8-bit LGT8XM
RISC Microcontroller with
In-System Programmable
FLASH Memory
LGT8F88P
LGT8F168P
LGT8F328P
Data book
Version 1.0.4.0A

Application area: appliance, motor driver, automation control

LGT8FX8P Series – EFLASH Based MCU Overview

功能概述/Feature

高性能低功耗 8 位 LGT8XM 内核

High Performance, Low Power 8-Bit LGT8XM Microcontroller Family

高级 RISC 构架

Advanced RISC Architecture

131 条指令，80%以上为单周期执行

131 Powerful Instructions – Most Single Clock Cycle Execution

32x8 通用工作寄存器

32 x 8 General Purpose Working Registers

32MHz 工作时最高可达 32MIPS 的执行效率

Up to 32 MIPS Throughput at 32MHz

内部单周期乘法器(8x8)

Internal single cycle multiplier (8x8)

非易失程序与数据存储空间

High Endurance Non-volatile Memory Segments

32Kbytes 片上可在线编程 FLASH 程序存储器

32KBytes of In-System Self-Programmable Flash program memory

2Kbytes 内部数据 SRAM

2KBytes Internal SRAM

可编程 E2PROM 模拟接口，支持字节访问

Programmable E2PROM analog interface, support access in byte

全新的程序加密算法，保证用户代码安全

New programming Encryption Algorithm, ensure of user code security

外设控制器

Peripheral control

两个具有独立预分频器的 8 位定时器，支持比较输出模式

Two 8-bit Timer/Counters with Separate Prescaler and Compare Mode

两个具有独立预分频器的 16 位定时器，支持输入俘获和比较输出

Two 16-bit Timer/Counter with Separate Prescaler, Compare Mode, and Capture Mode

内部 32KHz 可校准 RC 振荡器实现实时计数器功能

Internal 32KHz is calibrated, Real Time Counter with Separate Oscillator

最多可支持 9 路 PWM 输出，三组互补可编程死区控制

Max. 9 PWM Channels output, 3 sets complementary programmable dead time control

12 通道 12 位高速模数转换器(ADC)

12-channel 12-bit high speed anlogy-digital-converter (ADC)

12-bit high speed ADC

- 可选内部、外部参考电压

Internal and external voltage for option

- 可编程增益(X1/8/16/32)差分放大输入通道

input channel for Programmable gain (X1/8/16/32) differential amplifier

- 自动阀值电压监控模式

Automatic threshold voltage monitor mode

两路模拟比较器(AC)，支持来自 ADC 输入通道的扩展

2 channel Analog Comparator, support expansion of input channel from ADC

内部 1.024V/2.048V/4.096V ±1%可校准参考电压源

Internal calibrated reference voltage source :1.024V/2.048V/4.096V ±1%

一个 8 位可编程 DAC，可用于产生参考电压源

8-bit programmable DAC can generate reference voltage source

可编程看门狗定时器（WDT）

Programmable Watch Dog timer

可编程同步/异步串行接口（USART/SPI）

Programmable USART/SPI

同步外设接口(SPI)，可编程主/从工作模式

Master/Slave SPI Serial Interface

双线串行接口(TWI)，兼容 I2C 主从模式

TWI，Master/Slave I2C compatible

16 位数字运算加速单元(DSC)，支持直接 16 位数据存取访问

16-Bit DSC, support 16-bit data access

C 特殊处理器功能

Special Microcontroller Features

SWD 双线片上调试/量产接口

SWD dual line on-chip debug/Mass production interface

外部中断源与 I/O 电平变化中断支持

Supporting external interrupt source and I/O pin level change interrupt

内置上电复位电路（POR）与可编程低电压检测电路（LVD）

Internal POR and programmable LVD

内置 1%可校准 32MHz RC 振荡器，支持倍频输出

Internal calibrated 1% 32MHz RC oscillator, support frequency doubling output

内置 1%可校准 32KHz RC 振荡器

Internal calibrated 1% 32MHz RC oscillator

外部支持 32.768KHz 以及 400K~32MHz 晶振输入

Support external oscillator input of 32.768KHz and 400K~32MHz

6x 大电流推挽驱动 IO，支持高速 PWM 应用

6x big current driven IO, support high speed PWM application.

I/O 与封装：QFP48/32L, SSOP20L

I/Q and Package: QFP48/32L, SSOP20L

最低功耗：1uA@3.3V

Min. power: 1uA@3.3V

工作环境

Working environment:

工作电压： 1.8V ~ 5.5V

Working voltage: 1.8V ~ 5.5V

工作频率： 0 ~ 32MHz

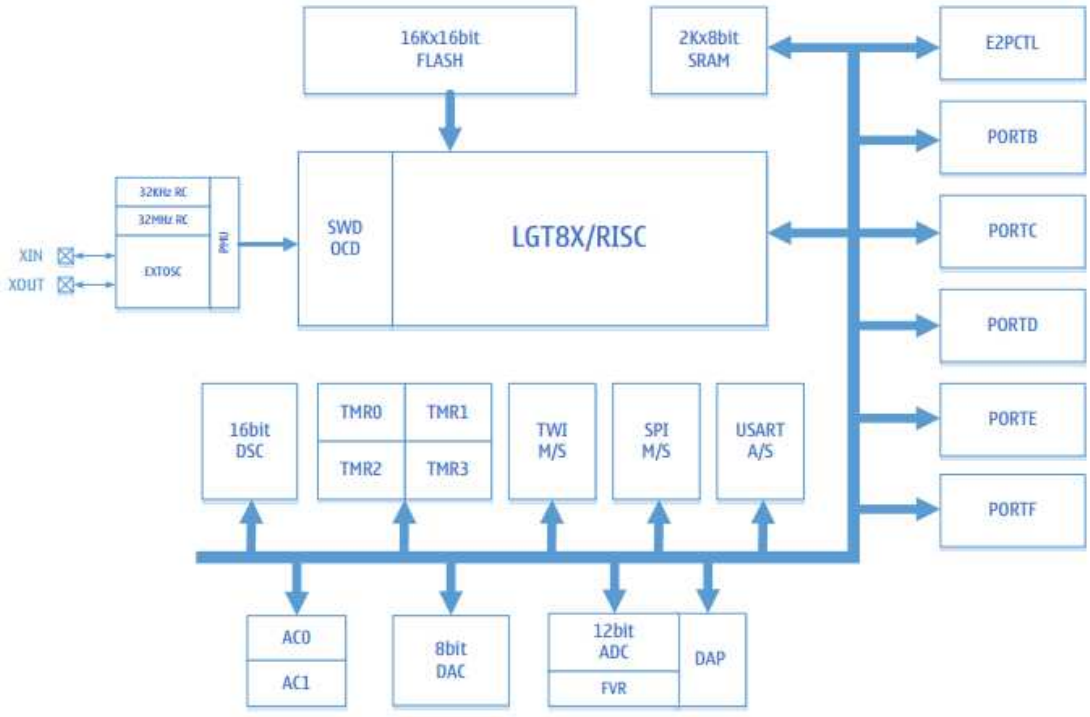
Working frequency: 0 ~ 32MHz

工作温度： -40C ~ +85C

Working temperature: -40C ~ +85C

HBM ESD : > 4K

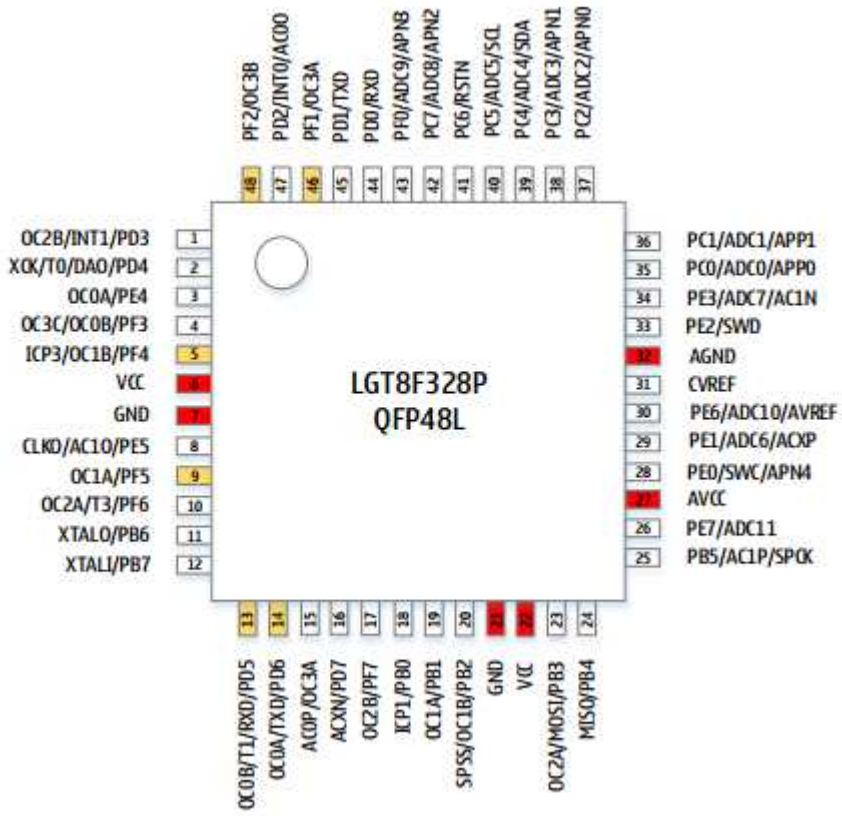
System Frame:

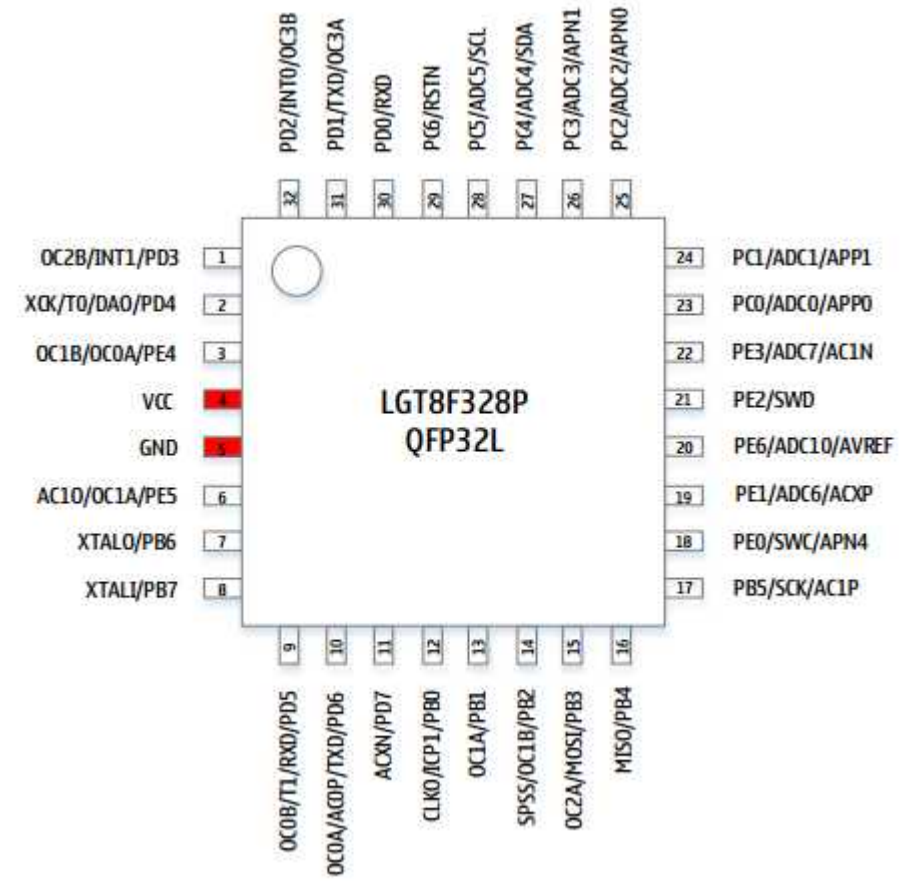


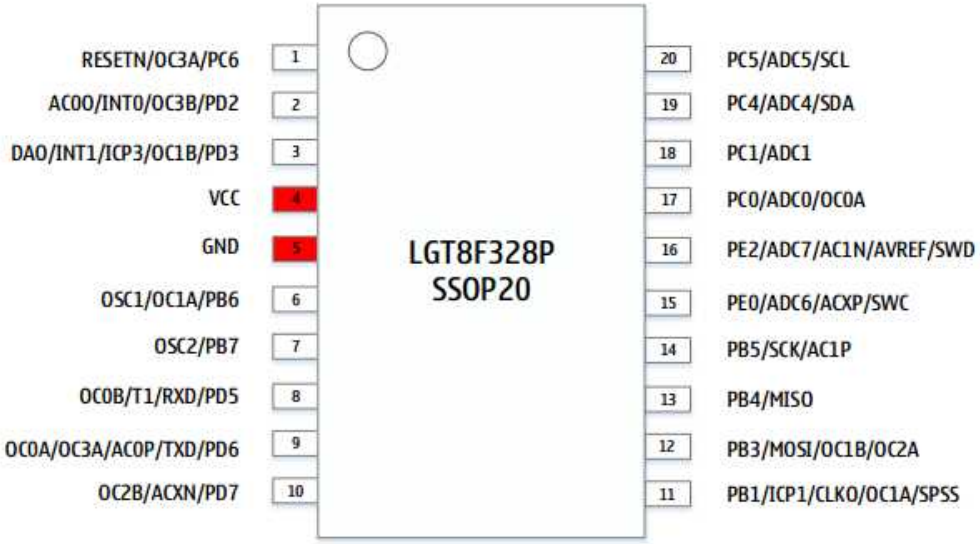
Module Name	Function
SWD	Debug module, on-line debugging and ISP at the same time
LGT8X	8 bit high performance RISC core
E29CTL	Data flash access interface controller
PMU	Power management module, manage to switch working state
PORTB/C/D/E/F	General programmable input and output
DSC	16-bit DSC
ADC DAP IVREF	8-bit 12 channel ADC Programmable gain differential amplifier 1.024V/2.048V/4.096V internal reference

AC0/1	analog comparator
TMR0/1/2/3	8/16 bit timer/counter, PMW control
WDT	Watch dog reset module
SPI M/S	Mater/Slave SPI controller
TWI M/S	TWI，Master/Slave I2C compatible
USART	USART/SPI
DAC	8-bit ADC

Package:







引脚说明

Pinout Configuration

LGT8FX8P 系列封装中，QFP48L 封装引出全部引脚。其他封装均为在 QFP48 基础上将多个内部 I/O 绑定到一个引脚上产生。配置引脚方向时需特别注意。下表列出各种封装引脚的绑定情况：

For packing of LGT8FX8P, QFP48L packing leads out all pins. Other packages is to bundle several internal I/O to one pin based on QFP48. Please pay high attention to pin out configuration.

Below tables illustrate pin bundle of all kinds of packing:

QFP48	QFP32	SSOP20	Comment
01	01	03	PD3/INT1/OC2B*
			PD3: programmable interface D3 INT1: external interrupt input 1 OC2B: timer 2, compare and match output B
02	02		PD4/DAO/T0/XCK
			PD4: programmable interface D4 DAO: internal DAC output TO: Timer0, external clock input XCK: USART synchronization clock transmission
03	03	-	PE4/0C0A*
			PE4: programmable interface E4 OCOA: Timer 0, compare and match output A
04	-	-	PF3/OC3C/OC0B*
			PF3: programmable interface F3 OC3C: Timer 3 compare and match output C OCOB: Timer 0 compare and match output B
05	03	03	PF4/OC1B*/ICP3
			PF4: programmable interface F4 OC1B: Timer 1 compare and match output B 1CP3: Timer 3 Capture input
06	04	04	VCC
07	05	05	GND
08	06	-	PE5/AC1O/CLKO*
			PE5: programmable interface E5 C10: Analog Comparator AC1 output CLKO: system clock output
09		06	PF5/OC1A*
			PF5: programmable interface F5 OCIA: Timer 1 compare and match output A
10	-	-	PF6/T3/OC2A*
			PF6: programmable interface F6 T3: Timer 3, external clock input OC2A: Timer 2, compare and match output A
11	07	06	PB6/XTALO
			PB6: programmable interface B6 XTALO: oscillator IO output interface
12	08	07	PB7/XTALI
			PB7: programmable interface B7

			XTALI: oscillator IO output interface
13	09	08	PD5/RXD*/T1/OC0B
			PD5: programmable interface D5 RXD: USART Data receive (option) T1: Timer 1, external clock input OCOB: Timer 0, compare and match output B
14	10	09	PD6/TXD*/OC0A
			PD6: programmable interface D6 TXD: USART Data sending (option) OCOA: Timer 0, compare and match output A
15			AC0P/0C3A
			AC0P: analog comparator 0 positive input OC3A: Timer 3, compare and match output A
16	11	10	PD7/ACXN
			PD7: programmable interface D7 AXCN: analog compactor 0/1 public negative input
17	-		PF7/OC2B
			PF7: programmable interface F7 OC2B: Timer 2, compare and match output B
18	12	11	PB0/ICP1
			PB0: programmable interface B0 ICP1: Timer 1, capture input
19	13		PB1/OC1A
			PB1: programmable interface B1 OC1A: Timer 1, compare and match output A
20	14	12	PB2/OC1B/SPSS
			PB2: programmable interface B2 OC1B: Timer 1, compare and match output B SPSS: SPI Slave mode chip select
21	-	-	GND
22	-	-	VCC
23	15	12	PB3/MOSI/OC2A
			PB3: programmable interface B3 MOSI:SPI Master Output/Slave Input OC2A: Timer 2 compare and match output A
24	16	13	PB4/MISO
			PB4: programmable interface B4 MISO: SPI Master Output/Slave Input
25	17	14	PB5/SPCK/AC1P
			PB5: programmable interface B5 SPCK: SPI Serial Clock AC1P: analog comparator 1 positive input
26	-	-	PE7/ADC11
			PE7: programmable interface E7 ADC11: ADC analog input channel 11

27	-	-	AVCC: internal analog circuit power
28	18	15	PE0/SWC/APN4
			PE0: programmable interface E0 SWC: SWD debug interface clock APN4: differential amplifier reverse input channel 4
29	19		PE1/ADC6/ACXP
			PE1: programmable interface E1 ADC6: ADC analog input channel 6 ACXP: analog comparator 0/1 public positive input
30	10	16	PE6/ADC10/AVREF
			PE6: programmable interface E6 ADC10: ADC analog input channel 10 AVREF: ADC external reference input
31	-	-	CVREF: ADC reference voltage output Only used to connect 0.1uF filter capacitor externally
32	-	-	AGND: internal analog circuit ground
33	21	16	PE2/SWD
			PE2: programmable interface E2 SWD: SWD debug interface cable
34	22		PE3/ADC7/AC1N
			PE3: programmable interface E3 ADC7: ADC analog input channel 7 AC1N: analog comparator, negative input
35	23	17	PC0/ADC0/APP0
			PC0: programmable interface C0 ADC0: ADC analog input channel 0 APP0: differential amplifier, forward input channel 0
36	24	18	PC1/ADC1/APP1
			PC1: programmable interface C1 ADC1: ADC analog input channel 1 APP1: differential amplifier, forward input channel 1
37	25	-	PC2/ADC2/APN0
			PC2: programmable interface C2 ADC2: ADC analog input channel 2 APN0: differential amplifier, reverse input channel 0
38	26	-	PC3/ADC3/APN1
			PC3: programmable interface C3 ADC3: ADC analog input channel 3 APN1: differential amplifier, reverse input channel

			1
39	27	19	PC4/ADC4/SDA
			PC4: programmable interface C4 ADC4: ADC analog input channel 4 SDA: I2C controller cable
40	28	20	PC5/ADC5/SCL
			PC5: programmable interface C5 ADC5: ADC analog input channel 5 SCL: 12C control clock cable
41	29	1	PC6/RESETN
			PC6: programmable interface C6 RESETN: external reset input
42	-	-	PC7/ADC8/APN2
			PC7: programmable interface C7 ADC8: ADC analog input channel 8 APN2: differential amplifier, reverse input channel 2
43	-	-	PF0/ADC9/APN3
			PF0: programmable interface F0 ADC9: ADC analog input channel 9 APN3: differential amplifier, reverse input channel 3
44	30	-	PD0/RXD
			PD0: programmable interface D0 RXD: USART Data receive input
45	31	-	PD1/TXD
			PD1: programmable interface D1 TXD: USART Data sending output
46		1	PF1/OC3A
			PF1: programmable interface F1 0C3A:Timer 3, compare and match output A
47	32	2	PD2/INT0/AC0O
			PD2: programmable interface D2 INT0: external interrupt input 0 AC00: analog compare 0 output
48			PF2/OC3B
			PF2: programmable interface F2 0C3B: Timer 3 compare and match output B

LGT8XM 内核

LGT8XM Core

- 低功耗设计

  Low power design

- 高效率 RISC 构架

  Enhanced RISC architucture

- 16 位 LD/ST 扩展(uDSU 专用)

  16-bite LD/ST expansion (especial for uDSU)
- 130 条指令，其中 80%以上为单周期

  130 pcs instructions , more than 80% are single circle
- 内嵌在线调试(OCD)支持

  Support Embed on-chip debugging

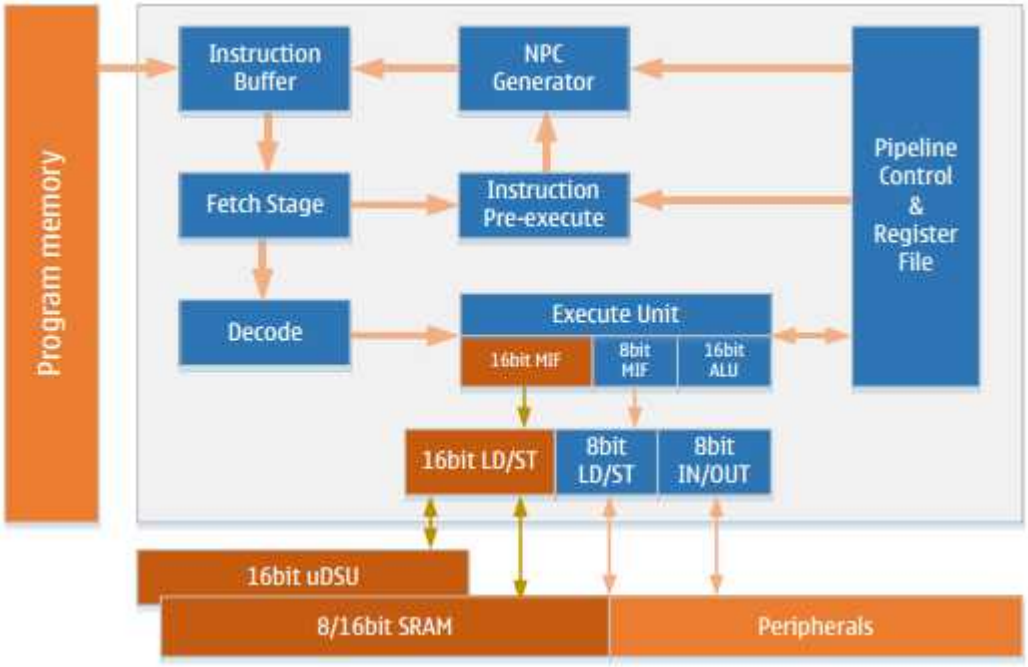
概述 Overview

本章节主要描述 LGT8XM 内核构架和功能。 内核是 MCU 的大脑，负责保证程序的正确执行，因此内核必须能够准确的执行计算，控制外设以及处理各种中断

Overview

This chapter will mainly descript core frame and function of LGT8XM. Core, brain of MCU, is in charge of right executive of software. So it is a must for core to be able to accurately execute algorithm, control external device and deal with all kinds of interrupt.

LGT8XM Core Construction Frame



为了实现更大的效率和并行性，LGT8XM 内核采用哈弗构架 – 独立的数据和程序总线。

指令通过一个优化的两级流水线执行，两级流水线能够减少流水线中无效指令的个数，减少了对 FLASH 程序存储器的访问量，因此可以降低内核运行的功耗。同时 LGT8XM 内核在取指令的前级中增加了指令缓存（可以同时缓存 2 条指令），通过在取指令周期的预执行模块，进一步减少了对 FLASH 程序存储器的访问频率；经大量测试，LGT8XM 可以比其他同类构架的内核减少约 50%对 FLASH 的访问，大大降低了系统的运行功耗。

In order to maximize performance and parallelism, the AVR uses a Harvard architecture – with separate buses for program and data.

Instructions in the program memory are executed with a double level pipelining, double level pipelining can reduce qty of invalid instruction in pipeline, and reduce page view to FLASCH memory, so that core power consumption is reduced. Meanwhile LGT8XM core add an instruction cache in the previous instruction-fetching (can cache 2 instruction at the same time), via pre-executive module of instruction-fetching cycle, it can further reduce page view frequency to FLASH program memory. After large scale testing, LGT8XM can

reduce 50% page view to FLASH compare with other core of same architecture, which reduce system power consumption.

LGT8XM 内核具有 32 个 8 位高速访问的通用工作寄存器(Register file)，有助于实现单周期的算术逻辑运算(ALU)。一般情况下， ALU 运算的两个操作数均来自与通用工作寄存器， ALU 运算的结果也会在一个周期内写入到寄存器文件中。32 个通过工作寄存器中的 6 个用于两两结合构成三个 16 位寄存器，可用于间接寻址地址指针，用于访问外部存储空间以及 FLASH 程序空间。LGT8XM 支持单周期的 16 位算术运算，极大的提高了间接寻址的效率。LGT8XM 内核中这三个特殊的 16 位寄存器被命名为 X, Y, Z 寄存器，将在后面详细介绍。

The fast-access Register File contains 32 x 8-bit general purpose working registers; this allows single-cycle Arithmetic Logic Unit (ALU) operation. In a typical ALU operation, two operands are output from the Register File, the operation is executed, and the result is stored back in the
Register File – in one clock cycle.
Six of the 32 registers can be used as three 16-bit indirect address register pointers for Data Space addressing – used to look up external space and FLASH program memory.
LGT8XM support single clock 16-bit ALU—enabling efficient indirect address calculations. These added function registers are the 16-bit X-, Y-, and Z-register, described later in this section.

ALU 支持寄存器之间以及常数与寄存器之间的算术逻辑运算，单个寄存器的运算也可以在 ALU 中执行。ALU 运算完成后，运算结果对内核状态的影响更新到状态寄存器中(SREG)。程序流程控制通过条件和无条件跳转/调用实现，可以寻址到所以的程序区域。大部分 LGT8XM 指令为 16 位。每个程序地址空间对应一个 16 位或者 32 位的 LGT8XM 指令。内核响应中断或子程序调用后，返回地址(PC)被存储在堆栈中。堆栈被分配在系统的一般数据 SRAM 中，因此堆栈的大小仅受限于系统中 SRAM 的大小和用法。所有的支持中断或子程序调用的应用，必须首先初始化堆栈指针寄存器(SP)，SP 可以通过 IO 空间访问。数据 SRAM 可以通过 5 种不同的寻址模式访问。LGT8XM 的内部存储空间都被线性的映射到一个统一的地址空间。具体请参考存储章节的介绍。

The ALU supports arithmetic and logic operations between registers or between a constant and a register. Single register operations can also be executed in the ALU. After an arithmetic operation, the Status Register is updated to reflect information about the result of the operation.
Program flow is provided by conditional and unconditional jump and call instructions, able to directly address the whole address space. Most LGT8XM instructions have a single 16-bit word format. Every program memory address contains a 16- or 32-bit LGT8XM instruction.

During interrupts and subroutine calls, the return address Program Counter (PC) is stored on the Stack. The Stack is effectively allocated in the general data SRAM, and consequently the Stack size is only limited by the total SRAM size and the usage of the SRAM. All user programs must initialize the SP in the Reset routine (before subroutines or interrupts are executed). The Stack Pointer (SP) is read/write accessible in the I/O space. The data SRAM can easily be accessed through the five different addressing modes. The memory spaces in the AVR architecture are all linear and regular memory maps. Detailed description please refer to chapter for Storage

LGT8XM 内核包含了一个灵活的中断控制器，中断功能可以通过状态寄存器中的一个全局中断使能位控制。所有的中断都有一个独立的中断向量。中断的优先级与中断向量地址有对应关系，中断地址越小，中断的优先级就越高。

A flexible interrupt module has its control registers in the I/O space with an additional Global Interrupt Enable bit in the Status Register. All interrupts have a separate Interrupt Vector in the Interrupt Vector table. The interrupts have priority in accordance with their Interrupt Vector position. The lower the Interrupt Vector address, the higher the priority.

I/O 空间包含了 64 个可以通过 IN/OUT 指令直接寻址的寄存器空间。这些寄存器现实对
内核控制以及状态寄存器，SPI 以及其他 I/O 外设的控制功能。这部分空间可以通过 IN/OUT 指令直接访问，也可以通过他们映射到数据存储器空间的地址访问(0x20 – 0x5F)。

The I/O memory space contains 64 addresses for CPU peripheral functions as Control Registers, SPI, and other I/O functions. The I/O Memory can be accessed directly, or as the Data Space locations following those of the Register File, 0x20 - 0x5F.

另外，LGT8FX8P 也包含扩展的 I/O 空间，他们被映射到数据存储空间 0x60 – 0xFF，这里只能使用 ST/STS/STD 以及 LD/LDS/LDD 指令访问。
Additionally, LGT8FX8P also have extended I/O, they are mapped to Data Space 0x60 – 0xFF, and instructions can only be accessed via ST/STS/STD and LD/LDS/LDD

为增强 LGT8XM 内核的运算能力 ，指令流行线中增加了 16 位的 LD/ST 扩展。此 16 位 LD/ST 扩展配合 16 数字运算加速单元(uDSU)工作，实现高效的 16 位数据运算。同时内核也增加对 RAM 空间的 16 位访问能力。因此 16 位 LD/ST 扩展可以在 uDSU，RAM，以及工作寄存器之间传递 16 位的数据。具体细节请参考”数字运算加速器”章节。
To improve algorithm of LGT8XM ACR, 16-bit LD/ST expension, which combines with 16-bit Digital Algorithm Accelerator unit (uDSU) to acheive highly efficient 16-bit data operaton, is increased to command line. Meanwhile core increase 16-bit access to RAM. So 16-bit LD/ST can transfer 16-bit data in uDSU, RAM and registers. For details, refer to section of “Algorithm Accelerator ”

算术逻辑运算单元（ALU）
ALU – Arithmetic Logic Unit
LGT8XM 内部包含了一个 16 位的算术逻辑运算单元，能够在一个周期内完成 16 为数据
的算术运算。高效的 ALU 与 32 个通用工作寄存器相连。能够在一个周期内完成两个寄存器或者寄存器与立即数之间的算术逻辑运算。ALU 的运算分为三种：算术，逻辑以及位运算。

LGT8XM has a 16-bit ALU, within a single clock cycle can finish 16-bit arithmetic operations. The high-performance ALU operates in direct connection with all the 32 general purpose working registers. Within a single clock cycle, arithmetic operations between 2 registers and an immediate are executed. The ALU operations are divided into three main categories – arithmetic, logical, and bit-functions.

同时 ALU 部分也包含了一个单周期的硬件乘法器，能够在一个周期内实现两个 8 位寄存器直接的有符号或者无符号运算。请参考指令集部分的详细介绍。

Meanwhile part of ALU also provide a single cycle hardware multiplier, that support signed/unsigned multiplication in a single cycle between 8 bit registers. See the “Instruction Set” section for a detailed description.

状态寄存器 (SREG)
Status Register
状态寄存器中主要保存了因执行最近一次 ALU 运算而产生的结果信息。这些信息用于
控制程序执行流程。状态寄存器是在 ALU 操作完全结束后更新，这样就可以省去了使用单独的比较指令，可以带来更加紧凑高效的代码实现。状态寄存器的值在响应中断和从中断中退出时并不会自动保存和恢复，这需要软件去实现

The Status Register contains information about the result of the most recently executed arithmetic

instruction. This information can be used for control program execution flow.
Status Register is updated after all ALU operations, this will in many cases remove the need for using the dedicated compare instructions, resulting in faster and more compact code.
The Status Register is not automatically stored when entering an interrupt routine and restored when returning from an interrupt. This must be handled by software.

SREG 寄存器定义

SREG Rgister Definition

SREG 系统状态寄存器								
地址: 0x3F (0x5F)				默认值: 0x00				
Bit	7	6	5	4	3	2	1	0
Name	I	T	H	S	V	N	Z	C
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

(0)	C	进位标志，表示算术或逻辑操作导致了进位，具体请参考指令描述 The Carry Flag C indicates a carry in an arithmetic or logic operation. See the “Instruction Set Description” for detailed information.
(1)	Z	零标志，表示算术或逻辑运算的结果为零，请参考指令描述部分 The Zero Flag Z indicates a zero resultin an arithmetic or logic operation. See the “Instruction Set Description” for detailed information.
(2)	N	负标志，表示算术或逻辑运算产生了一个负数，请参考指令描述部分 The Negative Flag N indicates a negative result in an arithmetic or logic operation. See the “Instruction Set Description” for detailed information
(3)	V	溢出标志，表示二进制补码运算结果产生溢出，请参考指令描述部分 The Two’s Complement Overflow Flag V supports two’s complement arithmetic. See the “Instruction Set Description” for detailed information.
(4)	S	符号位，等效于 N 与 V 的异或运算结果，具体请参考指令描述部分 The S-bit is always an exclusive or between the Negative Flag N and the Two’s Complement Overflow Flag V. See the “Instruction Set Description” for detailed information
(5)	H	半进位标志，在 BCD 运算中有用，表示字节运算产生了的半进位 Half Carry Flag :The Half Carry Flag H indicates a Half Carry in some arithmetic operations. Half Carry Is useful in BCD arithmetic.
(6)	T	临时位，位复制(BLD)和位存储(BST)指令中使用，T 位将作为一个临时的存储位，用于临时存放通用寄存器中的某一位的值。具体请参考指令描述部分 Bit Temporary Copy Storage: The Bit Copy instructions BLD (Bit Load) and BST (Bit Store) use the T-bit as source or destination for the operated bit temporarily. A bit from a register in the Register File can be copied into T. See the “Instruction Set Description” for detailed information
(7)	I	全局中断使能位，必须设置此位为 1 才能使能内核响应中断事件。不同的中断源是由独立的控制位控制。全局中断使能位是控制中断信号进入内核的最后一道屏障。 I 位在内核响应中断向量后由硬件自动清除，在执行中断返回指令(RETI)后自动置位。I 位也可以使用 SEI 和 CLI 指令改变，请参考指令描述部分 The Global Interrupt Enable bit must be set as 1 for the interrupts to be enabled.

		The individual interrupt enable control is then performed in separate control registers. If the Global Interrupt Enable Register is cleared, none of the interrupts are enabled independent of the individual interrupt enable settings. The I-bit is cleared by hardware after an interrupt has occurred, and is set by the RETI instruction to enable subsequent interrupts. The I-bit can also be set and cleared by the application with the SEI and CLI instructions, as described in the instruction set reference.

通用工作寄存器

General Purpose Register File

通用工作寄存器根据 LGT8XM 指令集构架优化。为了达到内核执行需要的效率和灵活性，

LGT8XM 内部的通用工作寄存器支持一下几种访问模式：

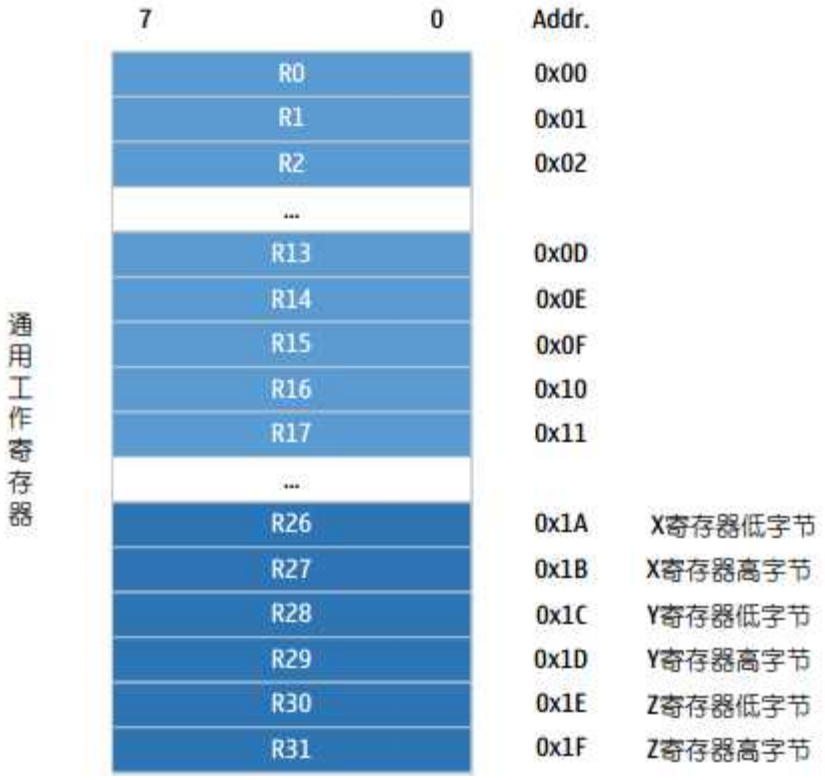
- 一个 8 位的读同时一个 8 位的写操作
- 两个 8 位的读同时一个 8 位的写操作
- 两个 8 位的读同时一个 16 位的写操作
- 一个 16 位的读同时一个 16 位的写操作

The Register File is optimized based on LGT8XM Enhanced RISC instruction set. In order to achieve the required performance and flexibility, the following input/output schemes are supported by the Register File:

- One 8-bit output operand and one 8-bit result input

Two 8-bit output operands and one 8-bit result input

- Two 8-bit output operands and one 16-bit result input
- One 16-bit output operand and one 16-bit result input



大部分指令能够直接访问到全部的通用工作寄存器，他们大部分也都是单周期指令。如上图所示，每一个寄存器都对应一个数据存储空间的地址，这

些通用工作寄存器被映射到数据储存空间。尽快他们不没有真正的存在于 SRAM 中，但这种统一映射的存储组织给访问
他们带来了很大的灵活性。X/Y/Z 寄存器可以作为指针索引到任何通用寄存器。

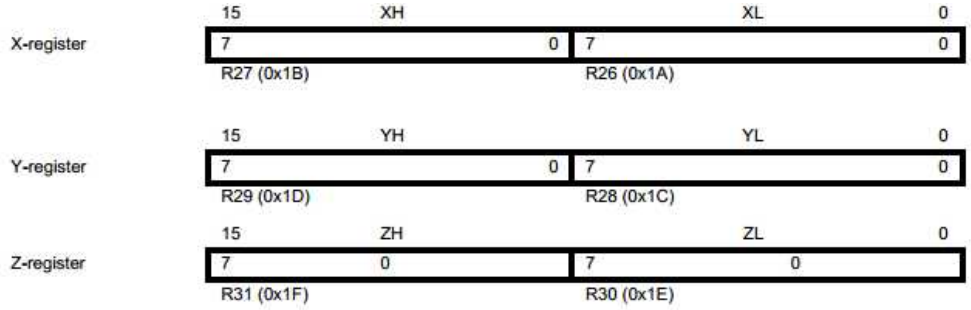
Most of the instructions operating on the Register File have direct access to all registers, and most of them are single cycle instructions. As shown above, each register is also assigned a data memory address, mapping them directly into the Data Space. Although not being physically implemented as SRAM locations, this memory organization provides great flexibility in access of the registers, as the X-, Y- and Z-pointer registers can be set to index any register in the file.

X/Y/Z 寄存器

The X-register, Y-register, and Z-register

寄存器 R26…R31 可以两两组合，构成三个 16 位寄存器。这三个 16 位寄存器主要用于间接寻址访问的地址指针，X/Y/Z 寄存器结构如下：

The registers R26...R31, grouping between two of them, is used as three 16-bit register. The three indirect address registers X, Y, and Z are defined as described as below:



在不同的寻址模式下，这些寄存器被用作固定偏移，自动递增以及自动递减的地址指针，具体细节请参考指令描述部分。

In the different addressing modes, these address registers have functions as fixed displacement, automatic increment, and automatic decrement (see the instruction set reference for details).

堆栈指针

Stack Pointer

堆栈用于存储临时数据，局部变量以及中断和子程序调用的返回地址。需要特别注意的
是，堆栈别设计为从高地址向低地址生长。堆栈指针寄存器(SP)总是指向堆栈的顶部。堆栈
指针指向数据 SRAM 所在的物理空间，这里存放子程序或中断调用必须的堆栈空间。 PUSH 指令将会使得堆栈指针递减。

The Stack is mainly used for storing temporary data, for storing local variables and for storing return addresses after interrupts and subroutine calls. Note that the Stack is implemented as growing from higher to lower memory locations. The Stack Pointer Register always points to the top of the Stack. The Stack Pointer points to the data SRAM Stack area where the Subroutine and Interrupt Stacks are located. A Stack PUSH command will decrease the Stack Pointer.

堆栈在 SRAM 中的位置必须在子程序执行或者中断使能之前由软件正确的设置。一般情况下是将堆栈指针初始化指向 SRAM 的最高地址处。堆栈指针必须设置为高位 SRAM 开始地址。SRAM 在系统数据存储映射的地址请参考系统数据存储部分。

The Stack in the data SRAM must be defined by the program before any subroutine calls are executed or interrupts are enabled. Initial Stack Pointer value equals the last address of the internal SRAM and the Stack Pointer must be set to point above start of the SRAM. Mapping address of SRAM in data storage pls. refers to “storage set” section.

Instruction	Stack pointer	Description
PUSH	Decremented by 1	Data is pushed onto the stack
CALL ICALL RCALL	Decremented by 2	Return address is pushed onto the stack with a subroutine call or interrupt
POP	Incremented by 1	Data is popped from the stack
RET RETI	Incremented by 2	Return address is popped from the stack with return from subroutine or return from interrupt

堆栈指针由分配在 I/O 空间的两个 8 位的寄存器构成。堆栈指针的实际长度与系统实现相

关。在 LGT8XM 构架的有些芯片实现中，数据空间非常小，以至于仅仅 SPL 就能满足寻址需要，这种情况下，SPH 寄存器将不会出现。

The AVR Stack Pointer is implemented as two 8-bit registers in the I/O space. The number of bits actually used is implementation dependent. Note that the data space of LGT8XM in some implementations of the AVR architecture is so small that only SPL is needed. In this case, the SPH Register will not be present.

SPH/SPL Stack Pointer Register		
SPH: 0x3E (0x5E	Default value: RAMEND	
SPL: 0x3D (0x5D)		
SP	SP[15:0]	
R/W	R/W	
Location define		
[7:0]	SPL	Stack Pointer lower than 8 bit
[15:8]	SPH	Stack Pointer higher than 8 bit

指令执行时序

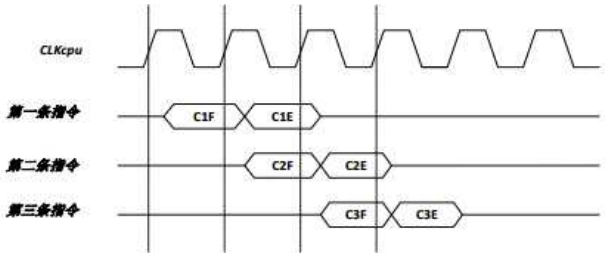
Instruction Execution Timing

这一章节描述指令执行的一般时序概念。 LGT8XM 内核由内核时钟(CLKcpu)驱动，这个时

钟直接来自与系统的时钟源选择电路。下图展示了哈弗构架与快速访问寄存器文件概念基础上的指令流水线执行时序。这是使得内核能够获得 1MIPS/MHz 的执行效率的物理保证。

This section describes the general access timing concepts for instruction execution. The LGT8XM CPU is driven by the CPU clock clkCPU, directly generated from the selected clock source for the chip. No internal clock division is used. Below picture shows the parallel instruction fetches and instruction executions enabled by the Harvard architecture and the fast-access Register File concept. This is the basic pipelining concept to obtain up to 1 MIPS per MHz

with the corresponding unique results for functions per cost, functions per clocks, and functions per power-unit.



从上图可以看出，第一条指令的执行期间同时会读出第二条指令。当第二条指令进入执行期间，同时又会读出第三条指令。这样在整个执行期间，并不

需要为读取指令花费额外的

周期，从流水线上看，实现了每个周一执行一条指令的效率。

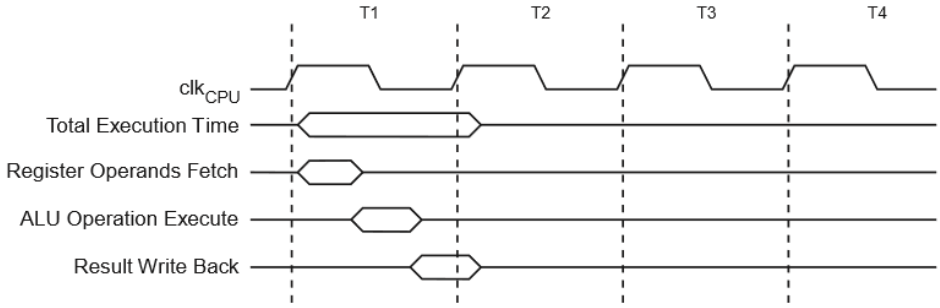
As shown above, 1st instruction execute meanwhile 2nd instruction is fetched. When 2nd instruction enters into execution, 3rd instruction is fetched at the same time. In a whole execution cycle, no extra cycle is needed for instruction execution and fetch. Viewing from the pipeline, one cycle for one instruction is efficiently achieved.

下图展示通用工作寄存器的访问时序，在一个周期内， ALU 操作使用到两个寄存器作为

操作数，并在这个周期内将 ALU 执行结果写入到目标寄存器中。

Below picture shows the internal timing concept for the Register File. In a single clock cycle an ALU operation using two register operands is executed, and the result is stored back to the destination register.

Single Cycle ALU Operation



复位与中断处理

Reset and Interrupt Handling

LGT8XM 支持多个中断源。这些中断以及复位向量在程序空间都对应一个独立的程序向量入口。一般而言，所有的中断都有单独的控制位控制。当设置了该控制位，并且使能了内 核的全局中断使能位后，内核才能响应这个中断.

The AVR provides several different interrupt sources. These interrupts and the separate Reset Vector each have a separate program vector in the program memory space. All interrupts are assigned individual enable bits, which must be written logic one together with the Global Interrupt Enable bit in the Status Register in order to enable the interrupt.

最低的程序空间默认保留为复位以及中断向量区域。LGT8FX8P 支持的完整的中断列表请参考中断章节的介绍。这个列表同时也决定了不同中断的优先级。向量地址越低的中断， 对应的中断优先级就越高。复位(RESET)具有最高的优先级，然后是 INT0 – 外部中断请求 0. 中断向量表的起始地址（复位向量除外）可以被重新定义到任何 256 字节对齐的开始处，需要通过 MCU 控制寄存器(MCUCR)中的 IVSEL 位以及 IVBASE 向量基地址寄存器实现。

The lowest addresses in the program memory space are by default defined as the Reset and Interrupt Vectors. The complete list of vectors is shown in ”Interrupts”section. The list also determines the priority levels of the different interrupts. The lower the address the higher is the priority level. RESET has the highest priority, and next is INT0 – the External Interrupt Request 0. The Interrupt Vectors can be moved to the start of any 256 byte alignment by setting the IVSEL bit in the MCU Control Register (MCUCR) and setting the IVBASE vector address Register.

当内核响应中断后，全局中断使能标志为 I 会被硬件自动清除。用户可以通过将 I 位使能实现中断嵌套。这样任何随后发生的中断都会中断当前的中断服务程序。I 位在执行中断返回指令(RETI)后自动置位，从而可以正常响应随后发生的中断。

When an interrupt occurs, the Global Interrupt Enable I-bit is cleared by hardware automatically The user software can write logic one to the I-bit to enable interrupt nesting. All enabled interrupts afterwards can then interrupt the current interrupt routine. The I-bit is automatically set when a Return from Interrupt instruction –

RETI – is executed, than interrupt afterwards can be enabled as normal.

有种基本的中断类型。第一种类型由事件触发，中断事件发生后置位中断标志位。对于 这种中断来说，内核响应中断请求后，当前的 PC 值被直接替换为实际的中断向量地址，执 行对应的中断服务子程序，同时硬件自动清除掉中断标志位。中断标志位也可以通过向中断标志位的位置写 1 清除。如果在发生中断时，中断使能位被清除，中断标志位仍然会被设置以记录中断事件。等到中断使能后，这个记录的中断事件会被立即响应。

同样，如果在中断发生时，全局中断使能位(SERG.I)被清除，对应的中断标志位也会被设置以记录中断事件，等到全局中断使能位被设置后，这些被记录的中断将会依照优先级依次执行。

第二种中断类型是当中断条件一直存在时，中断就一直响应。这种中断不需要中断标志 位。如果中断条件在中断使能之前消失，这个中断将不会得到响应。

There are basically two types of interrupts. The first type is triggered by an event that sets the Interrupt Flag. For these interrupts, the Program Counter is vectored to the actual Interrupt Vector in order to execute corresponding interrupt handling subroutine, and hardware clears the corresponding Interrupt Flag. Interrupt Flags can also be cleared by writing a logic one to the flag bit position(s) to be cleared. If an interrupt condition occurs while the corresponding interrupt enable bit is cleared, the Interrupt Flag will be set and triggered until the interrupt is enabled.

Similarly, if one or more interrupt conditions occur while the Global Interrupt Enable bit is cleared, the corresponding Interrupt Flag(s) will be set and triggered until the Global Interrupt Enable bit is set, and will then be executed by order of priority.

The second type of interrupts will trigger as long as the interrupt condition is present. These interrupts do not necessarily have Interrupt Flags. If the interrupt condition disappears before the interrupt is enabled, the interrupt will not be triggered.

当 LGT8XM 内核从中断服务子程序中退出后，执行流程会返回到主程序中。在主程序中 执行一条或几条指令后，才能响应其他等待的中断请求。

When the AVR exits from an interrupt subroutine, it will always return to the main program and execute one more instruction before any pending interrupt is served.

需要注意的是，系统状态寄存器(SREG)在进入中断服务后并不会自动保存，也不会在从 中断服务返回后自动恢复。它必须由软件负责处理。

Note that the Status Register is not automatically stored when entering an interrupt routine, nor restored when returning from an interrupt routine. This must be handled by software.

当使用 CLI 指令禁止中断后，中断将会被立即禁止。在 CLI 指令之后发生的所以中断都 不会得到响应。即使是和 CLI 指令执行时同时发生的中断，也不会被响应。下面的例子中说 明如何利用 CLI 避免中断打乱 EEPROM 的写时序：

When using the CLI instruction to disable interrupts, the interrupts will be immediately disabled. No interrupt will be executed after the CLI instruction, even if it occurs simultaneously with the CLI instruction. The following example shows how this can be used to avoid interrupts during the timed EEPROM write sequence

中断响应时间

Interrupt Response Time

LGT8XM 内核针对中断响应进行了优化，使得任何中断在 4 个系统时钟周期内一定得到 响应。4 个系统时钟周期后，中断服务子程序进入执行周期。在这 4 个时钟内，中断之前的 PC 值被压入堆栈，系统执行流程跳转到中断向量对应中断服务程序。如果中断发生在 一个多周期指令执行期间，内核将保证当前指令正确的执行结束。如果中断发生在系统处于 休眠状态下(SLEEP)，中断响应需要额外增加 4 个时钟周期。这增加的时钟周期用于从选择的 休眠模式下唤醒操作的同步周期。休眠模式的具体描述，请参考功耗管理的相关章节。

从中断服务子程序中返回需要 2 个时钟周期。在这 2 个时钟周期内，PC 从堆栈中恢复， 堆栈指针加 2，并自动使能全局中断控制位。

LGT8XM optimize its interrupt in order that any interrupt can be responded within 4 clock cycles.

After 4 clock cycles interrupt handling subroutine is executed. During this four clock cycle period, the Program Counter is pushed onto the Stack.

Execution flow is normally a jump to interrupt handing routine corresponding to interrupt vector. If interrupt occurs during execution of a multi-cycle instruction, current instruction will be guaranteed executed rightly by AVR.

If an interrupt occurs when the MCU is in sleep mode, the interrupt execution response time is increased by four clock cycles. This increase comes in addition to the start-up time from the selected sleep mode. Details about sleeping mode, please refer to power consumption section.

A return from an interrupt handling routine takes 2 clock cycles. During these 2 clock cycles, the Program Counter (two bytes) is popped back from the Stack, the Stack Pointer is incremented by two, and Global Interrupt Enable bit is automatically enabled.

存储单元

Memories

概述

Overview

本章节主要描述 LGT8FX8P 系列内部不同的存储单元。 LGT8XM 构架支持两种主要的内部

存储空间，分别是数据存储空间和程序存储空间。 LGT8FX8P 内部也包含了数据 FLASH，通过内部的控制器可以实现 EEPROM 接口的数据存储功能。另外， LGT8FX8P 系统中还包含了特殊的存储单元，用于存放系统配置信息以及芯片的全局设备号(GUID)。

This section describes the different memories of LGT8FX8P series. The LGT8XM architecture has two main memory spaces, the Data Memory and the Program Memory space. LGT8FX8P internally has data Flash, by internal controller, it can realize EEPROM Memory for data storage. Besides, LGT8FX8P has special memory which is used to storage system configuration and GUID of chip.

