.
RESV	6:5	0x0	RO	Reserved
EPS	4	0x0	RW	Parity check select, 1'b1 : even parity check; 1'b0 : odd parity check.
PEN	3	0x0	RW	Parity check enable
STOP	2	0x0	RW	Stop bit number, 1'b0 : one stop bit, 1'b1 : two stop bits.
DLS	1:0	0x0	RW	Data length select: 2'b00 : 5 bits 2'b01 : 6 bits 2'b10 : 7 bits 2'b11 : 8 bits

9.4.8 SEC_UART_LSR_REG

Address 0x4000_061C		SEC_UART_LSR_REG		
Field Name	Field Bits	Reset Value	SW Access	Field description
RESV	31:7	0x0	RO	Reserved
TEMT	6	0x0	RW	1'b1 : Both THR and transmit shift register is empty
THRE	5	0x0	RW	1'b1 : THR is empty
RESV	4	0x0	RO	Reserved
FE	3	0x0	RW	1'b1 : frame error
PE	2	0x0	RW	1'b1 : parity check error
OE	1	0x0	RW	1'b1 : active data is override error
DR	0	0x0	RW	1'b1 : receive data is active

9.4.9 SEC_UART_USR_REG

Address 0x4000_0620		SEC_UART_LSR_REG		
Field Name	Field Bits	Reset Value	SW Access	Field description
RESV	31:1	0x0	RO	Reserved
BUSY	0	0x0	RO	1'b1 : UART is busy

10 DP/DM

10.1 Overview

The device contains two pairs of DP/DM pins for charging detection to detect conventional battery chargers conforming to BC1.2 and QC, AFC etc.

10.2 Description

10.3 DP/DM Registers

10.3.1 SW_CTRL_REG

Address 0x4000_00CC		SW_CTRL_REG		
Field Name	Field Bits	Reset Value	SW Access	Field description
DP1SW500	7	0x0	RW	1: enable 500 Kohm pull-down resister on DP1
DP1SW20	6	0x0	RW	1: enable 20 Kohm pull-down resister on DP1
DM1SW20	5	0x0	RW	1: enable 20 Kohm pull-down resister on DM1
DP0SW500	4	0x0	RW	1: enable 500 Kohm pull-down resister on DP0
DP0SW20	3	0x0	RW	1: enable 20 Kohm pull-down resister on DP0
DM0SW20	2	0x0	RW	1: enable 20 Kohm pull-down resister on DM0
SW1R20	1	0x0	RW	short DP0 and DM0
SW0R20	0	0x0	RW	short DP1 and DM1

11 GPIO

11.1 Overview

General Purpose I/O features:

- APB accessible configure registers.
- 32 independent configurable I/O for each port.
- Each I/O has individual data and direction control registers.
- Configurable interrupt mode.
- Fast IO is supported.

11.2 Description

For detailed description of each GPIO control register, see 5.2.7 and 5.2.8.

11.3 GPIO Registers

11.3.1 GPIO Enable Register

Address 0x4000_0048		GPIO Enable Register		
Field Name	Field Bits	Reset Value	SW Access	Field description
GPIO_EN	31:0	0xFF87FE7F	RW	GPIO enable, 1: GPIO mode is selected, 0: function mode is selected, pin work as UART or I2C and so on.

11.3.2 GPIO_DR_REG

Address 0x4000_0B00		GPIO_DR_REG		
Field Name	Field Bits	Reset Value	SW Access	Field description
P0_DR	31:0	0x0	RW	Port 0 output data register, SW could write this register to decide GPIO's output values.

11.3.3 GPIO_DDR_REG

Address 0x4000_0B04		GPIO_DDR_REG		
Field Name	Field Bits	Reset Value	SW Access	Field description
P0_DDR	31:0	0x0	RW	Port 0 direction register, 0 : input; 1 : output.

11.3.4 GPIO_DSR_REG

Address 0x4000_0B08		GPIO_DSR_REG		
Field Name	Field Bits	Reset Value	SW Access	Field description
P0_DSR	31:0	0x0	RW	Port 0 data source register, these register's value is fixed as 0, which indicate all P0's data value is decide by SW.

11.3.5 GPIO_INT_EN_REG

Address 0x4000_0B0C		GPIO_INT_EN_REG		
Field Name	Field Bits	Reset Value	SW Access	Field description

P0_INT_EN	31:0	0x0	RW	Port 0 interrupts enable bits.