LGT8FX8P 系列芯片包含了 LGT8F88P/168P/328P 四种不同的型号；四种型号的外设以及

封装完全兼容，所不同是 FLASH 程序存储空间以及内部数据 SRAM，下面的表格比较清楚的描述了 LGT8FX8P 系列芯片不同的存储空间配置：

LGT8FX8P series contain 4 types: LGT8F88P/168P/328P. Peripheral design and package of these 4 types are compatible, only FLASH program space and internal DATA SRAM are different, below table descript the different memory configuration of LGT8FX8P series chips:

DEVIE	FLASH	SRAM	EZPROM	Interrupt vector
LGT8F88P	8KB	1KB	2KB	1 instruction word
LGT8F168P	16KB	1KB	4KB	2 instruction word
LGT8F328P	32KB	2KB	0/1/2/4/8KB are available (share with FLASH)	2 instruction word

LGT8F328P 内部没有独立用于模拟 E2PROM 接口的 FLASH 空间；用于模拟 E2PROM 的存储空间与程序 FLASH 共享，用户可以根据应用需求，选择合适的配置。由于模拟 E2PROM 接口采用的独特实现，系统需要两倍的程序 FLASH 空间模拟 E2PROM 存储空间，比如对于 LGT8F328P， 如果用户配置了 1KB 的 E2PROM 空间，将会有 2KB 字节的程序空间被保留，剩下 30KB 的 FLASH 空间用于存储程序。

There is no independent FLASH space for analog ERPROM interface, which is shared with program FLASH, so user should choose suitable configuration as per individual demand.

Because of this special method for analog E2PROM interface, system need 2 times program FLASH space to analog E2PROM storage space. Taking LGT8F328P as example, if user choose s 1KB for E2PROM, 2KB program space will be kept, the rest 30KB of FLASH space will be used as program storage.

LGT8F328P 程序 FLASH 与 E2PROM 共享配置表：

FLASH VS E2PROM configuration of LGT8F328P:

DEVICE	FLASH	E2PROM
LGT8F328P	32KB	0KB
	30KB	1KB
	28KB	2KB
	24KB	4KB
	16KB	8KB

系统可编程 FLASH 程序存储单元

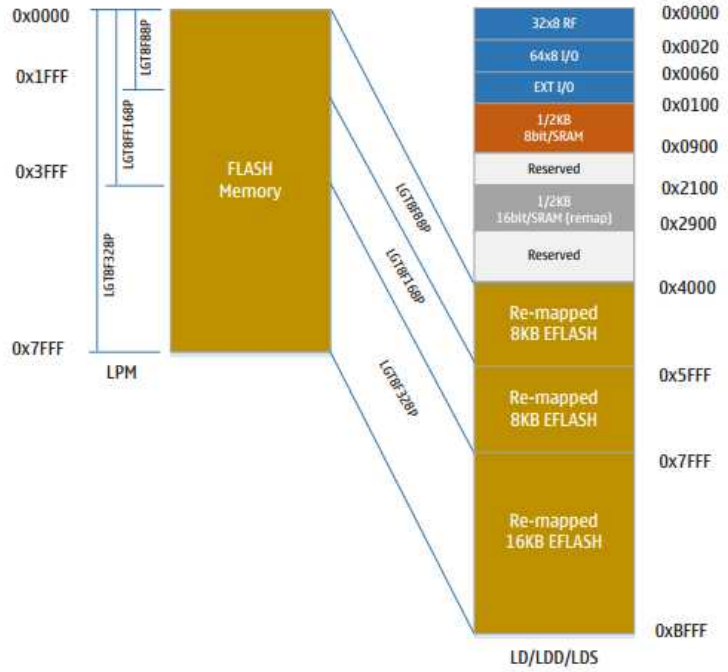
In-System Reprogrammable Flash Program Memory

LGT8FX8P 系列微控制器内部分别包括 8K/16K/32K 字节的片上在线可编程 FLASH 程序存储单元。程序 FLASH 能保证至少 100,000 次以上的擦写周期。LGT8FX8P 内部集成 FLASH 接口控制器，能够实现在系统编程(ISP)以及程序的自升级功能。具体实现细节请参考本章在关于 FLASH 接口控制器部分的描述。程序空间也可以通过 LPM 指令直接访问(读取)，这个特点可以实现应用相关的常数查找表。同时 FLASH 程序空间也被映射到系统数据存储空间内，用户也可以使用 LD/LDD/LDS 实现对 FLASH 空间的访问。程序空间被映射到数据存储空间 0x4000 开始的地址范围内。如下图所示：

LGT8FX8P series microcontroller contains 8K/16K/32K on-chip In-System Reprogrammable Flash memory for program storage. The Flash memory has an endurance of at least 10,000 write/erase cycles. Through FLASH interface cotroller, LGT8FX8P can upgrade In-system programming and program by itself. Details please refer to section of “FLASH Interface Controller”.

Program space can be access (read and fetch) through LPM instruction, this feature can realize constant table of relevant application.

Meanwhile program space maps into data storage space, so user can access FLASH space by LD/LDD/LDS. Program space is mapped into data storage space addressing starting from 0x 4000.



SRAM 数据存储单元

SRAM Data Memory

LGT8FX8P 系列微控制器是一种相对复杂的微控制器，它支持多种不同类型的外设，这些外设的控制器被分配在 64 个 I/O 寄存器空间内。可以直接通过 IN/OUT 指令访问。另一些外设的控制寄存器分配在 0x60 ~ 0xFF 区域内，由于这部分空间是映射到数据存储空间内，只能通过 ST/STS/STD 以及 LD/LDS/LDD 等指令访问。

LGT8FX8P is a complex microcontroller with more peripheral units than can be supported within the 64 locations reserved in the Opcode for the IN and OUT instructions. For the Extended I/O space from 0x60 - 0xFF in SRAM, only the ST/STS/STD and LD/LDS/LDD instructions can be used.

LGT8FX8P 的系统数据存储空间从 0 地址开始，分别映射了通用工作寄存器文件， I/O 空间，扩展 I/O 空间以及内部数据 SRAM 空间。最开始的 32 个字节地址对应 LGT8XM 内核 32 个通用工作寄存器。接下来的 64 个地址是可以通过 IN/OUT 指令直接访问的标准 I/O 空间. 然后的 160 个地址是扩展 I/O 空间，在接下来就是最多 2K 字节的数据 SRAM。从 0x4000 开始到 0xBFFF 结束的这部分空间，映射了 FLASH 程序存储单元。

LGT8FX8P data memory location address from 0, mapping general register files, I/O memory, Extended I/O memory, and the internal data SRAM. The first 32 byte locations address the 32 Register Files, the next 64 location are the standard I/O memory, which can be directly access through IN/OUT instruction, the following 160 locations is extended I/O memory, afterwards are max. 2K byte DATA SRAM. Space starting from 0x4000 to 0XBFFF map FLASH program storage.

系统内 1K/2K 字节 SRAM 被分别映射到两个空间。从 0x0100 开始到 0x0900 结束的这个空间被内核以 8 位字节的宽度读写。 从 0x2100 开始到 0x2900 结束这个区域为 16 位宽度的访问空间。系统 RAM 被映射到 0x2100 开始的高位地址主要用于配合 uDSU 模块工作，实现高效的 16 位数据存储。在编程时，将普通的 8 位寻址变量地址加上 0x2000 的偏移量，即可切换到 16 位访问模式。

1/2K byte SRAM address to 2 spaces. Space from 0x0100 to 0x0900 is read at 8 bit wide, while space from 0x2100 to 0x2900 is ready at 16 bit wide. High location, RAM mapping to 0x2100, is used to work with uDSU to achieve enhanced 16 bit DATA storage. Adding normal 8 bit addressing vetor with 0X2000 Displacement, access mode is changed to 16 bit

系统支持 5 种不同的寻址模式可以覆盖到整个数据空间：直接访问，带偏移的间接访问，间接访问，访问前递减地址的间接访问，访问后递增地址的间接访问。通用工作寄存器 R26 到 R31 用于间接访问的地址指针。间接访问可以寻址整个数据存储空间。带偏移地址的间接访问能够寻址到以 Y/Z 寄存器为基地址的附近 63 个地址空间。

The five different addressing modes for the data memory cover: Direct, Indirect with Displacement, Indirect, Indirect with Pre-decrement, and Indirect with Post-increment. In the Register File, registers R26 to R31 feature the indirect addressing pointer registers. The indirect addressing reaches the entire data space. The Indirect with Displacement mode reaches 63 address locations from the base address given by the Y- or Z-register.

当使用支持地址自动递增/递减的寄存器间接访问模式，地址寄存器 X/Y/Z 会在访问发生前/后自动由硬件递减/递增。具体请参考指令集描述部分。16 位寄存器 X/Y/Z 以及与之相关的自动寻址模式(递增、递减)，在 16 位扩展模式下也有着非常重要的作用。16 位扩展模式可以使用 LD/ST 的递增/递减模式，实现带变量的自动递增、递减寻址。这种模式在对数组进行运算操作时，将非常有效。具体实现请参考“数字运算加速器(uDSU)”相关章节。

When using register indirect addressing modes with automatic pre-decrement and post-increment, the address registers X, Y, and Z are decremented or incremented automatically by hardware. For details please refer to section of “instruction”, 16 Register X/Y/Z and related automatic addressing mode(increment, decrement) plays an important role in 16 bit Extend mode. 16 bit Extend mode can use the increment / decrement of LD/ST, to achieve automatic increment or automatic decrement with variable. This model is very useful for arithmetic array operation. For details, please refer to section of “uDSU”

通用 I/O 寄存器

LGT8FX8P 的 I/O 空间有三个通用 I/O 寄存器 GPIOR2/1/0，这三个寄存器可以使用 IN/OUT 指令访问，用于存放用户自定义数据。

General I/0 Register

LGT8FX8P I/O space has 3 general I/O register GPIOR2/1/0, which can be access via IN/OUT instruction and is used to storage user-defined data.

外设寄存器空间

Peripheral register space

I/O 空间的详细定义，请参考 LGT8FX8P 数据手册中“寄存器概述”章节。

Regarding detailed definition of I/O space, please refer to section of “Register Overview” of LGT8FX8P data sheet.

LGT8FX8P 所以的外设都被分配到 I/O 空间。所有的 I/O 空间地址都可以被 LD/LDS/LDDD

以及 ST/STS/STD 指令访问。访问的数据都是通过 32 个通用工作寄存器传递。在 0x00 ~ 0x1F 之间的 I/O 寄存器可以通过位寻址指令 SBI 和 CBI 访问。在这些寄存器中，某一个位的值可以使用 SBIS 和 SBIC 指令检测，用以控制程序的执行流程。具体请参考指令集描述部分。

All peripheral s of LGT8FX8P are placed in the I/O space. All I/O locations may be accessed by the LD/LDS/LDD and ST/STS/STD instructions, transferring data between the 32 general purpose working registers and the I/O space. I/O Registers within the address range 0x00 - 0x1F are directly bit-accessible using the SBI and CBI instructions. In these registers, the value of single bits can be checked by using the SBIS and SBIC instructions. Refer to the instruction set section for more details.

当使用 IN/OUT 指令访问 I/O 寄存器时，必须寻址 0x00 ~ 0x3F 之间的地址。当使用 LD 或 ST 指令访问 I/O 空间时，必须通过 I/O 空间在系统数据存储器统一映射空间的映射地址访问(加上 0x20 的偏移)。其他一些分配在扩展 I/O 空间的外设寄存器(0x60 ~ 0xFF)， 只能够使用 ST/STS/STD 和 LD/LDS/LDD 指令访问。为了与未来的设备兼容，保留位在写操作时必须写 0。不能在保留的 I/O 空间上执行写操作。一些寄存器中包括了状态标志，需要被写 1 才能清零。需要注意的是，CBI 和 SBI 指令仅仅支持特定的位，因此 CBI/SBI 也只能工作在包含这些状态标志的寄存器上。除此之外，CBI/SBI 指令只能工作在 0x00 到 0x1F 这个地址范围内的寄存器。

When using IN/OUT instruction access to I/O register, must address location between 0x00 and 0x3F. When addressing I/O Registers as data space using LD and ST instructions to access I/O memory, 0x20 must be added to these addresses.

For the Extended I/O space from 0x60 - 0xFF in SRAM, only the ST/STS/STD and LD/LDS/LDD instructions can be used.

For compatibility with future devices, reserved bits should be written to zero if accessed. Reserved I/O memory addresses should never be written. Some register include status flag, which are cleared only by writing a logical one to them. Note that, the CBI and SBI instructions will only operate on the specified bit, and can therefore be used on registers containing such Status Flags. The CBI and SBI instructions work with registers 0x00 to 0x1F only.

FLASH 控制器(E2PCTL)

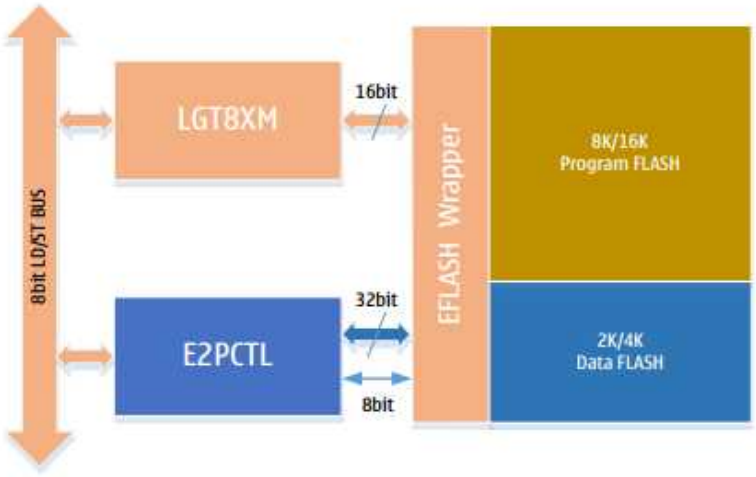
LGT8FX8P 内部实现集成了一个灵活可靠的 EFLASH 读写控制器，可以利用系统中已有的数据 FLASH 存储空间，实现字节读写访问的存储空间，实现类似 E2PROM 的存储应用； E2PROM 接口模拟采用擦写均衡的算法，可以将数据 FLASH 的使用周期提高 1 倍左右，能够保证 100,000 次以上的擦写周期。

LGT8FX8P internally integrate a flexible and reliable EFLASH read & write controller, it can use current data FLASH memory, to realize storage space accessed by byte read and write, similar to E2PROM storage. E2PROM interface analog adopt erase balancing algorithm, which can increase the lifespan of data FLASH for 1 time and can ensure more than 100,000 write/eras cycle.

E2PCTL 控制器也实现了对 FLASH 程序空间的在线擦写操作，可以通过软件实现在线自动升级固件的功能。通过 FLASH 控制器访问程序 FLASH 程序空间，只支持页擦除(1024 字节)以及 32 位宽度的读写访问。

E2PCTL controller also features in-system write/erase operation to FLASH program space, with function of in-system automatic upgrade via software. If FLASH controller access program FLASH memory, read and write access of page erase (1024 byte) and 32 bit wide is supported.

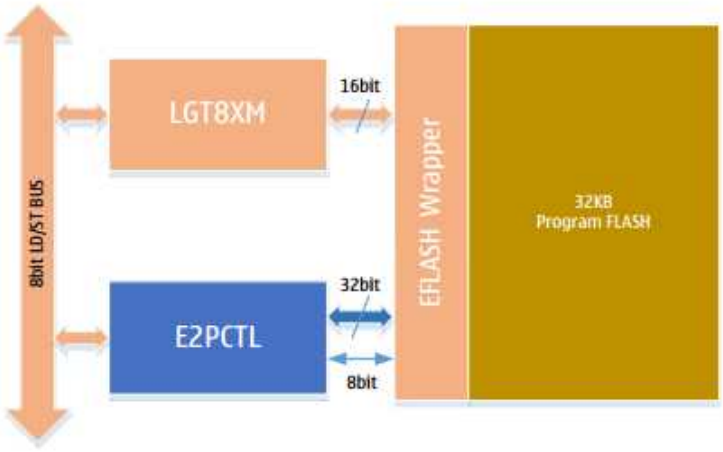
LGT8F88D/168D E2PCTL 控制器结构图



E2PCTL 模拟 E2PROM 功能访问数据 FLASH 空间时，可以支持 8 位、32 位读写宽度。访问程序 FLASH 空间时，支持页擦除和 32 位数据读写。由于 LGT8FX8P 内部 FLASH 的最小存储单元为 32 位，因此建议用 32 位访问方式，特别是对于写操作。32 位访问的读写操作不仅效率高，也有利于保护 FLASH 存储单元的擦写寿命。

When E2PCTL access data FLASH memory by analog E2PROM, it support 8 bit, 32 bit wide. When access to FLASH memory, support page erase and 32 bit data read and write. Because of the min. FLASH storage unit of LGT8FX8P is 32 bit, we suggest to use 32 bit access, especially for write, 32 bit read and write access is not only highly efficient, but also well protect erase/write lifespan of FLASH memory unit.

LGT8F328P E2PCTL 控制器结构图



LGT8F328P 内部没有多余的数据 FLASH. 因此，LGT8XM 内核与 E2PCTL 共享内部 32K 字节 FLASH 存储空间。用户可以根据需要，将 32K 字节 FLASH 空间划分为程序空间和数据空间。通过配置 E2PCTL 控制器，可以设置模拟 E2PROM 的空间大小。E2PCTL 使用页交换模式实现模拟 E2PROM 逻辑，算法以页(1K 字节)为单位。因此模拟 1K 字节的 E2PROM 空间，需要占用 2K 字节的 FLASH 空间，以此类推，实现 4K 字节的 E2PROM，需要占用 8K 字节的 FLASH 空间。具体实现方式，请参考 E2PCTL 算法实现的描述。

There is no extra data FLASH in LGT8F328P, so its ARU and E2PCTL share 32 byte FLASH memory, as per individual demand, user can divide 32 byte FLASH memory into program memory and data memory. Analog E2PROM memory size can be set-up via E2PCTL configuration. E2PCTL analog E2PROM logic by using page exchange mode, the algorithm unit is page (1 byte). So to analog 1K byte E2PROM memory, it need 2K byte FLASH memory, similarly, for 4K byte E2PROM, it need 8K byte FLASH memory. Detailed method please refer to section of “E2PCTEL Algorithm”

E2PCTL 数据寄存器

E2PCTL Data Register

E2PCTL 控制器内部有 4 个字节的数据缓存(E2PD0~3)，此 4 字节的缓存组成最终访问 FLASH 空间的 32 位数据接口。当 E2PCTL 控制器工作在字节读写模式时，EEDR 作为读写字节数据的接口， E2PCTL 更加 EEARL[1:0]的地址信息加载数据到正确的数据缓存中，并根据当前 FLASH 目标地址的数据补齐另外三个字节的数据， 最终将组合的完整 32 位数据更新到 FLASH 中。

E2PCTL controller has 4 byte data cache internally(E2PD0~3), this 4 byte cache consist of 32 bit data interface of final access to FLASH memory. When E2PCTL controller work in mode of byte read & write, EEDR is used as the data interface for byte read and write.

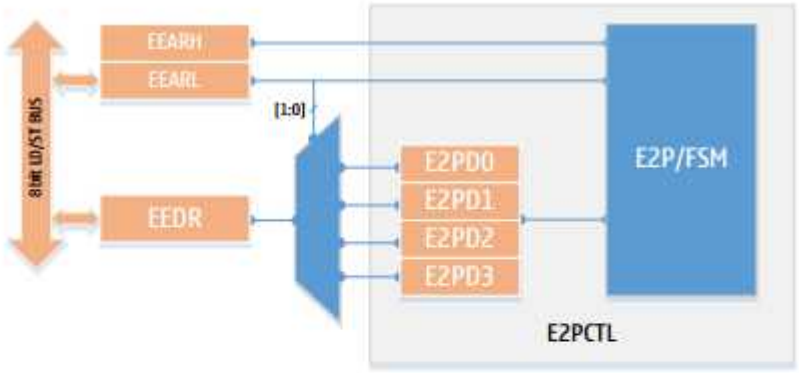
Adding E2PCTL and address info of EEARL[1:0], load data to correct data cache, based on current FLASH target address to complete other 3 byte data, finally complete 32 byte data will be updated to FLASH.

当 E2PCTL 工作在 32 位读写模式时, 此时仍然可以使用 EEDR 寄存器作为一个公用的数据接口， 通过 EEARL[1:0]作为地址寻址内部数据缓存，实现读写一个完整的 32 位数据。此外，还可以直接使用数据缓存映射到 IO 空间的寄存器直接访问(E0~3)。E2PCTL 工作在 8 位字节读写模式时的数据访问示意图：

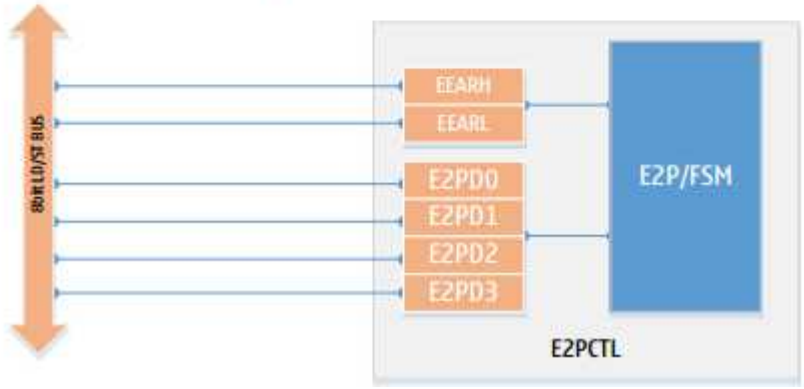
When E2PCTL work in pattern of 32 byte read and write, EEDR register here still can be used as public data interface.

Addressing internal data cache via EEARL[1:0] location, a complete 32 byte data is finished. Besides, directly using data cache to map to register direct access of IO memory. When E2PCTL is working in pattern of 8 byte read and write, its data access routine shows as below:

E2PCTL 工作在 8 位字节读写模式时的数据访问示意图：



E2PCTL 工作在 32 位字读写模式时的数据访问示意图：



字节模式用于向下兼容 LGT8FX8D 的字节读写模式。LGT8FX8P 的内置 FLASH 为 32 位接口宽度， 使用 32 位读写模式将给读写效率和 FLASH 的擦写寿命带来极大的好处，因此建议使用 32 位读写模式。

Byte pattern of LGT8FX8D features downwards compatible byte read and write mode. LGT8FX8D internal intergrated FLASH is in the wide of 32 bit, which greatly benefit read/ write efficiency and FLASH erase/ write lifespan, we highly recommend 32 byte pattern.

E2PCTL 模拟 E2PROM 接口算法

E2PCTL analogy E2PROM interface algorithm

我们知道， FLASH 存储器在写之前必须先擦除，而擦除操作是以页面为单位的。LGT8FX8P 内置 FLASH 存储器一个页面的大小为 1K 字节。因此为了更新页面中的一个字节数据，也需要首先擦除掉整个页面的数据， 然后更新目标地址数据，并同时恢复页面中其他字节的数据，整个操作不仅仅耗时，也同时带来因电源意外丢失数据风险。

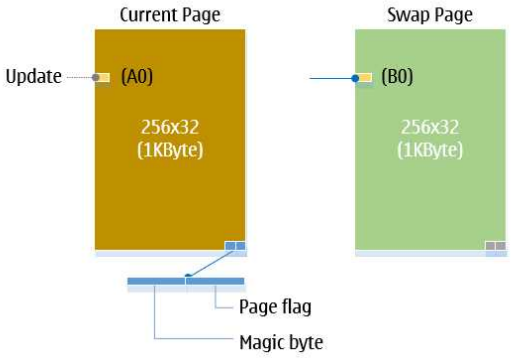
All know that Flash register must be erased before write, and erase operation unit is page. Internal FLASH register is 1K byte per page. So to update 1 byte data in one page, before update target address data it is a must to erase date of the whole page, meanwhile resume other byte data of the page. Such operation is not only time-consuming, but also risk data loss caused by power cut-off.

E2PCTL 内部采用页交换算法实现模拟 E2PROM。页交换算法模式可以保证在执行页擦除操作时， 不会因为掉电等意外情况导致原有数据的丢失。同时也交换算法使用 2 个页面空间互为交换的方式交替使用， 也增加了模拟 E2PROM 空间的使用寿命。

在效率方面， E2PCTL 控制器实现了一种连续数据更新模式，减少了因反复更新数据带来的重复擦写过程。在实现方面， E2PCTL 对每一个页面单独管理，并占用一个页面最后 2 个字节作为页面状态的信息。因此用户在使用大于 1K 的 E2PROM 模拟空间时，需要注意地址跨过 1K 空间的特殊处理。因为每 1K 空间的最后 2 个字节保留给 E2PCTL 使用，用户无法对这 2 个字节的空间进行正常的读写。

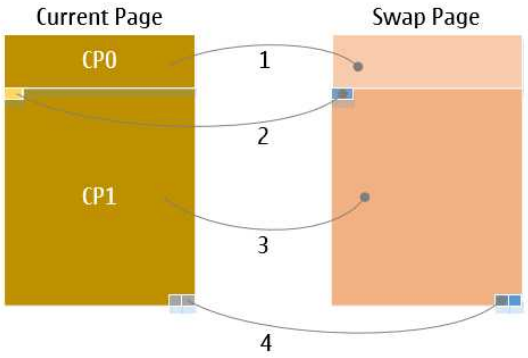
E2PCTL use page exchange algorithm to analog E2PROM, this kind of algorithm ensure that original data will not loss because of accidents such as power off during page erase operation. Meanwhile page exchange algorithm alternatively use 2 page space, which increase the lifespan of analog E2PROM memory.
In the aspect of efficiency, E2PCTL controller features a continuous data update pattern, this can reduce repeated erase/write because of repeated data update. In aspect of execution, E2PCTL manage individual page, taking up the last 2 byte of one page as page status information. Note that user must skip 1K memory when E2PROM analog is bigger than 1K. Because the last 2 byte of each 1K memory is kept for E2PTCL, so user cannot read and write this 2 byte memory as normal.

下图为 E2PCTL 基于页交换算法的逻辑示意图：



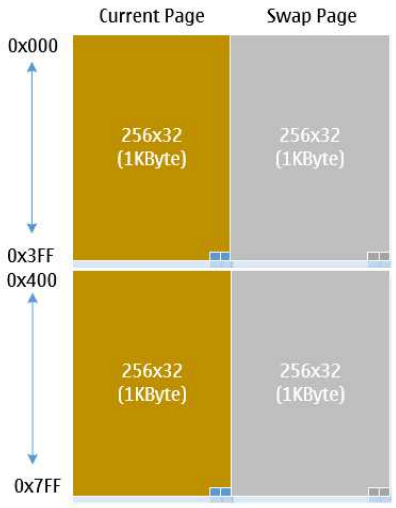
如图所示，E2PCTL 内部使用 2 个页面模拟一个页面大小的 E2PROM 空间。这两个页面 一个被标记为当前页面，另外为交换页。E2PCTL 使用页面最后 2 个字节存储页面信息。当 我们需要更新页面中的某一个字节时，比如上图中的 A0 字节。首先，我们不会擦除当前页 面，而是擦除交换页。然后将当前页面分为 3 个部分操作。首先是在 A0 之前的数据，我们 把这部分空间成为 CP0，然后是 A0 之后的数据，这部分空间为 CP1。
E2PCTL 会根据用户配 置，将 CP0 对应的数据复制到交换页的对应地址，然后将需要更新的数据写到交换页对应的 地址上(B0)，最后是复制 CP1 的数据到交换页。 完成上述操作后，数据已经完成交换，但并没有更新页面状态。因此如果在此之前发生 掉电或者其他异常，本次更新操作因为并没有完成，之前的数据并不会被破坏，保证了数据 的完整性。如果一切顺利，E2PCTL 会在 CP1 交换数据的最后，将更新的页面状态写到之前 的交换页面的页面信息中，实现当面页面的更换。此后，交换页面成为当前页。 E2PCTL 页面交换过程如下图所示(1->2->3->4)：

As above shown, E2PCTL use 2 page to analog 1 page 2EPROM memory . One page is marked as current page, one is swap page. E2PCTL use the last 2 byte to store page information. When we need to update some byte of the page, saying A0 byte as above picture, firstly we will erase swap page instead of current page. We make 3 operations for current page. For data before A0, we name it CP0, data after A0 is CP1. Based on user configuration , E2PCTL copy data in CP0 to corresponding location in swap page, than to write data, that need to be updated, in corresponding location (B0) in swap page, at last is to copy data in CP1 to swap page. After the above operation, data exchange is finished, but page status is not yet updated. If power off or other abnormal situation happen before page update, the update operation is actually not yet finished, while the odd data is not damaged, complete data is ensured.
If everything goes well, after data swap of CP1, updated page status will be written in previous swap page, than exchange of current page is finished, afterwards sway page is current page.



当系统配置的 E2PROM 模拟空间大于 1K 时，E2PCTL 还是以页面为最小单位实现 E2PROM 空间的模拟算法。比如如果用户配置了 2K 的 E2PROM 区域，实际上 E2PCTL 将会占用 4 个页 面（4K）的空间。其中 2 个页面为一组，用于实现模拟一个页面大小的 E2PROM 空间

If E2PROM analog memory assigned by system is smaller than 1K, E2PCTL will still use page as unit to realize analog algorithm of E2PROM. For example, user assign 2K for E2PROM, actually E2PCTL will take up 4 page memory (4K), among which 2 page is one group to be used to analog 1 page of E2PROM memory.



需要注意的是用户配置的 2K 字节的 E2PROM 空间并不连续，因为每个页面的最后 2 个 字节将会被用来保存页面状态信息。

Note that 2K byte of E2PROM assigned by user is not continuous, because the last 2 byte of each page is used to save page status information.

E2PCTL 连续编程模式

E2PCTL Continuous programming mode

由于通过 E2PCTL 更新会导致页面交换，页面交换过程中将会对交换页进行擦除，页擦 除不仅耗时，也同时会增加 FLASH 寿命的损耗。因此 E2PCTL 增加了连续写模式。在连续写 模式中，用户可以连续更新 E2PROM 区域，只有在连续地址的最后，才会进行页交换操作，对于需要连续更新一整块数据的应用， 连续模式更加有效。

连续编程模式 E2PCTL 控制寄存器 ECCR 的 SWM 位使能。连续模式使能后，后续的写操 作将直接将数据写到交换页对应的地址上，在 SWM 模式下， 写操作不会执行 CP0/1 区域数 据复制操作。在写最后一个字节前，软件通过 SWM 禁止连续模式，然后执行写， 此后 E2PCTL 将执行完整的 CP0/1 复制操作，并更新页面状态信息。

Using E2PCTL to update will cause page swap, during which swap page will be erased. Page erase is not only time consuming, but also reduce lifespan loss of FLASH. So continuous write mode is featured. In this mode, user can update E2PRROM area continuously, page swap operation is only executed at the last of continuous location.Continuous mode is more useful for continuous update of a whole lot data. Continuous programming

mode, E2PCTL control register ECCR enable SWM-bit. Enable bit after continuous mode, following write operation will directly write data in corresponding location of swap page. In SWM mode, write operation will not copy CP0/1 data. Before writing the last byte, software will prohibit continuous mode through SWM, than to execute write operation. Afterwards E2PCTL will execute complete copy of CP0/1, while update page status information.

E2PCTL 读写 FLASH 程序空间 通过 E2PCTL 控制器，可以实现对程序 FLASH 空间的读写访问。与模拟 E2PROM 不同的 是，通过 E2PCTL 对程序 FLASH 空间的访问完全需要软件控制。步骤如下： 1. 擦除目标页面，更新数据前需要首先擦除目标页面，页面地址通过 EEAR 寄存器给 出。对 FLASH 页面的擦除命令控制，请参考 EECR 寄存器的定义； 2. 写程序 FLASH 空间必须以 32 位为最小单位。通过 E2PD0~3 设置数据； 3. 目标地址由 EEAR 寄存器给出，地址 EEAR[1:0]将会被忽略；

通过 E2PCTL 读写程序 FLASH 空间，可以实现在线程序更新(IAP)功能，在一些需要现场 更新应用数据以及需要提供产品自定义更新的应用中，非常有用。

E2PCTL read and write FLASH program memory. Via E2PCTL controller, it features read/write access to program FLASH memory. Different from analog E2PROM, this access depend completely on software. Steps as belows:

1: Erase target page. Before data update, it is a must to erase target page, page location is provided by EEAR register. About erase instruction control on FLASH page, please refer to definition of EECR register.

2: The min. Unit of programming FLASH memory writing must be 32 byte. Setting data via E2PDO~3.

3: Target location is provided by EEAR register, location EEAR[1:0] will be ignored.

By using E2PCTL to read/write programming FLASH memory, in-system programming system function is realize. This is very useful for situation that need on-spot application data updating and self-definition updating must be provided.

E2PCTL 接口操作流程

E2PCTL 控制器是工作主要通过 4 个寄存器实现， 分别为 E2PCTL 控制状态寄存器 EECR、 ECCR；数据寄存器 EEDR(E2PD0~E2PD3)以及地址寄存器 EEAR(EEARL/EEARH)。

ECCR 寄存器用于设置 E2PCTL 的工作状态，大部分状态需要在 E2PCTL 工作前设置完成， 这个过程一般在系统初始化过程中实现。

ECCR 寄存器中的 SWM 位用于使能连续写模式， 这个控制位需要在实现连续写操作过程中设置。

EECR 寄存器用于控制选择操作类型，用于选择操作指令，比如设置读、擦除命令。

EEDR 寄存器用于 8 位字节模式接口，E2PD0~3 用于 32 位模式的读写操作；

EEAR 寄存器用于设置读，写的目标地址，也用于设置页擦除操作的页地址。页地址是 已页位单位对齐的，一页的大小为 1K 字节，需要注意 EEAR 指定的地址是字节地址。

E2PCTL Interface Operation Flow

E2PCTL register feature 4 register, that is E2PCTL status register EECR, ECCR; Data register EEDR(E2PD0~E2PD3) and address register EEAR(EEARL/EEARH)

ECCR register is used to set up status of E2PCTL. Most status must be set up before E2PCTL start to work. This process is realized normally during system initiation.

SWM bit of ECCR register is used to enable continuous write model. This control bit must be set up during continuous write.

EECR register is used to choose operation type, operation instructions such as set up write, erase instruction.

EEDR register is used in 8 bit byte mode interface,E2PD0~3 is used for 32 bit read/write operation.

EEAR register is used to set up target address for read and write, also used to set up page address for page erase operation. Page address is aligned by unit of page, one page is 1 K byte, note that appointed address

by EEAR is byte address.

通过 E2PCTL 接口 访问 FLASH 程序 空间：

通过 E2PCTL 接口可以实现对 FLASH 程序空间的读写和擦除。对 FLASH 空间的读写仅支 持 32 位访问宽度。擦除操作以页位单位，每页的大小的 1K 字节(256x32)。

在写 FLASH 程序空间之前， 首先擦除目标地址所在的页面。E2PCTL 写 FLASH 程序空间 不支持连续模式， 用户需要按顺序完成写操作。以下为擦写 FLASH 程序空间的流程：

Access to FLASH programming memory via E2PCTL interface

Through E2PCTL interface, it features read/write and erase FLASH programming memory. Read/write of FLASH memory support only access of 32 bit wide. Erase operation take page as unit, each page is 1K byte (256x32).

Before writing FLASH programming memory, at first page of target address must be erased. Continuous mode is not supported when E2PCTL write FLASH programming memory, user need to finish write operation by order. Below is the flow of erase FLASH programming memory:

1. 程序 FLASH 页 擦除操作 ◠

- 设置 EEAR[14:0]为需要擦除的目标页地址，程序 FLASH 一页大小为 1K 字节， 因此 EEAR[14:10]将作为页地址，EEAR[9:0]设置为 0 ◠
- 设置 EEPM[3:0] = 1X01，其中 EEPM[2]可设置为 0 或 1 ◠
- 设置 EEMPE = 1， 同时 EEPE = 0 ◠ 在四个周期内，设置 EEPE = 1，启动程序 FLASH 擦除流程

1. Erase FLASH page

- Set EEAR[14:0] as target page address that need to be erased, programming FLASH page is 1K byte. So to set EEAR[14:10] as page address, EEAR[9:0] as 0
- Set EEPM[3:0] = 1X01, EEPM[2] can be 0 or 1
- Set EMPE = 1 while EEPE = 0,
- In 4 clock cycles, set EEPE = 1, enable programming FLASH erase flow.

2. 程序 FLAS H 编程操作 ◠

- 写 E2PD0~3，准备 32 位编程数据 ◠
- 设置 EEAR 为目标地址，此处地址为 4 字节对齐 ◠
- 设置 EEPM[3:0] = 1X10， 其中 EEPM[2]可设置为 0 或 1 ◠
- 设置 EEMPE = 1，同时 EEPE = 0 ◠
- 在四个周期内，设置 EEPE = 1，启动 FLASH 编程流程

3. FLASH Programming operation

- Write E2PD0~3, prepare 32 bit programming data
- Set EEAR as target address, here address should be 4 byte aligned
- Set EEPM[3:0] = 1X10, EEPM[2] can be 0 or 1
- Set EEMPE = 1 while EEPE = 0
- In 4 clock cycles, set EEPE = 1, enable FLASH programming flow

通过 E2PCTL 接口 访问 E2PROM 模拟 空间：

Access to E2PROM analog memory via E2PCTL interface

E2PCTL 控制器通过模拟 E2PROM 接口逻辑访问数据 FLASH 空间。模拟 E2PROM 支持 8 位、16 位以及 32 位数据宽度的读写访问。8 位字节模式对 E2PROM 接口具有更好的兼容性。32 位模式有利于提高存储效率和 FLASH 的使用寿命，因此 32 位读写模式为建议的读写模 式。E2PROM 模拟接口支持连续读写模式，在需要一次更新多个连续地址的数据应用中，优 势明显，建议采用。

E2PCTL controller logic access to data FLASH memory by analog E2PROM. Analog E2PROM support read/write access in wide of 8 bit, 18 bit and 32 bit. 8 bit byte mode is highly compatible with E2PROM interface. 32 bit mode is useful to enhance storage efficiency and FLASH lifespan. So 32 bit read/write mode is recommend. E2PROM analog interface support continuous read/write mode and in case that need to update several continuous address at one time, E2PROM analog interface is of more advantage and recommend.

对于 LGT8F88P/168P，数据 FLASH 为独立的存储空间。无需通过 ECCR 寄存器配置和使 能 FLASH 数据空间。LGT8F328P 并没有独立的数据 FLASH 空间，数据 FLASH 与程序 FLASH 共 享 32K 字节 FLASH 空间。需要通过 ECCR 寄存器使能数据 FLASH 分区功能，并通过 ECCR 寄存器的 ECS[1:0]位配置数据 FLASH 的大小。配置生效后，其他使用方法与 LGT8F88P/168P 相同。

For LGT8F88P/168P, data FLASH is dependant storage memory, no need to configure and enable FLASH data memory through ECCR register. While LGT8F328P has no dependant storage memory, so data FLASH and program FLASH share 32K memory. DATA flash partitioning function is enabled by ECCR register, and data FLASH volume is configured by ECS[1:0] bit of ECCR register. Once configuration is effected, other usage is same as LGT8F88P/168P.

FLASH 控制器在实现 E2PROM 接口时，内部已经实现了在必要时自动擦除数据 FLASH 的 逻辑，所以 EPROM 擦除命令是可选的，这个命令只在用户需要单独执行擦除时使用。

When FLASH controller features E2PROM interface, logic of automatically data FLASH erase when necessary is realized internally. So EPROM erase instruction is optional, which is only used when user need to execute erase exclusively.

EECR 寄存器控制 FLASH 的擦/写时序，包括程序 FLASH 和 E2PROM。具体的操作类型需要通 过 EECR 寄存器的 EEPME 和 EEPM[3:0]设定。对 E2PROM 的读操作比较简单，

在设置好目标地 址和模式后，写 EERE 位即将目标地址对应的 32 位数据读入 FLASH 控制器内部，用户可以通 过 EEDR 寄存器读取感兴趣的字节 。FLASH 控制器并没有实现对程序 FLASH 空间的读操作， 用户可以方便的使用 LPM 或者通过程序 FLASH 在数据统一映射空间的地址处使用 LD/LDD/LDS 指令读取。

FLASH Erase/write timing of EECR register controller include program FLASH and E2PROM. Specific operation type need to be set up via EEPME and EEPM[3:0] of EECR register. Read operation of E2PROM is very simple,

After setting up target address and pattern, write EERE bit will read 32 bit data of target address into FLASH controller, than user can read and fetch interesting byte via EEDR register.

FLASH controller do not execute read operation to program FLASH memory, it is convenient for user by using LPM or via locations where program FLASH map in DATA mapping memeory to read and fetch LD/LDD/LDS instruction.

1. 8 位模式，编程 E2PROM

- 设置目标地址到 EEARH/L 寄存器
- 设置新的数据到 EEDR 寄存器
- 设置 EEPM[3:1] = 000，EEPM[0]可设置为 0 或 1
- 设置 EEMPE = 1， 同时 EEPE = 0
- 在四个周期内，设置 EEPE = 1

当设置完成后，FLASH 控制器将启动编程操作，编程期间 CPU 将保持在当前的指令地址上，直到操作完成后才会继续运行。在编程过程中， 如果需要擦除数据 FLASH，FLASH 控制器将会自动启动擦除流程。

1. 8 byte mode, programming E2PROM

- Set target address to EEARH/L register
- Set new data to EEDR register
- Set EEPM[3:1] = 000, EEPM[0] can be 0 or 1

- Set EEMPE = 1 while EEPE = 0
- In 4 clock cycles, set EEPE = 1

2. 32 位模式，编程 E2PROM

- 通过 E2PD0~3，准备 32 位数据
- 设置目标地址到 EEARH/L 寄存器。注意这里是字节对齐的地址，FLASH 控制器用 EEAR[15:2]作为访问 FLASH 的地址。
- 设置 EEPM[3:1] = 010，EEPM[0]可设置为 0 或 1
- 设置 EEMPE = 1， 同时 EEPE = 0
- 在四个周期内，设置 EEPE = 1

2. 32 bit mode, programming E2PROM

Prepare 32 bit data via E2PD0`3

- Set target address to EERAH/L. Note that here it is the address of byte alignment. Flash controller use EEAR[15:2] as address access to FLASH.
- Set EEPM[3:1] = 010, EEPM[0] can be 0 or 1
- Set EEMPE = 1 while EEPE = 0
- In 4 clock cycles, set EEPE = 1

3. 8 位模式，读 E2PROM

- 设置目标地址到 EEARH/L 寄存器
- 设置 EEPM[3:1] = 000
- 设置 EERE = 1 启动 E2PROM 读操作
- 等待 2 个周期 (执行两个 NOP 操作)
- 目标地址对应的数据被更新到 EEDR 寄存器

3. 8 bit mode, read E2PROM

- Set target address to EEARH/L register
- Set EEPM[3:1] = 000
- Set EERE = 1, enable E2PROM read operation
- Wait for 2 cycles(execute two NOP operation)
- Data corresponding to target address is updated to EEDR register.

4. 32 位模式，读 E2PROM

- 设置 EEARH/L 为目标地址，地址为 4 字节对齐
- 设置 EEPM[3:1] = 010，开启 32 位接口模式
- 设置 EERE = 1，启动 E2PROM 读操作
- 等待 2 个系统时钟周期 (执行两个 NOP 指令)

4. 32 bit mode, read E2PROM

- Set EEARH/L as target address, address is 4 byte alignment
- Set EEPM[3:1] = 010, open 32 bit interface mode
- Set EERE = 1, enable E2PROM read operation
- Wait for 2 system clock cycle (execute two NOP instruction)

E2PCTL 访问模拟 E2PROM 空间，支持连续编程模式，连续访问模式对于需要一次更新一个数据块的应用非常高效，也有利于提高 FLASH 的使用寿命。连续编程模式仅支持 32 位宽度的数据编程操作。连续访问模式通过 ECCR 寄存器的 SWM 位使能。SWM 使能后，接下来通过 E2PCTL 写模拟 E2PROM 空间的操作都在连续编程模式。在连续编程模式下， E2PCTL 控制器会根据目标地址内的数据情况自动处理换页。但在连续编程模式过程中如果发生换页，控制器在连续编程过程中，不会自动将 CP0/1 区域的数据交换，也不会更新页面信息。

当连续编程到最后一次操作前，通过清零 SWM 位关闭连续编程模式，然后在非 SWM 模式下启动最后一次编程操作，编程结束后，E2PCTL 会自动

将 CP0/1 区域的数据复制到交换页，并更新交换页的信息，使之成为当前有效页，从而完成整个连续编程操作

E2PCTL access to analog E2PROM memory, and support continuous programming mode, which is very useful in case a data module need to be updated 1 time, and also can increase the lifespan of FLASH. Continuous programming mode only supports 32 bit wide data programming operation. SWM bit of ECCR register enable continuous access mode, after that under continuous programming mode to write analog E2PROM memory via E2PCTL.

Under continuous programming mode, E2PCTL controller will handle page sway automatically according to data of target address, if page swap, controller will neither automatically swap data in CP0/1, nor update page information.

Before the last operation of continuous programming, continuous programming mode is closed by clearing SWM bit. Than under non-SWM mode, enable the last programming operation. After programming, E2TCTL will automatically copy data of CP0/1 to swap page and update swap page information, so that it can become current page, than a complete continuous programming operation is finished.

5. 连续编程模式操作流程：

1. 通过 ECCR 配置数据 FLASH 的大小，并使能 SWM 位

2. 使用 32 位模式编程模拟 E2PROM 区域

3. 如果不是最后一次操作，回到步骤 2 继续编程下一个数据

4. 如果达到最后一次编程，首先通过 SWM 禁止连续编程模式，然后使用步骤 2 的操作流程完成最后一次编程

5. Flow of continuous programming mode

1. **Via ECCR to configure data Flash volume, enable SWM bit**
2. **Via 32 bit mode, programming to analog E2PROM area**
3. **If not the last operation, return to step 2 to continue programming next data**
4. **If it is the last operation, prohibit continuous programming mode via SWM, than to finish the last programming by using step 2.**

E2PCTL 高效 FLASH 数据管理

E2PCTL 控制器除了实现连续编程模式，也可以通过 ECCR 寄存器的 CP0/1 位对页交换过程数据交换复制进行独立控制。ECCR 寄存器的 CP0/1 分别用于控制页交换过程中对于当前页面中 CP0/1 区域数据的交换操作。

清零 CP0/1 位， 在页交换过程中将不会交换当前页中对应区域的数据。本节提供的一种高效管理方法，将会利用这一特性。

在 FLASH 数据更新过程中， 最为耗时的操作发生在交换页擦除过程。因此我们可以寻址一种最大限度减小页擦除次数的数据管理方法，既能提高编程效率，也能减少寿命损耗。这里我们提供一种参考算法，适用于基于数据块数据管理应用：

1. 假定用户数据只是一个完整的数据块，数据块大小 4 字节的整数倍；

2. 每次数据更新将会更新一个完整的数据块

3. 数据块信息除了存放用户数据， 还需要存放一个块管理信息

E2PCTL enhanced FLASH data management

E2PCTL controller features continuous programming mode and it independently control on the swap and copy of data via CP0/1 page sway of EECR register. CP0/1 of ECCR register are separately used to control data swap operation of current page CP0/1 during page swap.

If clearing CP0/1 bit, it will not swap data in corresponding area of current page. In this section we introduce an efficient method to make use of this feature.

During FLASH data updating, the most time consuming operation is to erase swap page. So we have to address a data management method to reduce swap page erase time as much as possible, which is not only efficient and also increase lifespan. So here we provide a reference algorithm, that is suitable for data

management based on data block.

1. Assuming user data is only a complete data block, size of data block is times of 4 byte
2. Each data update will update a complete data block
3. Data block info save not only user data, but also block management info.

以上三个条件下，我们可以充分利用 E2PCTL 的连续编程模式和自动页交换机制，实现一个高效率的 FLASH 数据管理方法。由于是每次更新的数据为一个相同大小的数据块， 并且每块数据结构中保存有指向下一块数据的地址信息，因此我们可以每次更新数据时按地址顺序编程 FLASH，无需做 CP0/1 的数据复制。同时由于每次都是更新数据到一个已擦除的区域，也不会发生页擦除。当数据写完最后一块，其结构信息指向的下一块数据区回到页的起始地址。 此后再发生数据写操作，E2PCTL 将会启动一次页擦除过程，并更新当前活动页面。

Based on above 3 conditions, we can achieve a high efficient FLASH data management by making use of E2PCTL continuous programming mode and automatic page swap. We can program FLASH by order of address in each data updating, no need to copy data of CP0/1, because data to be updated each time is data block of same size and each data construction save address to appoint next data block.
Meanwhile each data update is to an area that has been erased, so page will not be erased. When write the last data block, its construction information appoint to the next data area which return to page starting address. Afterwards for each data write operation, E2PCTL will enable one page erase, and update current page.

FLASH 操作的保护措施

如果 VCC 电压偏低，FLASH 的擦写操作可能会因为电压太低而发生错误。FLASH/数据在低压下的擦写操作错误可能由两种原因。

首先， 正常的 FLASH 擦写操作需要一个最小工作电压，低于这个电压，操作将会失败而导致数据发生错误。第二个原因，是内核运行在某一频率下，也同样需要一个最小电压要求，当低于这个电压， 将会导致指令执行出错，从而使得 FLASH 的操作发生错误。可以通过下面简单的方法避免类似问题：

在供电电压较低时，让系统进入复位状态。这可以通过配置内部的低压检测电路(VDT)实现。如果 VDT 检测到当前的工作电压低于设置的阀值， VDT 将会输出一个复位信号。如果 VDT 的阀值不能满足应用的需要，可以考虑在外部增加一个复位电路。

Protection measurement for FLASH operation

If VCC voltage is too low, FLASH erase/ write operation will be wrong. Flash/Data corruption can be caused by two situations when the voltage is too low
First, a regular FLASH write/erase sequence requires a minimum voltage to operate correctly. Secondly, the CPU itself can execute instructions incorrectly, if the supply voltage is too low. FLASH data corruption can easily be avoided by following this design recommendation:
Keep the AVR RESET active during periods of insufficient power supply voltage. This can be done by enabling the internal low voltage detection circuit (VDT). If the VDT detect voltage level is lower than designed threshold value, VDT send out a RESET signal. If VDT threshold voltage cannot meet demand, we recommend a external RESET circuit.

寄存器描述

Register Description

FLASH 地址寄存器- EEARH/EEARL

EEARH and EEARL – The EEPROM Address Register

EEARH/EEARL	
EEARH: 0x22 (0x42)	Default value: 0x0000

EEARL: 0x21 (0x41)		
bits	EEAR[15:0]	
R/W	R/W	
Bit definition		
[7:0]	EEARL	EFLASH/E2PROM access address 8 bit lower
[14:8]	EEARH	EFLASH/E2PROM access address 7 bit higher
[15]	-	Kept as blank, no use

当使用 E2PCTL 控制器访问程序 FLASH 区域时，EEAR[14:2]用作访问以 4 字节对齐的整个程序空间。EEAR[1:0]只在访问数据寄存器 EEDR 时使用。具体请参考下面关于 EEDR 数据寄存器的描述。E2PCTL 控制器支持 8/16/32 位模式，无论是哪一种模式，此处的 EEAR 都是以字节对齐寻址。

When using E2PTL controller to access programming FLASH memory, EEAR[14:2] access the whole programming memory aligned with 4 byte. EEAR[1:0] is only used when access to data register EEDR. For details, please refer to description about EEDR data register. E2PCTL controller support 8/16/32 bit mode, whatever mode, EEAR address via byte alignment.