11.3.6 GPIO_INT_MASK_REG

Address 0x4000_0B10		GPIO_INT_MASK_REG		
Field Name	Field Bits	Reset Value	SW Access	Field description
P0_INT_MASK	31:0	0x0	RW	Port 0 interrupts mask bits.

11.3.7 GPIO_INT_TYPE_REG

Address 0x4000_0B14		GPIO_INT_TYPE_REG		
Field Name	Field Bits	Reset Value	SW Access	Field description
P0_INT_TYPE	31:0	0x0	RW	Port 0 interrupts trigger type control bits, 0: level trigger interrupt; 1: edge trigger interrupt.

11.3.8 GPIO_INT_POLAR_REG

Address 0x4000_0B18		GPIO_INT_POLAR_REG		
Field Name	Field Bits	Reset Value	SW Access	Field description
P0_INT_POLAR	31:0	0x0	RW	Port 0 interrupts trigger polarity control bits, 0: low level or falling edge trigger interrupt; 1: high level or rising edge trigger interrupt.

11.3.9 GPIO_INT_STATUS_REG

Address 0x4000_0B1C		GPIO_INT_STATUS_REG		
Field Name	Field Bits	Reset Value	SW Access	Field description
P0_INT_STATUS	31:0	0x0	RW	Port 0 masked interrupt status registers.

11.3.10 GPIO_RAW_INT_STATUS_REG

Address 0x4000_0B20		GPIO_RAW_INT_STATUS_REG		
Field Name	Field Bits	Reset Value	SW Access	Field description
P0_RAW_INT_STATUS	31:0	0x0	RW	Port 0 raw interrupt status registers.

11.3.11 GPIO_INT_CLR_REG

Address 0x4000_0B24		GPIO_INT_CLR_REG		
Field Name	Field Bits	Reset Value	SW Access	Field description
P0_INT_CLR	31:0	0x0	RW	Port 0 interrupt write clear registers, SW write this register bits to clear corresponding interrupt status bit. Write only, could not read.

11.3.12 GPIO_IDR_REG

Address 0x4000_0B28		GPIO_IDR_REG		
Field Name	Field Bits	Reset Value	SW Access	Field description
P0_IDR	31:0	0x0	RO	Port 0 input data register, SW could read this register to know GPIO's input values.

11.3.13 GPIO_INT_SYNC_REG

Address 0x4000_0B2C			GPIO_INT_SYNC_REG	
Field Name	Field Bits	Reset Value	SW Access	Field description
P0_INT_SYNC	31:0	0x0	RW	Registers to control Port 0 interrupts generate sync with clock or not.

11.3.14 GPIO_BIT_MASK_REG

Address 0x4000_0B30			GPIO_BIT_MASK_REG	
Field Name	Field Bits	Reset Value	SW Access	Field description
P0_BIT_MASK	31:0	0x0	RW	When GPIO direction is output, configure the mask bit could mask the write for P0_DR register by bit.

11.3.15 GPIO_PD_REG

Address 0x4000_0B34			GPIO_PD_REG	
Field Name	Field Bits	Reset Value	SW Access	Field description
PD	31:0	0x7	RW	GPIO Pull-Dn resistor Enable

11.3.16 GPIO_PU_REG

Address 0x4000_0B38			GPIO_PU_REG	
Field Name	Field Bits	Reset Value	SW Access	Field description
PU	31:0	0x0	RW	GPIO Pull-Up resistor Enable

11.3.17 GPIO_ODRV0_REG

Address 0x4000_0B3C			GPIO_ODRV0_REG	
Field Name	Field Bits	Reset Value	SW Access	Field description
ODRV0	31:0	0x0	RW	GPIO PAD output driver control, ODRV[1:0]: 2'b00 : Normal mode; 2'b01 : NMOS open-drain w/o internal pull-up; 2'b10 : NMOS open-drain with 10 KOhm pull-up; 2'b11 : NMOS open-drain with 1 Kohm pull-up.