FLASH 数据寄存器- EEDR/E2PD0

FLASH data register- EEDR/E2PD0

EEDR/E2PD0 – FLASH/E2PROM data register 0		
EEDR/E2PD0: 0x20 (0x40)	Default value: 0x00	
bits	EEDR[7:0]	
R/W	R/W	
Bit definition		
[7:0]	EEDR E2PD0	E2PCTL Data register When in 16/32 bit mode，used to save lowest byte

FLASH 数据寄存器- E2PD1

FLASH data register- E2PD1

E2PD1 – E2PCTL– data register 1		
E2PD1: 0x5A	Default value: 0x00	
bits	EEDR1[7:0]	
R/W	R/W	
Bit definition		
[7:0]	E2PD1	For 16 bit mode, used to save high 8 bit of 16 bit data For 32 bit mode, used to save high 8 bit that is smaller than 16 bit data

FLASH 数据寄存器- E2PD2

FLASH data register- E2PD2

E2PD2 – FLASH data register 2		
E2PD2 : 0x57	Default value: 0x00	
bits	E2PD2 [7:0]	
R/W	R/W	
Bit definition		
[7:0]	E2PD2	For 32 bit mode, used to save low 8 bit that is higher than 16

		bit data

FLASH 数据寄存器- E2PD3

FLASH data register - E2PD3

E2PD3 – FLASH data register 3		
E2PD3 : 0x5C	Default value: 0x00	
bits	E2PD23[7:0]	
R/W	R/W	
Bit definition		
[7:0]	E2PD3	For 32 bit mode, used to save high 8 bit that is higher than 16 bit data

FLASH 模式控制寄存器- ECCR

FLASH mode control register - ECCR

ECCR – FLASH/E2PROM configuration register								
ECCR: 0x36 (0x56)	Default value: 0x0C							
bits	WEN	EEN	ERN	SWM	CP1	CP0	ECS1	ECS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	1	1	0	0
Bit definition								
[7]	WEN	ECR write enable control Before modify ECCR, must write WEN as 1, than within 6ycles, UPDATE ECCR register						
[6]	EEN	E2PROM enable, only is valid for LGT8F328P 1: Enable E2PROM analog, will reserve partial space from 32KFLASH 0: Prohibit E2PROM analog, all 32KFLASH used for program memory						
[5]	ERN	Write 1 to reset E2PCTL controller						
[4]	SWM	Continuous write mode, only suitable to analog E2PROM controller operation						
[3]	CP1	Page SWAP CP1 Enable control						
[2]	CP0	Page SWAP CP0 Enable control						
[1:0]	ECS[1:0]	E2PROM memory configuration 00：1KB E2PROM, 30KB programming FLASH 01：2KB E2PROM, 28KB programming FLASH 10：4KB E2PROM, 24KB programming FLASH 11：8KB E2PROM, 16KB programming FLASH						

FLASH 访问控制寄存器- EECR

FLASH access control register-EECR

EECR – FLASH/E2PROM control register								
ECCR: 0x1F (0x3F)	Default value: 0x00							
bits	EEPM3	EEPM2	EEPM1	EEPM0	EERIE	EEMPE	EEPE	EERE
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

Bit definition						
[7:4]	EEPM[3:0]	ECR write enable control				
		[3]	[2]	[1]	[0]	Mode description
		0	0	0	x	8 bit mode read/write E2PROM (Default)
		0	0	1	x	16 bit mode read/write E2PROM
		0	1	0	x	32 bit mode read/write E2PROM
		1	x	0	0	ERPROM Erase (optional)
		1	x	0	1	Program FLASH Erase (page erase)
		1	x	1	0	Program FLASH programming
		1	x	1	1	Reset E2PROM/FLASH controller
[3]	EERIE	FLASH/E2PROM ready interrupt enable control. Write 1 enable, write 0 disable. If EEPE is automatically cleared by hardware, E2PROM ready interrupt is valid. During period of EPROM operation, interrupt will not be done.				
[2]	EEMPE	FLASH/E2PROM programming operation enable control EEMPE is used to control if EEPE is valid or not. When set EEMPE as 1 while EEPE as 0, than after 4 cycles, set EEPE as 1, programming operation is activated. Otherwise programming operation is invalid. 4 cycles later, EEMPE will be cleared automatically				
[1]	EEPE	FLASH/E2PROM programming operation enable bit				
[0]	EERE	E2PROM write enable bit，data becomes valid within 2 cyles				

通用 I/O 寄存器- GPIOR2

General I/O register- GPIOR2

GPIOR2 – General I/O register- 2		
GPIOR2: 0x2B (0x4B)	Default value: 0x00	
Bits	GPIOR2[7:0]	
R/W	R/W	
Initial value	0x00	
Bit definition		
[7:0]	GPIOR2	General I/O register- 2, used to save user-defined data

通用 I/O 寄存器- GPIOR1

General I/O register- GPIOR1

GPIOR1 – General I/O register- 1		
GPIOR2: 0x2A (0x4A)	Default value: 0x00	
Bits	GPIOR1[7:0]	
R/W	R/W	
Initial value	0x00	
Bit definition		
[7:0]	GPIOR1	General I/O register- 2, used to save user-defined data

通用 I/O 寄存器- GPIOR0

General I/O register- GPIOR0

GPIOR0 – General I/O register- 0		
GPIOR2: 0x1E (0x3E)	Default value: 0x00	
Bits	GPIOR0[7:0]	
R/W	R/W	
Initial value	0x00	
Bit definition		
[7:0]	GPIOR0	General I/O register- 2, used to save user-defined data

系统时钟与配置

System Clock and Clock Options

系统时钟分布

Clock Systems and their Distribution

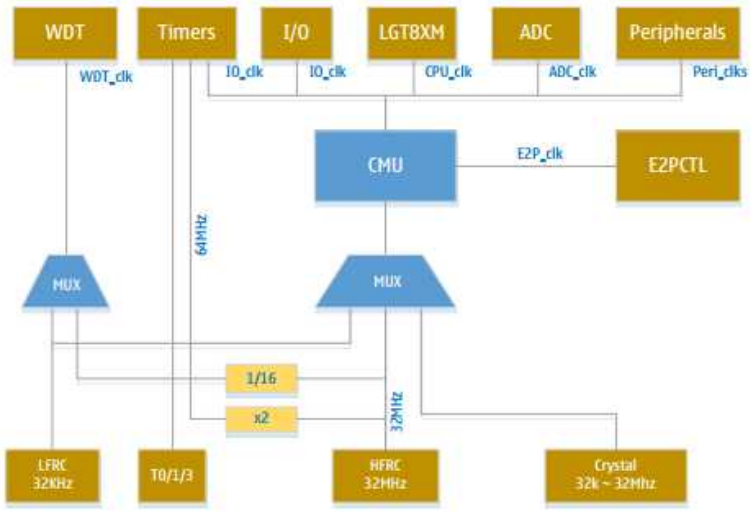
LGT8FX8P 支持多种时钟输入。系统可以工作在三种主要的时钟源，分别是内部 32KHz 可校准 RC 振荡器，内部 32MHz 可校准 RC 振荡器以及外部 400KHz ~ 32MHz 晶振输入。下图为 LGT8FX8P 时钟系统分布，CMU 是整个时钟管理的中心，负责系统时钟的分频，为不同的模块产生独立的时钟以及对时钟进行控制等等。一般的应用中，并不不要全部的时钟同时工作，为了减小系统功耗，系统功耗管理根据不同的休眠模式，关闭没有使用的模块时钟。

具体操作细节，请参考功耗管理相关章节。

LGT8FX8P support input of several clock. AVR can work with 3 main clock source, internal 32KHz adjustable RC oscillator, internal 32MHz adjustable oscillator and external 400KHz~32MHz crystal oscillator input. Below picture show distribution of LGT8FX8P clock.

CMU is the center of the whole clock system, in charge of frequency divider of clock system, make individual clock for different module and execute control on clock.

Normally, not all clock is required to work at the same time, to reduce system power consumption, power management start sleep mode as demand, to close clock that is not of use, detailed operation method, referring to section of power consumption management.



CPU_clk

用于驱动 LGT8XM 内核以及 SRAM 的运行。比如驱动通用工作寄存器，状态寄存器等。CPU 时钟停止后，内核将不会继续执行指令和进行计算。系统执行 SLEEP 指令进去休眠模式后，内核时钟将会被关闭。

The CPU clock is routed to parts of the system concerned with operation of the LGT8XM AVR and SRAM . Examples of such modules are the General Purpose Register File, the Status Register. Halting the CPU clock inhibits the core from performing general operations and calculations. AVR execute sleep instruction and enter into sleep mode, AVR clock is closed.

Peri_clk

用于驱动大部分外设模块，比如定时/计数器，SPI，USART 等。IO 时钟也用于驱动外部中断模块。当外设时钟因休眠而停止后，某些可以用了唤醒系统的外设部分工作在独立的时钟或异步模式。比如 TWI 的地址识别功能可以唤醒大部分休眠模式，此时的地址识别部分工作在异步模式。

Is used to drive most peripheral module, for example Timer/Counter, SPI, USART, etc. I0 Clock is also used to drive external interrupt module. When external clock is halted because of sleep, some peripherals, that can use wake up system, can work with independent clock or Async mode. For example, TWI Address identify function, which work under Async mode, can wake up most sleep mode.

E2P_clk

E2P_clk 时钟用于产生 FLASH 接口访问时序。E2P_clk 产生访问 E2PCTL 访问 FLASH 接口的时序。E2P_clk 固定来自内部 32MHz HFRC 振荡器的 32 分频(1MHz)。如果用户需要使用 E2PCTL 模块读写内部程序 FLASH 或者数据 FLASH 空间，需要提前使能内部 32MHz 振荡器。

E2P_clk clock is used to generate FLASH access timing sequence. E2P_clk generate timing sequence for access to E2PCTL and FLASH. E E2P_clk is eternally from 32 frequency divider (1MHz) from 32MHz HFRC oscillator. If user need to use E0PCTL module to read internal programming FLASH or data FLASH memory, it is a must to enable internal 32MHZ oscillator in advance.

Asy_clk

异步定时器时钟。

Asynchronous Timer Clock

定时/计数器可以直接使用外部时钟或晶振(32.768K)驱动。这种独立的时钟模式，可以在系统处理休眠模式时，定时器仍然保持运行。

The Asynchronous Timer clock allows the Asynchronous Timer/Counter to be clocked directly from an external

clock or an external 32kHz clock crystal. The dedicated clock domain allows using this Timer/Counter as a real-time counter even when the device is in sleep mode.

WDT _clk

内部看门狗定时器时钟源，可以配置选择内部 32KHz LFRC 振荡器，或者来自内部 32MHz HFRC 的 16 分频（2MHz）。系统上电后，看门狗默认时钟源为 32KHz LFRC 振荡器。

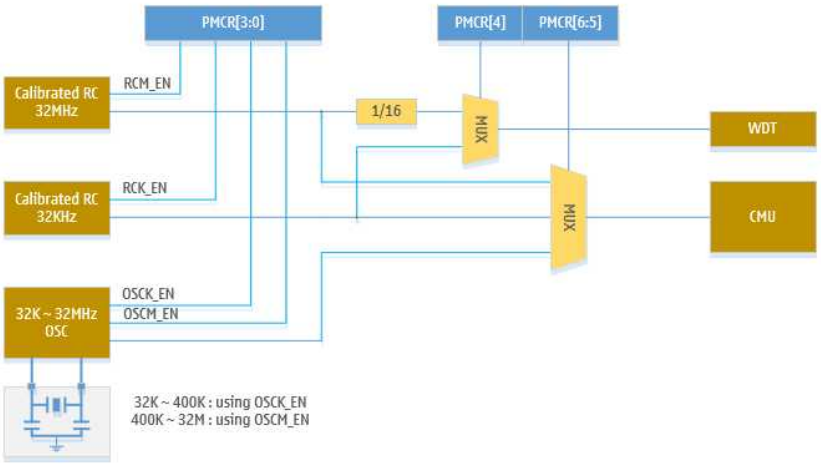
Internal Watch Dog Timer clock source, can be set up to choose internal 32KHz LFRC oscillator, or 16 frequency divide of 32MHz HFRC (2MHz). After power on., default clock source of WDT is 32KHz LFRC oscillator.

时钟源选择

LGT8FX8P 支持 4 种时钟源输入，用户可以通过 PMCR 寄存器实现对时钟源的使能控制 以及完成主时钟的切换。下面是 PMCR 的控制结构图：

Clock Source Choice

LGT8FX8P support 4 clock source, user can enable clock source via PMCR register, and realize swap of mater clock. Below picture shows PMCR control structure:



LGT8FX8P 内部 OSC 振荡器可以工作在高频和低频两种模式下，用户需要根据外接晶振 的实际大小控制内部 OSC 振荡器工作在正确的模式下。同样内部的 RC 振荡器也分为高频和 低频两种。PMCR 寄存器的最低 4 位用于控制这四种时钟源。控制关系如下：

LGT8FX8P internal OSC oscillator can work with both high and low frequency mode, according to size of external crystal oscillator, user chose the right mode that internal OSC oscillator work with. Meanwhile internal RC oscillator has two categories: high frequency and low frequency. The lowest 4 bit of PMCR register are used to control these 4 clock source. Control relations as below:

PMCR	Clock Source
PMCR[0]	32MHz RC Enable control, 1 enable, 0 close
PMCR[1]	32MHz RC Enable control, 1 enable, 0 close
PMCR[2]	400K ~ 32MHz OSC mode enable, 1 enable, 0 close
PMCR[3]	32K ~ 400K OSC mode enable, 1 enable, 0 close

LGT8FX8P 系统上电后，默认使用 32MHz RC 作为系统时钟源，内核工作在时钟源的 8 分 频(4MHz)。用户可以通过设置 PMCR 寄存器以及系统预分频寄存器(CLKPR)改变默认配置。

如果用户需要更改主时钟源配置，需要在切换时钟前保证切换后的时钟源处于稳定的工作状态。因此需要在切换主时钟源之前，通过 PMCR[3:0]使能所需时钟源，并等待到时钟稳 定后才能进行切换。

当用户切换主时钟到外部晶振时，虽然用户使能了外部晶振，但也不排除因配置错误或 晶振失效导致晶振无法起振。如果在此时切换到外部晶振，切换后系统将停止工作。因此， 从系统可靠性考虑，建议打开看门狗定时器，从软件设计的角度避免此类问题

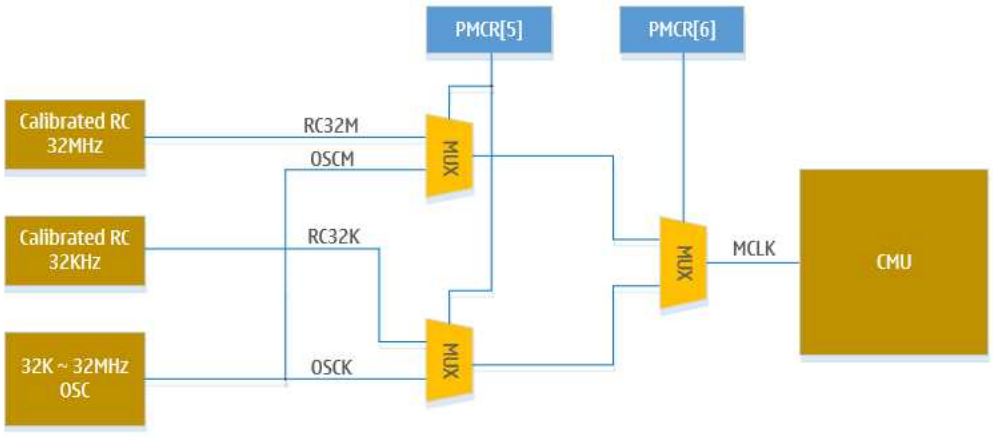
时钟源使能并等待稳定后，可以通过 PMCR[6:5]切换主时钟。其中 PMCR[5]用于选择是内 部 RC 振荡器和外部晶振，PMCR[6]用于选择高速时钟源和低速时钟源

When LGT8FX8P is power on, 32MHZ RC is considered as system clock source by default, core work at 8 frequency division domain (4MHz) of clock source. User can change default configuration via PMCR register and system clock prescaler register(CLKPR).

If user need to change configuration of master clock source, it is a must to ensure that clock source should be under stable working status before such change. Before change the clock source, to enable clock source via PMCR[3:0], only start to change till clock is stable.

When user switch master clock to external oscillator, it is possible that crystal oscillator cannot start to work even external crystal oscillator is enabled, reason could be configuration error or crystal oscillator is out of work, in this case if user switch to external crystal oscillator, system will stop working. So in consideration of stability, we suggest to activate watch dog timer and to avoid such issue from the aspect of software design.

Till clock source is enabled and stable, user can switch master clock via PMCR[6:5]. PMCR[5] is used to decide internal RC oscillator or external crystal oscillator, PMCR[6] is used to decide high speed or low speed clock source.



Master Clock Source

PMCR[6]	PMCR[5]	Master Clock Source
0	0	Internal RC oscillator (system default)
0	1	External 400K ~ 32MHz high speed oscillator
1	0	Internal 32KHz RC oscillator
1	1	External 32K ~ 400KHz low speed oscillator

时钟源控制时序

为保护 PMCR 寄存器被意外修改，对 PMCR 寄存器的修改需要严格安装指定的时序进行。PMCR 寄存器的最高位(PMCR[7])用于实现时序控制。用户在修改 PMCR 其他位之前，必须首 先要将 PMCR[7]置 1，在置 1 操作后的 6 个周期内，更改 PMCR 其他寄存器的值。6 个周期之 后，对 PMCR 的直接修改将失效。

Clock source timing sequence control

To protect PMCR register from modification by accident, modification to PMCR register should follow strictly timing sequence appointed. Highest bit (PMCR[7]) of PMCR register is used for timing sequence control. User must write PMCR[7] bit as 1 before modification of its other bit, than change value of other PMCR register within 6 cycles. After this 6 cycles, modification on PMCR will become invalid.

下面以切换到外部高速晶振为例，列出建议的操作步骤：

(1) 使能时钟源

设置 PMCR[7] = 1

在六个周期内，设置 PMCR[2] = 1，使能外部高速模式外部晶振

等待外部晶振稳定(等待时间因晶振不同而不同，一般 us 级等待即可)

(2) 切换主时钟源

设置 PMCR[7] = 1

在六个周期内，设置 PMCR[6:5] = 01，系统将工作时钟自动切换至外部晶振

执行几个 NOP 操作，提高稳定性(可选操作)

[注意]：在以上切换主时钟的操作中，要保证当前系统时钟正常工作，在切换到外部晶 振以后，才可以关闭之前的内部 RC 振荡器。

Taking example of switching to external high speed crystal oscillator, recommend steps is as below:
(1) Enable clock source
Set PMCR[7] = 1
Within 6 cycles, set PMCR[2] = 1, enable external high speed mode, external crystal oscillator
Wait till external crystal oscillator is stable (waiting period is different as crystal oscillator type, normally US level is enough)
Switch master clock source
Set PMCR[7] = 1
Within 6 cycles, set PMCR[6:5] = 01, system will switch working clock automatically to external crystal oscillator.
Execute several NOP operation, to improve stability (optional operation)
Note: Must ensure current system clock work normal when executing the above operation of switching master clock operation. Can only close previous internal RC oscillator after switching to external crystal oscillator,

系统时钟预分频控制

LGT8FX8P 内部有一个系统时钟预分频器，可以通过时钟预分频寄存器(CLKPR)进行控制。 这种功能可以用于当系统不需要非常高的处理能力时，减小系统功耗。预分频设置对系统支 持的时钟源都有效。时钟预分频能够影响到内核执行时钟以及所以同步外设。 当在不同的时钟预分频设置之间切换时，系统时钟预分频确保在切换过程中不会产生毛 刺，而已会保证不会有过高频的中间状态。分频切换是立即执行的，当寄存器改变生效后， 最多在 2~3 个当前系统时钟周期后，系统时钟就切换到了新的分频时钟。

为了避免对时钟分频寄存器的误操作，对 CLKPR 的修改也必须遵循一个特殊的时序流 程：
设置时钟预分频更改使能位(CLKPCE)为 1，CLKPR 其他所以位为 0 在四个周期内，把需要的值写入 CLKPS，同时 CLKPCE 写 0

在更改时钟预分频寄存器前， 需要禁止中断功能，以保证写时序能够完整的进行。

System Clock Prescaler

LGT8FX8P has a system clock prescaler, and the system clock can be divided by setting the ”CLKPR – Clock Prescale Register”. This feature can be used to decrease the power consumption when the requirement for processing power is low. This can be used with all clock source options, and it will affect the clock frequency of the CPU and all synchronous peripherals. When switching between different prescaler settings, the System Clock Prescaler ensures that no glitches occurs in the clock system. It also ensures that no intermediate frequency is higher than neither the clock frequency corresponding to the previous setting, nor the clock frequency corresponding to the new setting.
Prescaler swtiching is immediate once change is done on register. System clock will change to new prescaler clock no longer than 2~3 current system clock cycles.
To avoid unintentional changes of clock frequency, a special write procedure must be followed to change the CLKPR :
1. Write the Clock Prescaler Change Enable (CLKPCE) bit to one and all other bits in CLKPR to zero.
2. Within four cycles, write the desired value to CLKPS while writing a zero to CLKPCE.
3. Interrupts must be disabled when changing prescaler setting to make sure the write procedure is not interrupted

内部 RC 振荡器校准

LGT8FX8P 内部包含两个可校准 RC 振荡器，经过校准后，均可达到±1%以内的精度。其中 32MHz RC 默认用于系统工作时钟。 LGT8FX8P 出产前，内部 32MHz HFRC 和 32KHz LFRC 都进行了校准，并把校准值写入系 统配置信息区域。系统省电过程中，这些校准值将会被读入到内部寄存器中，通过寄存器实 现对 RC 频率的重新校准。 校准寄存器位于 IO 地址空间，用户程序可以读写。对于频率有特殊需求的应用，可以 通过修改校准寄存器方式调整内部振荡器的频率输出。修改校准寄存器不会改变出厂配置信 息，系统重新上电或者用户启动的配置位重新加载操作，校准寄存器将会恢复

到出厂设置。

RC Oscillator Calibration

LGT8FX8P has two RCC oscillator to be calibrated, its tolerance can be within ±1%. 32MHz RC is used on system working clock by default. Before delivery, 32MHz HFRC and 32KHz LFRC have been calibrated, and calibration value has been written into system configuration. In power-save mode, these calibration values will be read into internal register, than re-calibrate RC frequency via register. Calibration register locate IO address memory, user program can be read and write. For special requirement on frequency, to change register calibration method, frequency output of oscillator can be adjusted. While factory default is not changed by changing calibration register. Calibration register will restore factory default when system is re-start or user reload configuration bit.

寄存器定义

32MHz HFRC 振荡器校准寄存器- RCMCAL

Registration Definition

32MHz HFRC Oscillator Calibration Register- RCMCAL

RCMCAL – 32MHz HFRC Calibration Register		
RCMCAL: 0x66	Default value: factory default	
Bits	RCCAL[7:0]	
R/W	R/W	
Bit definition		
[7:0]	RCCAL	Once power on, RC calibration value of system configuration will replace register value

32KHz RC 振荡器校准寄存器- RCKCAL

32KHz RC Oscillator Calibration Register- RCKCAL

RCKCAL – 32KHz RC Oscillator Calibration Register		
RCKCAL: 0x67	Default value: factory default	
Bits	RCKCAL[7:0]	
R/W	R/W	
Bit definition		
[7:0]	RCKCAL	Writing calibration value to RCKCAL register to calibrate 32KHz RC Oscillator

时钟源管理寄存器- PMCR

Clock management register-PMCR

PMCR Clock management register							
PMCR: 0xF2				Default value: 0x03			
Bits	PMCE	CLKFS/ CLKSS	WCLKS	OSCKEN	OSCMEN	RCKEN	RCME
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit definition							
[0]	RCMEN	Internal 32MHz RC Oscillator enable control, 1 enable, 2 disable					
[1]	RCKEN	Internal 32KHz RC Oscillator enable control, 1 enable, 2 disable					
[2]	OSCMEN	External high frequency crystal oscillator enable control, 1 enable, 0 disable					
[3]	OSCKEN	External low frequency crystal oscillator enable control, 1 enable, 0 disable					

[4]	WCLKS	WDT Clock source choices: 0- Internal 32MHz HFRC oscillator 16 frequency division 1- Internal 32KHz LFRC oscillator
[5]	CLKSS	Master clock source choice, for clock source type, referring to section of clock source
[6]	CLKFS	Master source frequency control, for clock frequency contro, referring to section of clock source
[7]	PMCE	PMCR Register Change Enable bit, this bit must be set firstly before other bits. The other bits value should be set within 4 cycles.

主时钟预分频寄存器- CLKPR

Master Clock Prescale Register

CLKPR Master Clock Prescale Register								
CLKPR: 0x61					Default value: 0x03			
Bits	WCE	CKOEN1	CKOEN0	-	PS3	PS2	PS1	PS0
R/W	R/W	R/W	R/W	-	R/W	R/W	R/W	R/W
Bit Definition								
[3:0]	CLKPS	Clock Prescaler Select Bits						
		PS3	PS2	PS1	PS0	division factor		
		0	0	0	0	1		
		0	0	0	1	2		
		0	0	1	0	4		
		0	0	1	1	8(default value)		
		0	1	0	0	16		
		0	1	0	1	32		
		0	1	1	0	64		
		0	1	1	1	128		
		1	0	0	0	256		
		Other value				reserved		
[4]	-	Kept for no use						
[5]	CKOEN0	To set if system clock is output from PBO pin						
[6]	CKOEN1	To set if system clock is output from PE5 pin						
[7]	WCE	Use Clock Prescale to change clock control Before changing other bits of CLKPR register, must first set CKWEN to one, than to set other bits in the following 4 cycles. After 4 cycles, CKWEN is clear automatically.						

功耗管理

Power Management

概述

Overview

休眠模式通过关闭系统时钟以及时钟模块，从而减小系统功耗。LGT8FX8P 提供了非常 灵活多样的休眠模式和模块控制器，用户可以根据应用，实现最理想的低功耗配置。

LGT8FX8P 在进入休眠模式时，并不会自动关闭模拟功能模块，比如 ADC，DAC，比较器 (AC)，低电压复位模块(LVD)等等，软件需根据应用要求，

在进入休眠前关闭不需要的模拟功 能，并在系统唤醒后恢复正确的状态。

LGT8FX8P 支持多种休眠模式，其中包括 ADC 专用的噪声消除模式，用于消除 ADC 转换 过程中数字部分对 ADC 电源的干扰。除此之外，其他均为功耗控制模式，共分为五种：

Sleep modes enable the application to shut down system clock and clock modules, thereby saving power. The LGT8FX8P provides various sleep modes and module controller allowing the user to tailor the power consumption to the application's requirements.

In sleep modes, LGT8FX8P will not automatically disable function of analog module, such as ADC, DAC, AC(analog comparator), LVD (low voltage detector) and so son. As per requirement, software will disable unused analog function and resume to correct status after system wake up before going into sleep mode.

LGT8FX8P support several sleep modes, including ADC Noise Reduction which is used to reduce interrupt to ADC power by digital part during the period of ADC conversion. Other modes are power consumption control mode, there are 5 modes as below:

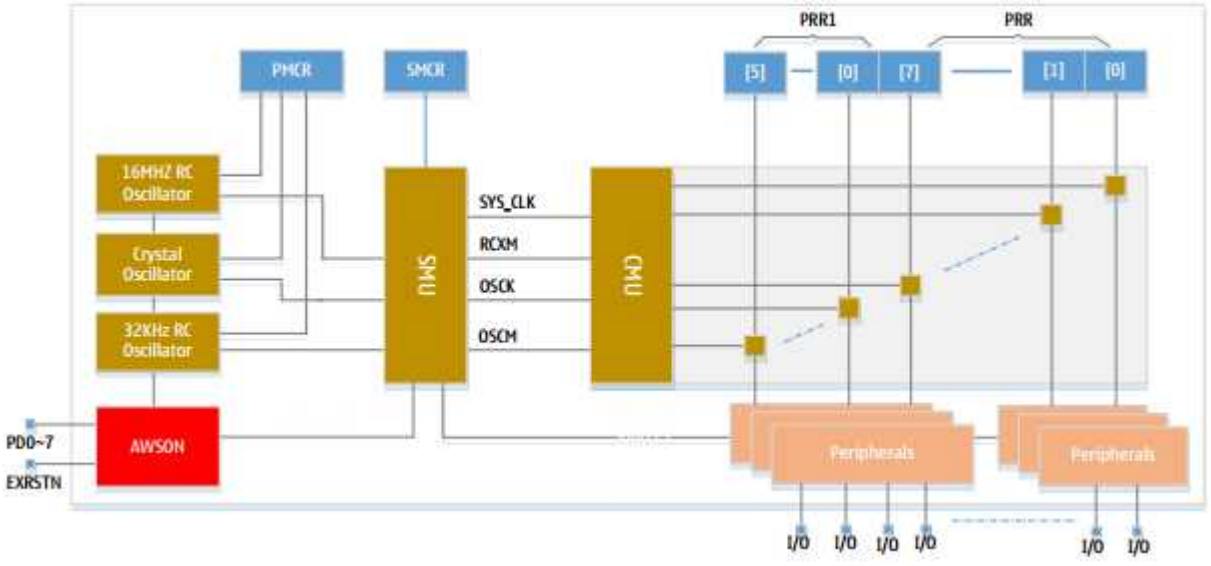
Sleep	Function
IDLE	Only disable core clock, other peripheral work as normal. Idle mode enables the MCU to wake up from valid interrupts
SAVE	Same as DPS0, save mode keep compatible with LGT8FX8D
DPS0	Same as SAVE mode, supported wake-up source include: Level change of all pins Watch dog timer wake-up TMR2 wake up of Asynchronous mode
DPS1	Disable all external oscillator, supported wake-up source include: External level change of all pins External interrupt 0/1 Watch dog timer that works with 32K LFRC
DPS2	Disable core power, lowest power consumption mode, supported wake-up source include: External rest PORTD pin level change LPRC Timer wake-up (128ms/256ms/512ms/1s)

LGT8FX8P 支持深度休眠 DPS2， 在该模式下， 系统内部 LDO 处于掉电状态， 内核寄存器，所有外设控制器以及 SRAM 等均处于掉电状态， 其中的数据将不会保持。FLASH 存储单元也将处于掉电状态，因此 DPS2 模式能够达到系统最小功耗。掉电模式可以通过端口 D(PORTD)引脚电平变化唤醒， 也可以选择 5 级定时唤醒。 用于唤醒的 DPS2 的定时器由于不支持校准，精度在 15%左右，只适合用于精度较低的定时唤醒应用。系统从 DPS2 模式唤醒，会首先开启 LDO， 这个过程和上电过程相同。 芯片将执行完整的上电复位启动过程，加载配置信息，然后从复位向量指向的地址运行程序。除 DPS2 以外的其他模式，不会关闭内部电源， 在休眠过程中， 所有寄存器信息以及 RAM 数据均不会丢失。唤醒后，内核从休眠前的最后一条指令继续执行。

LGT8FX8P support deep sleep DPS2, under this mode, system internal LDO is under power down status, also core register, all peripheral register and SRAM are all under power down status, while data will not be kept. FLASH storage unit is also under power down status, so power consumption is the lowest under DPS2 mode. a pin change interrupt of D(PORTD) can wake up power-down mode, so can 5-level timer can wake up power-down mode. Timer used to wake up DPS2 does not support calibration with tolerance of 15%, so it is only suitable for timer wake up application that requires low accuracy.

When system wake up from DPS2 mode, it will enable LDO firstly, this procedure is same as power-on. Chips

will execute complete power on and re-set, load configuration, than execute programming at address pointing to reset vector. Except DPS2, other mode will not enable internal power. During sleep mode, all register info and RAM data does not loss. After wake-up, core continues to execute programming starting from the last instruction before sleep mode.



如上图所示，LGT8FX8P 主要通过休眠模式控制器(SMU)以及时钟管理单元(CMU)控制整个系统的功耗。从节省功耗的级别上，我们可以把功耗分为 4 个等级:

第一级是通过 PRR 寄存器控制模块工作时钟，通过关闭没有使用模块的时钟，节省系统运行的动态功耗。一般情况下，这种级别能够节省的功耗并不明显。

第二级是通过切换主时钟源到低频时钟上，并关闭没有使用的时钟源模块以及其他模拟模块，这种模式基本上可以得到非常可观的系统运行功耗和休眠功耗。

第三级别是通过让系统进入到掉电模式(DPS1)，DPS1 模式下 LGT8FX8P 可以获得极地的待机功耗，从断电模式唤醒后，软件可以通过 MCUSR 寄存器读取复位前的状态。

第四级别是掉电模式(DPS2)，这个模式将关闭内核电源，可达到最低的系统功耗。因为关闭了内核电源， 这种模式下所有数据信息将会丢失。 唤醒后立刻执行一个上电复位流程，系统重新开始从复位向量处运行。

As shown above, LGT8FX8P mainly control system power consumption via SMU (Sleep mode unit) and CMU(Clock mode unit), From level of power-save, we can classify power consumption into below 4 levels:

Level 1: to control module working clock via PRR register. Disable clock that does not use module in order to save dynamic power consumption of system running. In general, power save consumption of this level is not obvious.

Level 2: switch master clock source to low frequency clock , and disable unused clock rouse module and other analog module, level 2 can achieve obvious power save consumption for system running and sleep mode.

Level 3: make system enter into power-down mode (DPS1), when in this mode, LGT8FX8P can achieve extremely low stand-by power consumption, after wake up from power-off mode, software can read and fetch status before reset via MCUSR register.

Level 4: Power-down mode (DPS2), in this mode, core power will be disable to achieve the lowest power consumption. Under this mode, all data will be lost. Once wake up, a process of power- on while re-set will be

immediately executed, system re-start from reset vector.

AWSON 电源管理

与 LGT8FX8D 相比，掉电模式 DPS2 为一个全新的功耗模式。DPS2 模式用于对休眠功耗有更高要求的应用。进入 DPS2 模式后， 系统仅维持一个静态的模块(AWSON)处于工作状态，其他电路均处于完全掉电状态。

AWSON 模块专用于负责 DPS2 模式的休眠和唤醒控制， AWSON 模块主要由 IO 唤醒控制逻辑以及一个低功耗的 LPRC 组成。软件可以通过 IOCWK 寄存器以及 DPS2R 寄存器实现对 AWSON 的控制。

IOCWK 寄存器用于控制 PD0~7 电平变化的唤醒功能。 DPS2R 寄存器用于控制 DPS2 模式以及 LPRC 的功能模式。具体信息请参考本节末寄存器定义部分。

使用 DPS2 模式前，软件设置 IOCWK 使能所需唤醒 IO， 或者通过 DPS2R 寄存器使能 LPRC 并配置定时唤醒周期， 然后通过 DPS2R 寄存器的 DPS2EN 位使能 DPS2 模式。设置完成后，软件需要通过 SMCR 寄存器设置 DPS2 休眠模式， 然后执行 SLEEP 指令进入休眠。

AWSON Power Management

Compare with LGT8FX8D, power-down mode DPS2 is a total new power consumption mode, it is used for application with very high requirement on sleep power consumption. When enter into DPS2 mode, system only keep a static module (AWSON) under working status, other circuit are under complete power-down status.

AWSON module is exclusively used to take care of sleep and wake up control of DPS2. It is consisted of IP wake-up control logic and a LPRC with low power consumption. Software can control AWSON via IOCWK register and DPS2R register.

IOCWK register is used to control wake-up of PD0~7 level change. DPS2R register is to control DPS2 mode and LPRC function mode. Detailed info, refer to section of register definition at the end of this chapter.

Before enabling DPS2 mode, set up IOCWK via software to enable wake-up IO, or enable LPRC via DPS2R register while set up timer wake up cycles, than enable DPS2 mode via DPS2EN bit of DPS2R register. After all these set up, software need to set up DPS2 sleep mode via SMCR register, onwards execute sleep instruction to enter into sleep.

休眠模式与唤醒源

LGT8FX8P 支持 5 种休眠模式，用户可以根据应用需求选择合适的休眠模式。SMCR 寄存器包含了休眠模式的控制设置，执行 SLEEP 指令后，内核进入休眠模式。为获得更加理想的休眠功耗，建议在内核进入休眠模式前，关闭所有没有使用的时钟以及模拟模块。但需要注意的是，某些唤醒源的产生需要工作时钟，如果需要使用这类唤醒源，请保持相关时钟源的工作状态。

Sleep mode and wake-up source

LGT8FX8P support 5 sleep mode, users can choose suitable one as per application requirement. SMCR register include set-up of sleep mode. After sleep instruction, core enters into sleep mode. Note that to achieve much better sleep power consumption, it is recommend to disable all unused clock and analog module before core enter into sleep mode. Also note of wake up source that need to be generated via working clock, in this case please keep related clock source under working status if you have to use such kind of wake up source

休眠模式与唤醒方式：

Sleep mode and wake up mode:

Sleep mode	Activate clock	Wake up source

	Core clock	Peripheral clock	ADC clock	Asynchronous clock	Pin level change	External interrupt 0/1	TWI address match	TWR2 interrupt	ADC conversion finish	Watch dog overflow	Peripheral interrupt	PD level change
idle		X	X	X	X	X	X	X	X	X	X	X
ADC Noise Reduction			X	X	X	X	X	X	X	X		X
save				X	X	X	X	X		X		X
Power down(DPS0) (With RC32K)				X	X	X		X		X		X
Power down (DPS1) (Without RC32K)				X	X	X		X				X
Power down (DPS2) (Without LDO)												X

如果需要进入以上 5 种休眠模式，SMCR 中的 SE 位必须置 1，使能休眠模式控制。然后执行一条 SLEEP 指令即可。SMCR 中的 SM0/1/2 用于选择不同的休眠模式。具体的信息请参考下面的描述。在 MCU 处于休眠模式下，如果唤醒源有效，MCU 将会在 4 个周期后被唤醒，继续执行指令。如果中断保持有效，中断也将立即响应，进入中断服务子程序。如果在 SLEEP 模式下发生了系统复位，MCU 也将会被唤醒，并从复位向量开始执行。当 MCU 处于 Power/Off 模式下，系统可以通过外部中断 INT0/1 唤醒，唤醒后 MCU 将从 sleep 前的位置继续执行。

To enter any of the five sleep modes, the SE bit in SMCR must be written to logic one, enable sleep mode control and a SLEEP instruction must be executed. The SM2, SM1, and SM0 bits in the SMCR Register select which sleep mode will be activated by the SLEEP instruction. See below description.
When MCU is under sleep mode, if wake source is effective, MCU will wake up in 4 cycles, and continue to execute instruction. If interrupt keep effective, interrupt will re-start immediately and enter into interrupt subroutine.
If system reset under sleep mode, MCU can also wake up and start to execute from reset vector.
When MCU is under power/off mode, system can wake up via external interrupt INT0/1, and then MCU continue to execute from location before sleep.

空闲模式(IDLE)

当 SM2…0 设置为 000，执行 SLEEP 指令后，MCU 进入到 IDLE 模式，IDLE 模式将会关闭掉内核工作时钟，除此之外的其他外设都能正常工作 。IDLE 模式可以通过外部中断以及内部中断等唤醒。如果不需要使用比较器以及 ADC 作为唤醒源，建议将其关闭。IDLE 模式因为仅仅关闭了内核运行的时钟，所以并不能得到明显的功耗降低。 IDLE 模式下，内核也将停止执行和取指令，因此可以降低内部程序 FLASH 的运行功耗。但 IDLE 模式拥有比较灵活的唤醒方式，用户可以通过降低系统主时钟以及关闭不需要的模块获取更加理想的运行功耗。

IDLE Mode

When the SM2...0 bits are written to 000, the SLEEP instruction makes the MCU enter Idle mode, stopping the core clock, while other peripherals work as normal. Idle mode can wake up via external interrupt and internal interrupt. If do not use comparator and ADC as wake up source ,suggest to disable them. Idle mode do not decrease power consumption obviously because only core clock is stopped. In idle, core will stop executing and fetching instructions, so power consumption for running programming flash can be decreased. Idle has

flexible wake-up, to achieve better power consumption, user can decrease system master clock and disable unnecessary module.

ADC 噪声抑制模式

当 SM2…0 设置为 001，执行 SLEEP 指令后，MCU 进入 ADC 噪声抑制模式。此模式下，内核以及大部分外设都将停止工作，ADC，外部中断，TWI 地址匹配，WDT 以及工作在异步时钟模式下的定时/计数器 2 都可以正常工作。

ADC 噪声一直模式主要用于为 ADC 转化提供一个良好的工作环境。降低数字模块对模拟

转换的高频干扰。进入这个模式后， ADC 将自动启动采样转换，转换的数据保存到 ADC 数据

寄存器后，ADC 转换结束中断将 MCU 从 ADC 噪声模式下唤醒。

ADC Noise Reduction Mode

When the SM2...0 bits are written to 001, the SLEEP instruction makes the MCU enter ADC Noise Reduction mode, stopping the core and most peripheral but allowing the ADC, the external interrupts, TWI address match, WDT as well as Timer/Counter2 under asynchronous clock mode to continue operating.

This improves the noise environment for the ADC, reducing high frequency interrupt of digital module to analog converter. (Enabling higher resolution measurements). If the ADC is enabled, a conversion starts automatically when this mode is entered, swap data save in ADC data register. After ADC swap disable interrupt, MCU wake up from ADC Noise Reduction Mode

省电模式(Save)

当 SM2…0 设置为 010，执行 SLEEP 指令后，MCU 进入到 Save 模式。这种模式下，系统将关闭掉所有模块的工作时钟。此模式因为关闭了所有模块的工作时钟，因此只能通过异步模式唤醒，外部中断，TWI 地址匹配以及工作在独立时钟源模式下的 WDT 都可以产生此模式下的唤醒信号。此种模式可以关闭除主时钟源以为的所有模块。为实现更加理想的运行功耗，建议在进入此中模式前，将系统主时钟切换到内部 32K RC 或者外部 32KHz 低频晶振，然后关闭掉所以没有被使用的时钟源以及模拟模块。

Power SAVE Mode:

When the SM2...0 bits are written to 011, the SLEEP instruction makes the MCU enter Power-save mode. Under this mode, system disable working clocks of all module, so can only wake up via asynchronous mode. External interrupt, TWI address match as well as WDT working under independent clock source mode can all generate wake up source for this mode.

Under this mode, all modules except master clock source can be disabled. To achieve perfect running power consumption, it is recommend to switch system master clock to internal 32KRC or external 32KHz low frequency crystal oscillator, than to disable all unused clock source and analog modules before entering into this mode.

掉电模式 DPS0

当 SM[2:0]设置为 110，执行 SLEEP 指令后，MCU 将进入到 DPS0 模式。进入 DPS0 后，除内部 32KHz RC 外，其他时钟源均被关闭。此种模式可以通过外部中断 INT0/1 唤醒；如果使能了 WDT 的中断功能，也可以通过 WDT 实现定时唤醒。

Power-down mode DPS0

When the SM2...0 bits are written to 010, the SLEEP instruction makes the MCU enter DPS0 mode.

In this mode, except internal 32KHz RC, all other clock source disable. This mode can wake up from external interrupt INT0/1. If enable interrupt of WDT, it can also wake up on time via WDT.

掉电模式 DPS1

当 SM[2:0]设置为 011，执行 SLEEP 指令后，MCU 将进入到 DPS1 模式。进入 DPS1 后，系统所有时钟源均被关闭。此种模式可以使用 IO 的电平变化，看门狗唤醒。

Power down mode DPS1

When the SM2...0 bits are written to 011, the SLEEP instruction makes the MCU enter DPS1 mode. In this mode, all system clock source are stopped. This mode can wake up via IP level change and watch dog.

掉电模式 DPS2

设置 SM[2:0]为 111， 并通过 DPSR2 寄存器的 DPS2EN 使能 AWSON 模块，执行 SLEEP 指令后将进入 DPS2 模式。进入 DPS2 模式后，系统关闭内核电源。所以寄存器以及 RAM 数据将会丢失。从 DSP2 唤醒过程与上电复位过程相同。DPS2 模式下， 由于关闭了内核电压，寄存器信息丢失，因此端口的控制状态也将全部恢复到输入状态，所有 IO 的输出驱动以及上拉控制也将关闭。

Power down mode DPS2

When the SM2...0 bits are written to 111, while enable AWSON module via DPS2EN of DPSR2 register, the SLEEP instruction makes the MCU enter DPS2 mode. In this mode, system disables core power. All register and RAM data will loss. Wake up from DSP2 and power-on while reset is identical. In DPS2 mode, because of core voltage disable and register data lost, interface control status all resume to input status, all IO output drive and pull-up will disable.

FLASH 电源控制以及快速唤醒

当系统处于 SLEEP 模式后，内核将不会继续执行指令，此时可以选择关闭 FLASH 的电源，以获得更低的待机功耗。这个功能可以通过 MCUCR 寄存器的 FPDEN 位控制实现；在掉电模式下， 系统可以使用外部中断或者 WDT 唤醒，为了滤除外部信号可能的干扰，内部唤醒电路包含了一个可配置的滤波电路，用户可以根据需要选择合适的滤波宽度。滤波电路的配置可以通过 MCUCR 寄存器的 FWKPEN 实现。MCUCR[FWKPEN]滤波宽度控制：

FLASH power control and quick wake up

When in sleep mode, instructions will not be executed by core, FLASH power can be off to achieve lower standby power consumption, which can be done via enable FPDEN bit of MCUCR register. In power-down mode, system can use external interrupt or WDT wake up. To get rid of possible interference from external signal, there is a filter circuit, that can be set-up, in the internal wake up circuit. User can choose suitable filter width according to specific requirement. Set-up of filter circuit can be done via FWKPEN of MCUCR register. MCUCR[FWKPEN] filter width control as below:

FWKPEN	Filter Width
0	260us (by default)
1	32us

寄存器描述

休眠模式控制寄存器- SMCR

Register Description

Sleep mode control register-SMCR

SMCR- Sleep mode control register					
SMCR: 0x33(0x53)		Default value: 0x00			
Bits		SM2	SM0	SM1	SE
R/W	-	R/W	R/W	R/W	R/W
Bit definition					
[0]	SE	Sleep mode enable bit, after write to logic 1, execute sleep instruction, core enter into sleep mode. SE bit protects system from entering into sleep mode by accident. After wake up, suggest to clear SE bit.			

[3:1]	SM	Sleep Mode Selection			
		SM2	SM1	SM0	Mode description
		0	0	0	IDLE
		0	0	1	ADC Noise reduction
		0	1	0	Save mode
		0	1	1	DPS1
		1	1	0	DPS0
		1	1	1	DPS2
		Others			Kept for no use
[7:4]	-	Kept for no use			

省电控制寄存器- PRR

Power Reduce Register

Power Reduce Register -PRR								
PRR: 0x64					Default value: 0x00			
PRR	PRTWI	PRTW2	PRTW0	-	PRTIM1	PRSPI	RUART0	PRADC
R/W	R/W	R/W	R/W	-	R/W	R/W	R/W	R/W
Bit definition								
[0]	PRADC	Writing a logic one to this bit ,shuts down ADC by stopping the clock to the module						
[1]	RUART0	Writing a logic one to this bit shuts down the USART by stopping the clock to the module						
[2]	PRSPI	Writing a logic one to this bit, shuts down the Serial Peripheral Interface by stopping the clock to the module.						
[3]	PRTIM1	Writing a logic one to this bit shuts down the clock to the Timer/Counter1 module.						
-		Kept for no use						
[5]	PRTW0	Writing a logic one to this bit shuts down the clock to the Timer/Counter0 module						
[6]	PRTW2	Writing a logic one to this bit shuts down the clock to the Timer/Counter2 module						
[7]	PRTWI	Writing a logic one to this bit shuts down the TWI by stopping the clock to the module.						
Bit definition								

省电控制寄存器- PRR1

Power Reduce Register-PRR1

Power Reduce Register-PRR1								
PRR1: 0x65				Default value: 0x00				
PRR1			PRWDT	-	PRTIM3	PREFL	PRPCI	-
R/W			R/W	-	R/W	R/W	R/W	-
Bit definition								
[0]	-	Reserved for no use						
[1]	PRPCI	Write to logic one, shut down external pin change and clock to external interrupt module						
[2]	PREFL	Write to logic one, shut down clock to FLASH controller interface						
[3]	PRTIM3	Write to logic one, shut down clock to TIM3 controller						

[4]	-	Reserved for no use
[5]	PRWDT	Write to logic one, shut down clock to WDT Counter
[7:6]	-	Reserved for no use

MCU 控制寄存器- MCUCR

MCU control register-MCUCR

MCU control register-MCUCR								
MCUCR: 0x35(0x55)				Default value: 0x00				
MCUCR	FWKEN	FPDEN	EXRFD	PUD	IRLD	IFAIL	IVSEL	WCE
R/W	R/W	R/W	R/W	R/W	W/O	R/O	R/W	R/W
Bit definition								
[0]	WCE	MCUCR update enable bit. Before update MCUCR, must write this bit, than update MCUCR register within 6 cycles.						
[1]	IVSEL	Interrupt vector selection bit, write this bit to logic one, interrupt vector address map into new address based on IVBASE Register value						
[2]	IFAIL	Fail to load bit for system configuration bit 0= pass configuration calibration 1= fail configuration calibration						
[3]	IRLD	Write logic one to re-load system configuration						
[4]	PUD	Global pull up forbidden bits 0= enable global pull up forbidden bits 1= shut down all pull up resister of IO						
[5]	EXRFD	External Reset Filter Forbidden Bit 0= Enable external reset digital filter (190us) 1= Disable external reset digital filter circuit						
[6]	FPDEN	Flash Power/Down Enable control 0= When system in sleep mode, FLASH is under power on status 1=When system in sleep mode, FLASH is under power off status						
[7]	FWKEN	Quick Wake Up Mode Enable Control, only effective for Power/Off Mode 0: 260us filter delay 1: 32us filter delay						

PD 组电平变化唤醒控制寄存器- IOCWK

PD Group Level Change Wake Up Register_IOCWK

PD Group Level Change Wake Up Register								
IOCWK: 0xAE				Default value: 0x00				
Bits	IOCD7	IOCD6	IOCD5	IOCD4	IOCD3	IOCD2	IOCD1	IOCD0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit definition								
[7:0]	IOCWK	Write 1 to corresponding bit, enable wake up of IO pin level change of PD group						

DPS2 模式控制寄存器- DPS2R

DPS2 Mode Control Register- DPS2R

DPS2 Mode Control Register- DPS2R	
DPS2R: 0xAF	Default value: 0x00

Bits	-	-	-	-	DPS2E	LPRCE	TOS1	TOS0
R/W	-	-	-	-	R/W	R/W	R/W	R/W
Bit definition								
[1:0]	TOS	LPRC Timer Wake Up Configuration： 00 = 128ms 01 = 256ms 10 = 512ms 11 = 1s						
[2]	LPRCE	LPRC Enable Control 0 = Disable LPRC Timer 1 = disable LPRC Timer [						
[3]	DPS2E	DPS2 Mode Enable Control 0 = Disable DPS2 Mode 1 = Enable DPS2 Mode						
[7:4]	-	Reserved						

系统控制与复位

System Control and Reset

概述

Introduction

系统复位以后，所有的 I/O 寄存器都会被设置为它们的初始值，程序从复位向量处开始 执行。LGT8FX8P 的中断向量地址上， 必须用一个 RJMP – 相对跳转指令跳转到复位处理程 序。如果程序没用使用到中断，没有使能中断源，中断向量也就不会被使用，中断向量区域 就可以用来存放用户的程序代码。

During reset, all I/O Registers are set to their initial values, and the program starts execution from the Reset Vector.

For LGT8FX8P the instruction placed at the Interrupt Vector address must be a RJMP – Relative Jump – instruction to the reset handling routine. If the program never enables an interrupt and an interrupt source, the Interrupt Vectors are not used, and regular program code can be placed at these locations.

复位有效后，所有 I/O 端口立即进入它们的初始状态。大部分 I/O 的初始化状态为输入 并关闭掉内部上拉电阻。有模拟输入功能的 I/O，也初始化为数字 I/O 功能。

The I/O ports are immediately reset to their initial state when a reset source goes active. This does not require any clock source to be running. Initial status of most I/O is to input while shut down internal pull-up resistor. There are I/O with analog input function and with digital I/O function.

当复位变为无效后，LGT8FX8P 内部的定时计数器开始启动，用于展宽复位。展宽复位 信号的宽度用于保证系统中的电源以及时钟等模块进入到稳定的状态。

Once reset is disable LGT8FX8P internal timer start to activate and used for widening reset, which ensure system power and clock modules stable operation.

复位源

Reset Source

LGT8FX8P 共支持六种复位源：

LGT8FX8P support below 6 reset sources:

- 上电复位：当系统的工作电压低压内部 POR 模块的复位阀值时，上电复位有效。
- 外部复位：在芯片的外部复位引脚上一定宽度的低电平脉冲，外部复位有效。
- 看门狗复位：使能看门狗模块后，如果看门狗定时器超时，系统将会复位。
- 低电压复位： LGT8FX8P 内部有一个低电压检测模块(LVD)，当系统工作电源低于 LVD 设定的复位阀值时，MCU 也将会被复位。
- 软件复位：LGT8FX8P 内部有一个专用的软件触发的复位寄存器，用户可以通过这个寄存器随时复位 MCU。
- OCD 复位：OCD 复位是有调试器模块发出的，用于直接复位 MCU 内核。

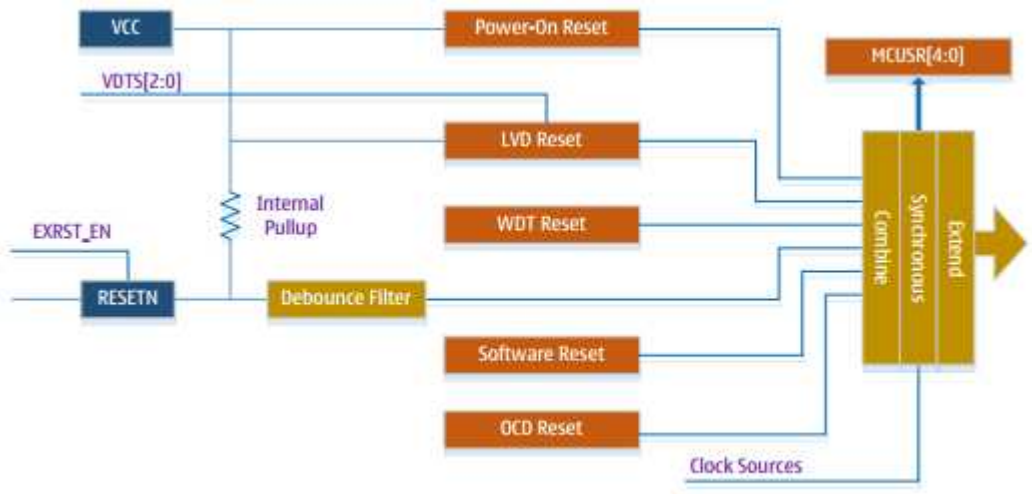
Power-on Reset: The POR is activated whenever system working voltage is below the threshold reset level of internal POR module

External Reset: The POR is activated if there is low level pulse on external reset pin

Watch dog Reset：Enable watch dog module, when watch dog time out, system is reset

Low voltage reset: LGT8FX8P has a LVD (low voltage defection), when system power is lower than designed reset threshold, MCU is reset

Software reset: LGT8FX8P has a reset register specially triggered by software. User can reset MCU via this software.

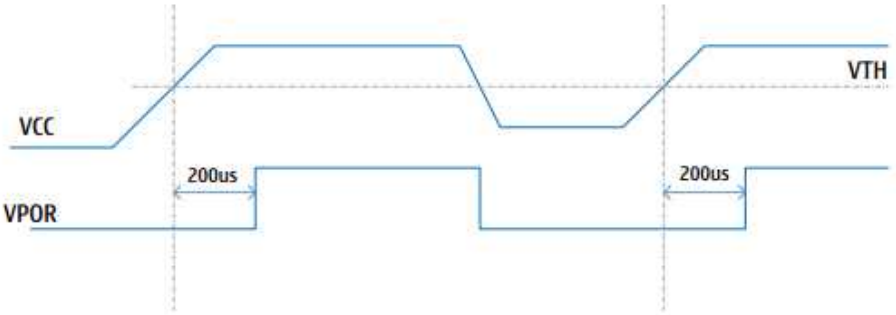


上电复位

上电复位信号由内部的电压检测电路产生。当系统电源(VCC)低于检测阀值时，上电复位信号有效。上电复位的检测阀值，请参考电气参数部分。上电复位电路能够保证芯片在上电过程中处于复位状态，芯片上电后能够从一个已知的稳定的状态开始运行。上电复位信号也会被芯片内部的计数器展宽，以保证上电后内部的各种模拟模块，比如 RC 振荡器等能够进入稳定的工作状态。

Power on Reset

A Power-on Reset (POR) pulse is generated by an On-chip detection circuit. The POR is activated whenever VCC is below the detection level. For power on reset detection threshold, please refer to section of electric data sheet. Power-on Reset circuit ensure chip is reset during period of power on and start to run from a known stable status. Power-on Reset (POR) pulse can be widen by chip internal counter, so that all kinds of internal analog module such as RC oscillator can enter into stable working status.



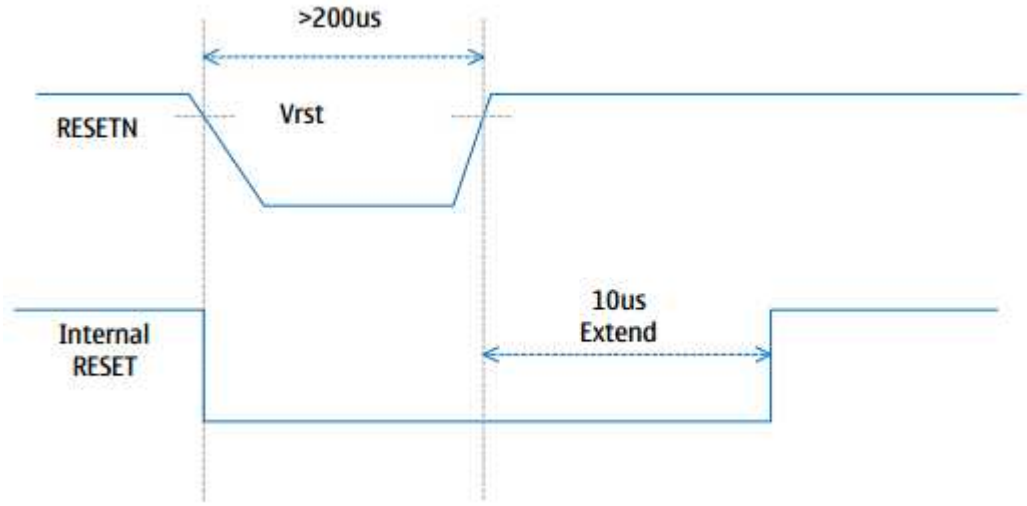
外部复位

在外部复位引脚(RSTN)上施加一个低电平，外部复位立即有效。低电平的宽度要大于一个最小复位脉冲宽度要求。外部复位为异步复位，即使芯片没有时钟工作，外部复位仍然能够对芯片进行复位。LGT8FX8P 的外部复位引脚同时也可以作为通用 I/O 使用。在芯片上电以后， 默认作为外部复位功能。用户可以通过寄存器配置，关闭该引脚的外部复位功能，从而可以当作普通的 I/O 使用。具体使用请参考 IOCR 寄存器的描述部分。

External Reset

External rest is activated once a low level is applied on external reset pin (RESTN). Reset pulses longer than the minimum pulse width will generate a reset, even if the clock is not running.

When power on, chip takes External Reset as default. Via shut down of external rest by write on register, LGT8FX8P external reset pins can be used as general IO. Details please refer to section of IOCR register.

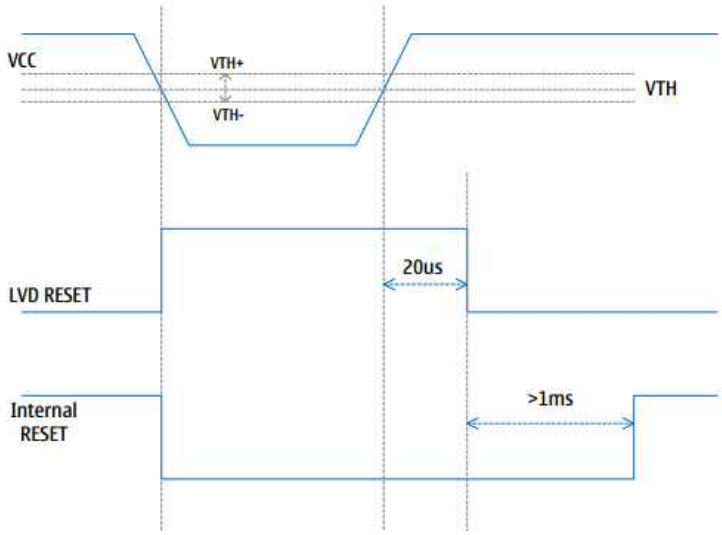


低电压检测(LVD)复位

LGT8FX8P 内部包含一个可编程低电压检测(LVD)电路。LVD 同样是检测 VCC 的电压变化，但与上电复位不同的是，LVD 可以选择检测电压的阀值。用户可以通过直接通过操作 VDTCR 寄存器在不同的电压阀值之间选择。LVD 的电压检测电路具有 ±10mV~±50mV 的迟滞特性，用于滤除 VCC 电压的抖动。当 LVD 使能后，如果 VCC 的电压下降到设定的复位阀值，LVD 复位将立刻有效。当 VCC 增加到复位阀值以上后，内部的复位展开电路启动，将复位继续展宽至少 1 毫秒。

LVD Reset (Low Voltage Detection)

LGT8FX8P includes a programmable LVD circuit. Different from power-on reset, LVD not only defect VCC voltage but also voltage threshold. User can choose different voltage thresholds directly via VDTCR register. LVD has a delay of ±10mV~±50mV to filter VCC voltage fluctuate. Enable LVD, LVD reset is activated if VCC is lower than designed reset threshold. If VCC is higher than reset threshold, reset widen circuit is activated, reset will be widen for at least 1 millisecond.

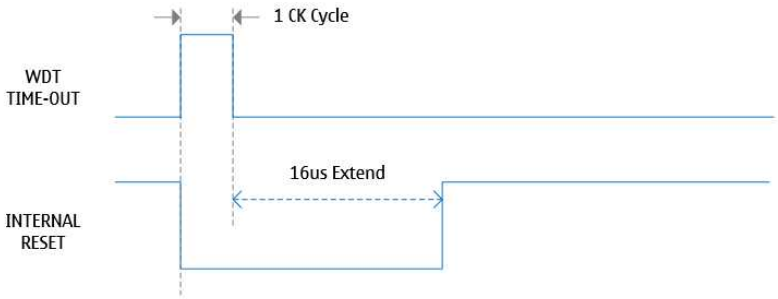


看门狗复位

当看门狗定时器溢出时，如果使能了看门狗系统复位功能，将立刻产生一个周期的系统 复位信号。看门狗复位信号通用也会被内部的延时计数器展宽。看门狗控制器的详细操作， 请参 考下面的详细介绍部分。

Watchdog Reset

When the Watchdog times out, it will generate a reset pulse of one CK cycle duration if watchdog reset is enable. Watchdog reset pulse is also widen by internal delay counter. Details on operation of the Watchdog Timer is as below.



软件复位、OCD 复位

软件复位是用户通过操作 VDTCR 寄存器的第六位触发，软件复位的时序与看门狗复位 完全相似。内部将复位信号展宽 16us。 OCD 复位由芯片内部的调试器单元产生，OCD 复位一般是由调试器控制，用户软件无法 触发 OCD 复位

Software Reset, OCD Reset

Software Reset is triggered via operation at 6th big of VDTCR register, its timing sequence is the same as watchdog reset. Reset pulse is widen by 16us. OCD Reset is generated and controlled by internal debugger, it cannot be triggered by software.

看门狗定时器

时钟可选内部 32KHz RC 或内部 32MHz RC 的 16 分频(2MHz)

支持中断模式，复位模式以及复位中断模式

定时器超时最大可到 8 秒

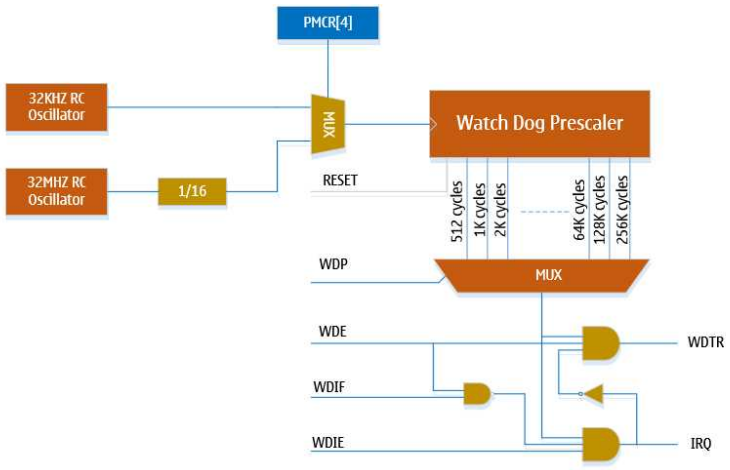
LGT8FX8P 内部包含一个增强的看门狗定时器(WDT)模块。WDT 定时器的工作时钟可以 是内部的 32KHz RC 振荡器，也可以是内部 32MHz RC 振荡器的 16 分频。WDT 计数器溢 出后，可以输出一个中断或者一个系统复位信号。在正常使用时，需要软件执行一个 WDR –看门狗定时器复位指令在

溢出之前重启计数器。如果系统没有即使的执行 WDR 指令，WDT 将会产生中断或系统复位。

Watch dog timer

- **Clock can select internal 32KHz RC or 16 frequency division (2MHz) of 32MHz RC**
- **Support interrupt, reset and reset interrupt mode**
- **Timer time out max. To 8 seconds**

LGT8FX8P has an enhanced Watch Dog Timer module. Its clock can be internal 32KHz RC oscillator or 16 frequency division of 32MHZ RC oscillator. When WDT time out, it will output interrupt or system reset signal. In normal operation mode, it is required that the system uses the WDR - Watchdog Timer Reset - instruction to restart the counter before the time-out value is reached. If the system doesn't restart the counter, an interrupt or system reset will be issued.



在中断模式下，WDT 溢出后会产生一个中断请求信号。可以使用这个中断作为休 眠模式的唤醒信号，也可以作为一个一般的系统定时器使用。比如可以使用这个中断限 制某个操作的执行时间，在溢出中终止当前某一个任务。在系统复位模式下，WDT 在计 数器溢出后立刻产生一个系统复位信号。最典型的用途就是用于防止系统死机或跑飞。 第三种模式，就是复位中断模式，结合了中断和复位两种功能。首先系统将响应 WDT 中断功能，退出 WDT 中断复位程序后，立刻切换到复位模式。这个功能可以支持在复 位之前保存一些比较关键的参数信息。

为了防止 WDT 被意外禁止，关闭 WDT 的操作必须按照一个严格定义的时序进行。下面 的代码描述如何关闭看门狗定时器。下面的例子假设中断已经被禁止，这样整个操作流 程就不会被中断。

In Interrupt mode, the WDT gives an interrupt when the timer expires. This interrupt can be used to wake the device from sleep-modes, and also as a general system timer. . One example is to limit the maximum time allowed for certain operations, stopping current task during time-out. In System Reset mode, the WDT gives a reset when the timer expires. This is typically used to prevent system hang-up in case of runaway code. The third mode, Interrupt and System Reset mode, combines the other two modes by first giving an interrupt and then switch to System Reset mode. This mode will for instance allow a safe shutdown by saving critical parameters before a system reset.

To prevent WDT from disable by accident, WDT Enable operation must follow timed sequences. The sequence for clearing WDE is as follows, note that in the below example interrupt has been disable, than the whole process is not interrupted.

寄存器定义

低压检测(LVD)控制寄存器- VDTCR

Register Definition

LVD Control Register-VDTCR

VDTCR -LVD Control Register								
VDTCR: 0x62				Default value: 0x00				
Bits	WCE	SWR	-	VDTS2	VDTS1	VDTS0	VDREN	VDTEN
R/W	R/W	W/R	-	R/W	R/W	R/W	R/W	R/W
Bit definition								
[0]	VDTEN	LVD module enable control, write one to enable, 0 to disable						
[1]	VDREN	LVD Reset enable control, write one to enable, 0 to disable						
[4:2]	VDTS	Low voltage testing threshold configuration bit 000 = 1.8V 001 = 2.2V 010 = 2.5V 011 = 2.9V 100 = 3.2V 101 = 3.6V 110 = 4.0V 111 = 4.4V						
[5]	-	Reserved for no use						
[6]	SWR	Software reset enable bit, clear this bit to generate software reset						
[7]	WCE	VDTCR Value Change Enable Bit Must write this bit logic one before change the value of VDTCR register, than to change other bits of VDTCR in the following 6 clock cycles. After 4 cycles, WCE is clear automatically and update operation to VDTCR register become invalid						

IO 功能复用寄存器- PMX2

IO Reset Register-PMX2

IO Reset Register-PMX2								
PMX2: 0xF0				Default value: 0x00				
Bits	WCE	STSC1	STSC0	-	-	XIEN	E6EN	C6EN
R/W	R/W	R/W	R/W	-	-	R/W	R/W	R/W
Bit definition								
0	C6EN	By default PC6 pin is taken as reset, write logic one to this bit to disable external reset, than PC6 is used as normal I/O						
1	E6EN	By default PE6 is taken as analog input, write logic one to this bit to disable analog input, than this pin can be used as GPIO						
2	XIEN	External clock input enable control						
4:3	-	reserved						
5	STSC0	Low speed oscillator enable control						
6	STSC1	High speed oscillator enable control						
7	WCE	IOCR value change enable bit Must write logic one to this bit before changing value of IOCR register, in following 6 clock cycles, change other bits of IOCR. After 4 cycles WCE is clear automatically, by than update operation to IOCR register becomes invalid.						

MCU Status Register- MCUSR

MCUSR – IO Special Function Control Register								
MCUSR: 0x34(0x54)				By default: 0x00				
Bits	SWDD	-	PDRF	OCDRF	WDRF	BORF	EXTRF	PORF
R/W	R/W	-	R/W	R/W	R/W	R/W	R/W	R/W
Bit definition								
[0]	PORF	Power on reset flag, write logic zero to clear						
[1]	EXTRF	External reset flag, automatic clear once power on rest, or write logic zero to clear						
[2]	BORF	Low voltage testing reset, automatic clear once power on rest, or write logic zero to clear						
[3]	WDRF	Watch dog reset flag, automatic clear once power on rest, or write logic zero to clear						
[4]	OCDRF	OCD Debugger Reset flag, automatic clear once power on rest, or write logic zero to clear						
[5]	PDRF	Wake up flag from Power/off mode, details refer to section of power consumption management.						
[6]	-	reserved						
[7]	SWDD	SWD port disable bit. Write logic one to shut down SWD port. After shut down of SWD port, debug and ISP operation is not possible. If SWD is shut down in user program, internal program should disable to execute via pull-down RESET during power on, than debug and ISP operation can be executed. When SWD port is shut down, 2 I/O, taken up by SWD, can be used as general I/O. To avoid SWDD mistake operation, in 4 cycles after first update on SWDD bit, use must write SWDD bit again.						

[使用提示]：

为了更加准确有效的使用复位标志信息，建议用户尽量在程序的初始化前期读取复位标志然后将其清零。

Note: To use reset flag information more accurate, it is recommend to clear it only after user read and fetch reset flag in the initialization period.

看门狗控制状态寄存器- WDTCSR

Watch dog control status register- WDTCSR

WDTCSR – WDT Control and Status Register											
Location: 0x60				By default: 0x00							
Bit	7	6	5	4	3	2	1	0			
Name	WDIF	WDIE	WDP3	WDTOE	WDE	WDP2	WDP1	WDP0			
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W			
Bit	Name	Description									
[7]	WDIF	WDT Interrupt Flag If WDT is in interrupt mode and overflow occurs, WDIF is set. If WDT Interrupt enable WDIE to logic one and is changed to global interrupt, WDT interrupt will be generated.									

<table>
<tr><td></td><td></td><td>WDIF bit will be cleared during execution of WDT interrupt, or write logic one to WDIF</td></tr>
<tr><td>[6]</td><td>WDIE</td><td>WDT 中断使能控制位。
当设置 WDIE 位为“1”，且全局中断置位时，WDT 中断被使能。当设置 WDIE 位为“0”时，WDT 中断被禁止。
WDIE 位和 WDE 位一起决定看门狗的工作模式，如下表所示。
WDT interrupt enable control bit
If write WDIE bit to logic one, and global interrupt is set, WDT interrupt is enable.
If write WDIE bit as logic zero, WDT interrupt is disable.
WDIE and WDE in combine to determine the working mode of watch dog, see below
<table>
<tr><th>WDE</th><th>WDIE</th><th>Mode</th><th>Action after time out</th></tr>
<tr><td>0</td><td>0</td><td>Stop</td><td>No</td></tr>
<tr><td>0</td><td>1</td><td>Interrupt mode</td><td>Interrupt</td></tr>
<tr><td>1</td><td>0</td><td>Reset mode</td><td>Reset</td></tr>
<tr><td>1</td><td>1</td><td>Interrupt reset mode</td><td>Reset after intterupt</td></tr>
</table></td></tr>
<tr><td>[5]</td><td>WDP3</td><td>WDT Prescale factor selection control 3rd bit.
WDP[3] and WDP[2:0] consist of WDT Presale factor choose bit WDP[3:0], used to configure WDT time out cycle.</td></tr>
<tr><td>[4]</td><td>WDTOE</td><td>WDT 关闭使能控制位。
当要把 WDE 位清零时， WDTOE 位须置位，否则 WDT 不会被关闭。
当 WDTOE 位被置位后，硬件会在 4 个时钟周期后清零 WDTOE 位
WDT shut down enable control bit
If clear WDT bit, WDTOE must be set, otherwise WDT will not be shut down.
When WEDOT bit is changed, hardware will clear WEDTOE bit in 4 clock cycles.</td></tr>
<tr><td>[3]</td><td>WDE</td><td>WDT Enable control bit
If write logic one to WDE, WDT is enable. If write logic zero to WDE, WDT is disable.
WDE can be clear only if WDTOE Bit is set. To shut down WDT that is enable, must follow the below operation by sequence:
1. Setting WDTOE and WDE bit at the same time, even WDE bit has been written, must write logic one to WDE before shut down operation
2. Write logic zero to WDE in the following 4 clock cycles. This will shut down WDT
If WDE bit is logic one, and WDT overflow reset system, WDT reset system flog WDRF (in MUSR register)is reset. When WDRF is set, WDE is set. So to clear WDE, must clear WDRF bit.</td></tr>
<tr><td>[2:0]</td><td>WDP</td><td>WDT Prescale choose control
Used to configure WDT time out cycle. Suggest to change WDP value before WDT start to count. Unexpected WDT time out will occur if changing WDP value during counting.</td></tr>
</table>

看门狗预分频选择列表

Watch dog prescale list：

WDP3	WDP2	WDP1	WDP0	Watchdog Timer Time-out Cycle	32KHz Clock	2MHz Clock
0	0	0	0	2K cycles	64ms	1ms

0	0	0	1	4K cycles	128ms	2ms
0	0	1	0	8K cycles	245ms	4ms
0	0	1	1	16K cycles	512ms	8ms
0	1	0	0	32K cycles	1s	16ms
0	1	0	1	64K cycles	2s	32ms
0	1	1	0	128K cycles	4s	64ms
0	1	1	1	256K cycles	8s	128ms
1	0	0	0	512K cycles	16s	256ms
1	0	0	1	1024K cycles	32s	512ms
1	0	1	0	reserved		
1	0	1	1			
1	1	0	0			
1	1	0	1			
1	1	1	0			
1	1	1	1			

中断与中断向量

28 个中断源

可编程向量起始地址

LGT8F88P/168P/328P 的中断资源基本相同，主要的区别为：LGT8F88P 的中断向量 为 1 个指令字(16 位)，而 LGT8F168P/328P 的中断向量为 2 个指令字。

Interrupt and Interrupt Vector

28 Interrupt source

Programmable vector start address

Interrupt source of LGT8F88P/168P/328P are in general the same. Main difference is: LGT8F88P interrupt vector is 1 instruction word(16bits) while LGT8F168P/328P is 2 instruction words

LGT8F 88 P 中断向量列表

LGT8F88P Interrupt Vector List

Vector No.	Vector Address	Interrupt Source Pulse	Interrupt Source Description
1	0x0000	RESET	External reset, Power-on reset, watchdog reset, SWD calibration reset, low voltage reset
2	0x0001	INT0	External Interrupt Request 0
3	0x0002	INT1	External Interrupt Request 1
4	0x0003	PCI0	Pin Change Interrupt Request 0
5	0x0004	PCI1	Pin Change Interrupt Request 1
6	0x0005	PCI2	Pin Change Interrupt Request 2
7	0x0006	WDT	Watchdog Time-out Interrupt
8	0x0007	TC2 COMPA	Timer/Counter2 Compare Match A
9	0x0008	TC2 COMPB	Timer/Counter2 Compare Match B

10	0x0009	TC2 OVF	Timer/Counter2 Overflow
11	0x000A	TC1 CAPT	Timer/Counter1 Capture Event
12	0x000B	TC1 COMPA	Timer/Counter1 Compare Match A
13	0x000C	TC1 COMPB	Timer/Coutner1 Compare Match B
14	0x000D	TC1 OVF	Timer/Counter1 Overflow
15	0x000E	TC0 COMPA	Timer/Counter0 Compare Match A
16	0x000F	TC0 COMPB	Timer/Counter0 Compare Match B
17	0x0010	TC0 OVF	Timer/Counter0 Overflow
18	0x0011	SPI STC	SPI Serial Transfer Complete
19	0x0012	USART RXC	USART Rx Complete
20	0x0013	USART UDRE	USART, Data Register Empty
21	0x0014	USART TXC	USART, Tx Complete
22	0x0015	ADC	ADC Conversion Complete
23	0x0016	EE_RDY	EEPROM Ready
24	0x0017	ANA_COMP	Analog Comparator
25	0x0018	TWI	2-wire Serial Interface
26	0x0019	ANA_COMP1	Analog Comparator 1
27	0x001A	-	Reserved
28	0x001B	PCI3	Pin Change Interrupt Request 3
29	0x001C	PCI4	Pin Change Interrupt Request 4
30	0x001D	TC3_INT	Timer/counter 3

LGT8F168P/328P 中断向量列表

LGT8F168P/328 Interrupt Vector List

Vector No.	Vector Address	Interrupt Source Pulse	Interrupt Source Description
1	0x0000	Reset	External reset, power on reset, watchdog reset, SWD calibration reset, low voltage reset
2	0x0002	INT0	External Interrupt Request 0
3	0x0004	INT1	External Interrupt Request 1
4	0x0006	PCI0	Pin Change Interrupt Request 0
5	0x0008	PCI1	Pin Change Interrupt Request 1
6	0x000A	PCI2	Pin Change Interrupt Request 2
7	0x000C	WDT	Watchdog Time-out Interrupt
8	0x000E	TC2COMPA	Timer/Counter2 Compare Match A
9	0x0010	TC2COMPB	Timer/Counter2 Compare Match B
10	0x0012	TC2 OVF	Timer/Counter2 Overflow

11	0x0014	TC1 CAPT	Timer/Counter 1 capture event
12	0x0016	TC1 COMPA	Timer/Counter 1 Compare Match A
13	0x0018	TC1 COMPB	Timer/Counter 1 Compare Match B
14	0x001A	TC1 OVF	Timer/Counter1Overflow
15	0x001C	TC0 COMPA	Timer/Counter0 Compare Match A
16	0x001E	TC0 COMPB	Timer/Counter0 Compare Match B
17	0x0020	TC0 OVF	Timer/CounterO Overflow
18	0x0022	SPI STC	SPI Serial Transfer Complete
19	0x0024	USART RXC	USART Rx Complete
20	0x0026	USART UDRE	USART Data Transfer Emptys
21	0x0028	USART TX	USART Tx Complete
22	0x002A	ADC	ACD Conversion Complete
23	0x002C	EE_RDY	EEPROM Ready
24	0x002E	ANA_COMP	Analog Comparator
25	0x0030	TWI	2 Wire Serial Interface
26	0x0032	ANA_COMP1	Analog Comparator 1
27	0x0034	-	reserved
28	0x0036	PCI3	Pin Change Interrupt 3
29	0x0038	PCI4	Pin Change Interrupt 4
30	0x003A	TC3_INT	Timer/Counter 3 Interrupt

LGT8FX8P 的复位向量从地址 0x0000 开始执行。除复位向量外，其他向量地址都可以通过 MCUCR 寄存器中的 IVSEL 以及 IVBASE 寄存器重新定向到 512 字节对齐的起始地址。

LGT8FX8P reset vector start to execute starting from 0x0000. Other vector address, via setting IVSEL of MCUCR register and IVBASE register, are re-appointed to start address of 512 byte aligned

中断向量处理

下面代码仅以 LGT8F88P 为例，用于说明复位以及中断向量编程，仅供参考：

Interrupt Vector Handling

Below codes are taking LGT8F88P as example, used to descript programming of reset and interrupt vector for your reference:

Programming Code Example: LGT8F88P		
Address	Code	Comments
0x000	RJMP RESET	Reset vector handler
0x001	RJMP EXT_INT0	External interrupt 0 handler
0x002	RJMP EXT_INT1	External interrupt 1 handler
0x003	RJMP PCINT0	Pin Change Interrupt 0 handler
0x004	RJMP PCINT1	Pin Change Interrupt 1 handler

0x005	RJMP PCINT2	Pin Change Interrupt 2 handler
0x006	RJMP WDT	Watchdog timer interrupt handler
0x007	RJMP TIM2_COMPA	Timer/Counter 2 Compare Match A interrupt handler
0x008	RJMP TIM2_COMPB	Timer/Counter 2 Compare Match B interrupt handler
0x009	RJMP TIM2_OVF	Timer/Counter 2 Overflow handler
0x00A	RJMP TIM1 CAPT	Timer/Counter 1 Capture handler
0x00B	RJMP TIM1_COMP A	Timer/Counter 1 Compare Match A interrupt handler
0x00C	RJMP TIM1_COMP B	Timer/Counter 1 Compare Match B interrupt handler
0x00D	RJMP TIM1_OVFR	Timer/Counter 1 Overflow handler
0x00E	RJMP TIM0_COMP A	Timer/Counter 0 Compare Match A interrupt handler
0x00F	RJMP TIM0_COMP B	Timer/Counter 0 Compare Match B interrupt handler
0x010	RJMP TIM0_OVFR	Timer/Counter 0S Overflow handler
0x011	RJMP SPI_STC	SPI Transfer Complete Handler
0x012	RJMP USART_RXC	USART Rx Complete handler
0x013	RJMP USART UDRE	USART Date register empty handler
0x014	RJMP USART_TX	USART Tx complete handler
0x015	RJMP ADC	ADC Conversion complete handler
0x016	RJMP EE_RDY	EEprom ready handler
0x017	RJMP ANA_COMP	Comparator handler
0x018	RJMP_TWI	TWI Handler
0x019	NOP	Reserved address
0x01A	NOP	Reserved address
0x01B	RJMP PCI3	Pin change interrupt handler
;		
0x01C(RESET☺	LDI r16, high(RAMEND)	Main programming start
0x01D	OUT SPH, r16	Set stack pointer to top of RAM
0x01E	LDI r16, ow(RAMEND)	
0x01F	OUT SPL, r16	
0x020	SEI	Enable globe interrupt
0x021	……	

寄存器定义

MCU 控制寄存器- MCUCR

Register Definition: MCU Control register-MCUCR

MCUCR-MCU Control Registe								
MCUCR: 0x35(0x55)				Default value: 0x00s				
MCUCR	FWKEN	FPDEN	EXRFD	PUD	IRLD	IFAIL	IVSEL	WCE
R/W	R/W	R/W	R/W	R/W	W/0	R/O	R/W	R/W
Bit definition								
[0]	WCE	MCUCR Update enable bit, before any update, must set this bit, than within 6 cycles to update MCUCR register						
[1]	IVSEL	Interrupt vector selection. When set this bit to one, interrupt vector address will be						

		mapping to new address according to IVBASE register value
[2]	IFAIL	System configuration bit fail to load flag 0=configuration info verification pass 1=configuration info loading fail
[3]	IRLD	Write logic one to re-load system configuration info
[4]	PUD	Globe pull up disable bit 0=enable globe pull up control 1=shut down all IO pull-up resistance
[5]	EXRFD	External reset filter disable bit 0=enable digital filter of external reset 1=disable digital filter circuit of external reset
[6]	FPDEN	Flash power/down enable control 0: When system is sleep, flash is still under POWER-ON 1: When system sleep, flash is under power off
[7]	FWKENS	Quick wake up mode enable control, only valid for Power/Off mode 0: 260us filter delay 1: 32us filter delay

中断向量基地址寄存器 – IVBASE

Interrupt vector base register

IVBASE- Interrupt vector base register		
IVBASE:0X75	Default value: 0x00	
IVBASE	IVBASE[7:0]	
R/W	R/W	
Bit definition		
[7:0]	IVBASE	If IVSEL is logic one, interrupt vector (except reset vector), taking IVBASE as base address, re-map in 512-byte page After mapping, IVBASE is: (IVBASE<<8) + corresponding vector address in list 1

外部中断

2 个外部中断源

可配置的电平或边沿触发中断

可用作睡眠模式下的唤醒源

External Interrupts

2 external interrupt sources

Pin level change that can be configurable or edge trigger source

Can be used as wake up source under sleep mode

Overview:

外部中断由 INT0 和 INT1 引脚触发。只要外部中断被使能，即使这 2 个引脚配置为输出也能触发中断。这可以用来产生软件中断。外部中断可以由上升沿，下降沿或低电平触发，由外部中断控制寄存器 EICRA 来配置。当外部中断使能并且配置为电平触发（只有 INT0 和 INT1 引脚）时，只要引脚电平为低，中断就会一直产生。 INT0 和 INT1 引脚的上升沿或下降沿中断触发需要 IO 时钟正常工作，而 INT0 和 INT1 引脚的低电平触发中断都是异步检测的。除了空闲模式，其它睡眠模式下 IO 时钟都是停止工作的。因此，这 2 个外部中断都可用作除空闲模式外的其它睡眠模式下的唤醒源。

若电平触发中断用作省电模式下的唤醒源，改变的电平必须保持一定的时间来唤醒 MCU，以降低 MCU 对噪声的敏感程度。要求的电平必须保持足够长的时间使 MCU 结束唤醒过程，然后触发电平中断。

The External Interrupts are triggered by the INT0 and INT1 pins, Observing that, if enabled, the interrupts will trigger even if the INT0 and INT1 or PCINT23...0 pins are configured as outputs. This feature provides a way of generating a software interrupt. The External Interrupts can be triggered by a falling or rising edge or a low level. This is set up as indicated in the specification for the External Interrupt Control Registers – EICRA. When the external interrupt is enabled and is configured as level triggered (has only INT0 and INT1 pin), the interrupt will trigger as long as the pin is held low. Rising or falling edge of INT 0 and INT 1 interrupt can only be trigger in case IO clock work normally, while Low level interrupts on INT 0 and INT 1 are detected asynchronously. The I/O clock is halted in all sleep modes except Idle mode. This implies that these interrupts can be used for waking the part also from sleep modes other than Idle mode.

If a level triggered interrupt is used for wake-up from Power-down, changed lever must be held for some time to wake up MCU in order to reduce MCU sensibility degree to noise. The required level must be held long enough for the MCU to complete the wake-up to trigger the level interrupt .

Register Definition

Register list

Register	Address	Default value	Comments
EICRA	0X69	0X00	External interrupt control register A
EIMSK	0X3D	0X00	External Interrupt Mask Register
EIFR	0X3C	0X00	External interrupt flag register

External interrupt control register A-- EICRA

External interrupt control register A-- EICRA								
Address: 0X69				Default value: 0X00				
Bit	7	6	5	4	3	2	1	0
Name	-	-	-	-	ISC11	ISC10	ISC01	ISC00
R/W	-	-	-	-	R/W	R/W	R/W	R/W

Bit	Name	Comment
7:4	-	reserved
3	ISC11	INT1 pin interrupt trigger mode: control high bit (Bit 1)
2	ISC10	INT1 pin interrupt trigger mode: control low bit. (bit 0) 当全局中断置位且 GICR 寄存器的相应中断屏蔽控制位被置位时，外部中断 1 由 INT1 引脚激发。 When globe interrupt and corresponding mask control bit of GICR register are set, external interrupt 1 is triggered by INT1. Triger mode of interrupt refer to description in the list. Before edge is detected, The value on the INT1 pin is sampled. If edge or level change trigger mode is selected, pulses that last longer than one clock period will generate an interrupt. Shorter pulses are not ensured to generate an interrupt. If low level interrupt is selected, the low level must be held until the completion of the currently executing instruction to generate an interrupt.
1	ISC01	INT0 pin interrupt trigger mode: control high bit (bit 1)
0	ISC00	INT1 pin interrupt trigger mode: control low bit. (bit 0) 当全局中断置位且 GICR 寄存器的相应中断屏蔽控制位被置位时，外部中断 1 由 INT1 引脚激发。

		When globe interrupt and corresponding mask control bit of GICR register are set, external interrupt 0 is triggered by INT0. Triger mode of interrupt refer to description in the list. Before edge is detected, the value on the INT0 pin is sampled. If edge or level change trigger mode is selected, pulses that last longer than one clock period will generate an interrupt. Shorter pulses are not ensured to generate an interrupt. If low level interrupt is selected, the low level must be held until the completion of the currently executing instruction to generaste an interrupt.

External Interrupt 1 Trigger Mode as below table:

External interrupt 1 trigger control:

ISC1[1:0]	Comments
0	Triggered by low level of external pin INT1
1	Triggered by rising or falling edge of external pin INT1
2	Triggered by falling edge of external pin INT1
3	Triggered by rising edge of external pin INT1

External Interrupt 0 Trigger Mode as below table:

External interrupt 0 trigger control:

ISC1[1:0]	Comments
0	Triggered by low level of external pin INT0
1	Triggered by rising or falling edge of external pin INT0
2	Triggered by falling edge of external pin INT0
3	Triggered by rising edge of external pin INT0

External Interrupt Mask Register--EIMSK

External Interrupt Mask Register--EIMSK								
Address: 0X3D				Default value: 0X00				
Bit	7	6	5	4	3	2	1	0
Name	-	-	-	-	-	-	INT1	INT0
R/W	-	-	-	-	-	-	R/W	R/W
Bit	Name	Comment						
7:2	-	reserved						
1	INT1	External pin 1 interrupt enable control bits When the INT1 bit is set (one) and globe interrupt is set, external pin one is enable and wake up is enable. Activity on the pin will cause an interrupt request even if INT1 is configured as an output. When set INT1 to zero, external pin 1 interrupt is disable, so is wake up disable.						
0	INT0	External pin 0 interrupt enable control bit When set INT0 to logic one and globe interrupt is set, external pin 0 interrupt and wake up are enable. Activity on the pin will cause an interrupt request even if INT1 is configured as an output. When set INT0 to zero, external pin 0 interrupt is disable, so is wake up disable.						

External Interrupt Flag Register--EIFR

External Interrupt Flag Register--EIFR								
Address: 0X3C				Default value: 0X00				
Bit	7	6	5	4	3	2	1	0
Name	-	-	-	-	-	-	INT1	INT0
R/W	-	-	-	-	-	-	R/W	R/W
Bit	Name	Comment						
7:2	-	reserved						
1	INTF1	External pin 1 interrupt flag bit When an edge on the external pin 1 pin triggers an interrupt request, INTF1 becomes set. While if a low level on the external pin 1 triggers an interrupt, INTF1 will not be set. At this moment if external pin 1 interrupt enable INT1EN bit to logic one and globe interrupt flag is set, external pin 1 interrupt will be generated. By this interrupt execution or write logic one to INTF1, INTF1 will be clear automatically.						
0	INTF0	External pin 0 interrupt flag bit When an edge on the external pin 0 pin triggers an interrupt request, INTF0 becomes set. While if a low level on the external pin 0 triggers an interrupt, INTF1 will not be set. At this moment if external pin 0 interrupt enable INT0EN bit to logic one and globe interrupt flag is set, external pin 0 interrupt will be generated. By this interrupt execution or write logic one to INTF0, INTF0 will be clear automatically.						

运算加速器(uDSC)

16 位存储模式(LD/ST)

32 位累加器(DX)

单周期 16 位乘法器(MUL)

32 位算术逻辑运算单元(ALU)

16 位饱和运算(SD)

8 周期 32/16 除法器

单周期乘加/乘减运算(MAC/MSC)

Algorithm Accelerator (uDSC)

- **16 bit storage mode (LD/ST)**
- **32 bit Accumulator (DX)**
- **Single cycle 16-bit multiplier (MUL)**
- **bit algorithm logic operation unit (ALU)**
- **bit saturated algorithm**
- **8 cycle 32/16 divider**
- **Single cycle multiply&plus / multiply&minus algorithm (MAC/MSC)**

概述

数字运算加速器（uDSC）作为 LGT8XM 内核的一个运算协处理模块，配合 LGT8XM 内核

16 位 LD/ST 模式，实现一个 16 位的数字信号处理单元。可以满足大部分控制类数字信号的处理。

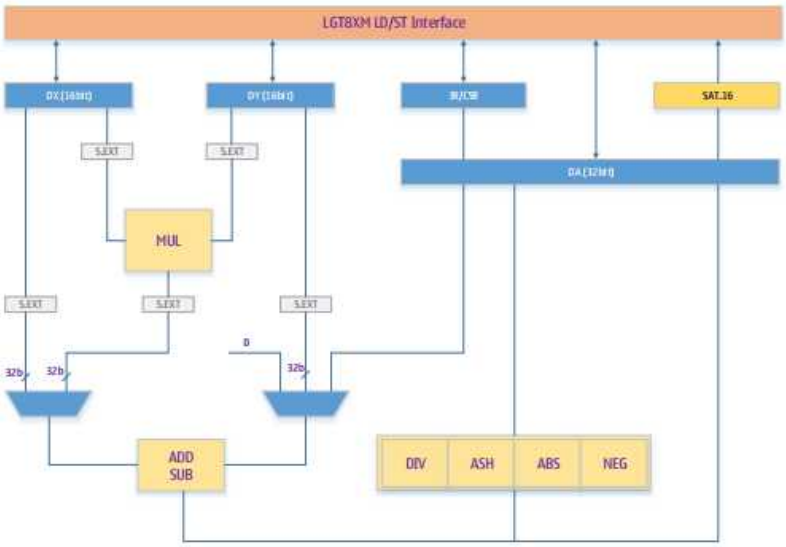
uDSC 功能内部以及功能：

1. 16 位操作数寄存器 DX/DY
2. 32 位累加寄存器 DA

3. 单周期 17 位乘法器（可以实现 16 位有/无符号乘法运算）
4. 32 位 ALU (可以实现 16/32 位的加法，减法以及移位运算)
5. 16 位饱和运算 （用于将运算结果存储到 RAM 空间）
6. 32/16 除法器，8 个周期内完成运算

Digital Arithmetic Accelerator (uDSC), an co-processor arithmetic module of LGT8XM, in combined with 16 bit LD/ST mode, features as a 16 bit digital pulse processing unit. It can deal with most control digital pulse. uDSC internal function as below:

1. **16 bit operand register DX/DY**
2. **32 BIT accumulator register DA**
3. **Single circle 17 bit multiplier (feature 16 bit multiply algorithm with/without symbol)**
4. **32 bit ALU (feature 16/32 bit plus, minus and shift operation)**
5. **16 bit saturation arithmetic (to save algorithm result to RAM memory)**
6. **32/16 divider, algorithm within 8 cycles**



16 位 LD/ST 工作模式

为提高 uDSC 处理大量数据运算的效率，LGT8XM 内核实现一个专用的 16 位 LD/ST 存储通道，可以使用 LDD/STD 指令高效的在 uDSC 与 SRAM 以及通用寄存器文件之间进行 16 位数据交换。

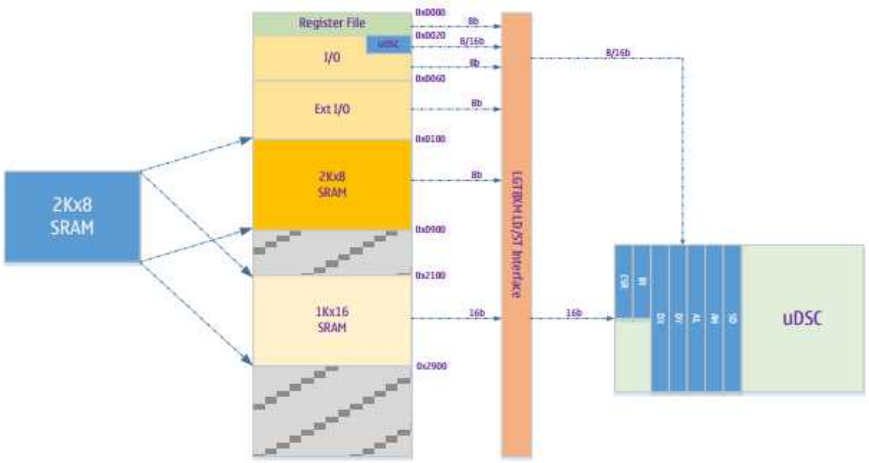
为了不破坏正常的 LD/ST 指令系统， LGT8XM 内核把 SRAM 空间重映射到 0x2100~0x28FF。

使用 LD/ST 指令从 0x2100~0x28FF 空间访问 SRAM 时， 内核自动开启 16 位 LD/ST 功能，打开 SRAM 与 uDSC 之间的直接存取通道。下图为 LGT8XM 内核的数据空间地址分布：

16 bit LD/ST operation mode:

To increase the efficiency of uDSC to operate big volume data, LGT8XM core features a special 16bit LD/ST storage channel, using LDD/STD instruction to operate 16 bit data switch between uDSC and SRAM as well as general register.

To not damage the normal LD/ST instructions system, LGT8XM core maps SRAM memory into 0x2100~0x28FF. When using LD/ST instruction to access SRAM via 0x2100~0x28FF space, LGT8XM core enable 16 bit LD/ST automatically, and open the direct storage channel between SRAM and uDSC. Below shows distribution of data memory address LGT8XM core:



如上图所示，LGT8XM 内核可以通过使用 LD/ST 指令，在 uDSC 的 DX/DY/DA 寄存器与 SRAM 之间直接进行 16 位的数据存存取访问。同时 uDSC 的内部寄存器也映射到 I/O 空间，访问 uDSC 寄存器分为 8/16 两种模式.

As shown above, LGT8XM ,using LDD/STD instruction, features 16 bit data access and storage directly between DX/DY/DA register of uDSC and SRAM. Meanwhile internal register of uDSC map into I/O space. Two modes, 8/16, can be access to uDSC register.

uDSC 内部除了用于运算的 DX/DY/DA 寄存器外， 还包含了另外 2 个 8 位的寄存器： uDSC 控制状态寄存器 CSR 以及运算指令寄存器 IR。CSR/IR 只能通过 I/O 空间以字节为单位 访问；访问 DX/DY/AL/AH 时为 16 位模式。可以使用 IN/OUT 以及 LD/ST/LDD/STD/LDS/STD 等 指令访问。
uDSC 相关的控制状态以及数据寄存器均映射到 IO 空间，直接使用 IN/OU 指令寻址，可以在一个指令周期内完成 8/16 位的数据访问。
CSR 用于控制 uDSC 的工作模式以及记录当前 uDSC 执行运算的状态标志位。IR 控制 uDSC 实现的具体运算。uDSC 支持的运算大部分都会在一个周期内完成，除法运行需要 7 个等待 周期，也可以通过 CSR 寄存器中的标志位判断当前的除法操作是否完成。

标准的 LD/ST 指令使用 LGT8XM 内部的通用工作寄存器作为 LD/ST 的数据，使用 X/Y/Z 作为目标地址。当目标地址落在 16 位 SRAM 映射空间时，此时 LD/ST 指令操作数的含义有 所变化，其中 X/Y/Z 仍然作为目标地址，通用工作寄存器寻址的含义根据 uDSC 映射模式将 有两种处理方式 。
uDSC 的映射模式只作用于对 0x2100~0x28FF 地址存取访问。映射模式通 过 CSR 寄存器的第 6 位(MM)设置。

16 位 LD/ST 模式下，指令”LDD Rn, Z+q” 表示的是把[Z]地址的 16 位数据加载到 uDSC 的 数据寄存器中，然后将 Z 的值增加一个偏移量”q”。此处 Rn 的含义与映射模式 CSR[MM]的关 系如下：

Except DS/DY/DA register for arithmetic, uDSC has another two 8-bit register: uDSC control status register (CSR) and arithmetic instruction register (IR). CSR/IR can only be access in unit of byte via I/O space, and it is 16-bit mode when access to DX/DY/AL/AH,which can be access via IN/OUT and LD/ST/LDD/STD/LDS/STD instructions, and so on.
uDSC related control status register and data register are mapped into IO space, addressing directly via IN/OU instruction and featuring 8/16 bit data access in one instruction cycle.
CSR is used to control working mode of uDSC and record status flag of current arithmetic executed by uDSC. IR is used to control specific arithmetic executed by uDSC. Arithmetic, supported by uDSC can be finished mostly in one cycle, while division operation need 7 waiting cycles. It is also possible t use flag of CSR register to judge if current division is finished or not.

Standard LD/ST instruction uses general working register of internal LGT8XM as LD/ST data while X/Y/Z as

target address. When target address locate in 16-bit SRAM mapping space, the meaning of LD/ST instruction operand will change accordingly, while X/Y/Z are still target address. According to uDSC mapping mode, there are 2 handling methods for meaning of general working register addressing. Note that uDSC mapping mode works only for storage access to 0x2100~0x28FF. Mapping mode is configured via the 6rd bit(MM) of CSR register.

In 16-bit LD/ST mode, instruction "LDD Rn, Z+q" indicate that 16-bit data of [Z] address is loaded to uDSC data register, and add an offset "q" to Z value. Here the relationship between Rn meaning and mapping mode CSR[MM] is as below:

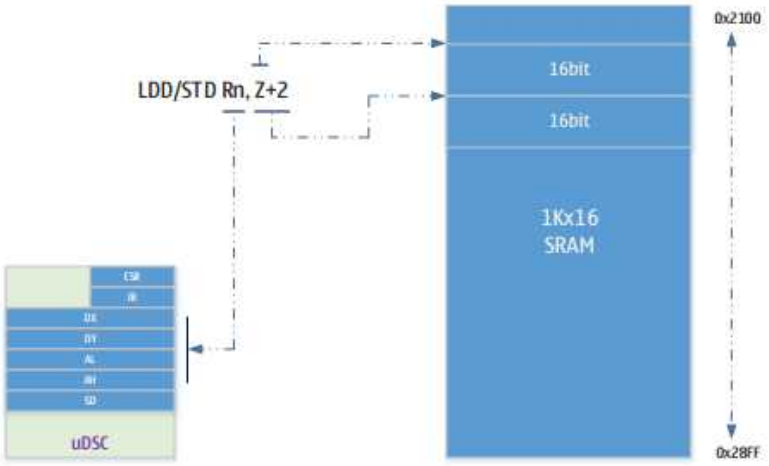
LDD Rn, Z/Y+q			
CSR[MM]	[Z+q]	Opcode	Operations
0	0x2100~0x28FF	LDD R0, Z+q	DX = [Z]; Z = Z + q; R0 kept unchanged
		LDD R1, Z+q	DY = [Z]; Z= Z + q; R1 kept unchanged
		LDD R2, Z+q	AL = [Z]; Z= Z + q; R2 kept unchanged
		LDD R3, Z+q	AH = [Z]; Z= Z + q; R3 kept unchanged
1	0x2100~0x28FF	LDD Rn, Z+q	{Rn} address for DX/DY/AL/AH in I/O region [DX/DY/AL/AY] = [Z]; Z = Z + q Rn keep unchanged
STD Rn, Z/Y+q			
0	0x2100~0x28FF	STD Z+q, R0	[Z] = DX; Z = Z + q; R0 kept unchanged
		STD Z+q, R1	[Z] = DY; Z = Z + q; R1 kept unchanged
		STD Z+q, R2	[Z] = AL; Z = Z + q; R2 kept unchanged
		STD Z+q, R3	[Z] = AH; Z = Z + q; R3 kept unchanged
		STD Z+q, R4	[Z] = SD; Z = Z + q; R4 kept unchanged
1	0x2100~0x28FF	STD Z+q, Rn	{Rn} address for DX/DY/AL/AH/SD in I/O region [Z] = [DX/DY/AL/AH/SD] addressed by {Rn} Rn keep unchanged

LGT8XM 指令集中的 LD/ST, LDS/STS 都可以访问到 0x2100~0x28FF 区域，但是 LDD/STD 的 Y/Z+q 寻址方式更加有效。LDD/STD 方式的寻址基于一个基地址，我们可以把 Y/Z 设置为 RAM 中数据的基地址，通过使用 LDD/STD 指令的 Y/Z+q 寻址方式，可以在一个周期内执行指 令和存取数据，并将地址指针自动移动到下一个目标地址。

LGT8XM 内核标准的 LDD/STD 指令的 Y/Z+q 偏移寻址模式，指令执行时使用[Y/Z+q]作为 8 位数据的地址，执行完成后 Y/Z 的值并不增加。当使用 LDD/STD 寻址 0x2100~0x28FF 区间 的地址时，LDD/STD 的指令行为发生改变：指令执行时，使用[Y/Z]作为 16 位数据寻址地址， 执行后，Y/Z 的值增加"q"指定的偏移量。这种特性可以提高我们连续寻址的效率， 通过 将"q=2"可以实现连续 16 位数据的寻址。

LD/ST and LDS/STS of LGT8XM instruction set are both able to access to 0x2100~0x28FF area, but it is more effective to use Y/Z+q addressing method of LDD/STD. Addressing via LDD/STD based on a base address, in this case we can set Y/Z as base address of RAM data. By using LDD/STD instruction for Y/Z+q addressing, instruction execution and data storage & fetch can be done in one cycle, and address pointer will be moved to next target address automatically.
Y/Z+q Offset addressing mode of LDD/STD instruction use [Y/Z+q] as 8-bit data addressing during instruction execution, after that Y/Z value will not increase. When using LDD/STD to address 0x2100~0x28FF area, instruction of LDD/STD will change. During instruction execution, [Y/Z] is used as 16-bit data addressing location, after instruction execution, value of Y/Z is increased by offset indicated by "q". Because of such

feature, continuous addressing becomes more efficient. If setting “q=2”, we can achieve continuous addressing of 16-bit data.



变量地址与 16 位模式地址之间的映射

LGT8XM 为 8 位处理器， 数据访问以字节为单位。LGT8F328P 内置 2K 字节的数据空间。这部分空间被映射到 0x0100~0x08FF 的地址。C/C++编译自动将变量分配到 0x0100~0x08FF 之间。如果我们在 C/C++中定义的一个 16 位的数组需要使用 uDSC 进行运算，就需要首先将 该变量的地址映射到 16 位 LD/ST 访问的地址区域(0x2100~0x28FF). 方法很简单，只需要将 变量的地址增加 0x2000 的偏移量即可。

Mapping between Variable address and 16-bit mode address

LGT8XM is a 8-bit processor with unit of byte for data access. LGT8F328P features 2K data memory internally, this space is mapped into address of 0x0100~0x08FF. C/C++ programming assign variant automatically to 0x0100~0x08FF. If a 16-bit array, defined by C/C++, need to use uDSC for arithmetic, it is a must to map the address of this variant into 0x2100~0x28FF, that is access to by 16-bit LD/ST. This is very easy: only need to increase variable address by 0x2000 offset.

uDSC 运算指令定义

软件通过 uDSC 的 IR 寄存器指定需要实现的操作。uDSC 的所有运算操作都在 DX/DY/DA 之间进行。用户可以使用 16 位 LD/ST 通道在 DX/DY/DA 以及 SRAM 直接快速的交换数据

uDSC Arithmetic Instruction Set

Software uses IR register of uDSC to appoint desired operations, all arithmetic operations of uDSC are done between DX/DY/DA. Via 16-bit LD/ST channel, users can make quick data swap in DX/DY/DA and SRAM.