11.3.18 GPIO_ODRV1_REG

Address 0x4000_0B40			GPIO_ ODRV1_REG	
Field Name	Field Bits	Reset Value	SW Access	Field description
ODRV1	31:0	0x0	RW	GPIO PAD output driver control, ODRV[1:0]: 2'b00 : Normal mode; 2'b01 : NMOS open-drain w/o internal pull-up; 2'b10 : NMOS open-drain with 10 KOhm pull-up; 2'b11 : NMOS open-drain with 1 Kohm pull-up.

11.3.19 GPIO_SEN_REG

Address 0x4000_0B44			GPIO_SEN_REG	
Field Name	Field Bits	Reset Value	SW Access	Field description
SEN	31:0	0x0	RW	GPIO PAD input schmiter buffer enable

11.3.20 GPIO_ES0_REG

Address 0x4000_0B48		GPIO_ES0_REG		
Field Name	Field Bits	Reset Value	SW Access	Field description
ES0	31:0	0x0	RW	GPIO PAD output driver control, ES[1:0]: 2'b00 : 4 mA; 2'b01 : 6 mA; 2'b10 : 8 mA; 2'b11 : 10 mA.

11.3.21 GPIO_ES1_REG

Address 0x4000_0B4C		GPIO_ES1_REG		
Field Name	Field Bits	Reset Value	SW Access	Field description
ES1	31:0	0x0	RW	GPIO PAD output driver control, ES[1:0]: 2'b00 : 4 mA; 2'b01 : 6 mA; 2'b10 : 8 mA; 2'b11 : 10 mA.

12 PWM

12.1 Overview

Total 6 channels PWM are implemented in device, PWM working clock frequency is 66 MHz.

12.2 Description

For detailed registers pls. see 5.2.5 section.

12.3 PWM Registers

12.3.1 PWM_EN_REG

Address 0x4000_0900		PWM_EN_REG		
Field Name	Field Bits	Reset Value	SW Access	Field description
RESV	31:6	0x0	RO	Reserved
PWM_EN	5:0	0x0	RW	PWM channels enable, high active

12.3.2 PWM0_CFG_REG_0

Address 0x4000_0904		PWM0_CFG_REG_0		
Field Name	Field Bits	Reset Value	SW Access	Field description
PWM_INIT_VAL	31	0x0	RW	PWM waveform initial value, 1'b1 : PWM waveform start as high; 1'b0 : PWM waveform start as low.
RESV	30:16	0x0	RO	Reserved
PWM_PER	15:0	0x0	RW	PWM waveform period.

12.3.3 PWM0_CFG_REG_1

Address 0x4000_0908		PWM0_CFG_REG_1		
Field Name	Field Bits	Reset Value	SW Access	Field description
PWM_CMP_1	31:16	0x0	RW	PWM counter compare point 1, when counter equal to this compare point value, the output is toggled.
PWM_CMP_2	15:0	0x0	RW	PWM counter compare point 2, when counter equal to this compare point value, the output is toggled.

12.3.4 PWM1_CFG_REG_0

Address 0x4000_090C		PWM1_CFG_REG_0		
Field Name	Field Bits	Reset Value	SW Access	Field description
PWM_INIT_VAL	31	0x0	RW	PWM waveform initial value, 1'b1 : PWM waveform start as high; 1'b0 : PWM waveform start as low.
RESV	30:16	0x0	RO	Reserved
PWM_PER	15:0	0x0	RW	PWM waveform period.

12.3.5 PWM1_CFG_REG_1

Address 0x4000_0910		PWM1_CFG_REG_1		
Field Name	Field Bits	Reset Value	SW Access	Field description
PWM_CMP_1	31:16	0x0	RW	PWM counter compare point 1, when counter equal to this compare point value, the output is toggled.
PWM_CMP_2	15:0	0x0	RW	PWM counter compare point 2, when counter equal to this compare point value, the output is toggled.

12.3.6 PWM2_CFG_REG_0

Address 0x4000_0914		PWM2_CFG_REG_0		
Field Name	Field Bits	Reset Value	SW Access	Field description
PWM_INIT_VAL	31	0x0	RW	PWM waveform initial value, 1'b1 : PWM waveform start as high; 1'b0 : PWM waveform start as low.
RESV	30:16	0x0	RO	Reserved
PWM_PER	15:0	0x0	RW	PWM waveform period.