Classify	IR[7:0]	Function Description

ADD/SUB	0	0	S^1	0	0	1	0	1	DA = DX + DY
	0	0	S^1	0	0	0	0	1	DA = DX – DY
	0	0	0	1	1	1	0	1	DA = DY
	0	0	S^1	1	1	0	0	1	DA = -DY
	0	0	S^1	1	0	1	1	1	DA = DA + DY
	0	0	S^1	1	0	0	1	1	DA = DA – DY
MAC/MSC	0	1	$S1^2$	$S0^2$	0	1	0	0	DA = DX * DY
	0	1	$S1^2$	$S0^2$	0	0	0	0	DA = -DX * DY
	0	1	$S1^2$	$S0^2$	1	1	0	0	DA = (DX * DY) >> 1
	0	1	$S1^2$	$S0^2$	1	0	0	0	DA = (-DX * DY) >> 1
	0	1	$S1^2$	$S0^2$	0	1	1	S	DA = DA + DX * DY
	0	1	$S1^2$	$S0^2$	1	1	1	S	DA = (DA + DX * DY) >> 1
	0	1	$S1^2$	$S0^2$	0	0	1	S	DA = DA – DX * DY
	0	1	$S1^2$	$S0^2$	1	0	1	S	DA = (DA – DX * DY) >> 1
MISC	1	0	0	0	0	0	0	0	DA = 0
	1	0	0	0	0	1	0	S	DA = NEG(DA)
	1	0	0	0	1	0	0	S	DA = DX^2
	1	0	0	0	1	0	1	S	DA = DY^2
	1	0	1	0	0	0	0	S	DA = ABS(DA)
	1	0	1	1	0	0	0	0	DA = DA/DY
	1	0	1	1	0	0	0	1	DA = DA/DY, DY = DA%DY
SHIFT	1	1	0	0	N3	N2	N1	N0	DA = DA << N
	1	1	S	1	N3	N2	N1	N0	DA = DA >> N

说明：

1. S 表示当然运算是有符号运算还是无符号运算
2. S1 表示 DX 是否为有符号数， S2 表示 DY 是否为有符号数
3. N3…0 为四位移位位数，可以实现最多 15 位移位操作
4. – 表示此位的值不无意义，可设置为 0 或 1，建议设置为 0

Note that:

- **S indicate if current arithmetic is with symbol or not.**
- **S1 indicate if DX is with symbolic number, while S2 indicate if DY is with symbolic number.**
- **N3…0 shift amount of 4-bit, can make 15-bit shift operation the most.**
- **– does not indicate “no meaning ” bit, it can be set to zero or one, we suggest zero.**

寄存器定义

Register Definition

Name	IO address	Comments
DCSR	0x20(0x00)	uDSC control status register
DSIR	0x21(0x01)	Arithmetic instruction register
DSSD	0x22(0x02)	Result of 16-bit saturation arithmetic of accumulator DSA
DSDX	0x10(0x30)	Operands DSDX, 16-bit read and write access
DSDY	0x11(0x31)	Operands DSDY, 16-bit read and write access
DSAL	0x38(0x58)	32-bit accumulator DSA[15:0], 16-bit read and write access
DSAH	0x39(0x59)	32-bit accumulator DSA[31:16, 16-bit read and write access

DSCR - 控制状态寄存器

DSCR---Control Status Register

DS CR – uDSC --Control Status Register								
Address: 0x20 (0x00)						Default value: 0010_xxxx		
Bit	7	6	5	4	3	2	1	0
Name	DSUEN	MM	D1	D0	-	N	Z	C
R/W	R/W	R/W	R/W	R/W	-	R/W	R/W	R/W
Bit	Name	Description						
7	DSUEN	uDSC module enable control: 1= enable,; 0=disable						
6	MM	uDSC register mapping mode, for details referring to introduction on 16-bit working mode 0=quick access mode; 1= IO mapping mode						
5	D1	Division arithmetic, 1= done						
4	D0	除法运算除 0 标志位 Division arithmetic by zero flag						
3	-	Unimplemented						
2	N	Arithmetic result is minus flag						
1	Z	Arithmetic result is zero flag						
0	C	32 加法器进位/借位标志 32 Adder carry/borrow						

DSIR – 运算指令寄存器

DSIR- Arithmetic Instruction Register

DSIR- Arithmetic Instruction Register								
Address: 0x21 (0x01)						Default Value: 0000_0000		
Bit	7	6	5	4	3	2	1	0
Name	DSIR[7:0]							
R/W	R/W							
Bit	Name	Description						
7:0	IR	uDSC arithmetic instruction, referring to section of “arithmetic instruction definition”						

DSDX - 操作数寄存器 DSDX

DSDX- Operands Register

DSDX- uDSC Operands Register- DX																
Address: 0x30 (0x10)												Default value: 0000_0000				
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	DSDX[15:0]															
R/W	R/W															
Bit	Name	Description														
15:0	DSDX	16-bit Operands Register														

DSAL - 32 位累加器 DA 的低 16 位

DSAL- Low 16-bit of 32-bit Accumulator DA

DSAL – uDSC operands register DSA low 16-bit																
Address: 0x58 (0x38)												Default value: 0000_0000				
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	DSA [15:0]															
R/W	R/W															
Bit	Name	Description														
15:0	DSAL	DSAL- Low 16-bit of 32-bit Accumulator DA														

DSAH - 32 位累加器 DA 的高 16 位

DSAH- High 16-bit of 32-bit accumulator DA

DSAH – uDSC operands register DSA high 16-bit																
Address: 0x59 (0x39)												Default value: 0000_0000				
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	DSA[31:16]															
R/W	R/W															
Bit	Name	Description														
15:0	DSAH	High 16-bit of 32-bit accumulator DA														

DSSD - DA 饱和运算寄存器

DSSD—DA Saturation Arithmetic Register

DSSD– 16-bit DA Saturation Arithmetic Result																
Address: 0x22 (0x02)												Default value: 0000_0000				
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	DSSD[15:0]															
R/W	R/W															
Bit	Name	Description														
15:0	DSSD	16-bit DA Saturation Arithmetic Result of 32-bit Accumulator DSA														

uDSC 应用实例

实例 1. 基本配置与运算

下面为一个简单子程序(AVRGCC)，实现一个 16 位的乘法运算，返回 32 位结果.:

uDSC Case Study:

Case 1: Basic Configuration and Arithmetic

Below is a very simple subroutine (AVRGCC), featuring a 16-bit multiply arithmetic while return a 32-bit result:

```
unsigned long dsu_xmuluu (unsigned short dy, unsigned short dx);
```

以下为该 C 函数的汇编实现代码:

```
#include    "udsc_def.inc"      ; opcode definitions
            .global dsu_xmuluu  ; declare for called from C/C++ code

dsu_xmuluu:
            out     DSDX, r24   ; load DX
            out     DSDY, r22   ; load DY
            ldi     r20, XMULUU ; load opcode
            out     DSIR, r20   ; do multiply
            in      r22, DSAL   ; {r23, r22} = AL
            in      r24, DSAH   ; {r25, r24} = AH
            ret
```

通用可编程端口(GPIO)

概述

所有基于 LGT8XM 内核系列实现的 MCU 都具有 I/O 端口读-改-写功能。这意味着，某一个端口的状态可以使用 SBI 和 CBI 指令单独的改变，而不会影响到其他任何 I/O。同样，改变一个端口的方向或者控制它的上拉电阻也可以如此。

LGT8FX8P 的大部分 I/O 拥有对称的驱动特性，能够驱动和吸收较大的电流。I/O 具有两级驱动能力，用户可以控制每组 I/O 的驱动能力。I/O 的驱动能力可以直接驱动一些 LED。

LGT8FX8P 的大部分 I/O 可以驱动高达 30mA 的电流，可直接用于驱动段码 LED。所有的 I/O 的 VCC 和 GND 直接都有独立的 ESD 保护二极管，设计至少可以承受高达 5000V 的 ESD 脉冲。

I/O 等效电路图：

General programmable port GPIO

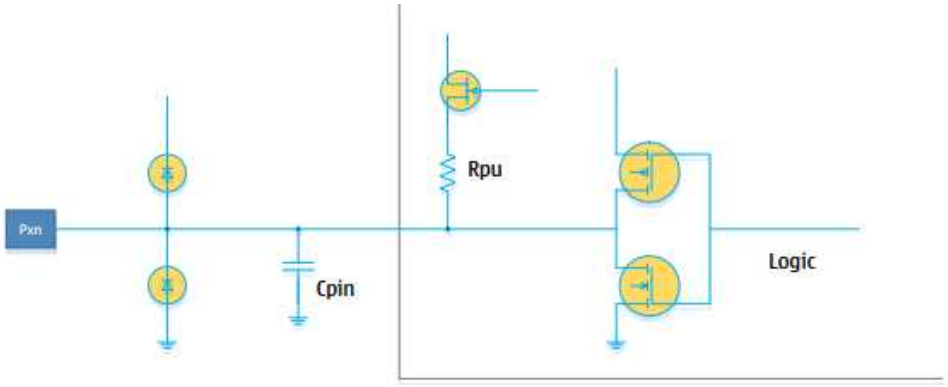
Overview

MCU, based on LGT8XM core, features Read-Modify-Write when used as I/O port, this signify that status of some port can be changed by SBI and CBI instruction alone, while other I/O will not be influenced. This can be also applied to change some port's direction or its pull-up resistance.

Most I/O of LGT8FX8P features symmetric driving characteristics and can drive or absorb big current. I/O has two level driving capability, both can be controlled by user individually. I/O driving capability can drive some LED.

Most I/O of LGT8FX8P can drive currency up to 30mA, can drive directly segment code LED. VCC and GND of all I/Os have its individual ESD protection diode, such design can withstand 5000V ESD pulse at least.

Below is I/O equivalent circuit:



本章下面所有寄存器采用统一描述方式，小写的"x"表示端口的字母序号名，小写的"n"表示端口中的位号。但当在程序中使用端口寄存器时，必须使用准确的寄存器名字。比如 PORTB3，它表示 PORTB 的第三位，这里则统一用 PORTxn 表示。 I/O 相关寄存器的详细定义，请参考寄存器描述部分。

每个端口分配有三个 I/O 寄存器空间，它们为：端口数据输出寄存器(PORTx)，端口方向寄存器(DDRx)，端口数据输入寄存器(PINx)。端口数据输入寄存器为只读寄存器。数据输出寄存器与端口方向寄存器可读也可以改写。

MCUCR 寄存器中的 PUD 位，用于控制所有 I/O 的上拉电阻，当 PUD 位为 1 时，将禁止所以 I/O 的上拉电阻。

大部分 I/O 除了具有通用输入/输出功能，也会被复用为其他外设功能。具体的复用功能请参考关于端口功能复用的章节。

需要注意的是，使能某些端口的复用功能并不会影响这些端口作为数字 I/O 使用。而且某些复用功能也可能需要通过 I/O 寄存器控制端口的输入/输出方向。具体的设置将会在各个复用模块的文档的介绍。

In this section we descript all register the same way, small "x" indicate alphabetic serial number, small "n" for bit number of I/O. However accurate register name must be used during programming. For example, PORTB3 indicate the 3rd bit of PORTB, in this case we adopt a unit way: PORT xn. For detailed definition of I.O related registers, please refer to section of register.

Each port is assigned 3 I/O register memory: port data output register(PORTx), port direction register (DDRx), port data input register (PINx). Port data input register is read-only, while port data output and port direction register are both readable and writable.

PUD bit of MCUCR register is used to control pull-up resistance of all I/O. Write logic one to PUD bit to disable pull-up resistance of all I/O.

Except for input/ output, most I/O can be reused for peripheral, for details please refer to section about port reuse.

Note that to enable reuse of some port, digital I/O function will not be influenced. Moreover some reuse functions need to pass through input/output of IO register control port. Specific setting will be introduced in each reuse module

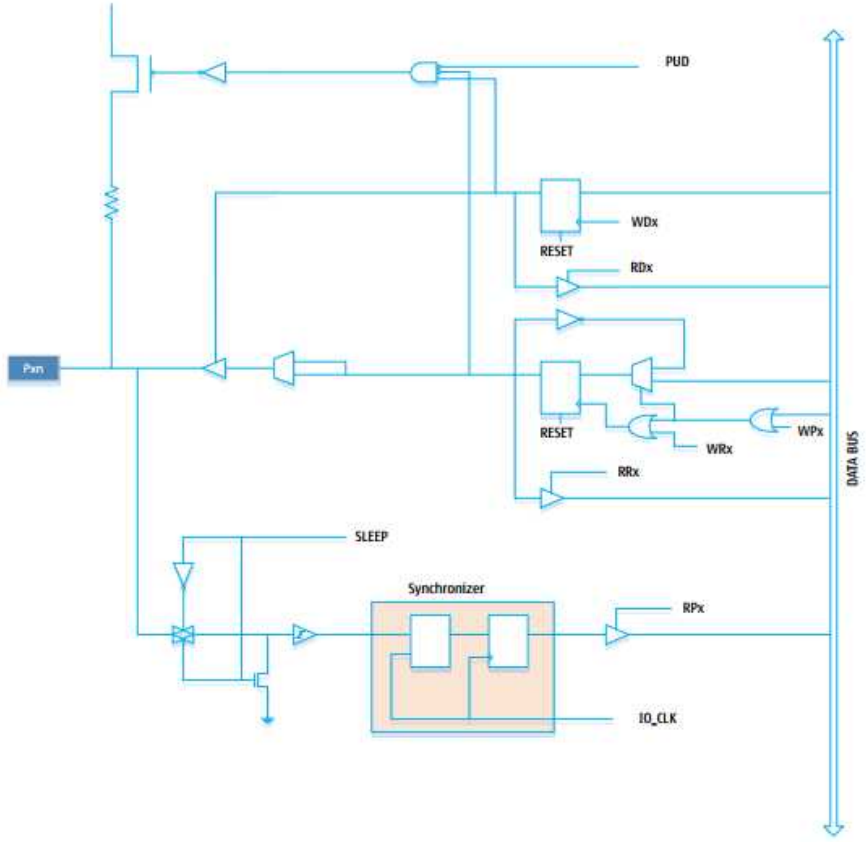
通用输入/输出端口

作为通用 I/O 时，端口为双向驱动 I/O 端口，内部可编程上拉。

下图为通用 I/O 端口的等效电路图：

General Input/output Port:

When it is used as general I/O, I/O port is bidirectional driving, pull-up can be done by programming internally. Below is an equivalent circuit of general I/O port:



PUD: PULLUP DISABLE
SLEEP: SLEEP CONTROL
IO_CLK: I/O CLOCK

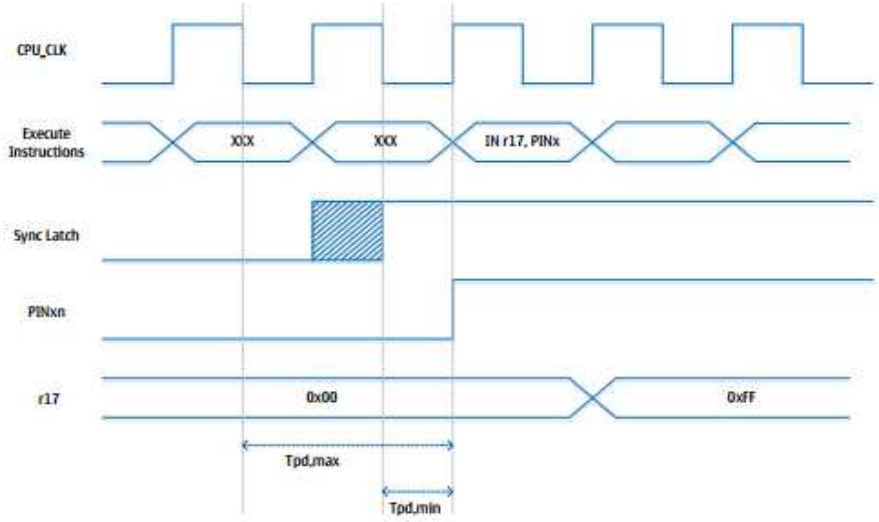
WDx: WRITE DDRx
RDx: READ DDRx
WRx: WRITE PORTx
RRx: READ PORTx REGISTER
RPx: READ PORTx PIN
WPx: WRITE PINx REGISTER

端口使用配置

每个端口由三个寄存器位控制： DDxn， PORTxn 和 PINxn。其中 DDxn 用于可以通过 DDRx 寄存器访问，PORTxn 可以通过 PORTx 寄存器访问，PINxn 可以通过 PINx 寄存器访问。DDRxn 寄存器位用于设置端口的输入/输出方向。如果 DDxn 设置为 1，Pxn 端口就被配置为一个输出端口。如果 DDxn 设置为 0，Pxn 就被配置为一个输入端口。

Port Configuration:

Each port is controlled by 3 register: DDxn， PORTxn and PINxn. DDxn can be access via DDRx register, PORTxn via PORTx REGISTER, and PINxn via PINx register. Register bit of DDRxn is used to design input/ output direction. When DDxn is set to 1, Pxn port is set to be one output port, while DDxn is set to 0, one input port is set to Pxn.



如果 PORTxn 位被写 1，同时这个端口被配置为输入端口，这个端口的上拉电阻有效。如果想要禁止端口的上拉电阻，PORTxn 必须写为 0 或者将这个端口配置为输出端口。端口的复位初始化状态为输入状态，上拉电阻无效。

PORTxn 设置为 1，同时这个端口被配置为输出端口，外部端口将会被驱动为高电平。如果 PORTxn 设置为 0，端口将会被驱动为低。

If PORTxn bit is written to 1 and this port is set to be an input port, pull-up resistor becomes valid. If user want to disable pull-up resistor of port, must write PORTxn to 0 or set this port as output. Reset Initial status of port is input, pull-up resistor is invalid.

Setting PORTxn to 1 and this port as output, external port will be driven as high-level. While setting PORTxn to 0, port will be driven as low-level.

输入/输出切换

当 I/O 状态在三态([DDxn, PORTxn]) = 0b00)和输出高电平([DDxn, PORTxn] = 0b11)之间切换时，将会出现一个端口上拉或者输出为低的中间状态。通常，上拉电阻是可以被接受的，因为在一个高阻环境下，驱动为高和上拉之间的区别并不重要。如果不是这种情况，可以通过 MCUCR 寄存器中的 PUD 位关闭所以端口的上拉功能。

同样，在上拉使能的输入与输出低电平之间切换时，也会出现同样的问题。用户必须使用三态 ([DDxn, PORTxn] = 0b00)或者输出高([DDxn, PORTxn] = 0b11)作为中间状态。

Input/output Switch

When I/O status switch between 3 status ([DDxn, PORTxn]) = 0b00) and output high-level ([DDxn, PORTxn] = 0b11), a intermediate status, port pulled up or low output, will happen. Normally pull-up resistance is acceptable because difference between high drive and pull-up does not matter under conditions of high resistance. Otherwise pull-up can be disable via PUD bit of MCUCR register.

Similarly, the same thing will occur for switch between pull-up enable input and output low level.

So users must use 3 status([DDxn, PORTxn] = 0b00) or high output ([DDxn, PORTxn] = 0b11) as intermediate status.

Port Configuration List

DDxn	PORTxn	PUD	Port Status	Pull-up	Comment
0	0	X	Input	Disable	Tri-Status(High Z)
0	1	0	Input	Enable	Input+ internal pull up mode
0	1	1	Input	Disable	Tri--Status(High Z)

1	0	X	Output	Disable	Low output (fan-in)
1	1	X	Output	Disable	High output (fan-out)

读端口值

无论端口方向位 DDxn 如何设置，都可以通过 PINxn 寄存器位读取到端口的当前状态。为避免直接读取端口产生的亚稳态，PINxn 寄存器位是端口经过一个同步器的结果。同步器为一个锁存器和一个寄存器共同组成，因此 PINxn 的值与当前端口之间有一个很小的延迟。这个延迟是因为同步器存在的结果，延迟时间最多为 1 个半系统周期。

我们假设系统周期从系统时钟的第一个下降沿开始，锁存器在时钟为低的时候锁存数据，时钟为高时数据直通过锁存器，如上图中阴影部分所示。在时钟为低电平时，端口数据被锁存，并且在下一个时钟的上升沿被寄存器到 PINxn 寄存器。上图中的 Tpd,max 以及 Tpd,min 为端口数据的最大和最小延迟，分为为 1.5 周期和 0.5 周期。

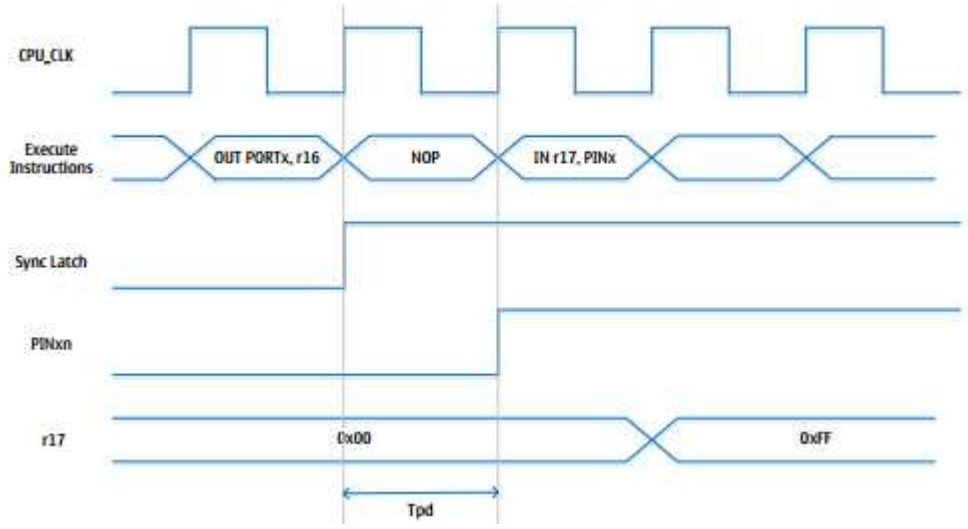
如果要读取到软件设置的端口值，需要在 I/O 的写和读字节支持插入一个空操作指令(NOP)。时序如下图所示：

Read Port Value

No matter how to set port direction bit DDxn, current status of port can be read via PINxn register bit. PINxn register bit is the result, that is from port going through a synchronizer, in order to avoid metastable from direct reading port.

The synchronizing unit is consisted of a latch and a register, so there is a very tiny delay between PINxn value and current port, such delay, resulted from synchronizer, is no more than 1 and a half system cycle.
Assuming that system cycle start from the 1st falling edge of system clock, data is locked in the latch when clock is low. While when clock is high, data goes through latch, for which referring to the shallow part of above picture.
When clock is at low level, port data is latched and is save in PINxn register in the rising edge of the next clock. As shown above, Tpd,max and Tpd,min are the largest and smallest delay of port data, individually 1.5 and 0.5 cycle.

To read port value set by software, a no-op instruction (NOP) should be inserted in write- and read -byte support of I/O. Timing sequence is shown as below:



下面的代码说明如何设置端口 B 的引脚 0/1 为高， 2/3 为低，定义引脚 4~7 为输入并且使能了引脚 6、7 的上拉电阻。然后引脚的值回读到通用工作寄

存器中，按照之前的描述，在引脚的输出和输入直接插入了一个 NOP 指令。

The following code example shows how to set port B pins 0 and 1 high, 2 and 3 low, and define the port pins from 4 to 7 as input with pull-ups assigned to port pins 6 and 7. The resulting pin values are read back again, but as previously discussed, a nop instruction is included to be able to read back the value recently assigned to some of the pins.

汇编代码

```
; Define Pull-ups and set outputs high
; Define directions for port pins
LDI r16, (1<<PB7)|(1<<PB6)|(1<<PB1)|1<<PB0)
LDI r17, (1<<DDB3)|(1<<DDB2)|(1<<DDB1)|(1<<DDB0)
OUT PORTB, r16
OUT DDRB, r17
; Insert nop for synchronization
NOP
; Read port pins
IN r16, PINB
```

C 语言代码

```
unsigned char I;
/* Define pull-ups and set outputs high */
/* Define directions for port pins */
PORTB = (1<<PB7)|(1<<PB6)|(1<<PB1)|(1<<PB0);
DDRB = (1<<DDB3)|(1<<DDB2)|(1<<DDB1)|(1<<DDB0);
/* Insert nop for synchronization */
__no_operation();
/* Read port pins */
I = PINB;
```

输入使能与休眠控制

从 I/O 的等效电路图中我们可以看到，数字输入可以在 SLEEP 信号的控制下被钳位到地电平。SLEEP 信号由 MCU 的休眠控制器以及各种休眠模式控制。这样可以保证在进入休眠后，系统不会因为端口输入浮空而造成漏电。

端口的 SLEEP 控制作用会被外部中断功能取代。如果外部中断请求无效，SLEEP 控制仍然可以起作用。SLEEP 控制功能也会被其他一些第二功能取代，具体请参考下面关于端口第二功能的介绍。

Input enable and sleep control

As shown in I/0 equivalent circuit, the digital input signal can be clamped to ground under the control of SLEEP signal. SLEEP signal is set by MCU Sleep Controller and all kinds of sleep mode, to avoid currency leakage if some input signals are left floating during sleep status.

SLEEP of port is overridden by external interrupt pins. If the external interrupt request is not enabled, SLEEP is active also for these pins. SLEEP is also overridden by various other alternate functions as described in "Alternate Port Functions" as below.

快速翻转端口状态

端口状态设置为输出的 IO，可以通过 PORTn 寄存器改变端口状态。如果需要翻转当前 端口的输出状态，通常需要首先读取当前端口状态 PINx，然后取反回写到 PORTn 寄存器完 成翻转。LGT8FX8P 提供另外一种更加高效的方式翻转端口状态，通过直接向 PINx 寄存器写 1 即可实现将指定的端口状态翻转。比如我们写 PINB[3]为 1，即可实现将 PB3 的端口状态 翻转。对于需要产生输出时钟的应用中，这种方式非常的实用。

Quick Flip Port Status

Setting port status as output IO, port status can be changed via PORTn register. To flip current port output status, in general at first need to read current port status PINx, than write return to PORTn register. LGT8FX8P provide another more efficient mehod to flip status of appointed port, that is writing logic one directly to PINx regsiter. For example: if we write PINB[3] as logic one, PB3 status will be flipped. For application in need of output clock, this is very pratical.

数字/模拟复用端口

LGT8FX8P 部分端口为数模功能混合复用端口。除内部 DAC 的输出 PD4 外，其他混合端 口的均作为模拟输入用。当端口作为模拟功能使用时，软件需要将该端口设置为输入模式， 并根据需要关闭内部上拉，以免对模拟收入产生影响。DIDR0~2 寄存器用于关闭混合功能端 口的数字输入通道，以避免模拟输入对数字电路造成多余功耗损失。DIDRx 不会关闭端口的 数字输出功能。

Digital/ Analog Alternative port

Som of LGT8FX8P ports are hybrid alternate ports with digital and analog function. Except for DAC output PD4, other hybrid port are all used as analog input. When ports are used as analog, software has to set it as input and shut down internal pull-up to avoid influence on analog input. DIDR0~2 register is used to shut down digital input of multiplex ports to avoid extra power consumption caused by analog input to digital circuit. DIDRx does not shut down port digital output.

大电流推挽驱动端口

LGT8FX8P 支持对多 6 路大电流推挽驱动端口，支持最大 80mA 的推挽驱动。考虑到芯 片 VCC 最大过电流能力限制，不建议同时开启 6 路大电流驱动。特别是对于只有一组电源端 口的 QFP32 封装，建议不要同时开启并驱动 4 路以上的大电流负载。

普通端口的驱动为 12mA，软件需要通过 HDR 寄存器开启端口的大电流驱动功能。 具备大电流驱动能力的端口如下：

Large currency push-pull drive port

LGT8FX8P supports no more than 6 circuits of large currency push-pull drive ports , and 80mA push-pull the most. Considering the currency limit of VCC for chip, it is not recommended to open 6 circuts of large currency driving simutanniously. Especially in case of QFP32 packing with only one power port, it is suggested not to open and drive 4 ciurcuits with large currency load at the same time.

Normal port is driven by 12mA, large currency drive of port can be set by software via HDR register. Ports with large currency drive capability are as below:

HDR port	QFP48	QFP32	HDR	Comments
PD5	PD5	PD5	HDR[0]	N/A
PD6	PD6	PD6	HDR[1]	N/A
PF1	PF1	PD1 PF1	HDR[2]	PD1 of QFP32 package equals to parallel between PD1 and PF1 of QFP48
PF2	PF2	PD2 PF2	HDR[3]	PD2 of QFP32 package equals to parallel between PD2 and PF2 of QFP48
PF4	PF4	PE4 PF4	HDR[4]	PE4 of QFP32 package equals to parallel between PE4 and PF4 of QFP48
PF5	PF5	PE5 PF5	HDR[5]	PE5 of QFP32 package equals to parallel between PE5 and PF5 of QFP48

空闲端口的处理

如果一些端口没有被使用，建议将他们驱动到一个固定的电平。在任何情况下，浮空的 引脚都会带来更多的功耗，并且会导致系统在强干扰下变的不稳定。 给端口一个固定电平最简单的方法就是打开端口的上拉电阻。需要注意的是，上拉电阻 在上电复位过程中是禁止的。上拉电阻的方式也会带来多余的漏电。因此建议使用外部的上 拉或者下拉电阻连接。直接将端口与电源或地连接是不建议的，因为如果这些引脚被配置为 输出，会有可能导致

非常大的电流由端口经过，对芯片造成破坏性的影响。

Free Port Handling

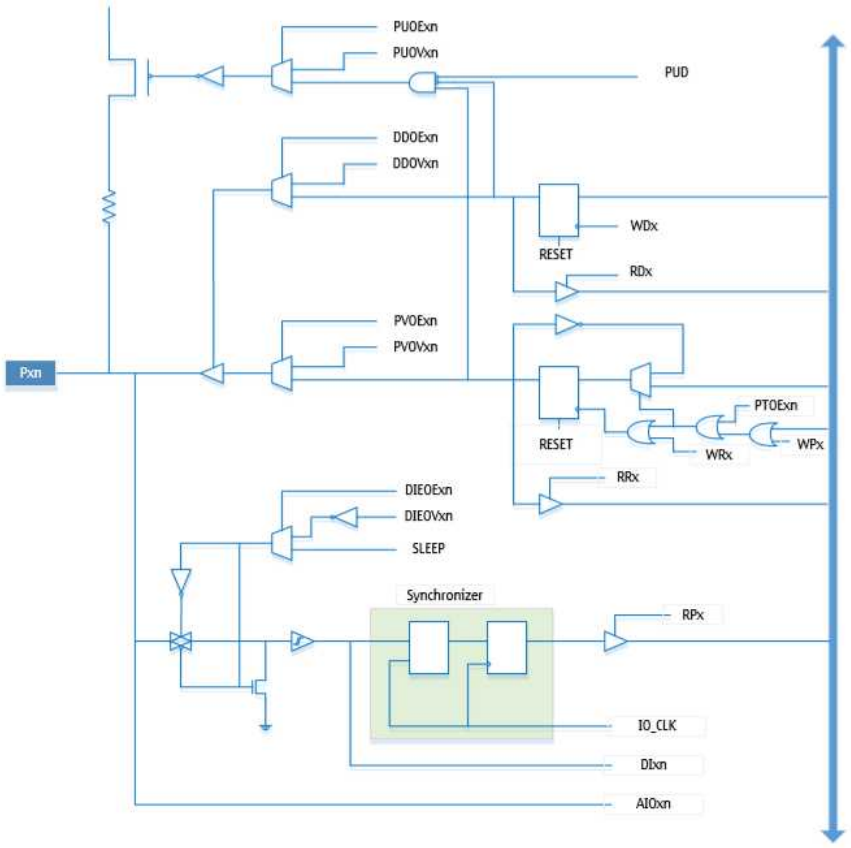
If port is not used, it is suggested to drive them to a fixed level. In any case, pins left floating will cause more power comsumption and make system unstable under strong interferrence. An easiest method of giving port a fixed level is to open its pull-up resistor. Note that: pull-up resistor is disabled during power-on reset. Pull-up resistor will causee extra electricity leagkage, so we suggest use external pull-up or pull-down resistor for connection. Direct connecting port with power or groundd is not recommended, because if these pins are set output, destructive influence will occur on chips, resulting from possible large currency going through port.

端口复用功能

大部分端口都有复用功能，下面的等效电路说明了端口复用功能对端口的控制。这些复 用功能并不一定存在与所以的端口引脚。

Port Multiplex

Most ports has Alternate function, below equivelent circuit show how port is controlled by port multiplex. These port alternate functions do not all exist in all port pins.



PUOExn:	Pxn PULL-UP OVERRIDE ENABLE	PUD:	PULLUP DISABLE
PUOVxn:	Pxn PULL-UP OVERRIDE VALUE	WDx:	WRITE DDRx
DDOExn:	Pxn DATA DIRECTION OVERRRIDE ENABLE	RDx:	READ DDRx
DDOVxn:	Pxn DATA DIRECTION OVERRIDE VALUE	RRx:	READ PORTx REGISTER
PVOExn:	Pxn PORT VALUE OVERRIDE ENABLE	WRx:	WRITE PORTx
PVOVxn:	Pxn PORT VALUE OVERRIDE VALUE	RPx:	READ PORTx PIN
DIEOExn:	Pxn INPUT-ENABLE OVERRIDE ENABLE	WPx:	WRITE PINx
DIEOVxn:	Pxn INPUT-ENABLE OVERRIDE VALUE	IO_CLK:	I/O CLOCK
SLEEP:	SLEEP CONTROL	DIxn:	INPUT PIN n ON PORTx
PTOExn:	Pxn PORT TOGGLE OVERRIDE ENABLE	AIOxn:	ANALOG I/O PIN n ON PORTx

s

复用功能控制信号一般描述

Generic Description of Overriding Signals for Alternate Functions

Signal	Name	Comments
PUOE	Pull-up Override Enable	If this signal is set, the pull-up enable is controlled by the PUOV signal. If this signal is cleared, the pull-up is enabled when {DDxn, PORTxn, PUD} = 0b010.
PUOV	Pull-up Override Value	If PUOE is set to 1, the pull-up is enabled, if it is set to zero to disabled pull-up
DDOE	Data Direction Override Enable	If this signal is set, the Output Driver Enable is controlled by the DDOV signal. If this signal is cleared, the Output driver is enabled by the DDxn Register bit.
DDOV	Data Direction Override Value	If DDOE and DDOV are set to 1, the Output of PIN is enabled, otherwise disable
PVOE	Port Value Override Enable	If this signal is set to 1 and the Output Driver is enabled, the port value is controlled by the PVOV signal. If PVOE is cleared, and the Output Driver is enabled, the port Value is controlled by the PORTxn Register bit
PVOV	Port Value Override Value	Referring to section of PVOE
PTOE	Port Toggle Override Enable	If PTOE is set to 1, the PORTxn Register bit is inverted.
DIEOE	Digital Input Enable Override Enable	If this bit is set to 1, the Digital Input Enable is controlled by the DIEOV signal. If this signal is cleared, the Digital Input Enable is determined by MCU state
DIEOV	Digital Input Enable Override Value	If DIEOE is set, the Digital Input is enabled/disabled when DIEOV is set/cleared, regardless of the MCU state.
DI	Digital Input	This is the Digital Input to alternate functions. In the figure, the signal is connected to the output of the Schmitt Trigger but before the synchronizer. If this singal is connected with peripheral modules, these modules make syschrnonized handling as per demand.
AIO	Analog Input	This is the Analog Input/output. The signal is connected directly to the pad, and can be used bidirectionally. This singal is connected directly with port of

		internal ADC, Compator and other analog modules.

下面一小节将会简短的描述每个引脚的复用功能和相关的控制信号。

The following subsections shortly describe the alternate functions for each port, and related control signals.

Port B Pins Alternate Functions

PB7	XTALI/TOSC2 (External Master Oscillator pin XI) PCINT7 (Pin Change Interrupt 7)
PB6	XTALO/TOSC1 (External Master Oscillator pin X0) PCINT6 Pin Change Interrupt 6)
PB5	SCK (SPI Bus Master clock Input) PCINT5 (Pin Change Interrupt 5)
PB4	MISO (SPI Bus Master Input/Slave Output) PCINT4 (Pin Change Interrupt 4)
PB3	MOSI (SPI Bus Master Output/Slave Input) OC2A (Timer/Counter2 Compare Match A Output) PCINT3 (Pin Change Interrupt 3)
PB2	SSN (SPI Bus input from device) SPI 总线从设备选择输入 OC1B (Timer/Counter1 Compare Match B Output) PCINT2 (Pin Change Interrupt 2)
PB1	OC1A (Timer/Counter1 Compare Match A Output) PCINT1 (Pin Change Interrupt 1)
PB0	ICP1 (Timer/Counter1 Capture Input) CLKO (System Clock Output) PCINT0 (Pin Change Interrupt 0)

XTALI/TOSC2/PCINT7 – 端口 B 引脚 7

XTALI：外部晶振引脚 XI。当用作晶振的时钟信号时，这个引脚将不能作为 I/O 使用。

TOSC2：定时器外部晶振引脚 2。当内部 RC 被配置为芯片的主工作时钟，并且使能了异步定时器功能(ASSR 寄存器配置)，此引脚将作为定时器的外部晶振引脚。当 ASSR 寄存器的 AS2 被设置为 1，EXCLK 为设置为 0，便使能了定时/计数器 2 使用外部晶振的异步时钟功能，PB7 将与内部 I/O 端口断开，成为内部振荡放大器的反向输出引脚。这种模式下，外部晶振与引脚相连接。

PCINT7：引脚电平变化中断 7。PB7 为外部中断源。如果 PB7 被用于晶振引脚，DDB7,PORTB7 和 PINB7 的值将没有任何意义。

XTALI/TOSC2/PCINT7 –Port B Pin 7

XTALI: external oscillator pin XI. Used as clock signal for crystal Oscillator the pin cannot be used as an I/O pin.

TOSC2: Timer external Oscillator pin 2. Used only if internal RC is selected as master working clock, and the asynchronous timer is enabled by the correct setting in ASSR register. When the AS2 bit in ASSR is set (one) and the EXCLK bit is cleared (zero) to enable asynchronous clocking of Timer/Counter2 using the Crystal Oscillator, PB7 is disconnected from the port, and becomes the inverting output of the Oscillator amplifier. In this mode, external crystal Oscillator is connected to this pin.

PCINT7: Pin Change Interrupt 7. The PB7 pin can serve as an external interrupt source.

If PB7 is used as a crystal oscillator pin, DDB7, PORTB7 and PINB7 will all read 0.

XTALO/TOSC1/PCINT6- 端口 B 引脚 6

XTALO：外部晶振引脚 XO。

TOSC1：定时器外部晶振引脚 1。当内部 RC 被配置为芯片的主工作时钟，并且使能了异步定时器功能(ASSR 寄存器配置)，此引脚将作为定时器的外部晶振引脚。当 ASSR 寄存器的 AS2 被设置为 1，EXCLK 为设置为 0，便使能了定时/计数器 2 使用外部晶振的异步时钟功能， PB6 将与内部 I/O 端口端口，成为内部振荡放大器的输入引脚。这种模式下，外部晶振与引脚相连接。

PCINT6：引脚电平变化中断 6。PB6 为外部中断源。如果 PB6 被用于晶振引脚，DDB6,PORTB6 和 PINB6 的值将没有任何意义。

XTALO/TOSC1/PCINT6- Port B Pin 6

XTALO: external crystal oscillator pin XO

TOSC1: Timer external Oscillator pin 1. Used only if internal RC is selected as master working clock, and the asynchronous timer is enabled by the correct setting in ASSR register.

When the AS2 bit in ASSR is set (one) and the EXCLK bit is cleared (zero) to enable asynchronous clocking of Timer/Counter2 using the Crystal Oscillator, PB6 is disconnected from internal I/O port, and becomes the input pin of the Oscillator amplifier. In this mode, external crystal Oscillator is connected to this pin.

PCINT6: Pin Change Interrupt 6. The PB6 pin can serve as an external interrupt source. If PB6 is used as a crystal oscillator pin, DDB6, PORTB6 and PINB6 will all read 0.

SCK/PCINT5- 端口 B 引脚 5

SCK: SPI 控制器主设备时钟输出，从设备时钟输入。当 SPI 控制器被配置为一个从设备，这个引脚将被配置为一个输入引脚，不受 DDB5 的控制。当 SPI 控制器被配置为主设备，这个引脚的方向由 DDB5 控制。当这个引脚被 SPI 强制为输入后，仍然可以通过 PORTB5 位控制上拉电阻。

PCINT5：引脚电平变化中断。PB5 为外部中断源。

SCK/PCINT5 – Port B, Bit 5

SCK: Master Clock output, Slave Clock input pin for SPI channel. When the SPI is enabled as a Slave, this pin is configured as an input regardless of the setting of DDB5. When the SPI is enabled as a Master, the data direction of this pin is controlled by DDB5. When the pin is forced by the SPI to be an input, the pull-up can still be controlled by the PORTB5 bit.

PCINT5: Pin Change Interrupt 5. The PB5 pin can serve as an external interrupt source.

MISO/PCINT4- 端口 B 引脚 4

MISO: SPI 控制主设备数据输入，从设备数据输出。当 SPI 被配置为主设备，这个引脚将会被强制为输入，并不受 DDB4 的控制。当 SPI 作为一个从设备时，这个引脚的数据方向由 DDB4 控制。当这个引脚被 SPI 控制器强制为输入后，它的上拉电阻仍然可以通过 PROTB4 控制。

PCINT4：引脚电平变化中断。PB4 为外部中断源。

MISO/PCINT4 – Port B, Bit 4

MISO: Master Data input, Slave Data output pin for SPI channel. When the SPI is enabled as a Master, this pin is configured by fource as an input regardless of the setting of DDB4. When the SPI is enabled as a Slave, the data direction of this pin is controlled by DDB4. When the pin is forced by the SPI to be an input, the pull-up can still be controlled by the PORTB4 bit.

PCINT4: Pin Change Interrupt source 4. The PB4 pin can serve as an external interrupt source.

MOSI/OC2A/PCINT3- 端口 B 引脚 3

MOSI: SPI 控制器主设备数据输出，从设备数据输入。当 SPI 被配置为从设备，这个引脚将会被强制为输入，并不受 DDB3 的控制。当 SPI 控制器被配置为主设备，这个引脚的方法由 DDB3 控制。当这个引脚被 SPI 控制强制为输入，仍然可以通过 PORTB3 控制它的上拉电阻。

OC2A：定时/计数器 2 的 A 组比较匹配输出。PB3 可以作为定时/计数器 2 比较匹配的外部输出。此时必须通过 DDB3 将引脚设置为输出。同时，OC2A 也是定时器 2 的 PWM 模式输出引脚。

PCINT3: 引脚电平变化中断。**PB3** 为外部中断源。

MOSI/OC2A/PCINT3 – Port B, Bit 3

MOSI: SPI Master Data output, Slave Data input for SPI channel. When the SPI is enabled as a Slave, this pin is configured as an input regardless of the setting of DDB3. When the SPI is enabled as a Master, the data direction of this pin is controlled by DDB3. When the pin is forced by the SPI to be an input, the pull-up can still be controlled by the PORTB3 bit.

OC2A: Timer/Counter 2 Compare Match Output A. The PB3 pin can serve as an external output for the Timer/Counter2

Compare Match. The PB3 pin has to be configured as an output (DDB3 set (one)) to serve this function. The OC2A pin is also the output pin for the PWM mode timer function.

PCINT3: Pin Change Interrupt source 3. The PB3 pin can serve as an external interrupt source.

SSN/OC1B/PCINT2- 端口 **B** 引脚 **2**

SSN: SPI 从设备片选输入。当 **SPI** 控制器配置为从设备，这个引脚将会被强制为输入，并不受 **DDB2** 的控制。作为一个从设备， **SPI** 控制器在 **SSN** 被驱动为低是有效。当 **SPI** 控制器配置为主设备，这个引脚的方向由 **DDB2** 控制。当这个引脚被 **SPI** 控制器强制为输入后，仍然可以通过 **PORTB2** 控制上拉电阻。

OC1B: 定时/计数器 **1** 的 **B** 组比较匹配输出。**PB2** 可以作为定时/计数器 **1** 比较匹配的外部输出。此时必须通过 **DDB2** 将引脚设置为输出。同时，**OC1B** 也是定时器 **1** 的 **PWM** 模式输出引脚。

PCINT2: 引脚电平变化中断。**PB2** 为外部中断源。

SSN/OC1B/PCINT2 – Port B, Bit 2

SSN: Slave Select input. When the SPI is enabled as a Slave, this pin is configured as an input regardless of the setting of DDB2. As a Slave, the SPI is activated when this pin is driven low. When the SPI is enabled as a Master, the data direction of this pin is controlled by DDB2.When the pin is forced by the SPI to be an input, the pull-up can still be controlled by the PORTB2 bit.

OC1B: Timer/Counter 1 Compare Match Output B. The PB2 pin can serve as an external output for the Timer/Counter 1. The PB2 pin has to be configured as an output (DDB2 set (one)) to serve this function. The OC1B pin is also the output pin for the PWM mode of Timer 1.

PCINT2: Pin Change Interrupt 2. The PB2 pin can serve as an external interrupt source.

OC1A/PCINT1- 端口 **B** 引脚 **1**

OC1A: 定时/计数器 **1** 的 **A** 组比较匹配输出。**PB1** 可以作为定时/计数器 **1** 比较匹配的外部输出。此时必须通过 **DDB1** 将引脚设置为输出。同时，**OC1A** 也是定时器 **1** 的 **PWM** 模式输出引脚。

PCINT1: 引脚电平变化中断。**PB1** 为外部中断源。

OC1A/PCINT1 – Port B, Bit 1

OC1A: Timer/Counter 1 Compare Match Output A. The PB1 pin can serve as an external output for the Timer/Counter 1 . The PB1 pin has to be configured as an output (DDB1 set (one)) to serve this function. The OC1A pin is also the output pin for the PWM mode of Timer 1 function.

PCINT1: Pin Change Interrupt. The PB1 pin can serve as an external interrupt source.

ICP1/CLKO/PCINT0- 端口 **B** 引脚 **0**

ICP1: 定时/计数器 **1** 的俘获输入引脚

CLKO: 系统工作时钟输出，当 **CLKPR** 寄存器中的 **CLKOE** 位为 **1**，这个引脚将会被强制为输出，不受 **DDB0** 的控制。输出频率为当前系统的工作时钟频率。

PCINT0: 引脚电平变化中断。**PB0** 为外部中断源。

ICP1/CLKO/PCINT0 – Port B, Bit 0
ICP1: Input Capture Pin of Timer/Counter 1
CLKO: System Clock Output. If CLKOE of CLKPR register is set to 1, this pin will be forced to output regardless of the setting of DDB0. Output frequency is that of working clock of current system
PCINT0: Pin Change Interrupt. The PB0 pin can serve as an external interrupt source.

Port C Alternate Function

Pin	Alternate Function
PC7	ADC8 (ADC input channel 8) APN2 (DAP invert input 2) PCINT15 (Pin change input 15)
PC6	RESETEN (Output reset input) PCINT14 (Pin change input 14)
PC5	ADC5 (ADC input channel 5) SCL （TWI clock wire） PCINT13 (Pin change input 13)
PC4	ADC4 (ACD input channel 4) SDA (TWI data cable) PCINT12 (Pin change input 12)
PC3	ADC3 (ACD input channel 3) PCINT11 (Pin change input 11)
PC2	ADC2 (ACD input channel 2) PCINT11 (Pin change input 10)
PC1	ADC1 (ACD input channel 1) PCINT9 (Pin change input 9)
PC0	ADC0 (ACD input channel 0) PCINT8 (Pin change input 8)

ADC8/APN2/PCINT15- 端口 C 引脚 6

ADC8: ADC 外部输入通道 8

APN2: 差分放大器的反向输入端口 **2**

PCINT15: 引脚电平变化中断。关闭这个引脚的外部复位输入功能后， **PC7** 可以做为外部中断源。

ADC8/APN2/PCINT15- Port C Pin 6

ADC8: ADC external input channel 8
APN2: inverting input port 2 of differential amplifier
PCINT15: pin change interrupt. If shut down the external reset input of this pin, PC7 can be used as external interrupt source.

RESETN/PCINT14- 端口 C 引脚 6

RESETN: 外部复位输入引脚。上电复位后，这个引脚默认为外部复位功能。可以通过 **IOCR** 寄存器关闭外部复位功能。关闭外部复位功能后，这个引脚可作为通用 **I/O** 使用。但需要注意的是，在上电和其他复位过程中，这个引脚默认为复位输入，所以如果用户需要用到这个引脚的通用 **I/O** 功能，外部电路不能影响到芯片的上电和复位过程，建议将这个引脚配置为输出功能的 **I/O**，并在外部加一个适当的上拉电阻。

PCINT14: 引脚电平变化中断。关闭这个引脚的外部复位输入功能后， **PC6** 可以做为外部中断源。

RESETN/PCINT14- Port C Pin 6

RESETN: External Reset Input Pin. When power on reset, this pin is set to external reset by default, which can be disable via IOCR register. After disable, pin can be used as general I/O. Note that: During power on and reset, the pin is configured as reset input by default, if use need to use general I/O of pin and external circuit cannot have effect power-on and reset of chips, it is recommend to configure the pin as I/O output and add a suitable external pull-up resistor.

PCINT14: Pin change interrupt. To disable external reset input of this pin, PC6 can be used as external interrupt source.

SCL/ADC5/PCINT13- 端口 **C** 引脚 **5**

SCL: TWI 接口时钟信号。**TWCR** 寄存器中的 **TWEN** 位置 **1** 后，使能 **TWI** 接口，**PC5** 将被 **TWI** 控制，成为 **TWI** 接口的时钟信号。

ADC5: ADC 输入通道 **5**。**DIDR** 寄存器用于关闭数模复用 **I/O** 的数字功能，以避免数字部分对模拟电路的影响。具体请参考 **ADC** 相关章节。

PCINT13: 引脚电平变化中断 **13**

SCL/ADC5/PCINT13- Port C Pin 5

SCL: TWI interface clock signal. Set TWEN of TWCR register to 1.enable TWI interface, PC5 is to be controlled by TWI and becomes TWI interface clock signal.

ADC5: ADC input channel 5. DIDIR register is used to disable digital function of Digital/ Analog Alternate I/O to avoid influence from digital part to analog circuit. For details please refer to sections related to ADC

PCINT13: Pin change interrupt 13.

SDA/ADC4/PCINT12- 端口 **C** 引脚 **4**

SDA: TWI 接口数据信号。**TWCR** 寄存器中的 **TWEN** 位置 **1** 后，使能 **TWI** 接口，**PC4** 将被 **TWI** 控制，成为 **TWI** 接口的数据信号。

ADC4: ADC 输入通道 **4**。**DIDR** 寄存器用于关闭数模复用 **I/O** 的数字功能，以避免数字部分对模拟电路的影响。具体请参考 **ADC** 相关章节。

PCINT12: 引脚电平变化中断 **12**

SDA/ADC4/PCINT12- Port C Pin 1

SDA：TWI interface digital signal. Set TWEN of TWCR register to 1. Enable TWI interface, PC4 is controlled by TWI and become digital signal of TWI interface.

ACD4: ACD input channel 5. DIDIR register is used to disable digital function of Digital/ Analog Alternate I/O to avoid influence from digital part to analog circuit. For details please refer to sections related to ADC

PCINT12: Pin change interrupt 12.

ADC3/APN1/PCINT11- 端口 **C** 引脚 **3**

ADC3: ADC 输入通道 **3**。**DIDR** 寄存器用于关闭数模复用 **I/O** 的数字功能，以避免数字部分对模拟电路的影响。具体请参考 **ADC** 相关章节。

APN1: 差分放大器反向输入 **1**

PCINT11: 引脚电平变化中断 **11**

ADC3/APN1/PCINT11- Port C Pin 3

ACD3: ACD input channel 3. DIDIR register is used to disable digital function of Digital/ Analog Alternate I/O to avoid influence from digital part to analog circuit. For details please refer to sections related to ADC

APN1: Differential amplifier inverting input 1

PCINT11: Pin change interrupt 11.

ADC2/APN0/PCINT10- 端口 **C** 引脚 **2**

ADC2: ADC 输入通道 2。DIDR 寄存器用于关闭数模复用 I/O 的数字功能，以避免数字部分对模拟电路的影响。具体请参考 ADC 相关章节。

APN0：差分放大器反向输入 0

PCINT10：引脚电平变化中断 10

ADC2/APN0/PCINT10- Port C Pin 2

ACD2: ACD input channel 2. DIDIR register is used to disable digital function of Digital/ Analog Alternate I/O to avoid influence from digital part to analog circuit. For details please refer to sections related to ADC

APN0: Differential amplifier inverting input 0

PCINT10: Pin change interrupt 10.