12.3.7 PWM2_CFG_REG_1

Address 0x4000_0918		PWM2_CFG_REG_1		
Field Name	Field Bits	Reset Value	SW Access	Field description
PWM_CMP_1	31:16	0x0	RW	PWM counter compare point 1, when counter equal to this compare point value, the output is toggled.
PWM_CMP_2	15:0	0x0	RW	PWM counter compare point 2, when counter equal to this compare point value, the output is toggled.

12.3.8 PWM3_CFG_REG_0

Address 0x4000_091C		PWM3_CFG_REG_0		
Field Name	Field Bits	Reset Value	SW Access	Field description
PWM_INIT_VAL	31	0x0	RW	PWM waveform initial value, 1'b1 : PWM waveform start as high; 1'b0 : PWM waveform start as low.
RESV	30:16	0x0	RO	Reserved
PWM_PER	15:0	0x0	RW	PWM waveform period.

12.3.9 PWM3_CFG_REG_1

Address 0x4000_0920		PWM3_CFG_REG_1		
Field Name	Field Bits	Reset Value	SW Access	Field description
PWM_CMP_1	31:16	0x0	RW	PWM counter compare point 1, when counter equal to this compare point value, the output is toggled.
PWM_CMP_2	15:0	0x0	RW	PWM counter compare point 2, when counter equal to this compare point value, the output is toggled.

12.3.10 PWM4_CFG_REG_0

Address 0x4000_0924		PWM4_CFG_REG_0		
Field Name	Field Bits	Reset Value	SW Access	Field description
PWM_INIT_VAL	31	0x0	RW	PWM waveform initial value, 1'b1 : PWM waveform start as high; 1'b0 : PWM waveform start as low.
RESV	30:16	0x0	RO	Reserved
PWM_PER	15:0	0x0	RW	PWM waveform period.

12.3.11 PWM4_CFG_REG_1

Address 0x4000_0928			PWM3_CFG_REG_1	
Field Name	Field Bits	Reset Value	SW Access	Field description
PWM_CMP_1	31:16	0x0	RW	PWM counter compare point 1, when counter equal to this compare point value, the output is toggled.
PWM_CMP_2	15:0	0x0	RW	PWM counter compare point 2, when counter equal to this compare point value, the output is toggled.

12.3.12 PWM5_CFG_REG_0

Address 0x4000_092C			PWM4_CFG_REG_0	
Field Name	Field Bits	Reset Value	SW Access	Field description
PWM_INIT_VAL	31	0x0	RW	PWM waveform initial value, 1'b1 : PWM waveform start as high; 1'b0 : PWM waveform start as low.
RESV	30:16	0x0	RO	Reserved
PWM_PER	15:0	0x0	RW	PWM waveform period.

12.3.13 PWM5_CFG_REG_1

Address 0x4000_0930			PWM3_CFG_REG_1	
Field Name	Field Bits	Reset Value	SW Access	Field description
PWM_CMP_1	31:16	0x0	RW	PWM counter compare point 1, when counter equal to this compare point value, the output is toggled.
PWM_CMP_2	15:0	0x0	RW	PWM counter compare point 2, when counter equal to this compare point value, the output is toggled.

13 Timers

13.1 Overview

The device includes three 32-bit counter/timers. The counter/timer is designed to count cycles of the system derived clock. It can optionally generate interrupts or perform other actions at specified timer values, based on three match registers.
The first two counter/timers also includes up to two capture inputs to trap the timer value when an input signal transitions, optionally generating an interrupt.

13.2 Description

TBD

13.3 Timer Registers

13.3.1 Hardware Timer Control Register

Address 0x4000_0014				HW Timer Control Register
Field Name	Field Bits	Reset Value	SW Access	Field description
HW_TIMER3_BUSY	18	0x0	RW	Hardware Timer 3 busy indication
HW_TIMER2_BUSY	17	0x0	RW	Hardware Timer 2 busy indication, SW could set this bit to 1 only when IN_CAPTURE1_EN is 0.
HW_TIMER1_BUSY	16	0x0	RW	Hardware Timer 1 busy indication, SW could set this bit to 1 only when IN_CAPTURE0_EN is 0.
IN_CAPTURE1_CHN_SEL	15:11	0x0	RW	Input capture GPIO channel selection
IN_CAPTURE1_EN	10	0x0	RW	1: Input Capture enable
IN_CAPTURE1_MODE	9:8	0x0	RW	2'b00: Capture the width between rising edges of input signal, the capture result will be latched into HW_TIMER2 registers 2'b01: Capture the width between falling edges of input signal 2'b10: Capture the width from rising edge to falling edge of input signal 2'b11: Capture the width from falling edge to rising edge of input signal
IN_CAPTURE0_CHN_SEL	7:3	0x0	RW	Input capture GPIO channel selection
IN_CAPTURE0_EN	2	0x0	RW	1: Input Capture enable
IN_CAPTURE0_MODE	1:0	0x0	RW	2'b00: Capture the width between rising edges of input signal, the capture result will be latched into HW_TIMER2 registers 2'b01: Capture the width between falling edges of input signal 2'b10: Capture the width from rising edge to falling edge of input signal 2'b11: Capture the width from falling edge to rising edge of input signal

13.3.2 Hardware Timer 1 Register

Address 0x4000_0018				HW Timer 1 Register
Field Name	Field Bits	Reset Value	SW Access	Field description

HW_TIMER_1	31:0	0x0	RW	Hardware Timer 1 or Input capture0 result, when in capture mode, this register is read only

13.3.3 Hardware Timer 2 Register

Address 0x4000_001C		HW Timer 2 Register		
Field Name	Field Bits	Reset Value	SW Access	Field description
HW_TIMER_2	31:0	0x0	RW	Hardware Timer 2 or input capture1 result, when in capture mode, this register is read only

13.3.4 Hardware Timer 3 Register

Address 0x4000_0020		HW Timer 3 Register		
Field Name	Field Bits	Reset Value	SW Access	Field description
HW_TIMER_3	31:0	0x0	RW	Hardware timer 3 timeout value, configured by Software.

14 Watchdog Timer

14.1 Overview

14.2 Description

14.3 WDT Registers

14.3.1 TIMER_LDCNT_REG

Address 0x4000_0E00		TIMER_LDCNT_REG		
Field Name	Field Bits	Reset Value	SW Access	Field description
TIMER_LOAD_CNT	31:0	0x0	RW	Timer counter load value registers

14.3.2 TIMER_CURVAL_REG

Address 0x4000_0E04		TIMER_CURVAL_REG		
Field Name	Field Bits	Reset Value	SW Access	Field description
TIMER_CURVAL_CNT	31:0	0xFFFFFFFF	RO	Timer counter current value registers

14.3.3 TIMER_CTRL_REG

Address 0x4000_0E08		TIMER_CTRL_REG		
Field Name	Field Bits	Reset Value	SW Access	Field description
RESV	31:4	0x0	RO	Reserved
WDT_RST_RECORD	3	0x0	RO	1'b1 : Indicate the watch dog timer timeout once occurred.
TIMER_INT_MASK	2	0x0	RW	Timer interrupt mask bit. 1: interrupt is masked.
TIMER_MODE	1	0x0	RW	0: Free run mode, reload value is 32'hFFFFFFFF; 1: reload value is from TIMER_LOAD_CNT
TIMER_ENA	0	0x0	RW	1: Timer is enabled; 0: Timer is disabled.

14.3.4 TIMER_EOI_REG

Address 0x4000_0E0C		TIMER_CURVAL_REG		
Field Name	Field Bits	Reset Value	SW Access	Field description
TIMER_EOI	31:0	0x0	RO	Read this register to clear timer interrupt.

14.3.5 TIMER_INT_STATUS_REG

Address 0x4000_0E10		TIMER_INT_STATUS_REG		
Field Name	Field Bits	Reset Value	SW Access	Field description
RESV	31:1	0x0	RO	Reserved

TIMER_INT_STATUS	0	0x0	RO	Timer interrupt status bit, could be masked by TIMER_INT_MASK bit

14.3.6 TIMER_INT_RAWSTATUS_REG

Address 0x4000_0E14		TIMER_INT_STATUS_REG		
Field Name	Field Bits	Reset Value	SW Access	Field description
RESV	31:1	0x0	RO	Reserved
TIMER_INT_RAWSTATUS	0	0x0	RO	Timer interrupt raw status bit

15 CRC Engine

15.1 Overview

The Cyclic Redundancy Check (CRC) engine with programmable polynomial settings supports several CRC standards commonly used.
Supports three common polynomials CRC-8, CRC-16, and CRC-32.