ADC1/APP1/PCINT9- 端口 C 引脚 1

ADC1：ADC 输入通道 1。DIDR 寄存器用于关闭数模复用 I/O 的数字功能，以避免数字部分对模拟电路的影响。具体请参考 ADC 相关章节。

APP1：差分放大器正向输入 1

PCINT9：引脚电平变化中断 9

ADC1/APP1/PCINT9- Port C Pin 1

ACD1: ACD input channel 1. DIDIR register is used to disable digital function of Digital/ Analog Alternate I/O to avoid influence from digital part to analog circuit. For details please refer to sections related to ADC

APP1: Differential amplifier non-inverting input 1

PCINT9: Pin change interrupt 9.

ADC0/APP0/PCINT8- 端口 C 引脚 0

ADC0: ADC 输入通道 0。DIDR 寄存器用于关闭数模复用 I/O 的数字功能，以避免数字部分对模拟电路的影响。具体请参考 ADC 相关章节。

APP0：差分放大器正向输入 0

PCINT8：引脚电平变化中断 8

ADC0/APP0/PCINT8- Port C Pin 0

ACD0: ACD input channel 0. DIDIR register is used to disable digital function of Digital/ Analog Alternate I/O to avoid influence from digital part to analog circuit. For details please refer to sections related to ADC

APP0: Differential amplifier non-inverting input 0

PCINT8: Pin change interrupt 8.

Port D Alternate Function

Pin	Alternate Function
PD7	ACXN (analog comparator 0/1 public negative terminal input) PCINT23 (Pin change interrupt 23)
PD6	ACOP (QFP32: analog comparator 0 positive terminal input) OCOA (Timer/Counter 0 Compare match output A) OC3A (QFP32: Timer/Counter 3 Compare match output A) PCINT22 (Pin change interrupt 22)
PD5	T1 (Timer/Counter 1 external counting clock input) OCOB (Timer/Counter 0 Compare match output B) PCINT21 (Pin change interrupt 21)
PD4	XCK (USART external clock input/output) DAO (internal 8-bit DAC analog output)

	TO (Timer/Counter 0 external counting clock input) PCINT20 (Pin change interrupt 20)
PD3	INT1 (external interrupt input 1) OC2B (Timer/ Counter 2 compare match output B) PCINT19 (Pin change interrupt 19)
PD2	INT0 (external interrupt input 0) AC00(Comparator 0 output) 0C3B (QFP32: Timer/counter 3, compare match output B) PCINT18 (Pin change interrupt 18)
PD1	TXD (USART Data output) OC3A (QFP32: Timer/ Counter 3 compare match output A) PCINT17 (Pin change interrupt 17)
PD0	RXD (USART Data input) PCINT16 (Pin change interrupt 16)

ACXN/OC2B/PCINT23- 端口 D 引脚 7

ACXN: 模拟比较器 0/1 公用负端输入

OC2B: 定时/计数器 2 的 B 组比较匹配输出。 PD7 可以作为定时/计数器 2 比较匹配的外部输出。此时必须通过 DDD7 将引脚设置为输出。同时，OC2B 也是定时器 2 的 PWM 模式输出引脚;

PCINT23: 引脚电平变化中断 23

ACXN/OC2B/PCINT23- Port D Pin 7

ACXN: Analog Comparator 0/1 public negative terminal input

OC2B: Time/Counter 2 compare match B. When PD7 is used as external output of Timer/ Counter 2 compare match, pin must be set to output via DDD7. OC2B is also output pin of Timer 2 PWM mode.

PCINT23: Pin change interrupt 23

AC0P/OC0A/PCINT22- 端口 D 引脚 6

AC0P: 模拟比较器 0 正端输入。

OC0A: 定时/计数器 0 的 A 组比较匹配输出。 PD6 可以作为定时/计数器 0 比较匹配的外部输出。此时必须通过 DDD6 将引脚设置为输出。同时，OC0A 也是定时器 0 的 PWM 模式输出引脚

PCINT22: 引脚电平变化中断 22

AC0P/OC0A/PCINT22- Port D Pin 6

AC0P: analog comparator 0 positive terminal input

OC0A: Timer/ Counter 0 compare match output A. When PD6 is used as external output of Timer/ Counter 0 compare match, pin must be set to output via DDD6. OC0A is also output pin of Timer 0 PWM mode.

PCINT23: Pin change interrupt 22

T1/OC0B/PCINT21- 端口 D 引脚 5

T1: 定时/计数器 1 的外部计数时钟输入

OC0B: 定时/计数器 0 的 B 组比较匹配输出。 PD5 可以作为定时/计数器 0 比较匹配的外部输出。此时必须通过 DDD5 将引脚设置为输出。同时，OC0B 也是定时器 0 的 PWM 模式输出引脚

PCINT21: 引脚电平变化中断 21

T1/OC0B/PCINT21- Port D Pin 5

T1: Timer/ Counter 1 external counting clock input
OCOB: Timer/ Counter 0 compare match output B. When PD5 is used as external output of Timer/ Counter 0 compare match, pin must be set to output via DDD5. OC0B is also output pin of Timer 0 PWM mode.
PCINT23: Pin change interrupt 21

XCK/T0/DAO/PCINT20- 端口 D 引脚 4
XCK: 同步模式 USART 的外部时钟信号
T0: 定时/计数器 0 的外部计数时钟输入
DAO: 内部 8 位 DAC 模拟输出
PCINT20: 引脚电平变化中断 20

XCK/T0/DAO/PCINT20- Port D Pin 4
XCK: external clock signal of USART in synchronize mode
T0: Timer/Counter 0 external counting clock input
DAO: internal 8-bit DAC analog output
PCINT20: Pin change interrupt 20

INT1/OC2B/PCINT19- 端口 D 引脚 3
INT1: 外部中断输入 1
OC2B: 定时/计数器 2 的 B 组比较匹配输出。 PD3 可以作为定时/计数器 2 比较匹配的外部输出。此时必须通过 DDD3 将引脚设置为输出。同时，OC2B 也是定时器 2 的 PWM 模式输出引脚
PCINT19: 引脚电平变化中断 19

INT1/OC2B/PCINT19-Port D Pin 3

INT1: external interrupt input 1
OC2B: Timer/ Counter 2 compare match output B. When PD3 is used as external output of Timer/ Counter 2 compare match, pin must be set to output via DDD3. OC2B is also output pin of Timer 2 PWM mode.
PCINT19: Pin change interrupt 19

INT0/OC3B/AC0O/PCINT18- 端口 D 引脚 2
INT0: 外部中断输入 0
OC3B: 定时计数器 3 比较匹配输出 B。 仅在 QFP32 封装时， PD2 与 QFP48/PF2 合并成一个 IO，因此 PF2 上的 OC3B 功能也将从 PD2 上输出
AC0O: 模拟比较器 0 比较结果直接输出。由 AC0FR 寄存器控制
PCINT18: 引脚电平变化中断 18

INT0/OC3B/AC0O/PCINT18- Port D Pin 2

INT1: external interrupt input 1
OC3B: Timer/ Counter 3 compare match output B. Only in QFP32 packing, PD2 and QFP48/PF2 are combined to one IO, and OC3B on PF2 is output from PD2
PCINT18: Pin change interrupt 18

TXD/OC3A/PCINT17- 端口 D 引脚 1
TXD: 传输数据(USART 数据输出)。USART 发送器使能后，PD1 将被强制为输出，不受 DDD1 的控制
OC3A: 定时计数器 3 比较匹配输出 A。 仅在 QFP32 封装时， PD1 与 QFP48/PF1 合并成一个 IO，因此 PF1 上的 OC3A 功能也将从 PD1 上输出
PCINT17: 引脚电平变化中断 17

TXD/OC3A/PCINT17- Port D Pin 1

TXD: Data transfer (USART data output). When USART transmitter is enable, PDI is forced to output regardless of setting of DDD1.

OC3A: Timer/ Counter 3 compare match output A. Only in QFP32 packing, PD2 and QFP48/PF1 are combined to one IO, and OC3A on PF1 is output from PD2

PCINT17: Pin change interrupt 17

RXD/PCINT16- 端口 D 引脚 0

RXD: 传输数据(USART 数据输入)。USART 接收器使能后，PD0 将被强制为输入，不受 DDD0 的控制。当引脚被 USART 强制为输入后，上拉电阻仍然可以通过 PORTD0 位控制

PCINT16: 引脚电平变化中断 16

RXD/PCINT16- Port D Pin 0

RXD: Data transfer (USART data input). When USART receiver is enable, PD0 is forced to input regardless of setting of DDD0. If pin is forced to input by USART, pull-up resistor is still control by PORTD0

PCINT16: Pin change interrupt 16

Port E Alternate Function

PIN	Alternate Function Description
PE7	ADC11 (ADC input channel 11) PCINT31 (Pin change interrupt 31)
PE6	AVREF (QFP32: ADC external referrence voltage) ADC10 (ADC input channel 10) PCINT30 (Pin change interrupt 30)
PE5	CLK0 (system clock output) AC10 （Analog Comparator 1 output) PCINT29 (Pin change interrupt 29)
PE4	OCOA (Timer/ Counter 0 Compare Configuration Output A) PCINT28 (Pin change interrupt 28)
PE3	ADC7 (ADC input channel 7) AC1N (Analog Comparator 1 Negative Terminal Input) PCINT27 (Pin change interrupt 27)
PE2	SWD (SWD Debugger Data Cable) PCINT26 (Pin change interrupt 26)
PE1	ADC6 (ADC input channel 6) ACXP (Analog Comparator 0/1 Public Positive Ternimal Input PCINT25(Pin change interrupt 25)
PE0	SWC (SWD Debugger Clock Input APN4 (Differential Amplifier Inverting Input 4) PCINT24(Pin change interrupt 24)

ADC11 /PCINT3 1 - 端口 E 引脚 7

ADC11: ADC 外部输入通道 11

PCINT31: 引脚电平变化中断 30

ADC11 /PCINT3 1 - Port E Pin 7
ADC11: ADC external input channel 11
PCINT31: Pin change interrupt 30

A VREF/ ADC10 / PCINT30 - 端口 E 引脚 6
AVREF: ADC 外部参考电源输入，用作模拟功能时，需要将对应的数字 I/O 设置为输入， 并关闭上拉电阻，以避免数字电路对模拟电路产生干扰
ADC10: ADC 模拟输入通道 10
PCINT30: 引脚电平变化中断 30

A VREF/ ADC10 / PCINT30 - Port E Pin 6
AVREF: ADC external reference voltage input. If used for analog, corresponding digital I/O should be set to input and shut down pull-up to avoid intervene of digital circuit on analog circuit.

CLKO/ AC1O / PCINT29 - 端口 E 引脚 5
CLKO: 此功能与 PB0 的 CLKO 功能相同。可作为 PB0/CLKO 的备用引脚
AC1O: 模拟比较器 1 输出
PCINT29: 引脚电平变化中断 29

CLKO/ AC1O / PCINT29 - Port E Pin 5
CLKO: function is same as PB0 of CLKO, can used as copy pin of PB0/CLKO
AC10: Analog Comparator 1 output
PCINT29: Pin change interrupt 29

OC0A/PCINT28 - 端口 E 引脚 4
OC0A: 定时/计数器 0 的 A 组比较匹配输出。PE4 可以作为定时/计数器 0 比较匹配的外部输出。此时必须通过 DDE4 将引脚设置为输出。同时，OC0A 也是定时器 0 的 PWM 模 式输出引脚。
PCINT28: 引脚电平变化中断 28

OC0A/PCINT28 - Port E Pin 4
OC0A: Time/Counter 0 Compare Match Output A. If set PE4 as external output of Timer/Counter 0 compare match, pin must be set to output via DDE4. Meanwhile OCOA is output pin of Timer 0 PWM mode.
PCINT28: Pin change interrupt 28

ADC7/ AC1N/ PCINT27 - 端口 E 引脚 3
ADC7: ADC 输入通道 7。DIDR 寄存器用于关闭数模复用 I/O 的数字功能，以避免数字部 分对模拟电路的影响。具体请参考 ADC 相关章节
AC1N: 模拟比较器 1 负端输入
PCINT27: 引脚电平变化中断 27
ADC7/ AC1N/ PCINT27 - Port E Pin 3
ADC7: ADC input channel 7. DIDR register is used to shut down digital function of digital moducle alternate I/O, in order that digital part do not influence analog circuit. For details please refer to related section about ADC
AC1N: Analog comparator 1 negative ternimal input
PCINT27: Pin change interrupt 27

SWD/PCINT26 - 端口 E 引脚 2
SWD: SWD 调试器数据线。PE2 默认为 SWD 功能。用户可以通过将 MCUSR 寄存器 SWDD 位置 1 关闭 SWD 调试器功能。SWD 被关闭后，调试功

能将不能使用。

PCINT26: 引脚电平变化中断 26

SWD/PCINT26 - Port E Pin 2

SWD: SWD Debugger data cable. PE is set to SWD function by detault. Write logic one to SWDD bit of MCUSR register to shut down SWD debugger function, than debug function can not be used.

PCINT26: Pin change interrupt 26

ADC6/ ACXP / PCINT25 - 端口 E 引脚 1

ADC6: ADC 输入通道 6。DIDR 寄存器用于关闭数模复用 I/O 的数字功能，以避免数字部 分对模拟电路的影响。具体请参考 ADC 相关章节

ACXP: 模拟比较器 0/1 公用正端输入

PCINT25: 引脚电平变化中断 25

ADC6/ ACXP / PCINT25 - PORT E Pin 1

ADC6: ADC input channel 6. DIDR register is used to shut down digital function of digital moducle alternate I/O, in order that digital part do not influence analog circuit. For details please refer to related section about ADC

ACXP: Analog comparator 0/1 public positive ternimal input

PCINT25: Pin change interrupt 25

SWC/ APN4 / PCINT24 - 端口 E 引脚 0

SWC: SWD 调试器时钟线。PE0 默认为 SWC 功能。用户可以通过将 MCUSR 寄存器 SWDD 位置 1 关闭 SWD 调试器功能。SWD 被关闭后，调试功能将不能使用

APN4: 差分放大器反向输入 4

PCINT24: 引脚电平变化中断 24

SWC/ APN4 / PCINT24 - Port E Pin 0

SWC: SWD debugger clock cable. PE0 is set to SWC by default. SWD debugger is shut down if write logic one to SWDD bit of MSUSR register, than debug function cannot be used.

APN4: Differential amplifier inverting input 4

PCINT24: Pin change interrupt 24

Terminal E Alternate Function

Pin	Alternat Function Description
PF7	OC2B (Timer/Counter 2 compare match output B) PCINT39 (Pin change interrupt 39)
PF6	T3 (Timer/ Counter 3 external clocj input) OC2A (Timer/Counter 2 compare match output A) PCINT38 (Pin change interrupt 38)
PF5	OC12A (Timer/Counter 1 compare match output A) PCINT37 (Pin change interrupt 37)
PF4	OC1B (Timer/Counter 1 compare configuration output B) ICP3 (Timer/ Counter 3 external capture input) PCINT36 (Pin change interrupt 36)
PF3	OC0B (Timer/Counter 0 compare configuration output B)

	PCINT35 (Pin change interrupt 35)
PF2	OC3B (Timer/Counter 3 compare configuration output B) PCINT34 (Pin change interrupt 34)
PF1	OC3A (Timer/Counter 3 compare configuration output A) PCINT33 (Pin change interrupt 33)
PF0	ADC9 (ADC external input channel 9) APN3 (Differential Amplifier Inverting Input 3) PCINT32 (Pin change interrupt 32)

OC2B /PCINT3 9 - 端口 F 引脚 7

OC2B: 定时/计数器 2 比较匹配输出 B。输出选择受 PMX1 寄存器控制

PCINT39: 引脚电平变化中断 39

OC2B /PCINT3 9 - Port F Pin 7

OC2B: Timer/Counter 2 compare match output B, output is decided by PMX1 register

PCINT39 (Pin change interrupt 39)

OC2A/T3 / PCINT3 8 - 端口 F 引脚 6

OC2A: 定时/计数器 2 比较匹配输出 A。输出选择受 PMX1 寄存器控制

T3: 定时/计数器 3 外部时钟输入

PCINT38: 引脚电平变化中断 38

OC2A/T3 / PCINT3 8 - Port F Pin 6

OC2A: Timer/Counter 2 compare match output A, output is decided by PMX1 register

T3: Timer/Counter 3 external clock input

PCINT38 (Pin change interrupt 38)

0C1A / PCINT 37 - 端口 F 引脚 5

OC1A: 定时/计数器 1 比较匹配输出 A。输出选择受 PMX0 寄存器控制

PCINT37: 引脚电平变化中断 37

0C1A / PCINT 37 - Port F Pin 5

OC1A: Timer/Counter 1 compare match output A, output is decided by PMX0 register

PCINT37: (Pin change interrupt 37)

ICP3 / OC1B/ PCINT 36 - 端口 F 引脚 4

OC1B: 定时/计数器 1 的 B 组比较匹配输出。输出选择受 PMX0 寄存器控制

ICP3: 定时/计数器 3 外部俘获输入

PCINT36: 引脚电平变化中断 36

ICP3 / OC1B/ PCINT 36 - Port F Pin 4

OC1B: Timer/Counter 1 compare match output B, output is decided by PMX0 register

ICP3: Timer/Counter 3 external capture input

PCINT36: (Pin change interrupt 36)

OC3C / OC0B / PCINT 35 - 端口 F 引脚 3

OC0B: 定时/计数器 0 的 B 组比较匹配输出。输出选择受 PMX0 寄存器控制

OC3C: 定时/计数器 3 的 C 组比较匹配输出

PCINT35: 引脚电平变化中断 35

OC3C / OC0B / PCINT 35 - Port F Pin 3

OC0B: Timer/Counter 0 compare match output B, output is decided by PMX0 register

OC3C: Timer/Counter 3 compare match output C

PCINT35: Pin change interrupt 35

OC3B /PCINT 34 - 端口 F 引脚 2

OC3B: 定时/计数器 3 的 B 组比较匹配输出

PCINT34: 引脚电平变化中断 34

OC3B /PCINT 34 - Port F Pin 2

OC3B: Timer/Counter 3 compare match output B

PCINT34: Pin change interrupt 34

OC3A / PCINT 33 - 端口 F 引脚 1

OC3A: 定时/计数器 3 的 B 组比较匹配输出。输出选择受 PMX1 寄存器控制

PCINT33: 引脚电平变化中断 33

OC3A / PCINT 33 - Port F Pin 1

OC3A: Timer/Counter 3 compare match output B, output is decided by PMX1 register

PCINT33: Pin change interrupt 33

ADC9 / APN 3 / PCINT 32 - 端口 F 引脚 0

ADC9: ADC 外部模式输入通道 9

APN3: 差分放大器反向输入 3

PCINT32: 引脚电平变化中断 32

ADC9 / APN 3 / PCINT 32 - Port F Pin 0

ADC9: ADC external mode input channel 9

APN3: Differential amplifier inverting input 3

PCINT32: Pin change interrupt 32

Register Definitioin

Port B output data register-PORTB

Port B output data register-PORTB								
PORTB: 0x05(0x25)					Default Value: 0x00			
Bit	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit definition								
[7:0]	PORTB	Port B output register						

Port B direction register-DDRB

Port B direction register-DDRB								
DDRB: 0x04(0x24)					Default Value: 0x00			
DDRB	DDB7	DDB6	DDB5	DDB4	DDB3	DDB2	DDB1	DDB0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit definition								
[7:0]	DDB	Port B direction register;1=output, 0= input						

Port B Input Data Register- PINB

Port B Input Data Register- PINB								
PINB: 0x03(0x23)					Default Value: 0x00			
PINB	PINB7	PINB6	PINB5	PINB4	PINB3	PINB2	PINB1	PINB0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit definition								
[7:0]	PINB	Port B status register. By reading PINB to get current port status directly. Write logic one to PINBn to flip output status of PORTBn						

Port C Output Data Register-- PORTC

Port C Output Data Register-- PORTC								
PORTC: 0x08(0x28)					Default Value: 0x00			
PORTC	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit definition								
[7:0]	PORTC	Port C output register						

Port C direction register--DDRC

Port C direction register--DDRC								
DDRC: 0x07(0x27)					Default Value: 0x00			
DDRC	DDC7	DDC6	DDC5	DDC4	DDC3	DDC2	DDC1	DDC0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit definition								
[7:0]	DDC	Port C direction control bit, 1= output, 0= input						

Port C input data register--PINC

Port C input data register--PINC								
PINB: 0x06(0x26)					Default Value: 0x00			
PINC	PINC7	PINC6	PINC5	PINC4	PINC3	PINC2	PINC1	PINC0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit definition								
[7:0]	PINC	Port C direction control bit, 1= output, 0= input						

Port D output data register-PORTD

Port D output data register-PORTD

PORTD: 0x0B(0x2B)					Default Value: 0x00			
Bit	PD7	PD6	PD5	PD4	PD3	PD2	PD1	PD0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit definition								
[7:0]	PORTD	Port D output register						

Port D dirction register- DDRD

Port C input data register--PINC								
DDRD: 0x0A(0x2A)					Default Value: 0x00			
DDRD	DDD7	DDD6	DDD5	DDD4	DDD3	DDD2	DDD1	DDD0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit definition								
[7:0]	DDD	Port D output direction control register						

Port D input data register- PIND

Port D input data register- PIND								
PIND: 0x09(0x29)					Default Value: 0x00			
PIND	PIND7	PIND6	PIND5	PIND4	PIND3	PIND2	PIND1	PIND0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit definition								
[7:0]	PIND	Port D status register Reading PIND to get current port level staus Write logic one to PINDn to flip corresponding bit status of PORTDn						

Port E output data register--PORTE

Port E output data register--PORTE								
PORTE: 0x0E(0x2E)					Default Value: 0x00			
Bits	PE7	PE6	PE5	PE4	PE3	PE2	PE1	PE0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit definition								
[7:0]	PINC	Port E output register						

Port E direction register-- DDRE

Port E direction register-- DDRE								
DDRE: 0x0D(0x2D)					Default Value: 0x00			
DDRE	DDE7	DDE6	DDE5	DDE4	DDE3	DDE2	DDE1	DDE0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit definition								
[7:0]	DDE	Port E direction control register						

Port E input data register--PINE

Port E input data register--PINE								
PINE: 0x0C(0x2C)					Default Value: 0x00			
PINE	PINE7	PINE6	PINE5	PINE4	PINE3	PINE2	PINE1	PINE0

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit definition								
[7:0]	PINE	Port E status register Reading PINE to get current port level Write logic one to PINEn to flip status of PORTEn bit						

Port F output register- PORTF

Port F output register- PORTF								
PORTF: 0x14(0x34)					Default Value: 0x00			
Bit	PF7	PF6	PF5	PF4	PF3	PF2	PF1	PF0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit definition								
[7:0]	PINE	Port F status register Port of input mode, write one to enable internal pull-up Port of output mode, write one to drive output high level						

Port F direction control register--DDRF

Port F direction control register--DDRF								
DDRF: 0x13(0x33)					Default Value: 0x00			
Bit	DDF7	DDF6	DDF5	DDF4	DDF3	DDF2	DDF1	DDF0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit definition								
[7:0]	DDF	Port F directioon control register						

Port F status register--PINF

Port F status register--PINF								
PINF: 0x12(0x32)					Default Value: 0x00			
Bit	PINF7	PINF6	PINF5	PINF4	PINF3	PINF2	PINF1	PINF0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit definition								
[7:0]	DDF	Port F status register Reading PINF to get current port F level Write logic one to PINFn to flip status of PORTFn bit						

Port drive control register—HDR

Port drive control register—HDR0								
HDR: 0xE0					Default Value: 0x00			
Bit	-	-	HDR5	HDR4	HDR3	HDR2	HDR1	HDR0
R/W	-	-	R/W	R/W	R/W	R/W	R/W	R/W
Bit definition								
[7:0]	-	Reserved						
5	HDR5	PF5 output drive control; 1=80mA drive, 0=12mA drive						
4	HDR4	PF4 output drive control; 1=80mA drive, 0=12mA drive						

3	HDR3	PF2 output drive control; 1=80mA drive, 0=12mA drive
2	HDR2	PF1 output drive control; 1=80mA drive, 0=12mA drive
1	HDR1	PF6 output drive control; 1=80mA drive, 0=12mA drive
0	HDR0	PF5 output drive control; 1=80mA drive, 0=12mA drive

Port Alternate Control Register—0-PMX0

Port Alternate Control Register—0-PMX0								
PMX0: 0xEE					Default Value: 0x00			
Bit	WCE	C1BF4	C1AF5	C0BF3	C0AC0	SSB1	TXD6	RXD5
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit definition								
7	WCE	PMX0/1 update enable; before update, must write logic one to WCE, in the following 6 system clock update will be finished.						
6	C1BF4	OC1B Auxiliary output control 1=OC1B output to PF4 0=OC1B output to PB2						
5	C1AF5	OC1A Auxiliary output control 1=OC1A output to PF5 0=OC1A output to PB1						
4	C0BF3	OC1B Auxiliary output control 1=OC0B output to PF3 0=OC0B output to PD5						
3	C0AC0	OC0A Auxiliary output control OCOA is co-controlled by COACO bit and TCCROB register COAS bit. { C0AC0, C0AS} = 00=OCOA output to PD6 01=OCOA output to PE4 10=OCOA output to PC0 11=OCOA simultaneous output to PE4 and PC0						
2	SSB1	SPSS Auxiliary output control 1= SPSS output to PB1 0= SPSS output to PB2						
1	TXD6	Serial port TXD Auxiliary output 1=TXD output to PD6 0= TXD output to pd1						
0	RXD5	Serial port RXD Auxiliary output 1=RXD input from PD5 0= RXD input from PD0						

Port Multiplex Control Register 1—PMX1

Port Multiplex Control Register 1—PMX1								
PMX1: 0xED					Default Value: 0x00			
Bit	-	-	-	-	-	C3AC	C2BF7	C2AF6
R/W	-	-	-	-	-	R/W	R/W	R/W

Bit definition		
[7:3]	-	Reserved
2	C3AC	OC3A Auxiliary output control 1=OC3A output to QFP48/AC0P 0=OC3A output to PF1
1	C2BF7	OC2B Auxiliary output control 1=OC2B output to PF7 0=OC2B output to PD3
0	C2AF6	OC2A Auxiliary output control 1=OC2A output to PF6 0=OC2A output to PB3
Remarks:		
PMX0/1 share PMX0[7], register update protection control bit. How to update PMX1, please refer to section about PMX0 register control to PMX0[7]		

Port Multiplex Control Register 2—PMX2

Port Multiplex Control Register 2—PMX2								
PMX2: 0xF0					Default Value: 0x00			
Bit	WCE	STSC1	STSC0	-	-	XIEN	E6EN	C6EN
R/W	R/W	R/W	R/W	-	-	R/W	R/W	R/W
Bit definition								
[7:3]	-	Reserved						
[7]	WCE	OC3A Auxiliary output control 1=OC3A output to QFP48/AC0P 0=OC3A output to PF1						
[6]	STSC1	High speed crystal oscillator IO enable circuit control When PMCR enable high speed oscillator, STSC1 is enable automatically. When system clock switch to external high speed oscillator, STSCI is clear automatically. When crystal oscillator is stable, software can clear STSC1 manually, shut down crystal oscillator power-on circuit to save power consumption.						
[5]	STSC0	Low speed crystal oscillator IO enable circuit control When PMCR enable low speed oscillator, STSC0 is enable automatically. When system clock switch to external low speed oscillator, STSC0 is clear automatically. When crystal oscillator is stable, software can clear STSC0 manually, shut down crystal oscillator power-on circuit to save power consumption.						
[4:3]	-	reserved						
[2]	XIEN	Enable external clock input meanwhile enable external crystal oscillator						
[1]	E6EN	Enable general IO of PE6, PE6 is set AVREF by default						
[0]	C6EN	Enable general IO of PC6, PC6 is set external reset by default						

引脚电平变化中断

- 40 个引脚电平变化中断源
- 5 个中断入口

综述

引脚电平变化中断由 PBn， PCn， PDn, PEn 和 PFn 引脚触发。只要引脚电平变化中断被使能，即使这些引脚配置为输出也能触发中断。这可以用来产生软件中断。

任何一个使能的 PBn 引脚翻转都会触发引脚电平中断 PCI0，使能的 PCn 引脚翻转将触发 PCI1，使能的 PDn 引脚翻转将触发 PCI2，使能的 PEn 引脚翻转将触发 PCI3。各个引脚变化中断的使能分别由 PCMSK0~4 寄存器来控制。所有的引脚电平变化中断都是异步检测的，可用作某些睡眠模式下的唤醒源。

Pin Change Interrupt

40 pcs pin change interrupt source

5 pcs interrupt inlet

Overview：

The External Interrupts are triggered by PBn， PCn， PDn, PEn and PFn. Observe that, if enabled, the interrupts will trigger even if these pins configured as outputs, this feature provides a way of generating a software interrupt. The pin change interrupt PCI0 will trigger if any enabled PBn pin toggles. The pin change interrupt PDn will trigger if any enabled PC12 pin toggles. The pin change interrupt PC13 will trigger if any enabled PEn pin toggles. The PCMSK0~4 Registers control enable of each pin change interrupt. All pin change interrupts are detected asynchronously. This implies that these interrupts can be used for waking the part also from sleep modes.

Register Definition

Register	Address	Default Value	Comments
PCICR	0x68	0x00	Pin Change Interrupt Control Register
PCIFR	0x3B	0x00	Pin Change Interrupt Flag Register
PCMSK0	0x6B	0x00	Pin Change Mask Register 0
PCMSK1	0x6C	0x00	Pin Change Mask Register 1
PCMSK2	0x6D	0x00	Pin Change Mask Register 2
PCMSK3	0x73	0x00	Pin Change Mask Register 3
PCMSK4	0x74	0x00	Pin Change Mask Register 4

PCICR- Pin Change Interrupt Control Register

PCICR- Pin Change Interrupt Control Register								
Address: 0x68					Default Value: 0x00			
Bit	7	6	5	4	3	2	1	0
Name	-	-	-	PCIE4	PCIE3	PCIE2	PCIE1	PCIE0
R/W	-	-	-	R/W	R/W	R/W	R/W	R/W
Bit	Name	Description						
7:5	-	Reserved						
4	PCIE4	Pin change interrupt enable control bit 4 When PCIE4 is set to logic one and global interrupt is enable, pin change interrupt 4 is enable. Pin level change of any enabled PFn pin, controlled by PCMSK4 register, will generate PCI4 interrupt. When PCIE3 is set to logic zero, pin change interrupt 3 is disable						
3	PCIE3	Pin change interrupt enable control bit 3						

		When PCIE3 is set to logic one and global interrupt is enable, pin change interrupt 3 is enable. Pin level change of any enabled PFn pin, controlled by PCMSK3 register, will generate PCI3 interrupt. When PCIE3 is set to logic zero, pin change interrupt 3 is disable
2	PCIE2	Pin change interrupt enable control bit 2 When PCIE2 is set to logic one and global interrupt is enable, pin change interrupt 2 is enable. Pin level change of any enabled PDn pin, controlled by PCMSK2 register, will generate PCI2 interrupt. When PCIE2 is set to logic zero, pin change interrupt 2 is disable
1	PCIE1	Pin change interrupt enable control bit 1 When PCIE1 is set to logic one and global interrupt is enable, pin change interrupt 1 is enable. Pin level change of any enabled PCn pin, controlled by PCMSK1 register, will generate PCI1 interrupt. When PCIE1 is set to logic zero, pin change interrupt 1 is disable
0	PCIE0	Pin change interrupt enable control bit 0 When PCIE0 is set to logic one and global interrupt is enable, pin change interrupt 0 is enable. Pin level change of any enabled PBn pin, controlled by PCMSK0 register, will generate PCI0 interrupt. When PCIE0 is set to logic zero, pin change interrupt 0 is disable

PCIFR- Pin Change Interrupt Flag Register

PCIFR- Pin Change Interrupt Flag Register								
Address: 0x3B					Default Value: 0x00			
Bit	7	6	5	4	3	2	1	0
Name	-	-	-	PCIF4	PCIF3	PCIF2	PCIF1	PCIF0
R/W	-	-	-	R/W	R/W	R/W	R/W	R/W
Bit	Name	Description						
7:5	-	Reserved						
4	PCIF4	Pin change interrupt Flag bit 4 PCIF4 is to be set when any PFn pin change is enabled. When PCIE4 and global interrupt are set, MCU will jump to inlet address of PCI4 interrupt. PFn pin interrupt is enable via PCMSK4 register. To execute interrupt routine program or to write logic one to PCIF4 will clear PCIF4						
3	PCIF3	Pin change interrupt Flag bit 3 PCIF3 is to be set when any PEn pin change is enabled. When PCIE3 and global interrupt are set, MCU will jump to inlet address of PCI3 interrupt. PEn pin interrupt is enable via PCMSK3 register. To execute interrupt routine program or to write logic one to PCIF3 will clear PCIF3						
2	PCIF2	Pin change interrupt Flag bit 2 PCIF2 is to be set when any PDn pin change is enabled. When PCIE2 and global interrupt are set, MCU will jump to inlet address of PCI2 interrupt. PEn pin interrupt is enable via PCMSK2 register. To execute interrupt routine program or to write logic one to PCIF2 will clear PCIF2						
1	PCIF1	Pin change interrupt Flag bit 1 PCI1 is to be set when any PCn pin change is enabled. When PCIE1 and global						

		interrupt are set, MCU will jump to inlet address of PCI1 interrupt. PCn pin interrupt is enable via PCMSK1 register. To execute interrupt routine program or to write logic one to PCIF1 will clear PCIF1
0	PCIF0	Pin change interrupt Flag bit 0 PCI1 is to be set when any PBn pin change is enabled. When PCIE0 and global interrupt are set, MCU will jump to inlet address of PCI0 interrupt. PBn pin interrupt is enable via PCMSK0 register. To execute interrupt routine program or to write logic one to PCIF0 will clear PCIF0

PCMSKO- Pin change interrupt mask register 0

PCMSKO- Pin change interrupt mask register 0								
Address: 0x6B					Default Value: 0x00			
Bit	7	6	5	4	3	2	1	0
Name	PCINT7	PCINT6	PCINT5	PCINT4	PCINT3	PCINT2	PCINT1	PCINT0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	Name	Description						
7	PCINT7	Pin change enable mask bit 7 PCINT7 is to be set to logic one, PB7 pin change interrupt is enabled, and pin change of PB7 is set to PCIF0. If PCIE0 and global bit are set, PCI0 interrupt will generate. When set PCINT7 to logic zero, PB7 pin change interrupt is disable.						
6	PCINT6	Pin change enable mask bit 6 PCINT6 is to be set to logic one, PB6 pin change interrupt is enabled, and pin change of PB6 is set to PCIF0. If PCIE0 and global bit are set, PCI0 interrupt will generate. When set PCINT6 to logic zero, PB6 pin change interrupt is disable.						
5	PCINT5	Pin change enable mask bit 5 PCINT5 is to be set to logic one, PB5 pin change interrupt is enabled, and pin change of PB5 is set to PCIF0. If PCIE0 and global bit are set, PCI0 interrupt will generate. When set PCINT5 to logic zero, PB5 pin change interrupt is disable.						
4	PCINT4	Pin change enable mask bit 4 PCINT4 is to be set to logic one, PB4 pin change interrupt is enabled, and pin change of PB4 is set to PCIF0. If PCIE0 and global bit are set, PCI0 interrupt will generate. When set PCINT4 to logic zero, PB4 pin change interrupt is disable.						
3	PCINT3	Pin change enable mask bit 3 PCINT3 is to be set to logic one, PB3 pin change interrupt is enabled, and pin change of PB3 is set to PCIF0. If PCIE0 and global bit are set, PCI0 interrupt will generate. When set PCINT3 to logic zero, PB3 pin change interrupt is disable.						
2	PCINT2	Pin change enable mask bit 2 PCINT2 is to be set to logic one, PB2 pin change interrupt is enabled, and pin change of PB2 is set to PCIF0. If PCIE0 and global bit are set, PCI0 interrupt will generate. When set PCINT2 to logic zero, PB2 pin change interrupt is disable.						
1	PCINT1	Pin change enable mask bit 1 PCINT1 is to be set to logic one, PB1 pin change interrupt is enabled, and pin change of PB1 is set to PCIF0. If PCIE0 and global bit are set, PCI0 interrupt will generate. When set PCINT1 to logic zero, PB1 pin change interrupt is disable.						
0	PCINT0	Pin change enable mask bit 0						

		PCINT0 is to be set to logic one, PB0 pin change interrupt is enabled, and pin change of PB0 is set to PCIF0. If PCIE0 and global bit are set, PCI0 interrupt will generate. When set PCINT0 to logic zero, PB0 pin change interrupt is disable.

PCMSK1—Pin change interrupt mask register 1

PCMSK1—Pin change interrupt mask register 1								
Address: 0x6c					Default Value: 0x00			
Bit	7	6	5	4	3	2	1	0
	PCINT15	PCINT14	PCINT13	PCINT12	PCINT11	PCINT10	PCINT9	PCINT8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Description
7	PCINT15	Pin change enable mask bit 15 PCINT15 is to be set to logic one, PC7 pin change interrupt is enabled, and pin change of PC7 is set to PCIF1. If PCIE1 and global bit are set, PCI1 interrupt will generate. When set PCINT15 to logic zero, PC7 pin change interrupt is disable.
6	PCINT14	Pin change enable mask bit 14 PCINT14 is to be set to logic one, PC6 pin change interrupt is enabled, and pin change of PC6 is set to PCIF1. If PCIE1 and global bit are set, PCI1 interrupt will generate. When set PCINT16 to logic zero, PC6 pin change interrupt is disable.
5	PCINT13	Pin change enable mask bit 13 PCINT13 is to be set to logic one, PC5 pin change interrupt is enabled, and pin change of PC5 is set to PCIF1. If PCIE1 and global bit are set, PCI1 interrupt will generate. When set PCINT15 to logic zero, PC5 pin change interrupt is disable.
4	PCINT12	Pin change enable mask bit 12 PCINT12 is to be set to logic one, PC4 pin change interrupt is enabled, and pin change of PC4 is set to PCIF1. If PCIE1 and global bit are set, PCI1 interrupt will generate. When set PCINT12 to logic zero, PC4 pin change interrupt is disable.
3	PCINT11	Pin change enable mask bit 11 PCINT11 is to be set to logic one, PC3 pin change interrupt is enabled, and pin change of PC3 is set to PCIF1. If PCIE1 and global bit are set, PCI1 interrupt will generate. When set PCINT11 to logic zero, PC3 pin change interrupt is disable.
2	PCINT10	Pin change enable mask bit 2 PCINT10 is to be set to logic one, PC2 pin change interrupt is enabled, and pin change of PC2 is set to PCIF1. If PCIE1 and global bit are set, PCI1 interrupt will generate. When set PCINT10 to logic zero, PC2 pin change interrupt is disable.
1	PCINT9	Pin change enable mask bit 1 PCINT9 is to be set to logic one, PC1 pin change interrupt is enabled, and pin change of PC1 is set to PCIF1. If PCIE1 and global bit are set, PCI1 interrupt will generate. When set PCINT9 to logic zero, PC1 pin change interrupt is disable.
0	PCINT8	Pin change enable mask bit 0 PCINT8 is to be set to logic one, PC0 pin change interrupt is enabled, and pin change of PC0 is set to PCIF1. If PCIE1 and global bit are set, PCI1 interrupt will generate. When set PCINT8 to logic zero, PC0 pin change interrupt is disable.

PCMSK2—Pin change interrupt mask register 2

PCMSK1—Pin change interrupt mask register 2								
Address: 0x6D					Default Value: 0x00			
Bit	7	6	5	4	3	2	1	0
	PCINT23	PCINT22	PCINT24	PCINT25	PCINT26	PCINT27	PCINT28	PCINT29
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Description
7	PCINT23	Pin change enable mask bit 23 PCINT23 is to be set to logic one, PD7 pin change interrupt is enabled, and pin change of PD7 is set to PCIF2. If PCIE2 and global bit are set, PCI2 interrupt will generate. When set PCINT23 to logic zero, PD7 pin change interrupt is disable.
6	PCINT22	Pin change enable mask bit 6 PCINT22 is to be set to logic one, PD6 pin change interrupt is enabled, and pin change of PD6 is set to PCIF2. If PCIE2 and global bit are set, PCI2 interrupt will generate. When set PCINT22 to logic zero, PD6 pin change interrupt is disable.
5	PCINT21	Pin change enable mask bit 21 PCINT21 is to be set to logic one, PD5 pin change interrupt is enabled, and pin change of PD5 is set to PCIF2. If PCIE2 and global bit are set, PCI2 interrupt will generate. When set PCINT21 to logic zero, PD5 pin change interrupt is disable.
4	PCINT20	Pin change enable mask bit 20 PCINT20 is to be set to logic one, PD4 pin change interrupt is enabled, and pin change of PD4 is set to PCIF2. If PCIE2 and global bit are set, PCI2 interrupt will generate. When set PCINT20 to logic zero, PD4 pin change interrupt is disable.
3	PCINT19	Pin change enable mask bit 19 PCINT19 is to be set to logic one, PD3 pin change interrupt is enabled, and pin change of PD3 is set to PCIF2. If PCIE2 and global bit are set, PCI2 interrupt will generate. When set PCINT19 to logic zero, PD3 pin change interrupt is disable.
2	PCINT18	Pin change enable mask bit 19 PCINT18 is to be set to logic one, PD2 pin change interrupt is enabled, and pin change of PD2 is set to PCIF2. If PCIE2 and global bit are set, PCI2 interrupt will generate. When set PCINT18 to logic zero, PD2 pin change interrupt is disable.
1	PCINT17	Pin change enable mask bit 17 PCINT17 is to be set to logic one, PD1 pin change interrupt is enabled, and pin change of PD1 is set to PCIF2. If PCIE2 and global bit are set, PCI2 interrupt will generate. When set PCINT17 to logic zero, PD1 pin change interrupt is disable.
0	PCINT16	Pin change enable mask bit 16 PCINT16 is to be set to logic one, PD0 pin change interrupt is enabled, and pin change of PD0 is set to PCIF2. If PCIE2 and global bit are set, PCI2 interrupt will generate. When set PCINT16 to logic zero, PD0 pin change interrupt is disable.

PCMSK3—Pin change interrupt mask register 3

PCMSK1—Pin change interrupt mask register 2								
Address: 0x73					Default Value: 0x00			
Bit	7	6	5	4	3	2	1	0

	PCINT31	PCINT30	PCINT29	PCINT28	PCINT27	PCINT26	PCINT25	PCINT24
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	Name	Description						
7	PCINT31	Pin change enable mask bit 31 PCINT31 is to be set to logic one, PE7 pin change interrupt is enabled, and pin change of PE7 is set to PCIF3. If PCIE3 and global bit are set, PCI3 interrupt will generate. When set PCINT31 to logic zero, PE7 pin change interrupt is disable.						
6	PCINT30	Pin change enable mask bit 30 PCINT30 is to be set to logic one, PE6 pin change interrupt is enabled, and pin change of PE6 is set to PCIF3. If PCIE3 and global bit are set, PCI3 interrupt will generate. When set PCINT30 to logic zero, PE6 pin change interrupt is disable.						
5	PCINT29	Pin change enable mask bit 29 PCINT29 is to be set to logic one, PE5 pin change interrupt is enabled, and pin change of PE5 is set to PCIF3. If PCIE3 and global bit are set, PCI3 interrupt will generate. When set PCINT29 to logic zero, PE5 pin change interrupt is disable.						
4	PCINT28	Pin change enable mask bit 28 PCINT28 is to be set to logic one, PE4 pin change interrupt is enabled, and pin change of PE4 is set to PCIF3. If PCIE3 and global bit are set, PCI3 interrupt will generate. When set PCINT28 to logic zero, PE4 pin change interrupt is disable.						
3	PCINT27	Pin change enable mask bit 27 PCINT27 is to be set to logic one, PE3 pin change interrupt is enabled, and pin change of PE3 is set to PCIF3. If PCIE3 and global bit are set, PCI3 interrupt will generate. When set PCINT27 to logic zero, PE3 pin change interrupt is disable.						
2	PCINT26	Pin change enable mask bit 26 PCINT26 is to be set to logic one, PE2 pin change interrupt is enabled, and pin change of PE2 is set to PCIF3. If PCIE3 and global bit are set, PCI3 interrupt will generate. When set PCINT26 to logic zero, PE2 pin change interrupt is disable.						
1	PCINT25	Pin change enable mask bit 25 PCINT25 is to be set to logic one, PE1 pin change interrupt is enabled, and pin change of PE1 is set to PCIF3. If PCIE3 and global bit are set, PCI3 interrupt will generate. When set PCINT25 to logic zero, PE1 pin change interrupt is disable.						
0	PCINT24	Pin change enable mask bit 24 PCINT24 is to be set to logic one, PE0 pin change interrupt is enabled, and pin change of PE0 is set to PCIF3. If PCIE3 and global bit are set, PCI3 interrupt will generate. When set PCINT24 to logic zero, PE0 pin change interrupt is disable.						

PCMSK4—Pin change interrupt mask register 4

PCMSK4—Pin change interrupt mask register 4								
Address: 0x74					Default Value: 0x00			
Bit	7	6	5	4	3	2	1	0
	PCINT39	PCINT38	PCINT37	PCINT36	PCINT35	PCINT34	PCINT33	PCINT32
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	Name	Description						
7	PCINT39	Pin change enable mask bit 39 PCINT39 is to be set to logic one, PF7 pin change interrupt is enabled, and pin						

		change of PF7 is set to PCIF4. If PCIE4 and global bit are set, PCI4 interrupt will generate. When set PCINT39 to logic zero, PF7 pin change interrupt is disable.
6	PCINT38	Pin change enable mask bit 38 PCINT38 is to be set to logic one, PF6 pin change interrupt is enabled, and pin change of PF6 is set to PCIF4. If PCIE4 and global bit are set, PCI4 interrupt will generate. When set PCINT38 to logic zero, PF6 pin change interrupt is disable.
5	PCINT37	Pin change enable mask bit 37 PCINT37 is to be set to logic one, PF5 pin change interrupt is enabled, and pin change of PF5 is set to PCIF4. If PCIE4 and global bit are set, PCI4 interrupt will generate. When set PCINT37 to logic zero, PF5 pin change interrupt is disable.
4	PCINT36	Pin change enable mask bit 36 PCINT36 is to be set to logic one, PF4 pin change interrupt is enabled, and pin change of PF4 is set to PCIF4. If PCIE4 and global bit are set, PCI4 interrupt will generate. When set PCINT36 to logic zero, PF4 pin change interrupt is disable.
3	PCINT35	Pin change enable mask bit 35 PCINT35 is to be set to logic one, PF3 pin change interrupt is enabled, and pin change of PF3 is set to PCIF4. If PCIE4 and global bit are set, PCI4 interrupt will generate. When set PCINT35 to logic zero, PF3 pin change interrupt is disable.
2	PCINT34	Pin change enable mask bit 34 PCINT34 is to be set to logic one, PF2 pin change interrupt is enabled, and pin change of PF2 is set to PCIF4. If PCIE4 and global bit are set, PCI4 interrupt will generate. When set PCINT34 to logic zero, PF2 pin change interrupt is disable.
1	PCINT33	Pin change enable mask bit 33 PCINT33 is to be set to logic one, PF1 pin change interrupt is enabled, and pin change of PF1 is set to PCIF4. If PCIE4 and global bit are set, PCI4 interrupt will generate. When set PCINT33 to logic zero, PF1 pin change interrupt is disable.
0	PCINT32	Pin change enable mask bit 32 PCINT32 is to be set to logic one, PF0 pin change interrupt is enabled, and pin change of PF0 is set to PCIF4. If PCIE4 and global bit are set, PCI4 interrupt will generate. When set PCINT32 to logic zero, PF0 pin change interrupt is disable.

定时/计数器 0 (TMR0)

- **8 位计数器**
- **两个独立的比较单元**
- **比较匹配发生时自动清零计数器并自动加载**
- **无干扰脉冲的相位修正的 PWM 输出**
- **频率发生器**
- **外部事件计数器**
- **10 位的时钟预分频器**
- **溢出和比较匹配中断**
- **带死区时间控制**
- **6 个可选触发源自动关闭 PWM 输出**
- **高速时钟模式下产生高速高分辨率（500KHz@7Bit）PWM**

- Timer/Counter 0
- 8-bit Timer/Counter
- Two Independent Compare Units
- Clear Timer on Compare Match (Auto Reload
- Glitch Free, Phase Correct Pulse, PWM output
- Frequency Generator
- External event counter
- 10-bit clock prescaler
- Overflow and Compare matchinterrupt
- Dead time control
- 6 trigger source for option, to shut down PWM output automatically
- High speed high resolution (500KHz@7Bit）PWM) in high speed clock mode

概述

TC0 是一个通用 8 位定时计数器模块，支持 PWM 输出，可以精确地产生波形。TC0 包含 1 个计数时钟产生单元，1 个 8 位计数器，波形产生模式控制单元和 2 个输出比较单元。同时，TC0 可与 TC1 共用 10 位的预分频器，也可以独立使用 10 位的预分频器。预分频器对系统时钟 clkio 或高速时钟 rcm2x（内部 32M RC 振荡器输出时钟 rc32m 的 2 倍频）进行分频来产生计数时钟 Clkt0。波形产生模式控制单元控制着计数器的工作模式和比较输出波形的产生。根据不同的工作模式，计数器对每一个计数时钟 Clkt0 实现清零、加一或减一操作。Clkt0 可以由内部时钟源或外部时钟源产生。

当计数器的计数值 TCNT0 到达最大值（等于极大值 0xFF 或输出比较寄存器 OCR0A，定义为 TOP，定义极大值为 MAX 以示区别）时，计数器会进行清零或减一操作。当计数器的计数值 TCNT0 到达最小值（等于 0x00，定义为 BOTTOM）时，计数器会进行加一操作。当计数器的计数值 TCNT0 到达 OCR0A/OCR0B 时，也被称为发生比较匹配时，会清零或置位输出比较信号 OC0A/OC0B，来产生 PWM 波形。

当使能插入死区时间时，设定的死区时间（DTR0 寄存器所对应的计数时钟数）将会插入到已产生的 PWM 波形中。软件可通过清除 COM0A/COM0B 位为零来关闭 OC0A/OC0B 的波形输出，或者设置相应的触发源，当触发事件发生时硬件自动清零 COM0A/COM0B 位来关闭 OC0A/OC0B 的波形输出。

Overview:

TC0 is a general purpose 8-bit Timer/Counter module, with PWM support. It allows accurate wave generation. TC0 consists of 1 counting clock generator unit, 1 8-bit counter, waveform generation control unit and 2 output compare unit. TC0 and TC1 share 10-bit prescaler , that can be used also individually. To generate counting clock CLKT0, prescaler divides frequency to system clock clkio or high speed clock rcm2x (internal 32M RC oscillator output 2 times frequency than rc32m).

Waveform generating control unit manage working mode of counter and generating of compare output waveform. For different mode, counter clear, minus 1 or plus 1 to each counting clock CLKt0 Clkt0 can be generated from internal or external clock source.

If counting number TCNT0 of counter is maximum (which is max. value OxFF, or define output compare register OCROA as TOP, define max. value as MAX for better identify), counter will clear or make minus 1 operation.

If counting number TCNT0 of counter is minimum (which is Ox00, or define as BOTTOM), counter will operate PLUS 1.

If counting number TCNT0 is OCROA/OCROB, indicate compare match occurs, counter will clear or is set to output compare signal 0C0A/0C0B to generate PMW waveform.

If enable to insert dead time which is corresponding counting clock number of DER0 register , it will be inserted to PWM waveform. Clearing COMOA/COMOB, software can shut down output of OCOA/OCOB waveform, or setting related trigger source, when it is triggered, hardware clear COMOA/COMOB automatically to shut down output of OCOA/OCOB waveform.

计数时钟可由内部或外部时钟源来产生，时钟源的选择及分频选择由位于 TCCR0B 寄存器的 CS0 位来控制，详细描述见 TC0 和 TC1 预分频器章节

Counting clock is generated from external or internal clock source, clock source selection and frequency division is control by CS0 of TCCROB register, for details please refer to section of TC0 and TC1 Prescaler.