- CRC-8: x8 + x5 + x4 + x3 + 1
- CRC-16: x16 + x15 + x2 + 1
- CRC-32: x32 + x26 + x23 + x22 + x16 + x12 + x11 + x10 + x8 + x7 + x5 + x4 + x2 + x + 1

15.2 Description

15.3 CRC Registers

15.3.1 CRC_CTRL_REG

Address 0x4000_00A0				CRC Calculate Control Register
Field Name	Field Bits	Reset Value	SW Access	Field description
RESV	31:4	0x0	RO	Reserved
CRC_BIT_ORDER	3	0x0	RW	1'b0: input data MSB first, 1'b1: input data LSB first.
RESV	2	0x0	RO	Reserved
CRC_CALC_MODE	1:0	0x0	RW	2'b00: Reserved; 2'b01: CRC8, x^8 + x^5 + x^4 + x^3 + 1 2'b10: CRC16, x^16 + x^15 + x^2 + 1 2'b11: CRC32, x^32 + x^26 + x^23 + x^22 + x^16 + x^12 + x^11 + x^10 + x^8 + x^7 + x^5 + x^4 + x^2 + x^1 + 1

15.3.2 CRC_INIT_REG

Address 0x4000_00A4				CRC Initial Value Register
Field Name	Field Bits	Reset Value	SW Access	Field description
CRC_INIT_VALUE	31:0	0x0	RW	CRC calculate register initial value

15.3.3 CRC_DATA_REG

Address 0x4000_00A8				CRC Data Register
Field Name	Field Bits	Reset Value	SW Access	Field description
RESV	31:8	0x0	RO	Reserved
CRC_DATA	7:0	0x0	RW	Input DATA, fixed 8 bit width

15.3.4 CRC_RSLT_REG

Address 0x4000_00AC		CRC Result Register		
Field Name	Field Bits	Reset Value	SW Access	Field description
CRC_RSLT_VALUE	31:0	0x0	RO	CRC calculate result register

16 ADC

16.1 Overview

16.2 Description

16.3 ADC Registers

16.3.1 ADC Read Control Register

Address 0x4000_0034				ADC Read Control Register
Field Name	Field Bits	Reset Value	SW Access	Field description
RESV	31:16	0x0	RO	Reserved
ADC_REPEAT	15:8	0x1	RW	ADC channel read repeat number.
ADC_RD	7	0x0	RW	SW set 1, and HW clear to this bit to 0 when ADC result is ready
ADC_SAM_CTRL	6:4	0x0	RW	ADC sample time control (in unit of clock cycle).
ADC_CHN	3:0	0x0	RW	ADC channel select.

16.3.2 ADC Result Register

Address 0x4000_0038				ADC Result Register
Field Name	Field Bits	Reset Value	SW Access	Field description
RESV	31:12	0x0	RO	Reserved
ADC_RSLT	11:0	0x0	RO	ADC result

17 TL431

17.1 Overview

TBD

17.2 Description

TBD

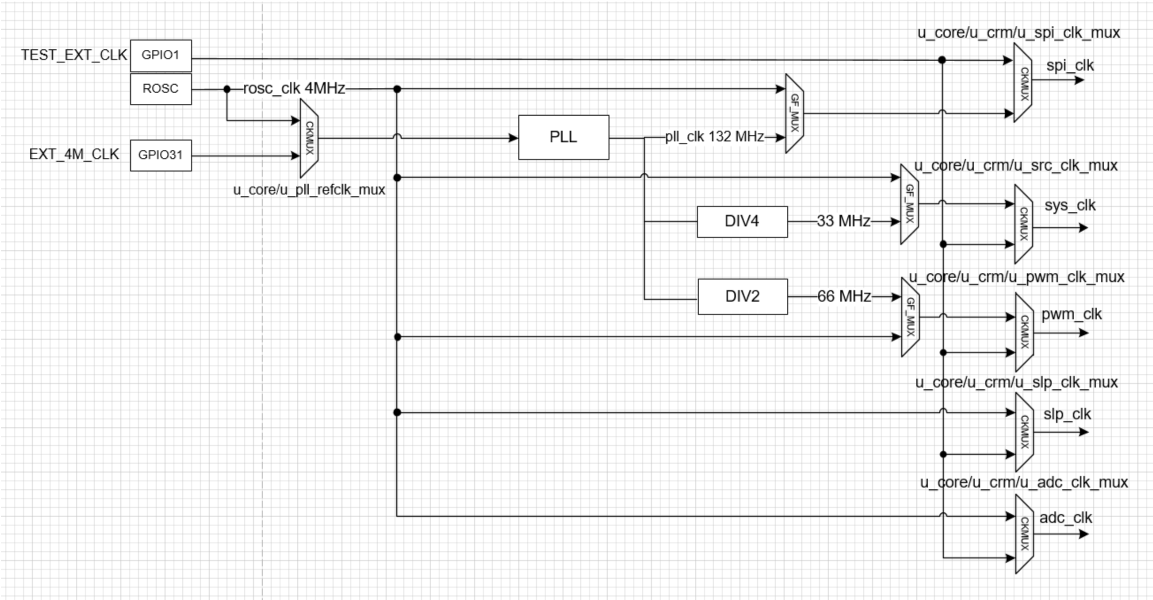
17.3 Registers

17.3.1 DA_CTRL_REG

Address 0x4000_00D4		DA_CTRL_REG		
Field Name	Field Bits	Reset Value	SW Access	Field description
PDDA2	31	0x1	RW	Power-down DAC, 1: power down, 0: power on
RESV	30:26	0x0	RO	Reserved
DAIN2	25:16	0x0	RW	DA input
PDDA3	15	0x1	RW	Power-down DAC, 1: power down, 0: power on
RESV	14:7	0x0	RO	Reserved
DAIN3	6:0	0x0	RW	DA input

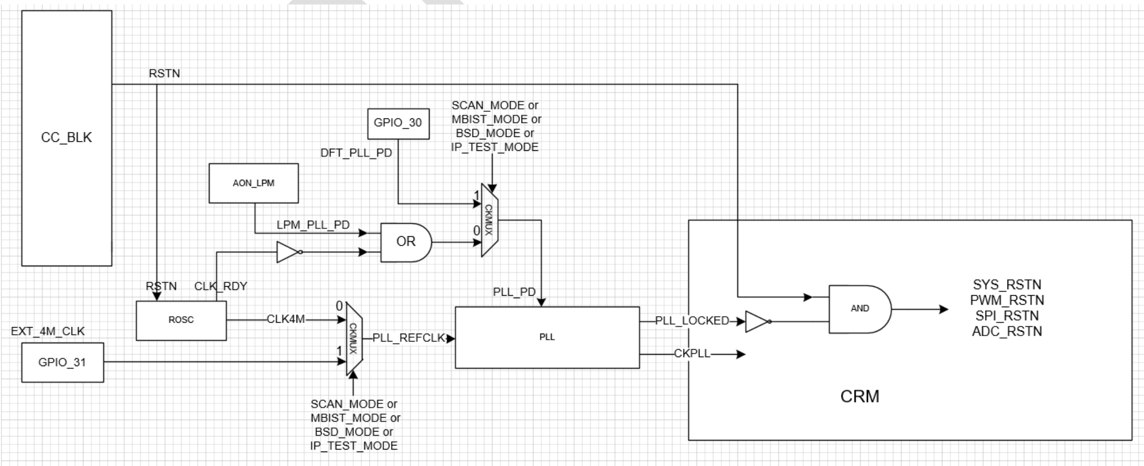
18 CLOCK&RST

18.1 CLK Overview



The device has one ROSC to generate 4 MHz clock, which works as both the sleep clock and also the PLL's input reference clock.
PLL is used to generate the 132 MHz clock, which is used by SPI flash interface logic.
PWM clock is 66 MHz which is PLL clock divided by 2.
MCU core and other system logic work at 33 MHz, which is the 132 MHz PLL clock divided by 4.