计数器的长度为 8 位，支持双向计数。波形产生模式即计数器的工作模式由位于 TCCR0A 和 TCCR0B 寄存器的 WGM0 位来控制。根据不同的工作模式，计数器对每一个计数时钟 Clkt0 实现清零、加一或减一操作。当计数发生溢出时，位于 TIFR0 寄存器的计数溢出标志 TOV0 位会被置位。当中断使能时可产生 TC0 计数溢出中断。

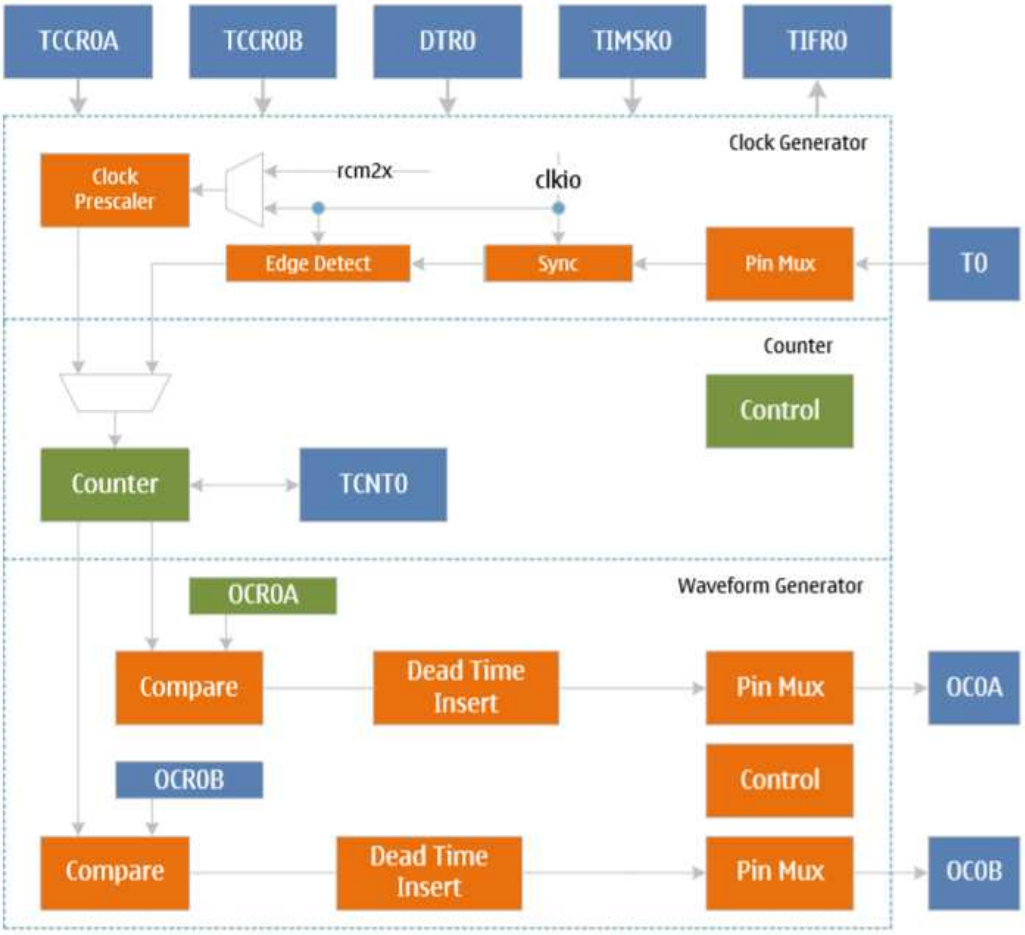
Counter is 8-bit, support bidirectional counting. Waveform generating mode, counter working mode, is controlled by WGM0 of TCCROA and TCCROB. For each mode, counter clear, plus 1 or minus 1 to each counting clock Clkt0. If counter overflows, counting overflow flag TOVO of TIFRO register will be set. TC0 counting overflow interrupt is generated when enable interrupt.

输出比较单元对计数值 TCNT0 和输出比较寄存器 OCR0A 和 OCR0B 的值进行比较，当 TCNT0 等于 OCR0A 或 OCR0B 时称为发生比较匹配，位于 TIFR0 寄存器的输出比较标志 OCF0A 或 OCF0B 位会被置位。当中断使能时可产生 TC0 输出比较匹配中断。
需注意的是，在 PWM 工作模式下，OCR0A 和 OCR0B 寄存器为双缓冲寄存器。在普通模式和 CTC 模式下，双缓冲功能失效。计数到达最大值或最小值时，缓冲寄存器中的值被同步更新到比较寄存器 OCR0A 和 OCR0B 中去。详见工作模式章节描述。

Output compare unit compare TCNT0 and value of output compare register OCROA and OCROB, if TCNT0 equals to OCROA or OCROB, compare match is happening, so output compare flag OCFOA or OCFOB of TIFR0 register is set. When interrupt is enable, TC0 output compare match interrupt is generated.
Note that in PWM mode, OCRA and OROB are doble buffered register, which is useess in normal and CTC mde. When counting numbe is max. Or min, value of buffered register is update sychronizedto compare registerOCROA and OCROB. For details please refering to section of working mode.

波形产生器根据波形产生模式控制和比较输出模式控制,使用比较匹配和计数溢出等来产生输出比较波形信号 OC0A 和 OC0B。具体产生方式见工作模式和寄存器章节描述。要把输出比较波形信号 OC0A 和 OC0B 输出到相应引脚上时，还必须设置该引脚的数据方向寄存器为输出。下图为 TC0 的内部结构图。TC0 包含 1 个计数时钟产生单元，1 个 8 位计数器，2 个输出比较单元和 2 个波形产生控制单元。

Waveform generator, basd on waveform generating mode control and compare output mode control, use compare match and counting overflow to generate output compare waveform signal OCOA and OCOB. For detais refering to section of working mode and register. To output compare waveform signal OCOA and OCOB to related pins, data direction register of pin must be set to output. Below shows internal construction diagram TC0, it includes 1 counting cloc generator unit,1 8-bit counter, 2 output compare unit and2 waveform generating control unit.



工作模式

定时计数器 0 有四种不同的工作模式，包括普通模式（Normal），比较匹配时清零（CTC） 模式，快速脉冲宽度调制（FPWM）模式和相位修正脉冲宽度调制（PCPWM）模式，由波形 产生模式控制位 WGM0[2:0]来选择。下面具体来描述这四种模式。由于有两个独立的输出比 较单元，分别用“A”和“B”来表示，用小写的“x”来表示这两个输出比较单元通道。

Working Mode

Timer/Counter 0 has 4 working modes, Normal mode (Normal), compare match clear mode (CTC), Fast Pulse Width Modulation mode(FPWM) and Phase correct pulse width modulation mode (PCPWM), mode is selected by waveform generating mode control bit WGM0[2:0]. Below is the details description of these 4 modes. Two individual output compare unit is indicated seperately by “A” and “B”, small “x” is used to indicate channel of these 2 output compare units.

普通模式

普通模式是定时计数器最简单的工作模式，此时波形产生模式控制位 WGM0[2:0]=0，计 数的最大值 TOP 为 MAX（0xFF）。在这种模式下，计数方式为每一个计数时钟加一递增，当 计数器到达 TOP 溢出后就回到 BOTTOM 重新开始累加。在计数值 TCNT0 变成零的同一个计 数时钟里置位定时计数器溢出标志 TOV0。这种模式下 TOV0 标志就像是第 9 计数位，只是 只会被置位不会被清零。溢出中断服务程序会自动清除 TOV0 标志，软件可以用它来提高定 时计数器的分辨率。普通模式下没有特殊情形需要考虑，可以随时写入新的计数值。 设置 OC0x 引脚的数据方向寄存器为输出时才能得到输出比较信号 OC0x 的波形。当 COM0x=1 时，发生比较匹配时会翻转 OC0x 信号，这种情况下波形的频率可以用下面的公式来计算：

foc0xnormal = fsys/(2*N*256) 其中，N 表示的是预分频因子（1，8，64，256 或者 1024）。输出比较单元可以用来产生中断，但是在普通模式下不推荐使用中断，这样会占用太多 CPU 的时间。

Normal Mode

The simplest mode of operation is the Normal mode (WGM02:0 = 0), its maximum TOP is MAX（0xFF). In this mode, counting increase by each counting clock, when it passes TOP and overflow, it will then restart to accumulate from the bottom (0x00). The Timer/Counter Overflow Flag (TOV0) will be set in the same timer clock cycle as the TCNT0 becomes zero. The TOV0 Flag in this case behaves like a ninth bit, except that it is only set, not cleared. However, combined with the timer overflow interrupt that automatically clears the TOV0 Flag, the timer resolution can be increased by software. There are no special cases to consider in the Normal mode, a new counter value can be written anytime. Waveform of output compare signal OCOx is generated only if data direction register of OCOx is set. If COMOx=1, OCOx singal will be toggled when compare match is happening, in this case waveform frequency can be calculated by below formula:

foc0xnormal = fsys/(2*N*256), N indicates prescaler factor (1， 8， 64， 256 or 1024). Using the Output Compare to generate waveforms in Normal mode is not recommended, since this will occupy too much of the CPU time.

CTC 模式

设置 WGM0[2:0]=2 时，定时计数器 0 进入 CTC 模式，计数的最大值 TOP 为 OCR0A。在 这个模式下，计数方式为每一个计数时钟加一递增，当计数器的数值 TCNT0 等于 TOP 时计 数器清零。OCR0A 定义了计数的最大值，亦即计数器的分辨率。这个模式使得用户可以很容易的控制比较匹配输出的频率，也简化了外部事件计数的操作。

当计数器到达计数的最大值时，输出比较匹配标志 OCF0 被置位，相应的中断使能置位时将 会产生中断。在中断服务程序里可以更新 OCR0A 寄存器即计数的最大值。在这个模式下 OCR0A 没有使用双缓冲，在计数器以无预分频器或很低的预分频器工作下将最大值更新为接 近最小值的时候要小心。如果写入 OCR0A 的数值小于当时的 TCNT0 值时，计数器将丢失一 次比较匹配。在下一次比较匹配发生之前，计数器不得不先计数到 TOP，然后再从 BOTTOM 开始计数到 OCR0A 值。和普通模式一样，计数值回到 BOTTOM 的计数时钟里置位 TOV0 标志。设置 OC0x 引脚的数据方向寄存器为输出时才能得到输出比较信号 OC0x 的波形。当 COM0x=1 时，发生比较匹配时会翻转 OC0x 信号，这种情况下波形的频率可以用下面的公式来计算：

foc0xctc = fsys/(2*N*(1+OCR0x))

其中，N 表示的是预分频因子（1，8，64，256 或者 1024）。

从公式可以看出，当设置 OCR0A 为 0x0 且无预分频器时，可以获得最大频率为 fsys/2 的输出 波形。

CTC Mode

Set WGM0[2:0]=2, timer/ counter 0 is in CTC mode, max. Counting number TOP is OCROA. In this mode, counting increase by each counting clock, counter is cleared when TCNTO is TOP. The OCR0A defines the top value for the counter, hence also its resolution. This mode allows greater control of the compare match output frequency. It also simplifies the operation of counting external events.

When counter value reaches TOP value, output compare match flag OCF0 is set, corresponding interrupt enable will generate interrupt. If the interrupt is enabled, the interrupt handler routine can be used for updating the TOP value of OCROA. However, changing TOP to a value close to BOTTOM when the counter is running with none or a low prescaler value must be done with care since the CTC mode does not have the double buffering feature. If the new value written to OCR0A is lower than the current value of TCNT0, the counter will miss the compare match. Before next compare match, the counter will then have to count to TOP and re-start from BOTTOM to count to OCROA value.

As for the Normal mode of operation, the TOV0 Flag is set when counting value return to counting clock of BOTTOM.

Waveform of output compare signal OCOx is generated only if data direction register of OCOx pin is set output. When COMOx=1, compare match occures and OCOCx signal is to be toggled. In this c ase waveform frequency can be calculated by below formula:

foc0xctc = fsys/(2*N*(1+OCR0x))

Here, N indicates prescaler factor (1，8，64，256 or1024)

Seeing from above formular, set OCROA to 0x0 meanwhile without prescaler, output waveform of max. fsys/2

can be gained.

快速 PWM 模式

设置 WGM0[2:0]=3 或 7 时，定时计数器 0 进入快速 PWM 模式，可以用来产生高频的 PWM 波形，计数最大值 TOP 分别为 MAX（0xFF）或 OCR0x。快速 PWM 模式和其他 PWM 模 式不同在于它是单向操作。计数器从最小值 0x00 累加到 TOP 后又回到 BOTTOM 重新计数。

当计数值 TCNT0 到达 OCR0x 或 BOTTOM 时，输出比较信号 OC0x 会被置位或清零，取决于比 较输出模式 COM0x 的设置，详情见寄存器描述。

由于采用单向操作，快速 PWM 模式的操作 频率是采用双向操作的相位修正 PWM 模式的两倍。高频特性使得快速 PWM 模式适用于功 率调节，整流以及 DAC 应用。高频信号可以减小外部元器件（电感电容等）的尺寸，从而降 低系统成本。 当计数值到达最大值时，定时计数器溢出标志 TOV0 将会被置位，并把比较缓冲器的值更新到比较值。如果中断使能，在中断服务程序中可以更新比较缓冲器 OCR0x 寄存器。

设置 OC0x 引脚的数据方向寄存器为输出时才能得到输出比较信号 OC0x 的波形。波形的频 率可用下面的公式来计算：

foc0xfpwm = fsys/(N*(1+TOP))

其中，N 表示的是预分频因子（1，8，64，256 或者 1024）。

当 TCNT0 和 OCR0x 发生比较匹配时，波形产生器就置位（清零）OC0x 信号，当 TCNT0 被清 零时，波形产生器就清零（置位）OC0x 信号，以此来产生 PWM 波。由此 OCR0x 的极值将会 产生特殊的 PWM 波形。当 OCR0x 设置为 0x00 时，输出的 PWM 为每(1+TOP)个计数时钟里 有一个窄的尖峰脉冲。当 OCR0x 设置为最大值时，输出的波形为持续的高电平或低电平

Fast PWM Mode

Fast PWM mode of Timer/Counter 0 (WGM0[2:0]= 3 or 7) provides a high frequency PWM waveform generation option.

TOP is defined as MAX(0xFF) or OCROx. The fast PWM differs from the other PWM option by its single-slope operation.

The counter counts from min. Value 0x00 to TOP then restarts from BOTTOM. When counting value reaches OCROx or BOTTOM, output compare OCOx is set or cleared, which depends on configuration of compare output mode, details referring to sections about register. Due to the single-slope operation, the operating frequency of the fast PWM mode can be twice as high as the phase correct PWM mode that use dual-slope operation. This high frequency makes the fast PWM mode well suited for power regulation, rectification, and DAC applications. High frequency allows physically small sized external components (capacitors), and therefore reduces total system cost. When counting value reaches maximum, timer counter overflow flag TOV0 will be set and value of compare compare buffer is updated to compare value. If interrupt is disable, compare buffer OCROx register can be updated in interrupt routine program.

Waveform of output compare signal OCOx is generated only if data direction register of OCOx pin is set output. Waveform frequency can be calculated by below formular:

foc0xfpwm = fsys/(N*(1+TOP))

Here, N indicates prescaler factor (1，8，64，256 and 1024)

When compare match occur between TCNT0 and OCR0x, waveform generator is set to OC0x (cleared). When TCT0 is cleared, waveform generator is (cleared) set to OC0x to generate PWM waveform, so extreme value of OCR0x will generate very special PWM waveform. If the OCR0A is set equal to 0x00 the output PWM will be a narrow spike for each

TOP+1 timer clock cycle. Setting the OCR0A equal to MAX will result in a constantly high or low output

相位修正 PWM 模式

当设置 WGM0[2:0]=1 或 5 时，定时计数器 0 进入相位修正 PWM 模式，计数的最大值 TOP 分别为 MAX（0xFF）或 OCR0A。计数器采用双向操作，由 BOTTOM 递增到 TOP，然后又递减到 BOTTOM，再重复此操作。计数到达 TOP 和 BOTTOM 时均改变计数方向，计数值在 TOP 或 BOTTOM 上均只停留一个计数时钟。在递增或递减过程中，计数值 TCNT0 与 OCR0x 匹配时，输出比较信号 OC0x 将会被清零或置位，取决于比较输出模式 COM0x 的设置。与单向操作相比，双向操作可获得的最大频率要小，但其极好的对称性更适合于电机控制。

相位修正 PWM 模式下，当计数到达 BOTTOM 时置位 TOV0 标志，当计数到达 TOP 时把比较缓冲器的值更新到比较值。如果中断使能，在中断服务

程序中可以更新比较缓冲器 OCR0x 寄存器。

设置 OC0x 引脚的数据方向寄存器为输出时才能得到输出比较信号 OC0x 的波形。波形的频率可用下面的公式来计算：

foc0xpcpwm = fsys/(N*TOP*2)

其中，N 表示的是预分频因子（1，8，64，256 或者 1024）。

在递增计数过程中，当 TCNT0 与 OCR0x 匹配时，波形产生器就清零（置位）OC0x 信号。在递减计数过程中，当 TCNT0 与 OCR0x 匹配时，波形产生器就置位（清零）OC0x 信号。由此 OCR0x 的极值会产生特殊的 PWM 波。当 OCR0x 设置为最大值或最小值时， OC0x 信号输出会一直保持低电平或高电平。

为了保证输出 PWM 波在最小值两侧的对称性，在没有发生比较匹配时，有两种情况下也会翻转 OC0x 信号。第一种情况是，当 OCR0x 的值由最大值 0xFF 改变为其他数据时。当 OCR0x 为最大值，计数值达到最大时，OC0x 的输出与前面降序计数时比较匹配的结果相同，即保持 OC0x 不变。此时会更新比较值为新的 OCR0x 的值（非 0xFF），OC0x 的值会一直保持，直到升序计数时发生比较匹配而翻转。此时 OC0x 信号并不以最小值为中心对称，因此需要在 TCNT0 到达最大值时翻转 OC0x 信号，此即没有发生比较匹配时翻转 OC0x 信号的第一种情况。第二种情况是，当 TCNT0 从比 OCR0x 高的值开始计数时，因而会丢失一次比较匹配，从而引起不对称情形的产生。同样需要翻转 OC0x 信号去实现最小值两侧的对称性。

Phase Correct PWM Mode

Setting WGM0[2:0] to 1 or 5, Timer/Counter 0 enter into Phase Correct PWM mode, each TOP value is MAX（0xFF）or OCR0A. The phase correct PWM mode is based on a dual-slope operation. The counter counts repeatedly from BOTTOM to TOP and then from TOP to BOTTOM. On the compare match between TCNT0 and OCR0x while up-counting and down-counting, the Output Compare (OC0x) is cleared or set depending on the configuration of compare output mode COM0x. The dual-slope operation has lower maximum operation frequency than single slope operation. However, due to the symmetric feature of the dual-slope PWM modes, these modes are preferred for motor control applications.

In phase correct PWM mode, the TOV0 is set each time the counter reaches BOTTOM. When the counter reaches TOP, value of compare buffer is updated to compare value. If interrupt enable, compare buffer OCR0x register can be updated in interrupt routine program.

Setting OC0x pin data direction register to output, it will generate waveform of output compare OC0x. Waveform frequency can be calculated via below formula:

foc0xpcpwm = fsys/(N*TOP*2)

Here, N indicates prescaler factor (1，8，64，256 or 1024)

During up-counting, when compare match occur between TCNT0 and OCR0x, waveform generator clear (set) OC0x . During down-counting, when compare match occur between TCNT0 and OCR0x, waveform generator set (clear) OC0x, so extreme value of OCR0x will generate very special PWM waveform. Setting the OCR0x equal to MAX or BOTTOM will result in a constantly high or low output.

OC0x will be toggled in two cases when compare match does not occur while to ensure symmetry of both side of PMW waveform at its bottom value.

OCR0x changes its value from MAX (0xFF), When the OCR0x value is MAX, counting value reach maximum, OC0x output is the same as the result of previous down-counting Compare Match, that is to say OC0x keep unchanged. In this case Compare value is updated to new OCR0x value (not 0xFF). OC0x keeps unchanged till it is toggled in up-counting compare match. At this moment OC0x is not symmetric with bottom value, so OC0x need to be toggled when TCNT0 reach max. Value.

The 2nd case is that when TCNT0 start to count from value higher than OCR0x, one compare match will loss, which will result in asymmetric. Same is that to achieve symmetric at both side of bottom value, OC0x is needed to be toggled.

PWM 输出的自动关闭与重启

当设置 TCCR0A 寄存器的 DOC0x 位为高时， PWM 输出的自动关闭功能会被使能，满足触发条件时，硬件会清零相应的 COM0x 位，将 PWM 输出信号 OC0x 与其输出引脚断开，切换成通用 IO 输出，实现 PWM 输出的自动关闭。此时，输出引脚的状态可由通用 IO 口的输出来控制。PWM 输出的自动关闭被使能后，还需要设置其触发条件，由 TCCR0C 寄存器的 DSX0n 位来选择触发源。触发源有模拟比较器中断，外部中断，引脚电平变化中断以及定时器溢出中断，具体情形请参考 TCCR0C 寄存器描述。当某个或某些触发源被选用作为触发条件后，在这些
中断标志位被置位的同时，硬件会清零 COM0x 位来关闭 PWM 的输出。

当发生了触发事件关闭 PWM 输出后，定时器模块没有相应的中断标志位，软件需要通过读取触发源的中断标志位来得知触发条件和触发事件。

当 PWM 输出被自动关闭而需要再次重启输出时，软件只需要重新设置 COM0x 位，来切换 OC0x 信号输出到相应的引脚上。需要注意的是，发生自动关闭后，定时器并未停止工作，OC0x 信号的状态也一直在更新。软件可在定时器发生溢出或比较匹配后，再设置 COM0x 位来输出 OC0x 信号，这样可以获得明确的 PWM 输出状态。

Auto shut down and re-start of PWM output

If DOC0x of TCCROA register is set high, PWM output enable to shut down automatically. Once trigger is satisfied, hardware will clear corresponding COM0x, PWM output OC0x will be apart from output pin and switch to general IO output, by this way automate shut-down of PWM output is done. Under this situation, status of output pin is controlled via output of general IO port.
When automate shut-down of PWM output is enable, trigger condition must be set. Trigger source is set by DSX0n of TCCROC register. Trigger source has analog compare interrupt, external interrupt, pin change interrupt and timer overflow interrupt, for details please refer to section of TCCROC register. If some of these trigger source is selected and these interrupt flag is set at the same time, hardware will clear COM0x to shut down PMW output.

When trigger events occur and PWM output is shut down, if no interrupt flag in timer module, trigger event and trigger condition can only be known through software read interrupt flag of trigger source.

If PWM need to re-start output, it is only necessary to use software re-set COM0x to switch OC0x output into related pins. Note that after automate shut-down occur, timer will not stop working, OC0x status is keeping update. Software can set COM0x to output OC0x after overflow or compare match occurs on timer, by this way accurate PMW output can be gotten.

死区时间控制

设置 DTEN0 位为“1”时，插入死区时间的功能被使能，OC0A 和 OC0B 的输出波形将在 B 通道比较输出所产生的波形基础上插入设定的死区时间，时间的长度为 DTR0 寄存器的计数时钟数所对应的时间值。如下图所示，OC0A 和 OC0B 的死区时间插入均是以通道 B 的比较输出波形为基准。当 COM0A 和 COM0B 同为“2”或“3”时，OC0A 的波形极性与 OC0B 的波形极性相同，当 COM0A 和 COM0B 分别为“2”或“3”时，OC0A 的波形与 OC0B 的波形极性相反。

Dead time control

Setting DTEN0 to logic one, it enables of inserting dead time, OCOA and OCOB will insert designed dead time based on waveform generated by B channel compare output, length of time is the timing value corresponded to counter clock number of DTR0 register. As shown below, OCOA and OCOB will insert designed dead time based on waveform generated by B channel compare output. If COM0A and COM0B are set to 2 or 3 at the same time, waveform polarity of OC0A and OC0B are the same, if they are set to 2 or 3 individually, waveform polarity of OC0A is opposite to that of OC0B.

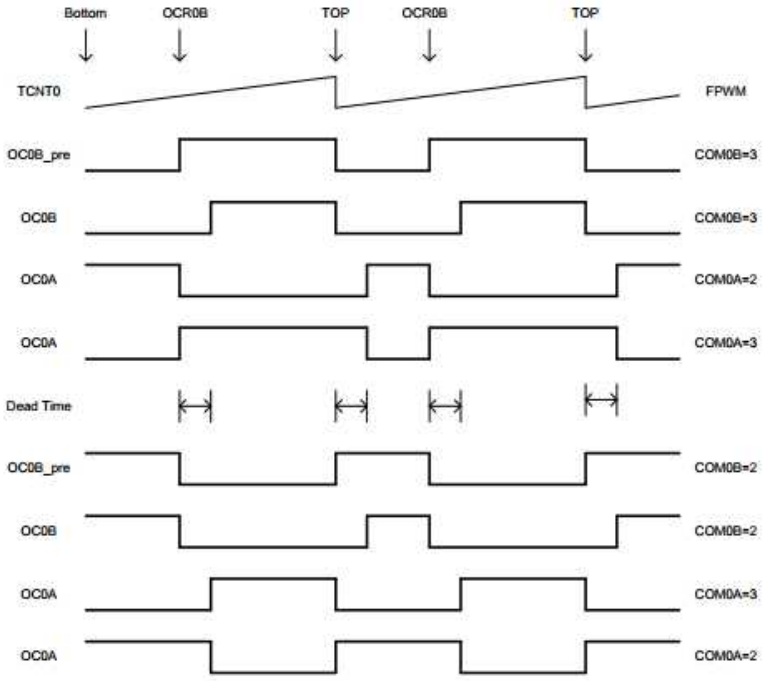


Figure 1　FPWM 模式下 TC0 死区时间控制

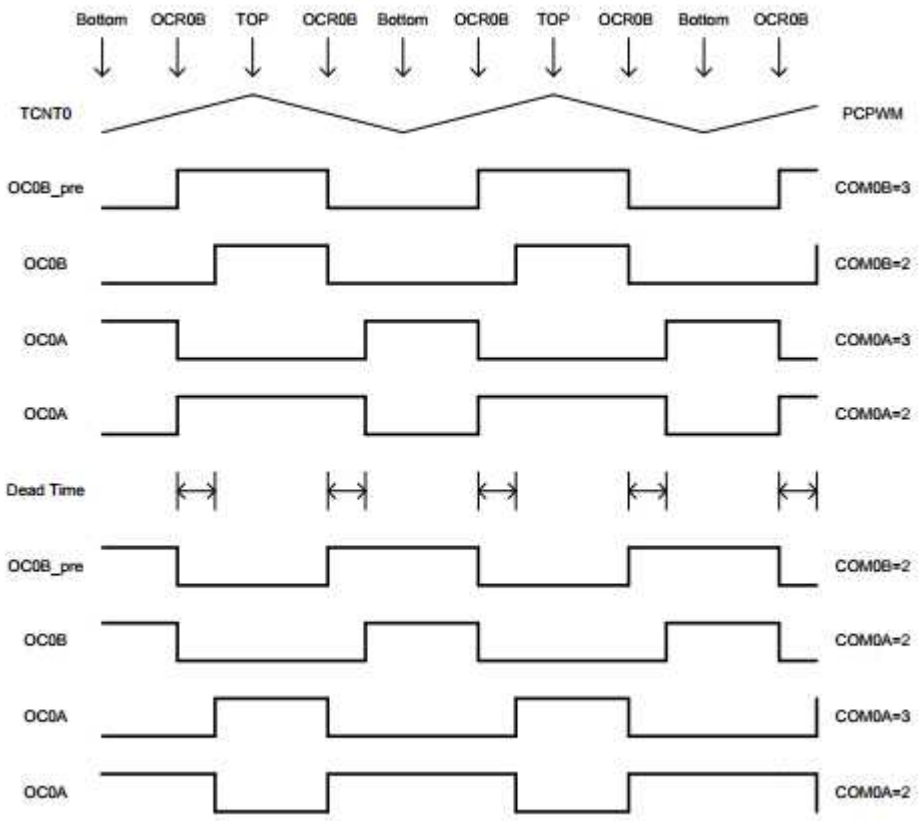


Figure 2　PCPWM 模式下 TC0 死区时间控制

设置 DTEN0 位为“0”时，插入死区时间的功能被禁止，OC0A 和 OC0B 的输出波形为各自比较输出所产生的波形。

Setting DTEN0 to logic zero, inserting dead time is disable. Waveform of OCOA and OCOB are separately generated from their register.

高速时钟模式

高速时钟模式下，采用更高频率的时钟作为计数的时钟源，用来产生更高速度和更高分辨率的 PWM 波形。此高频时钟是通过对内部 32M RC 振荡器的输出时钟 rc32m 进行 2 倍频来产生的。因此，在进入高频模式之前，需先使能内部 32M RC 振荡器的倍频功能，即置位 TCKCSR 寄存器的 F2XEN 位，并等待一定时间直到倍频时钟信号输出稳定。然后，可置位 TCKCSR 的 TC2XS0 位来使定时计数器进入高速时钟模式。

在此模式下，系统时钟与高速时钟是异步关系，而部分寄存器（见 TC0 寄存器列表）是工作在高速时钟域，因此，配置和读取这类寄存器时也是异步的，操作时需注意。

对高速时钟域下的寄存器进行非连续读写操作时无特殊要求，而进行连续读写操作时，需等待一个系统时钟，可按以下步骤

1） 写寄存器 A；

2） 等待一个系统时钟（NOP 或操作系统时钟下的寄存器）；

3） 读或写寄存器 A 或 B。

4） 等待一个系统时钟（NOP 或操作系统时钟下的寄存器）。

对高速时钟域下的寄存器进行读操作时，除 TCNT0 外的寄存器均可直接读取，当计数器还在进行计数时， TCNT0 的值会随高速时钟变化，可暂停计数器（设置 CS0 为零）再读取 TCNT0 的值。

High Speed Clock Mode

In high speed clock mode, higher frequency clock is used as clock source of timer, to generate higher speed and higher resolution PMW waveform. This high frequency clock is generated from 2 times frequency which is a result from output clock rc32m of internal 32M RC oscillator. Before high frequency mode, frequency doubling function of internal 32M RC oscillator must be enabled, set F2XEN of TCKCSR register, wait till signal of frequency doubling clock is stable, than set TC2XS0 bit of TCKCSR to make time counter enter into high speed clock mode.
In this mode, system clock and high speed clock is asynchronous, some registers (referring to TC0 register list) are working in high speed area, so pay attention during operation that configuration and reading of such registers is also asynchronous.

For non-continuous read-write on registers working in high speed clock area, there is no special requirement, while for continuous read-write operation, it need to wait for one system clock, operation step is as below:
1) Write register A
2) Wait for a system clock (NOP or operation system clock register)
3) Read or write register A or B
4) Wait for a system clock (NOP or operation system clock register)

Except TCNTO, other registers in high speed clock area can be read-written directly. When counter is couting, TCNT0 value is changing as per high speped clock, timer can be paused to read-fetch TCNT0 value (set CS0 to zero)

Register Definition

TC0 Register List

Register	Address	Default value	Comments
TCCR0A*	0x44	0x00	TC0 control register A
TCCR0B	0x45	0x00	TC0 control register B
TCNT0*	0x46	0x00	TC0 counting value register
OCR0A*	0x47	0x00	TC0 output compare register A
OCR0B*	0x48	0x00	TC0 output compare register B
DSX0*	0x49	0x00	TC0 trigger source control register
DTR0*	0x4F	0x00	TC0 dead time register
TIMSK0	0x6E	0x00	Timer/Counter 0 interrupt mask register
TIFR0	0x35	0x00	Timer/Counter 0 interrupt flag register
TCKCSR	0xEC	0x00	TC clock control and status register

[Note]
Registers with “*”work in system clock andd high speed clock area, that without “8” only in system clock area.

TC0 Control Register A-TCCROA

TC0 Control Register A-TCCROA								
Address: 0x44					Default value: 0x00			
Bit	7	6	5	4	3	2	1	0
	COM0A1	COM0A0	COM0B1	COM0B0	DOC0B	DOC0A	WGM01	WGM00
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Comment
7	COM0A1	TC0 compare match A output mode control high COMOA1 and COM0A0 combine to consist of compare output control COM0A[1:0], that is used to control 0C0A output waveform. If one or both of the COM0A bits are set, compare waveform takes up 0C0A pin, however whose data direction regester must be set to high. In different mode, C0M0A control compare waveform differently, for details referring to list on compare output control.
6	COM0A0	TC0 compare match A output mode control low COMOA0 and COM0A1 combine to consist of compare output control COM0A[1:0], that is used to control 0C0A output waveform. If one or both of the COM0A bits are set, compare waveform takes up 0C0A pin, however whose data direction regester must be set to high. In different mode, C0M0A control compare waveform differently, for details referring to list on compare output control.
5	COM0B1	TC0 compare match B output mode control high COMOB1 and COM0B0 combine to consist of compare output control COM0B[1:0], that is used to control 0C0B output waveform. If one or both of the COM0B bits are set, compare waveform takes up 0C0B pin, however whose data direction regester must be set to high. In different mode, C0M0B control compare waveform differently, for details referring to list on compare output control.
4	COM0B0	TC0 compare match B output mode control low COMOB0 and COM0B1 combine to consist of compare output control COM0B[1:0], that is used to control 0C0B output waveform. If one or both of the COM0B bits are set, compare waveform takes up 0C0B pin, however whose data direction regester must be set to high. In different mode, C0M0B control compare waveform differently, for details referring to list on compare output control.
3	DOC0B	TC0 shut-down output compare enable control high If D0C0B is set to logic one, compare signal 0C0B shutted down by trigger source is enable. When events is triggered, hardware will clear COMOB automatically and shut down OCOB waveform output. Software re-start PWM output when COMB is set. If D0C0B is set to zero, compare signal 0C0B shutted down by trigger source is disable.
2	DOC0A	TC0 shut-down output compare enable control low If D0C0A is set to logic one, compare signal 0C0B shutted down by trigger source is enable. When events is triggered, hardware will shut down OCOB waveform output automatically. If D0C0B is set to zero, compare match signal 0C0B shutted down by trigger source is disable. When event is triggered, 0C0A waveform output does not shut down.
1	WGM01	TC0 waveform generation control middle WGM01, WGM00 and WGM02 generate waveform generation control WGM0[2:0] together, couting method and waveform generation method of control counter refer to list on waveform generation mode.

0	WGM00	TC0 waveform generation control low WGM00, WGM01 and WGM02 generate waveform generation control WGM0[2:0] together, couting method and waveform generation method of control counter refer to list on waveform generation mode.

TC0 Control Register B-TCCR0B

TC0 Control Register B-TCCR0B								
Address: 0x45					Default value: 0x00			
Bit	7	6	5	4	3	2	1	0
	FOC0A	FOC0B	OC0AS	DTEN0	WGM02	CS02	CS01	CS00
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Comment
7	FOC0A	TC0 force output compare A control If not in PWM mode, writing logic one to force output compare F0C0A can result in compare match. Force compare match does not set 0CF0B flag, and not reload or clear timer, but will result in output pin 0C0B update according to C0M0B configuration, which seems a real compare match occur. Read return value of F0C0B till it is zero.
6	FOC0B	TC0 force output compare A control If not in PWM mode, writing logic one to force output compare F0C0A can result in compare match. Force compare match does not set 0CF0A flag, and not reload or clear timer, but will result in output pin 0C0A update according to C0M0A configuration, which seems a real compare match occur. Read return value of F0C0A till it is zero.
5	OC0AS	0C0A output port selection control. If 0C0AS is set to logic zero, waveform of 0C0A is output from pin PD6. If 0C0AS is set to logic one, waveform of 0C0A is output from pin PE4. (valid for QFP32 package)
4	DTEN0	TC0 dead time enable control Setting DTEN0 to logic one, inserting dead time is enable. Base on waveform generated in B channel compare output, both 0C0A and 0C0B can insert dead time, whose interval is decided by corresponding counter time of DTR0 register. Waveform polarity output by 0C0A is decided by relationship between C0M0 and COMOB, for details referring to list for waveform polarity after inserting dead time.
3	WGM02	TC0 waveform generation mode control high WGM02, WGM00 and WGM01 together consist of waveform generation control WGM0[2:0], counting method and waveform generation method of control counter refer to list on waveform generation mode.
2	CS02	TC0 clock selection control high Used to select clock source of timer/counter 0
1	CS01	TC0 clock selection control middle Used to select clock source of timer/counter 0
0	CS00	TC0 clock selection control low Used to select clock source of timer/counter 0

CS0[2:0]	comments
0	No clock source, stop couting

		1	clksys
		2	Clksys/8, from frequency prescaler
		3	Clksys/64, from frequency prescaler
		4	Clksys/256, from frequency prescaler
		5	Clksys/1024, from frequency prescaler
		6	External clock T0 pin, falling edge trigger
		7	External clock T0 pin, rising edge trigger

Below list show how compare output mode control output compare waveform in no PWM mode (normal mode and CTC mode)

COM0x[1:0]	Comments
0	0C0x disconnect, general IO port operation
1	Toggle 0C0x signal at compare match
2	Clear 0C0x signal at compare match
3	Set 0C0x signal at compare match

Below list show how compare output mode control output compare waveform in PWM mode

COM0x[1:0]	Comments
0	0C0x disconnect, general IO port operation
1	Reserved
2	Clear 0C0x signal at compare match, set 0C0x at max. value match
3	Set 0C0x signal at compare match, clear 0C0x at max. value match

Below list show how compare output mode control output compare waveform in Phase Correct Mode

COM0x[1:0]	Comments
0	0C0x disconnect, general IO port operation
1	Reserved
2	0C0x is cleared at compare match in up-counting; 0C0x is set at compare match in down-counting.
3	0C0x is set at compare match in up-counting; 0C0x is cleared at compare match in down-counting.

Below list show waveform generation mode control

WGM0[2:0]	Mode	TOP value	0CROX update timing	T0V0 set timing
0	Normal	0xFF	immediately	MAX
1	PCPWM	0xFF	TOP	BOTTOM
2	CTC	OCR0A	immediately	MAX
3	FPWM	0xFF	TOP	MAX
4	reserved	-	-	-
5	PCPWM	OCR0A	TOP	BOTTOM
6	reserved	-	-	-

7	FPWM	OCR0A	TOP	TOP

Below list show 0C0A signal output waveform polarity control when dead time is enable.

DTEN0	COM0A[1:0]	COM0B[1:0]	Comment
0	-	-	0C0A signal polarity is controlled by 0C0A compare output mode
1	0	-	0C0A disconnect, general IO port operation
1	1	-	Reserved
1	2	2	0C0A signal polarity is the same as that of 0C0B
		3	0C0A signal polarity is opposite to that of 0C0B
1	3	2	0C0A signal polarity is the same as that of 0C0B
		3	0C0A signal polarity is opposite to that of 0C0B

Note:
Waveform polarity of 0C0B signal output is decided by 0C0B compare output mode, it is same as to disable dead time mode

TC0 control register C- TCCORC

TC0 Control Register B-TCCROB								
Address: 0x49					Default value: 0x00			
Bit	7	6	5	4	3	2	1	0
	DSX07	DSX06	DSX05	DSX04	-	-	DSX01	DSX00
R/W	R/W	R/W	R/W	R/W	-	-	R/W	R/W
Bit	Name	Comment						
7	DSX07	7 bit of TC0 trigger source selection control is enable When setting DSX07 to logic one, TC1 overflow, used as trigger source to shut down output compare signal waveform 0C0A/0C0B, is enable. When setting DOC0A/DOC0B to logic one, rising edge of interrupt flag register bit of selected trigger source will automatically shutdown 0C0A/0C0B wave output When setting DSX07 to logic zero, TC1 overflow, used as trigger source to shut down output compare signal waveform 0C0A/0C0B, is disable.							
6	DSX06	7 bit of TC0 trigger source selection control is enable When setting DSX06 to logic one, TC2 overflow, used as trigger source to shut down output compare signal waveform 0C0A/0C0B, is enable. When setting DOC0A/DOC0B to logic one, rising edge of interrupt flag register bit of selected trigger source will automatically shutdown 0C0A/0C0B wave output When setting DSX06 to logic zero, TC2 overflow, used as trigger source to shut down output compare signal waveform 0C0A/0C0B, is disable.							
5	DSX05	5 bit of TC0 trigger source selection control is enable When setting DSX05 to logic one, pin change level 0, used as trigger source to shut down output compare signal waveform 0C0A/0C0B, is enable. When setting DOC0A/DOC0B to logic one, rising edge of interrupt flag register bit of selected trigger source will automatically shutdown 0C0A/0C0B wave output							

		When setting DSX05 to logic zero, pin change level 0, used as trigger source to shut down output compare signal waveform 0C0A/0C0B, is disable.
4	DSX04	4 bit of TC0 trigger source selection control is enable When setting DSX04 to logic one, external interrupt 0, used as trigger source to shut down output compare signal waveform 0C0A/0C0B, is enable. When setting DOC0A/DOC0B to logic one, rising edge of interrupt flag register bit of selected trigger source will automatically shutdown 0C0A/0C0B wave output When setting DSX04 to logic zero, external interrupt 0, used as trigger source to shut down output compare signal waveform 0C0A/0C0B, is disable.
3:2	-	reserved
1	DSX01	1 bit of TC0 trigger source selection control is enable When setting DSX01 to logic one, analog comparator 1, used as trigger source to shut down output compare signal waveform 0C0A/0C0B, is enable. When setting DOC0A/DOC0B to logic one, rising edge of interrupt flag register bit of selected trigger source will automatically shutdown 0C0A/0C0B wave output When setting DSX01 to logic zero, analog comparator 1, used as trigger source to shut down output compare signal waveform 0C0A/0C0B, is disable.
0	DSX00	0 bit of TC0 trigger source selection control is enable When setting DSX00 to logic one, analog comparator 0, used as trigger source to shut down output compare signal waveform 0C0A/0C0B, is enable. When setting DOC0A/DOC0B to logic one, rising edge of interrupt flag register bit of selected trigger source will automatically shutdown 0C0A/0C0B wave output When setting DSX00 to logic zero, analog comparator 0, used as trigger source to shut down output compare signal waveform 0C0A/0C0B, is disable.

Below list show waveform output trigger source selection control

Shut down waveform output trigger source selection control

DOC0x	DSX0n=1	Trigger source	Comment
0	-	-	Setting D0C0x to logic zero, function of trigger sources shut-down waveform output is disable
1	0	Analog comparator 0	Rising edge of ACIF0 shut down 0C0x waveform output
1	1	Analog comparator 1	Rising edge of ACIF1 shut down 0C0x waveform output
1	4	External interrupt 0	Rising edge of INTF0 shut down 0C0x waveform output
1	5	Pin change level 0	Rising edge of PCIF0 shut down 0C0x waveform output
1	6	TC2 overflow	Rising edge of T0V2 shut down 0C0x waveform output
1	7	TC1 overflow	Rising edge of T0V1 shut down 0C0x waveform output

Note:

If DSX0n=1, it is indicated when the 0 bit of DSX0 register is 1, every register bit can be set simultaneously.

TC0 counting value register –TCNT0

TC0 counting value register –TCNT0

Address: 0x46					Default value: 0x00			
Bit	7	6	5	4	3	2	1	0
	TCNT07	TCNT06	TCNT05	TCNT04	TCNT03	TCNT02	TCNT01	TCNT00
R/W	R/W	R/W	R/W	R/W	-	-	R/W	R/W
Bit	Name	Comment						
7:0	TCNT00	TC0 Counting value register Via TCNT0 register, counting value, 8 of counter, can be directly access to read-write. Write operation of CPU to TCNT0 register will prevent compare match from happening in the next timer clock cycle, even timer stop working, this allows that value of initialized TCNT0 register is consistent with 0CR0 value and interrupt is not triggered. If written value to TCNT0 equal to or override 0CR0 value, compare match will loss, which results in incorrect waveform. Time stops counting if clock source is not selected, but CPU can still access to TCNT0. CPU write counter has higher priority than clear or minus/plus operation.							

TC0 output compare register A-OCR0A

TC0 output compare register A-OCR0A								
Address: 0x47					Default value: 0x00			
Bit	7	6	5	4	3	2	1	0
	OCR0A7	OCR0A6	OCR0A5	OCR0A4	OCR0A3	OCR0A2	OCR0A1	OCR0A0
R/W	R/W	R/W	R/W	R/W	-	-	R/W	R/W
Bit	Name	Comment						
7:0	OCR0A	TC0 output compare register 0CR0A has a 8-bit data, compare with counter value TCNT0 continuously. Compare match is used to generate output compare interrupt or generate waveform in 0C0A pin. In PWM mode, 0C0R0A register uses dual buffer register, while in normal and match clear mode, dual buffer function is disable. Dual buffer function can make 0CR0A register and Max/ bottom counting timing synchronized; to prevent from asymmetric PMW pulse and clear pulse intervene. When dual bugger function is used, CPU is access to 0CR0A buffer register, otherwise it is access to 0CR0A itself.							

TC0 output compare register B-0CR0B

TC0 output compare register B-0CR0B								
Address: 0x48					Default value: 0x00			
Bit	7	6	5	4	3	2	1	0
	OCR0B7	OCR0B6	OCR0B5	OCR0B4	OCR0B3	OCR0B2	OCR0B1	OCR0B0
R/W	R/W	R/W	R/W	R/W	-	-	R/W	R/W
Initial	0	0	0	0	0	0	0	0
Bit	Name	Comment						
7:0	OCR0B	TC0 output compare B register 0CR0B has a 8-bit data, compare with counter value TCNT0 continuously. Compare match is used to generate output compare interrupt or generate waveform in 0C0B pin.						

		In PWM mode, 0C0R0B register uses dual buffer register, while in normal and match clear mode, dual buffer function is disable. Dual buffer function makes 0CR0B register and Max/ bottom counting timing synchronized; to prevent from asymmetric PMW pulse and clear pulse intervene. When dual bugger function is used, CPU is access to 0CR0B buffer register, otherwise it is access to 0CR0B itself.

TC0 Interrupt Mask Register-TIMSK0

TC0 Interrupt Mask Register-TIMSK0								
Address: 0x6E					Default value: 0x00			
Bit	7	6	5	4	3	2	1	0
	-	-	-	-	-	OCIE0B	OCIE0A	TOIE0
R/W	-	-	-	-	-	-	R/W	R/W
Initial	0	0	0	0	0	0	0	0

Bit	Name	Comment
7:3	OCIE0B	TC0 output compare B match interrupt is enable If 0CIE0B is set to logic one, and global interrupt is set, TC0 output compare B match interrupt is enable. When compare match occurs, 0CF0B bit of TIFR0 is set, interrupt is generated. If 0CIE0B is set to logic zero, TC0 output compare B match interrupt is disable
1	OCIE0A	TC0 output compare A match interrupt is enable If 0CIE0a is set to logic one, and global interrupt is set, TC0 output compare A match interrupt is enable. When compare match occurs, 0CF0A bit of TIFR0 is set, interrupt is generated. If 0CIE0A is set to logic zero, TC0 output compare A match interrupt is disable
0	TOIE0	TC0 overflow interrupt enable bit When TOIE0 is set to logic one, and global interrupt is set, TC0 overflow interrupt is enable. Once TC0 overflows, T0V0 bit of TIFR is set, interrupt is generated. If TOIE0 is set to logic zero, TC0 overflow interrupt is disable.

TC0 Interrupt Flag Register –TIFR0

TC0 Interrupt Flag Register –TIFR0								
Address: 0x35					Default value: 0x00			
Bit	7	6	5	4	3	2	1	0
	OC0A	OC0B	-	-	-	OCF0B	OCF0A	TOV0
R/W	R/0	R/0	-	-	-	-	R/W	R/W
Initial	0	0	0	0	0	0	0	0

Bit	Name	Comment
7	OC0A	Output compare waveform signal 0C0A Output compare waveform signal 0C0A, software is readable but not writable. Before software disable 0C0A signal output to corresponding IO pin, it can read 0C0A value firstly to gain polarity of compare waveform signal that is to be output, meanwhile it can change its polarity by configuration on C0M0A and setting F0C0A, so as to avoid extra intervene pulse after 0C0A signal output is enable to corresponding IO pins.

6	OC0B	Output compare waveform signal 0C0B Output compare waveform signal 0C0B, software is readable but not writable. Before software disable 0C0B signal output to corresponding IO pin, it can read 0C0B value firstly to gain polarity of compare waveform signal that is to be output, meanwhile it can change its polarity by configuration on C0M0B and setting F0C0B, so as to avoid extra intervene pulse after 0C0B signal output is enable to corresponding IO pins.
5:3		reserved
2	OCF0B	TC0 output compare B match flag bit If TCNT0 equals to 0CR0B, compare unit send out match signal, and set compare flag 0CF0B. In this case if output compare B interrupt enable OCIE0B to logic one and global interrupt flag is set, output compare B interrupt is generated. When to execute this interrupt routine program, OCF0B is cleared automatically, which can be done also by writing OCF0B to logic one.
1	OCF0A	TC0 output compare A match flag bit If TCNT0 equals to 0CR0A, compare unit send out match signal, and set compare flag 0CF0A. In this case if output compare A interrupt enable OCIE0A to logic one and global interrupt flag is set, output compare A interrupt is generated. When to execute this interrupt routine program, OCF0A is cleared automatically, which can be done also by writing OCF0A to logic one.
0	T0V0	TC0 overflow flag When counter overflows, overflow flag T0V0 is set. In this case if overflow interrupt enable TOIE0 to logic one and global interrupt flag is set, overflow is generated. When to execute this interrupt routine program, T0V0 is cleared automatically, which can be done also by writing T0V0 to logic one.

DTR0-TC0 dead time control register

DTR0-TC0 dead time control register								
Address: 0x4F					Default value: 0x00			
Bit	7	6	5	4	3	2	1	0
	DTR07	DTR06	DTR05	DTR04	DTR03	DTR02	DTR01	DTR00
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Comment
[7:4]	DTR0H	TC0 dead time register high If DTEN0 of TCCR0B is set to logic one, 0C0A and 0C0B become complementary output, inserting dead time control is enable. Inserted dead time in 0C0B channel is decided by DTR0H, length of time is the time corresponding to DTR0H counting clock.
[3:0]	DTR0L	TC0 dead time register low If DTEN0 of TCCR0B is set to logic one, 0C0A and 0C0B become complementary output, inserting dead time control is enable. Inserted dead time in 0C0A channel is decided by DTR0L, length of time is the time corresponding to DTR0H counting clock.

TCKSCR-TC Clock Control and Status Register

TCKSCR-TC Clock Control and Status Register

Address: 0xEC					Default value: 0x00			
Bit	7	6	5	4	3	2	1	0
	-	F2XEN	TC2XF1	TC2XF0	-	AFCKS	TC2XS1	TC2XS0
R/W	-	R/W	R	R	-	R/W	R/W	R/W

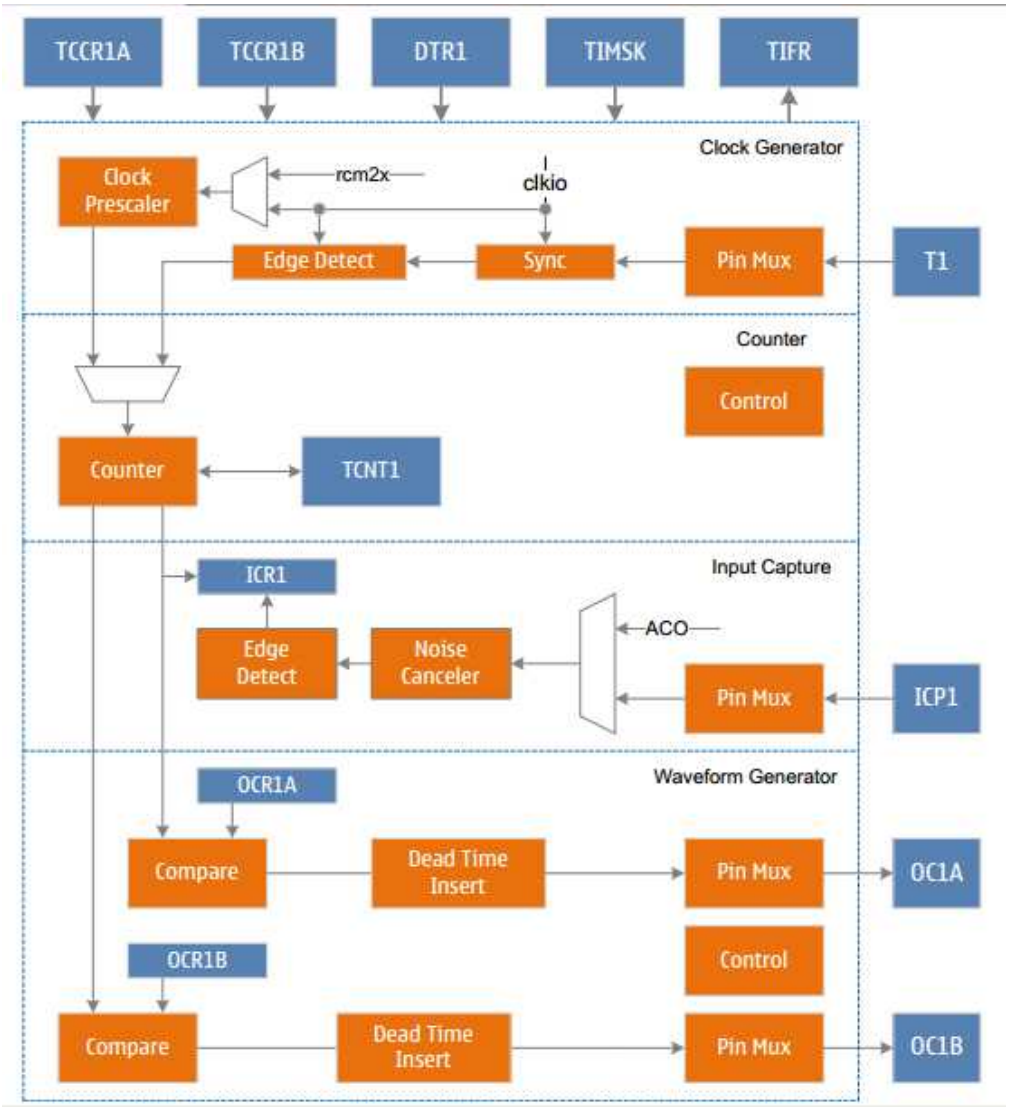
Bit	Name	Comment
7	-	reserved
6	F2XEN	RC 32M frequency doubling output enable control bit If F2XEN is set to logic one, RC 32M frequency doubling output is enable, output 64M high speed clock If F2XEN is set to logic one, RC 32M frequency doubling output is disable, cannot output 64M high speed
5	TC2XF1	TC High Speed Clock mode flag 1 Referring to section on timer/counter 1 register
4	TC2XF0	TC High Speed Clock mode flag 0 If TC2XF0 bit is read as logic one, it is indicated that timer counter 0 works in high speed clock mode. If it is logic zero, timer counter 0 works in system clock mode
3:2	-	reserved
1	TC2XS1	TC high speed clock mode selection control bit 1 eferring to section on timer/counter 1 register
0	TC2XS0	TC high speed clock mode selection control bit 0 If TC2XS0 bit is read as logic one, it is indicated that timer counter 0 works in high speed clock mode. If it is logic zero, timer counter 0 works in system clock mode

定时/计数器 1 (TMR1)

- 真正的 16 位设计，允许 16 位的 PWM
- 2 个独立的输出比较单元
- 双缓冲的输出比较寄存器
- 1 个输入捕捉单元
- 输入捕捉噪声抑制器
- 比较匹配时自动清零计数器并自动加载
- 无干扰脉冲的相位修正的 PWM
- 可变的 PWM 周期
- 频率发生器
- 外部事件计数器
- 4 个独立的中断源
- 支持死区时间控制的 PWM
- 6 个可选触发源自动关闭 PWM 输出
- 高速时钟模式下产生高速高分辨率(500KHZ@7BIT) PWM

Timer/Counter 1 (TMR1)

- Real 16-bit design, 16-bit PWM is allowed
- 2 dependant output compare unit
- Dual buffered output compare register
- 1 input capture unit
- Input capture noise suppression
- In compare match, clear counter and load automatically
- Non-intervene pulse correction PWM
- Variable PWM cycle
- Frequency generator
- External event counter
- 4 dependant interrupt source
- PWM supporting dead time control
- 6 trigger sources for option to shut down PWM output automatically
- In high speed clock mode to generate high speed high frequency PWM (500KHZ@7BIT)



TC1 Construction Diagram

TC1 是一个通用 16 位定时计数器模块，支持 PWM 输出，可以精确地产生波形。TC1 包含 1 个 16 位计数器，波形产生模式控制单元，2 个独立的输出比较单元和 1 个输入捕捉单元。同时，TC1 可与 TC0 共用 10 位的预分频器，也可以独立使用 10 位的预分频器。预分频器对系统时钟 clkio 或高速

时钟 rcm2x（内部 32M RC 振荡器输出时钟 rc32m 的 2 倍频）进行分频来产生计数时钟 Clkt1。波形产生模式控制单元控制着计数器的工作模式和比较输出波形的产生。根据不同的工作模式，计数器对每一个计数时钟 Clkt1 实现清零、加一或减一操作。Clkt1 可以由内部时钟源或外部时钟源产生。当计数器的计数值 TCNT1 到达最大值（等于极大值 0xFFFF 或固定值或输出比较寄存器 OCR1A 或输入捕捉寄存器 ICR1，定义为 TOP，定义极大值为 MAX 以示区别）时，计数器会进行清零或减一操作。当计数器的计数值 TCNT1 到达最小值（等于 0x0000，定义为 BOTTOM）时，计数器会进行加一操作。当计数器的计数值 TCNT1 到达 OCR1A 或 OCR1B 时，也被称为发生比较匹配时，会清零或置位输出比较信号 OC1A 或 OC1B，来产生 PWM 波形。当使能插入死区时间时，设定的死区时间（DTR1 寄存器所对应的计数时钟数）将会插入到已产生的 PWM 波形中。当开启输入捕捉功能时，计数器被触发即开始或停止计数， ICR1 寄存器会记录捕捉信号触发周期内的计数值。

软件可通过清除 COM1A/COM1B 位为零来关闭 OC1A/OC1B 的波形输出，或者设置相应的触发源，当触发事

件发生时硬件自动清零 COM1A/COM1B 位来关闭 OC1A/OC1B 的波形输出。

TC1 is a general 16-bit timer/counter module, supporting PWM output and generating accurate waveform. It has one 16-bit counter, waveform generation mode control unit, two independent output compare unit and one input capture unit. Meanwhile TC1 can use 10-bit frequency prescaler alone or share it with TC0. Prescaler divides frequency at system clock clkio or high speed clock rcm2x (internal 32M RC oscillator output 2 times frequency of rc32m clock)to generate counter clock CLKt1.