18.2 Reset Overview



18.3 Registers

18.3.1 PLL_CFG_REG

Address 0x4000_00C0		PLL_CFG_REG		
Field Name	Field Bits	Reset Value	SW Access	Field description
RESV	31:12	0x0	RO	Reserved
OSC_CLK_TEST	11	0x0	RW	1'b1: PLL clock div32 is connected to GPIO[0].
PLL_CLK_TEST	10	0x0	RW	1'b0: 4 MHz ROSC clock is selected as system clock,
PLL_CLK_SEL	9	0x0	RW	1'b1: PLL output clock is selected. SW should set this bit to high, when PLL_LOCKED is high.
PLL_LOCKED	8	0x1	RW	PLL LOCK indication.
DN	7:2	0x21	RW	Frequency of output clock: Fout = Fref*DN/DP
DP	1:0	0X1	RW	

18.3.2 ROSC_CFG_REG

Address 0x4000_00C4		ROSC_CFG_REG		
Field Name	Field Bits	Reset Value	SW Access	Field description
RESV	31:12	0x0	RO	Reserved
TUNE	5:1	0x0	RW	ROSC tune register
TSEL	0	0x1	RW	ROSC frequency tune select register, 1'b1: tuned by TRIM fuse. 1'b0: tuned by internal tune registers

19 Low Power Design

19.1 Overview

For the low power design, the device has no power gating design, only clock gating and logic reset is supported.

19.2 Description

SW set the "DPM_LPM_REG" register's SLP_MODE bit high to enter the sleep mode.
In sleep mode, flash, ADC, PLL will be set to power-down mode automatically. Except this, there're some independent registers to power-down some analog IPs individually. SW could configure these bits in the "PD_CTRL_REG" register before the chip entering sleep mode. Detailed information about these power-down control bits please see following LPM registers description.
In sleep mode, only ROSC (4 MHz) clock is running. Except the module "AON_LPM" is active, all the other modules, including MCU core is under reset, and clock is gate-off.
There're two ways to wake up the chip from sleep mode.

1) In sleep mode, any GPIO's toggle or CC IO toggle event will wake up the chip.
2) SW could set the auto-wakeup timer to wake up the chip in a specified period.

19.3 LPM Registers

19.3.1 DPM_LPM_REG

Address 0x4000_0094		DPM LPM Register		
Field Name	Field Bits	Reset Value	SW Access	Field description
RESV	31:25	0x0	RO	Reserved
T_RES	24:16	0x14E	RW	TRES1, timing parameter from FLASH power down release to CS drive low
RESV	15:9	0x0	RO	Reserved
SLP_MODE	8	0x0	RW	1'b1: chip enter sleep mode
SLP_AUTO_WAKEUP_TIMER	7:0	0x0	RW	HW sleep auto wake up timer

19.3.2 PD_CTRL_REG

Address 0x4000_00D0		PD_CTRL_REG		
Field Name	Field Bits	Reset Value	SW Access	Field description
IGAIN	6	0x0	RW	Current sense PGA Gain control. 1'b0: 15 times, 1'b1: 31 times.
PDISEN	5	0x0	RW	Power-down current sense control
PDREF	4	0x0	RW	Power-down reference control
PDCC	3	0x0	RW	Power-down CC wakeup
PD431	2	0x0	RW	Power-down 431 AMP control
PD10U	1	0x0	RW	AMP431 loading control. 1'b0: 10 uA loading 1'b1: No loading.
PDSAR	0	0x0	RW	Power-down SAR ADC, 1'b1 : Power-down, 1'b0 : Power-on

20 MISC.

20.1 Misc. Registers

20.1.1 Version ID and Reset register

Address 0x4000_0000		VERSION ID and Reset		
Field Name	Field Bits	Reset Value	SW Access	Field description
RESV	31:8	0x0	RO	Reserved
CHIP_RST	7	0x0	WC	Chip reset.
VERSION_ID	6:0	0x3	RO	Version ID

20.1.2 PROGRAM_ADDR_OFFSET Register

Address 0x4000_0030				PRG_ADDR_OFFSET Register
Field Name	Field Bits	Reset Value	SW Access	Field description
PRG_ADDR_OFFSET	31:12	0x0	RW	Program page address offset LSB.
RESV	11:0	0x0	RO	Reserved

20.1.3 ANA_TRIM_REG

Address 0x4000_00C8			ANA_TRIM_REG	
Field Name	Field Bits	Reset Value	SW Access	Field description
TURNBG	23:19	0x0	RW	LDO2, LDO3, 1.8V and 1.12v output, Band-Gap trim registers
TUNE33	18:14	0x0	RW	LDO1, 3.3V output, Band-Gap trim registers
ITRIM	13:8	0x0	RW	current source trim registers
RDTRIM2	7:4	0x0	RW	Rd1 trim registers
RDTRIM1	3:0	0x0	RW	Rd2 trim registers