Waveform generation mode control unit controls working mode of counter and generation of compare output waveform. In different mode, counter features operation of clear, plus one or minus one to each counter clock CIKt1, which is generated by either internal clock source or external clock source. Counter will be cleared or start operation of minus one when counter value TCNT1 reaches its maximum value(it equals to max. 0xFFFF or a fixed value or output compare register OCR1A or input capture register ICR1 defined as TOP, maximum value is defined as MAX to clear reference) While counter will make operation of plus on when counter value TCNT1 reaches minimum value (equals to 0x0000, defined as BOTTOM). If counter value TCNT1 reaches OCR1A or OCR1B, also called compare unit occurs, output compare signal OC1A or OC1B will be cleared or set. If inserting dead time is enable, a designed dead time, corresponding counter clock value of DTR1 register, will be inserted into the PWM waveform which has been generated. If input capture function is enable, once triggered counter will start or stop counting, ICR1 register records counting value in a capture signal trigger cycle. By clearing COM1A/COM1B to zero, software can shut down waveform output of OC1A/OC1B, or setting related trigger source, which is once triggered, hardware will automate clear COM1A/COM1B to shut down waveform output of OC1A/OC1B.

Counter clock is generated by internal or external clock source, whose selection and frequency division selection is set by CS1 of TCCR18 register, referring to section of TC0 & TC1 frequency prescaler.

Counter is 16-bit supporting bidirectional counting. Waveform generation mode, also called counter working mode, is set by WGM1 bit located in TCCR1A and TCCR1B register. In different mode, counter features operation of clear, plus one or minus one on each counter clock CLKT1. If counter overflows, counting overflow flag T0V1 located in TIFR1 register is to be set. If interrupt is enable, it will generate TC1 counting overflow interrupt.

Output compare unit compares counter value TCNT1 and OCR1A and OCR1B of output compare register, if TCNT1 equals to OCR1A or OCR1B, compare match occurs, it is resulted that output compare flag OCF1A or OCF1B located in TIFR1 register is set. If interrupt is enable, TC1 output compare match interrupt will occur. Noted that in PWM mode, OCR1A and OCR1B register are dual buffered, which is not activated if in normal mode and CTC mode. When counter value reaches maximum and minimum, value of buffer register is updated synchronized to compare register OCR1A and OCR1B. Details please refer to section of working mode.

Based on waveform generation mode control and compare output mode control, waveform generator uses compare match and counting overflow to generate output compare waveform signal OC1A and OC1B, for specific generation methods please refer to section on “working mode” and “register”. To output OC1A and OC1B to related pins, it is a must to set data direction register of this pin to output.

工作模式

定时计数器 1 有六种不同的工作模式，包括普通模式（Normal），比较匹配时清零 （CTC）模式，快速脉冲宽度调制（FPWM）模式，相位修正脉冲宽度调制（PCPWM）模式，相位频率修正脉冲宽度调制（PFCPWM）模式，和输入捕捉（ICP）模式。由波形产生模式控制位 WGM1[3:0]来选择。下面具体来描述这六种模式。由于有两个独立的输出比较单元，分别用“A”和“B”来表示，用小写的“x”来表示这两个输出比较单元通道。

Working Mode

Counter Time 1 has 6 different working mode, that is: normal mode (Normal), Clear Timer on Compare mode (CTC), Fast pulse width modulation (FPWM), Phase correction pulse width modulation mode (PCPWM), Phase frequency correction pulse width modulation mode (PFCPWM) AND Input capture mode (ICP), which are selected by waveform generation mode control bit WGM1[3:0], 6 modes are specified in the below. 2 independent output compare unit are indicated individually by “A” and “B”, small “x” is referring to channel of these 2 output compare unit.

普通模式

普通模式是定时计数器最简单的工作模式，此时波形产生模式控制位 WGM1[3:0]=0，计数的最大值 TOP 为 MAX（0xFFFF）。在这种模式下，计数方式为每一个计数时钟加一递增，当计数器到达 TOP 溢出后就回到 BOTTOM 重新开始累加。在计数值 TCNT1 变成零的同一个计数时钟里置位定时计数器溢出标志 TOV1。这种模式下 TOV1 标志就像是第 17 计数位，只是只会被置位不会被清零。溢出中断服务程序会自动清除 TOV1 标志，软件可以用它来提高定时计数器的分辨率。普通模式下没有特殊情形需要考虑，可以随时写入新的计数值。

设置 OC1x 引脚的数据方向寄存器为输出时才能得到输出比较信号 OC1x 的波形。当 COM1x=1 时，发生比较匹配时会翻转 OC1x 信号，这种情况下波形的频率可以用下面的公式来计算：

foc1xnormal = fsys/(2*N*65536)

其中，N 表示的是预分频因子（1，8，64，256 或者 1024）。

输出比较单元可以用来产生中断，但是在普通模式下不推荐使用中断，这样会占用太多 CPU 的时间。

Normal Mode

It is the simplest mode for timer counter, in this mode waveform generation mode control bit WGM1[3:0]=0, counter maximum value TOP is MAX (0xFFFF). In this mode the counting direction is always up (incrementing), when counter reaches TOP and overrun, it goes back to BOTTOM and restart to accumulate. In normal operation the Timer/Counter Overflow Flag(TOV1) will be set in the same timer clock cycle as the TCNT1 becomes zero. The TOV1 Flag in this case behaves like a 17th bit, except that it is only set, not cleared. However, combined with the timer overflow interrupt that automatically clears the TOV1 Flag, the timer resolution can be increased by software. There are no special cases to consider in the Normal mode, a new counter value can be written anytime.

Output compare signal OC1x waveform can only be generated when setting data direction register of OC1x pin as output. If COM1x=1, compare match occur, than OC1x signal is toggled. In this case waveform frequency is calculated by below formula:

foc1xnormal = fsys/(2*N*65536)

Here, N indicates prescaler factor (1，8，64，256 or 1024)

Output compare unit can generate interrupt, but it is not recommend in normal mode, because it takes too much CPU time.

CTC 模式

设置 WGM1[3:0]=4 或 12 时，定时计数器 1 进入 CTC 模式。当 WGM1[3]=0 时，计数最大值 TOP 为 OCR1A，当 WGM1[3]=1 时，计数最大值 TOP 为 ICR1。下面以 WGM1[3:0]=4 为例来描述 CTC 模式在这个模式下，计数方式为每一个计数时钟加一递增，当计数器的数值 TCNT1 等于 TOP 时计数器清零。这个模式使得用户可以很容易的控制比较匹配输出的频率，也简化了外部事件计数的操作。

当计数器到达 TOP 时，输出比较匹配标志 OCF1 被置位，相应的中断使能置位时将会产生中断。在中断服务程序里可以更新 OCR1A 寄存器。在这个模式下 OCR1A 没有使用双缓冲，在计数器以无预分频器或很低的预分频器工作下将最大值更新为接近最小值的时候要小心。如果写入 OCR1A 的数值小于当时的 TCNT1 值时，计数器将丢失一次比较匹配。在下一次比较匹配发生之前，计数器不得不先计数到 MAX，然后再从 BOTTOM 开始计数到 OCR1A。和普通模式一样，计数值回到 0x0 的计数时钟里置位 TOV1 标志。

设置 OC1x 引脚的数据方向寄存器为输出时才能得到输出比较信号 OC1x 的波形。波形的频率可以用下面的公式来计算：

foc1xctc = fsys/(2*N*(1+OCR1A))

其中，N 表示的是预分频因子（1，8，64，256 或者 1024）。

从公式可以看出，当设置 OCR1A 为 0x0 且无预分频器时，可以获得最大频率为 fsys/2 的输出波形。

当 WGM1[3:0]=12 时与 WGM1[3:0]=4 类似，只是把与 OCR1A 相关的换成 ICR1 即可。

Set WGM1[3:0]=4 or 12, timer/ counter 1 is in CTC mode, max. If WGM1[3:0]=0, counter maximum value TOP is OCR1A.

If WGM1[3:0]=1, counter maximum value TOP is OCR1. Below is an example when WGM1[3:0]=4, in this mode, the counting direction is always up by each counting clock (incrementing), when counter TCNT1 reaches TOP, it is cleared. This mode makes users quite easy to control frequency of compare match output, also operation of external events counting is simplified.

When counter reaches TOP , output compare match flag OCF1 is set, corresponding interrupt enable will generate interrupt, the interrupt handler routine can be used to update OCR1A register. In this mode, changing TOP to a value close to BOTTOM when the counter is running with none or a low prescaler value must be done with care since the CTC mode does not have the double buffering feature. If the new value written to OCR1A is lower than the current value of TCNT1, the counter will miss the compare match. Before next compare match, the counter will then have to count to MAX and re-start from BOTTOM to count to OCR1A value. As for the Normal mode of operation, the TOV0 Flag is set when counting value return to 0x0 of counting clock.

Waveform of output compare signal OC1x is generated only if data direction register of OC1x pin is set output. In this case waveform frequency can be calculated by below formula:

foc1xctc = fsys/(2*N*(1+OCR1A))

Here, N indicates prescaler factor (1，8，64，256 or 1024)

Seeing from above formular, set OCR1A to 0x0 meanwhile without prescaler, output waveform of max. fsys/2 can be gained.

It is similar case when WGM1[3:0]=12 and WGM1[3:0]=4, only need to change OCR1A to ICR1 accordingly.

快速 PWM 模式

设置 WGM1[3:0]=5，6，7，14 或 15 时，定时计数器 1 进入快速 PWM 模式，计数最大值 TOP 分别为 0xFF，0x1FF，0x3FF，ICR1 或 OCR1A，可以用来产生高频的 PWM 波形。快速 PWM 模式和其他 PWM 模式不同在于它是单向操作。计数器从 BOTTOM 累加到 TOP 后又回到 BOTTOM 重新计数。当计数值 TCNT1 到达 TOP 或 BOTTOM 时，输出比较信号 OC1x 会被置位或清零，取决于比较输出模式 COM1 的设置，详情见寄存器描述。由于采用单向操作，快速 PWM 模式的操作频率是采用双向操作的相位修正 PWM 模式的两倍。高频特性使得快速 PWM 模式适用于功率调节，整流以及 DAC 应用。高频信号可以减小外部元器件（电感电容等）的尺寸，从而降低系统成本。

当计数值到达 TOP 时，定时计数器溢出标志 TOV1 将会被置位，并把比较缓冲器的值更新到比较值。如果中断使能，在中断服务程序中可以更新 OCR1A 寄存器。

设置 OC1x 引脚的数据方向寄存器为输出时才能得到输出比较信号 OC1x 的波形。波形的频率可用下面的公式来计算：

foc1xfpwm = fsys/(N*(1+TOP))

其中，N 表示的是预分频因子（1，8，64，256 或者 1024）。

当 TCNT1 和 OCR1x 发生比较匹配时，波形产生器就置位（清零）OC1x 信号，当 TCNT1 被清零时，波形产生器就清零（置位）OC1x 信号，以此来产生 PWM 波。由此 OCR1x 的极值将会产生特殊的 PWM 波形。当 OCR1x 设置为 0x00 时，输出的 PWM 为每(1+TOP)个计数时钟里有一个窄的尖峰脉冲。当 OCR1x 设置为 TOP 时，输出的波形为持续的高电平或低电平。如果用 OCR1A 作为 TOP 并设置 COM1A=1，输出比较信号 OC1A 会产生占空比为 50%的 PWM 波。

Fast PWM Mode

Fast PWM mode of Timer/Counter 1 (WGM0[2:0]= 3 or 7) provides a high frequency PWM waveform generation option

When set WGM1[3:0]=5，6，7，14 or 15. TOP is defined as 0xFF,0x1FF,0x3FF, ICR1 or OCR1A which can generate high frequency PMW waveform. The fast PWM differs from the other PWM option by its single-slope operation. The counter counts from BOTTOM to TOP then restarts from BOTTOM. When counting value TCNT1 reaches TOP or BOTTOM, output compare OC1x is set or cleared, which depends on configuration of compare output mode, details referring to sections about register.

Due to the single-slope operation, the operating frequency of the fast PWM mode can be twice as high as the phase correct PWM mode. This high frequency makes the fast PWM mode well suited for power regulation, rectification, and DAC applications. High frequency allows physically small sized external components (capacitors), and therefore reduces total system cost.

When counting value reaches TOP, timer counter overflow flag TOV1 will be set and value of compare buffer is updated to compare value. If interrupt is disable, compare buffer OCR1A register can be updated in interrupt routine program.

Waveform of output compare signal OC1x is generated only if data direction register of OC1x pin is set output. Waveform frequency can be calculated by below formular:

foc1xfpwm = fsys/(N*(1+TOP))

Here, N indicates prescaler factor (1，8，64，256 or 1024)

When compare match occur between TCNT1 and OCR1x, waveform generator is set to OC1x (cleared). When TCNT1 is cleared, waveform generator is (cleared) set to OC1x to generate PWM waveform, so extreme value of OCR1x will generate very special PWM waveform. If the OCR1x is set to 0x00 the output PWM will be a narrow spike for each

1+TOP timer clock cycle. Setting the OCR1x as TOP will result in a constantly high or low change level. If OCR1A is TOP and setting COM1A=1, output compare signal OC1A will generate PWM waveform with 50% duty cycle.

相位修正 PWM 模式

当设置 WGM0[3:0]=1，2，3，10 或 11 时，定时计数器 1 进入相位修正 PWM 模式，计数的最大值 TOP 分别为 0xFF， 0x1FF， 0x3FF， ICR1 或 OCR1A。计数器采用双向操作，由 BOTTOM 递增到 TOP，然后又递减到 BOTTOM，再重复此操作。计数到达 TOP 和 BOTTOM 时均改变计数方向，计数值在 TOP 或 BOTTOM 上均只停留一个计数时钟。在递增或递减过程中，计数值 TCNT1 与 OCR1x 匹配时，输出比较信号 OC1x 将会被清零或置位，取决于比较输出模式 COM1 的设置。与单向操作相比，双向操作可获得的最大频率要小，但其极好的对称性更适合于电

机控制。

相位修正 PWM 模式下，当计数到达 BOTTOM 时置位 TOV1 标志，当计数到达 TOP 时把比较缓冲器的值更新到比较值。如果中断使能，在中断服务程序中可以更新比较缓冲器 OCR1x 存器。

设置 OC1x 脚的数据方向寄存器为输出时才能得到输出比较信号 OC1x 波形。波形的频率可用下面的公式来计算：
foc1xcpcpwm = fsys/(N*TOP*2)
其中，N 表示的是预分频因子（1，8，64，256 或者 1024）。

在递增计数过程中，当 TCNT1 与 OCR1x 匹配时，波形产生器就清零（置位）OC1x 信号。在递减计数过程中，当 TCNT1 与 OCR1x 匹配时，波形产生器就置位（清零）OC1x 信号。由此 OCR1x 的极值会产生特殊的 PWM 波。当 OCR1x 设置为 TOP 或 BOTTOM 时， OC1x 信号输出会一直保持低电平或高电平。如果用 OCR1A 作为 TOP 并设置 COM1A=1，输出比较信号 OC1A 会产生占空比为 50%的 PWM 波。
为了保证输出 PWM 波在 BOTTOM 两侧的对称性，在没有发生比较匹配时，有两种情况下也会翻转 OC1x 信号。第一种情况是，当 OCR1x 的值由 TOP 改变为其他数据时。当 OCR1x 为 TOP，计数值达到 TOP 时， OC1x 的输出与前面降序计数时比较匹配的结果相同，即保持 OC1x 不变。此时会更新比较值为新的 OCR1x 的值（非 TOP），OC1x 的值会一直保持，直到升序计数时发生比较匹配而翻转。此时 OC1x 信号并不以最小值为中心对称，因此需要在 TCNT1 到达最大值时翻转 OC1x 信号，此即没有发生比较匹配时翻转 OC1x 信号的第一种情况。
第二种情况是，当 TCNT1 从比 OCR1x 高的值开始计数时，因而会丢失一次比较匹配，从而引起不对称情形的产生。同样需要翻转 OC1x 信号去实现最小值两侧的对称性。

Phase Correct PWM Mode

Setting WGM0[3:0] to 1，2，3，10 or 11, Timer/Counter 1 enter into Phase Correct PWM mode, each TOP value is 0xFF, 0x1FF, 0x3FF, ICR1 or OCR1A. The phase correct PWM mode is based on a dual-slope operation. The counter counts repeatedly TOP (increment) and then to BOTTOM (decrement).
When counter reaches TOP or BOTTOM, it will both change counting direction. TOP or BOTTOM value only stay for one counting clock.
On the compare match between TCNT1 and OCR1x while up-counting and down-counting, the Output Compare (OC1x) is cleared or set depending on the configuration of compare output mode COM1. The dual-slope operation has lower maximum operation frequency than single slope operation. However, due to the symmetric feature of the dual-slope PWM modes, these modes are preferred for motor control applications.

In phase correct PWM mode, the TOV1 is set each time the counter reaches BOTTOM. When the counter reaches TOP, value of compare buffer is updated to compare value. If interrupt enable, compare buffer OCR1x register can be updated in interrupt routine program.
Setting OC1x pin data direction register to output, it will generate waveform of output compare OC1x. Waveform frequency can be calculated via below formula:

foc1xcpcpwm = fsys/(N*TOP*2)

Here, N indicates prescaler factor (1，8，64，256 or 1024)
During up-counting, when compare match occur between TCNT1 and OCR1x, waveform generator clear (set) OC1x . During down-counting, when compare match occur between TCNT1 and OCR1x, waveform generator set (clear) OC1x, so extreme value of OCR1x will generate very special PWM waveform. Setting the OCR1x equal to TOP or BOTTOM will result in a constantly high or low level change. If OCR1A is TOP and setting COM1A=1, output compare signal OC1A will generate PWM waveform with 50% duty cycle.

OC1x will be toggled in two cases when compare match does not occur while to ensure symmetry of both

side of PMW waveform at its bottom value.
The first case : OCR1x value is changed from TOP to others. When the OCR0x value is TOP, counting value reach TOP, OC1x output is the same as the result of previous down-counting Compare Match, that is to say OC0x keep unchanged.

In this case Compare value is updated to a new OCR1x value (not TOP). OC1x keeps unchanged till it is toggled in up-counting compare match. At this moment OC1x is not symmetric with minimum value, so OC1x need to be toggled when TCNT1 reach maximum value.
The 2nd case: when TCNT1 starts to count from value higher than OCR1x, one compare match will loss, which will result in asymmetric. Same is that to achieve symmetric at both side of bottom value, OC1x is needed to be toggled.

相位频率修正 PWM 模式
当设置 WGM0[3:0]=8 或 9 时，定时计数器 1 进入相位频率修正 PWM 模式，计数的最大值 TOP 分别为 ICR1 或 OCR1A。计数器采用双向操作，由 BOTTOM 递增到 TOP，然后又递减到 BOTTOM，再重复此操作。计数到达 TOP 和 BOTTOM 时均改变计数方向，计数值在 TOP 或 BOTTOM 上均只停留一个计数时钟。在递增或递减过程中，计数值 TCNT1 与 OCR1x 匹配时，输出比较信号 OC1x 将会被清零或置位，取决于比较输出模式 COM1 的设置。与单向操作相比，双向操作可获得的最大频率要小，但其极好的对称性更适合于电机控制。

相位频率修正 PWM 模式下，当计数到达 BOTTOM 时置位 TOV1 标志，并且把比较缓冲器的值更新到比较值，更新比较值的时间是相位频率修正 PWM 模式和相位修正 PWM 模式的最大不同点。如果中断使能，在中断服务程序中可以更新比较缓冲器 OCR1x 存器。当 CPU 改变 TOP 值即 ORC1A 或 ICR1 的值时，必须保证新的 TOP 值不小于已经在使用的 TOP 值，否则比较匹配将不会再发生。

设置 OC1x 脚的数据方向寄存器为输出时才能得到输出比较信号 OC1x 波形。波形的频率可用下面的公式来计算：
foc1xcpfcpwm = fsys/(N*TOP*2)
其中，N 表示的是预分频因子（1，8，64，256 或者 1024）。

在递增计数过程中，当 TCNT1 与 OCR1x 匹配时，波形产生器就清零（置位）OC1x 信号。在递减计数过程中，当 TCNT1 与 OCR1x 匹配时，波形产生器就置位（清零）OC1x 信号。由此 OCR1x 的极值会产生特殊的 PWM 波。当 OCR1x 设置为 TOP 或 BOTTOM 时， OC1x 信号输出会一直保持低电平或高电平。如果用 OCR1A 作为 TOP 并设置 COM1A=1，输出比较信号 OC1A 会产生占空比为 50%的 PWM 波。

因为 OCR1x 寄存器是在 BOTTOM 时刻更新的，所以 TOP 值两边升序和降序的计数长度是一样的，也就产生了频率和相位都正确的对称波形。

当使用固定 TOP 值时，最好采用 ICR1 寄存器作为 TOP 值，即设置 WGM1[3:0]=8，此时 OCR1A 寄存器只需用来产生 PWM 输出。如果要产生频率变化的 PWM 波，必须通过改变 TOP 值，OCR1A 的双缓冲特性会更适合于这个应用。

Phase Frequency Correct PWM Mode

Setting WGM0[3:0] to 8 or 9, Timer/Counter 1 enter into Phase Frequency Correct PWM mode, each TOP value is ICR1 or OCR1A. In this mode counter is based on a dual-slope operation. The counter counts repeatedly to TOP (increment) and then to BOTTOM (decrement).
When counter reaches TOP or BOTTOM, it will both change counting direction. TOP or BOTTOM value only stay for one counting clock.
On the compare match between TCNT1 and OCR1x while up-counting and down-counting, the Output Compare (OC1x) is cleared or set depending on the configuration of compare output mode COM1. The dual-slope operation has lower maximum operation frequency than single slope operation. However, due to the symmetric feature of the dual-slope PWM modes, these modes are preferred for motor control applications.

In phase frequency correct PWM mode, the TOV1 is set each time the counter reaches BOTTOM. When the

counter reaches TOP, value of compare buffer is updated to compare value. The time of updating compare value is the main difference between phase frequency correct PWM mode and phase correct PWM mode. If interrupt enable, compare buffer OCR1x register can be updated in interrupt routine program.
If CPU changes TOP, that is ORC1A or ICR1, it is a must to ensure that TOP value is no lower than TOP value that is under use, otherwise compare match will not occur.
Setting OC1x pin data direction register to output, it will generate waveform of output compare OC1x. Waveform frequency can be calculated via below formula:

foc1xcpfcpwm = fsys/(N*TOP*2)

Here, N indicates prescaler factor (1，8，64，256 or 1024)
During up-counting, when compare match occur between TCNT1 and OCR1x, waveform generator clear (set) OC1x . During down-counting, when compare match occur between TCNT1 and OCR1x, waveform generator set (clear) OC1x, so extreme value of OCR1x will generate very special PWM waveform. Setting the OCR1x to TOP or BOTTOM will result in a constantly high or low level change. If OCR1A is TOP and setting COM1A=1, output compare signal OC1A will generate PWM waveform with 50% duty cycle.

OCR1x register is updated at moment of BOTTOM, so counting length of up-counting and down-counting at both side of TOP value is the same, than symmetric waveform of same frequency and phase will be generated.

If using a fixed TOP value, had better use ICR1 register as TOP value, setting WGM1[3:0]=8, than OCR1A register is only used to generate PWM output. If PWM waveform with changing frequency is in need of, must change TOP value. Dual buffer of OCR1A is more suitable for this application.

输入捕捉模式

输入捕捉用来捕获外部事件，并为其赋予时间标记以说明此事件发生的时刻，可以在前面的计数模式下进行，不过要除去使用 ICR1 值作为计数 TOP 值的波形产生模式。

外部事件发生的触发信号由引脚 ICP1 输入，也可以通过模拟比较器单元来实现。当引脚 ICP1 上的逻辑电平发生变化，或模拟比较器的输出 ACO 电平发生变化，并且这个电平变化被输入捕捉单元所捕获，输入捕捉即被触发，此时 16 位的计数值 TCNT1 数据被复制到输入捕捉寄存器 ICR1，同时输入捕捉标志 ICF1 置位，若 ICIE1 位为“1”，输入捕捉标志将产生输入捕捉中断。

通过设置模拟比较控制与状态寄存器 ACSR 的模拟比较输入捕捉控制位 ACIC 来选择输入捕捉触发源 ICP1 或 ACO。需注意的是，改变触发源有可能造成一次输入捕捉，因此在改变触发源后必须对 ICF1 进行一次清零操作来避免出现错误的结果。

输入捕捉信号经过一个可选的噪声抑制器之后送入边沿检测器，根据输入捕捉选择控制位 ICES1 的配置，看检测到的边沿是否满足触发条件。噪声抑制器是一个简单的数字滤波，对输入信号进行 4 次采样，只有当 4 次采样值都相等时其输出才会送入边沿检测器。噪声抑制器由 TCCR1B 寄存器的 ICNC1 位控制其使能或禁止。

使用输入捕捉功能时，当 ICF1 被置位后，应尽可能早的读取 ICR1 寄存器的值，因为下一次捕捉事件发生后 ICR1 的值将会被更新。推荐使能输入捕捉中断，在任何输入捕捉工作模式下，都不推荐在操作过程中改变计数 TOP 值。

输入捕捉到的时间标记可用来计算频率、占空比及信号的其它特征，以及为触发事件创建日志。测量外部信号的占空比时要求每次捕捉后都要改变触发沿，因此读取 ICR1 值以后须尽快改变触发的信号边沿。

Input Capture Mode

Input capture mode is used to capture external events, it gives external events a time flag as record. This mode is used in previous counter modes but not include in waveform generation mode when using ICR1 as counting TOP value.

Trigger signal of external events is input via pin ICP1, it can also be realized via analog comparator unit. If logic change level on pin ICP1 changes, or output AC0 level change of analog comparator changes, input capture is triggered when the just mentioned level change is captured by input capture unit. At this moment, 16-bit counting value TCNT1 data is copied to input capture register ICR1, meanwhile input capture flag ICF1 is set. If setting ICIE1 to logic one, input capture flag will generate input capture interrupt.

By setting analog compare input capture control bit（ACIC）of analog compare control and status register (ACSR), ICPI or AC0, input capture trigger source, can be selected. Noted that one change of trigger source would result in one input capture, so after the change, must clear ICF1 to avoid a mistake.

Input capture signal is send to edge inspector after going through optional noise suppression, to see if the inspected edge satisfy trigger condition based on configuration of input capture selection control bit ICES1. Noise surpression is a very simple digital filter; it samples the input signal for 4 times. Its output is send to edge inspector only if the 4 sampling value are the same. Noice surpression is set for enable or disable via ICNC1 of TCCR1B register.

When using input capture, ICF1 is set, should read ICR1 register value ASAP, because ICR1 value will be updated in the next capture event. It is recommend to enable input capture interrupt, and in whatever input capture mode, we suggest not to chang TOP value during operation.

Input capture timig flag can be used to calculate frequency, empty percentage and other characteristic of signal, it can be also used to create log for trigger events. For measuring empty percentage of external signal, it is required to change trigger edge after each capture, so has to change edge of triggered signal ASAP after reading ICR1 value.

PWM 输出的自动关闭与重启

当设置 TCCR1C 寄存器的 DOC1x 位为高时， PWM 输出的自动关闭功能会被使能，满足触发条件时，硬件会清零相应的 COM1x 位，将 PWM 输出信号 OC1x 与其输出引脚断开，切换成通用 IO 输出，实现 PWM 输出的自动关闭。此时，输出引脚的状态可由通用 IO 口的输出来控制。

PWM 输出的自动关闭被使能后，还需要设置其触发条件，由 TCCR1D 寄存器的 DSX1n 位来选择触发源。触发源有模拟比较器中断，外部中断，引脚电平变化中断以及定时器溢出中断，具体情形请参考 TCCR1D 寄存器描述。当某个或某些触发源被选用作为触发条件后，在这些中断标志位被置位的同时，硬件会清零 COM1x 位来关闭 PWM 的输出。

当发生了触发事件关闭 PWM 输出后，定时器模块没有相应的中断标志位，软件需要通过读取触发源的中断标志位来得知触发条件和触发事件。

当 PWM 输出被自动关闭而需要再次重启输出时，软件只需要重新设置 COM1x 位，来切换 OC1x 信号输出到相应的引脚上。需要注意的是，发生自动关闭后，定时器并未停止工作，OC1x 信号的状态也一直在更新。软件可在定时器发生溢出或比较匹配后，再设置 COM1x 位来输出 OC1x 信号，这样可以获得明确的 PWM 输出状态。

Auto shut down and re-start of PWM output

If DOC1x of TCCR1C register is set high, PWM output enable to shut down automatically. Once trigger

conditins is satisfied, hardware will clear corresponding COM1x, PWM output OC1x will be apart from output pin and switch to general IO output, by this way automate shut-down of PWM output is done. Under this situation, status of output pin is controlled via output of general IO port.
When automate shut-down of PWM output is enable, trigger condition must be set. Trigger source is set by DSX1n of TCCR1D register. Trigger source has analog compare interrupt, external interrupt, pin change interrupt and timer overflow interrupt, for details please refer to section of TCCR1D register. If some of these trigger source is selected and these interrupt flag is set at the same time, hardware will clear COM1x to shut down PMW output.

When trigger events occur and PWM output is shut down, if no interrupt flag in timer module, trigger event and trigger condition can only be known when software read interrupt flag of trigger source.

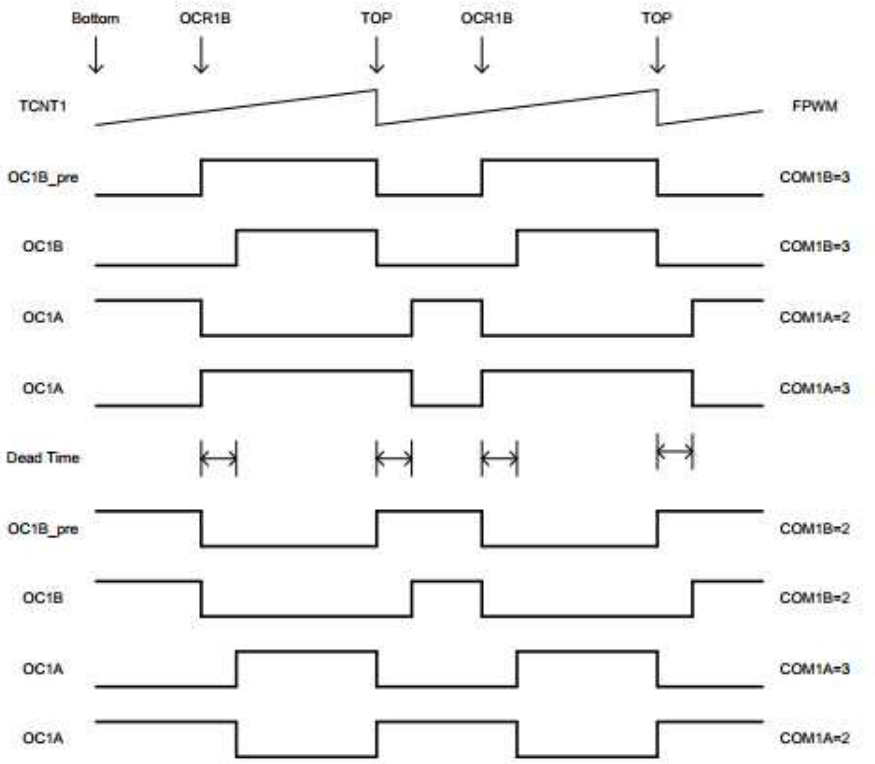
If PWM need to re-start output, it is only necessary to use software re-set COM01x to switch OC1x output into related pins. Note that after automate shut-down occur, timer will not stop working, OC1x status is keeping updating. Software can set COM1x to output OC1x after overflow or compare match occurs on timer, by this way accurate PMW output can be gotten.

死区时间控制

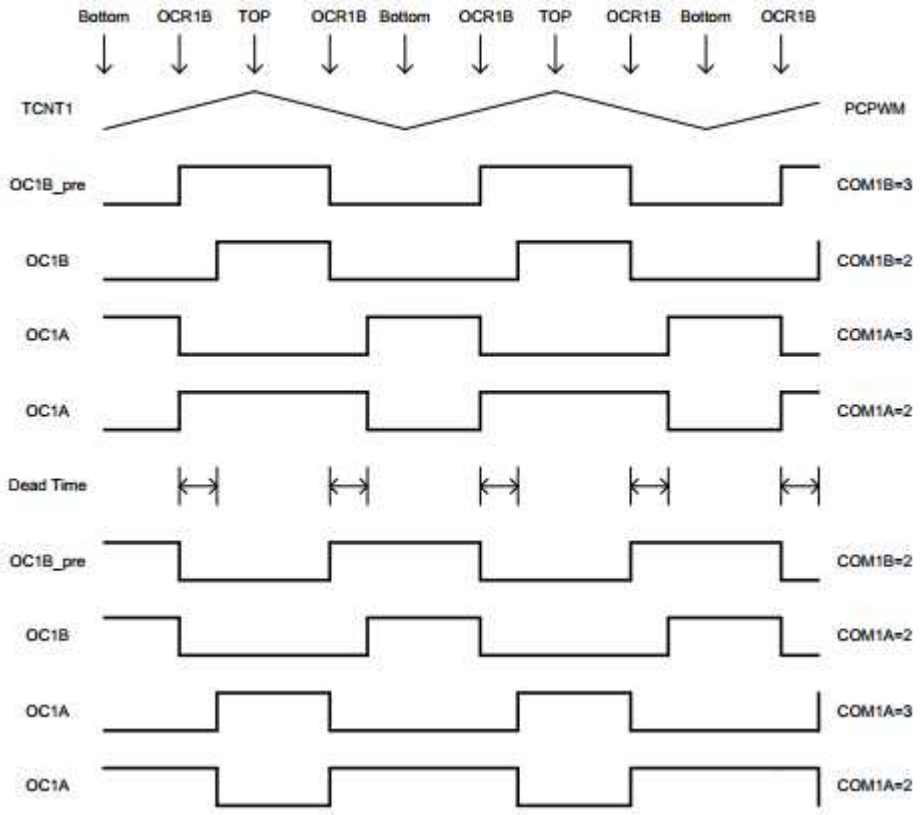
设置 DTEN1 位为“1”时，插入死区时间的功能被使能，OC1A 和 OC1B 的输出波形将在 B 通道比较输出所产生的波形基础上插入设定的死区时间，时间的长度为 DTR1 寄存器的计数时钟数所对应的时间值。如下图所示，OC1A 和 OC1B 的死区时间插入均是以通道 B 的比较输出波形为基准。当 COM1A 和 COM1B 同为“2”或“3”时，OC1A 的波形极性与 OC1B 的波形极性相同，当 COM1A 和 COM1B 分别为“2”或“3”时，OC1A 的波形与 OC1B 的波形极性相反。

Dead time control

Setting DTEN1 to logic one, it enables inserting dead time, OC1A and OC1B output wave will insert designed dead time based on waveform generated by B channel compare output, length of time is the timing value corresponded to counter clock number of DTR1 register. As shown below, OC1A and OC1B will insert designed dead time based on waveform generated by B channel compare output. If COM1A and COM1B are set to 2 or 3 at the same time, waveform polarity of OC1A and OC1B are the same, if they are set to 2 or 3 individually, waveform polarity of OC1A is opposite to that of OC1B.



Firgure 3　TC1 dead time control in FPWM mode



Firgure 4　TC1 dead time control in PCPWM mode

Setting DTEN1 to logic zero, function of inserting dead time is disabled. Output wave of OC1A and OC1B is

their own waveform generated by compare output.

高速计数模式

高速时钟模式下，采用更高频率的时钟作为计数的时钟源，用来产生更高速度和更高分辨率的 PWM 波形。此高频时钟是通过对内部 32M RC 振荡器的输出时钟 rc32m 进行 2 倍频来产生的。因此，在进入高频模式之前，需先使能内部 32M RC 振荡器的倍频功能，即置位 TCKCSR 寄存器的 F2XEN 位，并等待一定时间直到倍频时钟信号输出稳定。然后，可置位 TCKCSR 的 TC2XS1 位来使定时计数器进入高速时钟模式。

在此模式下，系统时钟与高速时钟是异步关系，而部分寄存器（见 TC1 寄存器列表）是工作在高速时钟域，因此，配置和读取这类寄存器时也是异步的，操作时需注意。

对高速时钟域下的寄存器进行非连续读写操作时无特殊要求，而进行连续读写操作时，需等待一个系统时钟，可按以下步骤：

5） 写寄存器 A；

6） 等待一个系统时钟（NOP 或操作系统时钟下的寄存器）；

7） 读或写寄存器 A 或 B。

8） 等待一个系统时钟（NOP 或操作系统时钟下的寄存器）。

对高速时钟域下的寄存器进行读操作时，宽度为 8 位的寄存器均可直接读取，而读取 16 位寄存器的值（OCR1A，OCR1B, ICR1, TCNT1）时，先读取低位寄存器的值，等待一个系统时钟后，再读取高位寄存器的值，而在读取 TCNT1 的值时，当计数器还在进行计数时，TCNT1 的值会随高速时钟变化，可暂停计数器（设置 CS1 为零）再读取 TCNT1 的值。

读取 OCR1A，OCR1B 和 ICR1 时，可按以下步骤：

1） 读取 OCR1AL/OCR1BL/ICR1L；

2） 等待一个系统时钟（NOP）；

3） 读取 OCR1AH/OCR1BH/ICR1H。

读取 TCNT1 时，可按以下步骤：

1） 置 CS1 为零；

2） 等待一个系统时钟（NOP）；

3） 读取 TCNT1L 的值；

4） 等待一个系统时钟（NOP）；

读取 TCNT1H 的值。

High Speed Clock Mode

In high speed clock mode, higher frequency clock is used as clock source of timer, to generate higher speed and higher resolution PMW waveform. This high frequency clock is generated from 2 times frequency which is a result from output clock rc32m of internal 32M RC oscillator. Before high frequency mode, frequency doubling function of internal 32M RC oscillator must be enabled, set F2XEN of TCKCSR register, wait till signal of frequency doubling clock is stable, than set TC2XS1 bit of TCKCSR to make time counter enter into high speed clock mode.

In this mode, system clock and high speed clock is asynchronous, some registers (referring to TC1 register list) are working in high speed area, so during operation must pay attention that configuration and reading of such registers is also asynchronous.

For non-continuous read-write on registers working in high speed clock area, there is no special requirement, while for continuous read-write operation, it needs to wait for one system clock, operation step is as below:

5) Write register A

6) Wait for a system clock (NOP or operation system clock register)

7) Read or write register A or B
8) Wait for a system clock (NOP or operation system clock register)

For read operation on registers in high speed clock area, register with 8 bit can be access to read and fetch directly, while for 16 bitregister (OCR1A, OCR1B, ICR1, TCNT1), must read lower register firstly, than read high value after a system clock. When reading TCNT1, counter is still counting number, TCNT1 value will change as per high speed clock, in this case can stop counter (setting CS1 to zero) to read TCNT1 value.

When reading OCR1A, OCR1B, ICR1 and TCNT1, below steps are as reference:
1） ready OCR1AL/OCR1BL/ICR1L
2） wait for one system clock (NOP)
3） read OCR1AH/OCR1BH/ICR1H

When reading TCNT1, below steps are as reference:
1) set CS1 to logic zero
2) wait for one system clock (NOP)
3) read TCNT1L
4) wait for one system clock (NOP)
Read TCNT1H value

Register defition

TC1 Register List

Register	Address	Default value	Comments
TCCR1A *	0x80	0x00	TC1 control register A
TCCR1B*	0x81	0x00	TC1 control register B
TCCR1C*	0x82	0x00	TC1 control register C
DSX1	0x83	0x00	TC1 trigger source control register
TCNT1L *	0x84	0x00	TC1 counting value register LOW byte
TCNT1H*	0x85	0x00	TC1 counting value register HIGH byte
ICR1L *	0x86	0x00	TC1 input capture register LOW byte
ICR1H*	0x87	0x00	TC1 input capture register HIGH byte
OCR1AL *	0x88	0x00	TC1 input capture register A low byte
OCR1AH*	0x89	0x00	TC1 output compare register A high byte
OCR1BL *	0x8A	0x00	TC1 output compare register B low byte
OCR1BH*	0x8B	0x00	TC1 output compare register B high byte
DTR1*	0x8C	0x00	TC1 dead time control register
TIMSK1	0x6F	0x00	Timer counter interrupt mask register
TIFR1	0x36	0x00	Timer counter interrupt flag register
TCKCSR1	0xEC	0x00	TC1 clock control status register
TCKCSR	0xEC	0x00	TC clock control and status register

Note that: Register with * are working in system clock and high speed clock area,while register without * are only woking in system clock area.

TCCR1A –TC1 Control Register A

TCCR1A –TC1 Control Register A

Address: 0x80					Default value: 0x00			
Bit	7	6	5	4	3	2	1	0
	COM1A1	COM1A0	COM1B1	COM1B0	-	-	WGM11	WGM10
R/W	R/W	R/W	R/W	R/W	-	-	R/W	R/W

Bit	Name	Comment
7	COM1A1	Compare match output A mode control high COM1A1 and COM1A0 combine to consist of COM1A[1:0], used to control 0C1A output compare waveform. If 1 or 2 of the COM0A bits are set, output compare waveform takes up 0C1A pin, however whose data direction regester must be set to high. In different mode, C0M1A control output compare waveform differently, for details referring to list on compare output control.
6	COM1A0	Compare match output A mode control low COM1A1 and COM1A0 combine to consist of COM1A[1:0], used to control 0C1A output compare waveform. If 1 or 2 of the COM0A bits are set, output compare waveform takes up 0C1A pin, however whose data direction regester must be set to high. In different mode, C0M1A control compare waveform differently, for details referring to list on compare output control.
5	COM1B1	Compare match output B mode control high COM1B1 and COM1B0 combine to consist of COM1B[1:0], used to control 0C1B output compare waveform. If 1 or 2 of the COM0B bits are set, compare waveform takes up 0C1B pin, however whose data direction regester must be set to high. In different mode, C0M1B control compare waveform differently, for details referring to list on compare output control.
4	COM1B0	Compare match output B mode control low COM1B1 and COM1B0 combine to consist of COM1B[1:0], used to control 0C1B output compare waveform. If 1 or 2 of the COM0B bits are set, compare waveform takes up 0C1B pin, however whose data direction regester must be set to high. In different mode, C0M1B control compare waveform differently, for details referring to list on compare output control.
3:2	-	reserved
1	WGM11	Waveform generation mode control sub-low WGM11, WGM13, WGM12 and WGM10 generate waveform generation control WGM1[3:0] together, counting method and waveform generation method of control counter is referring to list on waveform generation mode.
0	WGM10	Waveform generation mode control lowest WGM10, WGM13, WGM12 and WGM11 generate waveform generation control WGM1[3:0] together, counting method and waveform generation method of control counter is referring to list on waveform generation mode.

Below list shows how compare output mode controls output compare waveform in non-PWM mode (normal mode and CTC mode)

COM1x[1:0]	Comment
0	OC1x disconnect, general IO port operation
1	Toggle OC1x signal in compare match

2	Clear OC1x signal in compare match
3	Set OC1x signal in compare match

Below list shows how compare output mode controls output compare waveform in fast PWM mode

COM1x[1:0]	Comment
0	OC1x disconnect, general IO port operation
1	If WGM1 is 15, OC1A signal is toggled in compare match and OC1B is disconnected If WGM1 is other value, OC1x disconnected, general IO port operation
2	Clear OC1x signal in compare match, while set OC1x in maximum value match
3	Set OC1x signal in compare match, while clear OC1x in maximum value match

Below list shows how compare output mode controls output compare waveform in phase correction mode

COM1x[1:0]	Comment
0	OC1x disconnect, general IO port operation
1	If WGM1 is 9 or 11, OC1A signal is toggled in compare match and OC1B is disconnected If WGM1 is other value, OC1x disconnected, general IO port operation
2	OC1x is cleared in up-counting compare match, and set OC1x in down-counting compare match
3	OC1x is set in up-counting compare match, and clear OC1x in down-counting compare match

TCCR1B –TC1 control register B

TCCR1B –TC1 control register B								
Address: 0x81					Default value: 0x00			
Bit	7	6	5	4	3	2	1	0
	ICNC1	ICES1	-	WGM13	WGM12	CS12	CS11	CS10
R/W	R/W	R/W	-	R/W	R/W	R/W	R/W	R/W
Bit	Name	Comment						
7	ICNC1	Input capture noise supression enable control If setting ICNC1 to logic one, input capture noise surpression is enable, than input of external pin ICP1 is filtered, input signal is acitvaed if 4 continuous sampling are the same value, becaues of this funtion input capture is delayed for 4 clock cycles. If setting ICNC1 to logic zero, input capture noise supression is disable, than input of external pin ICP1 is activated directly.						
6	ICES1	Input capture trigger edge selection control If setting ICES1 to logic one, rising edge of selected pin change level will trigger input capture If setting ICES1 to logic zero, falling edge of selected pin change level will trigger input capture. When an event is captured, counter value will be copied to ICR1 register, meanwhile input capture flag ICF1 is set. If interrupt is enable, input capture interrupt is generated.						
5	-	reserved						
4	WGM13	Waveform generation mode control high						

		WGM13 and WGM12, WGM11,WGM10 together consist of waveform generation control WGM1[3:0], counting method and waveform generation method of control counter is referring to list on waveform generation mode.	
3	CS12	Clock selection control high Used to select clock source of timer/counter 1	
1	CS11	Clock selection control low Used to select clock source of timer/counter 1	
0	CS10	Clock selection control low Used to select clock source of timer/counter 1	
		CS1[2:0]	comments
		0	No clock source, stop couting
		1	clksys
		2	Clksys/8, from frequency prescaler
		3	Clksys/64, from frequency prescaler
		4	Clksys/256, from frequency prescaler
		5	Clksys/1024, from frequency prescaler
		6	External clock T1 pin, falling edge trigger
		7	External clock T1 pin, rising edge trigger

Below list shows waveform generation mode control

WGM1[3:0]	Working Mode	TOP Value	Update OCR0 Moment	Set T0V0 Moment
0	Normal	0xFFFF	Immediately	MAX
1	8-bit PCPWM	0x00FF	TOP	BOTTOM
2	9-bit PCPWM	0x01FF	TOP	BOTTOM
3	10-bit PCPWM	0x03FF	TOP	BOTTOM
4	CTC	OCR1A	Immediately	MAX
5	8-bit FPWM	0x00FF	BOTTOM	TOP
6	9-bit FPWM	0x01FF	BOTTOM	TOP
7	10-bit FPWM	0x03FF	BOTTOM	TOP
8	PFCPWM	ICR1	BOTTOM	TOP
9	PFCPWM	OCR1A	BOTTOM	TOP
10	PCPWM	CR1	TOP	BOTTOM
11	PCPWM	OCR1A	TOP	BOTTOM
12	CTC	ICR1	Immediately	MAX
13	reserved	-	-	-
14	FPWM	ICR1	TOP	TOP
15	FPWM	OCR1A	TOP	TOP

TCCR1C –TC1 Control Register C

TCCR1C –TC1 Control Register C								
Address: 0x82					Default value: 0x00			
Bit	7	6	5	4	3	2	1	0
	FOC1A	FOC1B	DOC1B	DOC1A	DTEN1	-	-	-
R/W	R/W	R/W	R/W	R/W	R/W	-	-	-

Bit	Name	Comment
7	FOC1A	Force output compare A In non-PWM mode, setting force input compare bit FOC1A to logice one, compare match will generate. Force compare match cannot set OCF1A flag, also not reload or clear counter, but output pin OC1A is to be updated according to COM1A configuration, it seems that a real compare match occurs. In PWM mode, must clear TCCR1A when writting it. Return value of read-fetch FOC1A is always zero.
6	FOC1B	Force output compare B In non-PWM mode, setting force input compare bit FOC1B to logice one, compare match will generate. Force compare match cannot set OCF1B flag, also not reload or clear counter, but output pin OC1B is to be updated according to related COM1B configuration, it seems that a real compare match occurs. In PWM mode, must clear TCCR1A when writting. Return value of read-fetch FOC1A is always zero.
5	DOC1B	TC1 shut down output compare enable control high If setting DOC1B to logic one, trigger source shut down output compare signal OC1B is enable. If trigger event occurs, hardware clear COM1B automatically and shut down waveform output of OC1B. By setting COM1B software will re-open PWM output. If setting DOC1B to logic zero, trigger source shut down output compare signal OC1B is disable
4	DOC1A	TC1 shut down output compare enable control low If setting DOC1A to logic one, trigger source shut down output compare signal OC1A is enable. If trigger event occurs, hardware will clear COM1A automatically and shut down waveform output of OC1A. By setting COM1A software will re-open PWM output. If setting DOC1A to logic zero, trigger source shut down output compare signal OC1B is disable
3	DTEN1	TC1 Dead time enable control bit If setting DTEN1 to logic one, inserting dead time is enable. Based on waveform generated in compare output channel B, OC1A and OC1B both insert dead time, whose intervals are decided by corresponding counting time of DTR1 register. Polarity of OC1A output waveform is decided by relationship between COM1A and COM1B, for details referring to list of waveform polarity after OC1A inserting dead time.
2:0	-	reserved

Below list shows control on polarity of OC1A output waveform when enable dead time

DTEN1	COM1A[1:0]	COM1B[1:0]	comment
0	-	-	OC1A polarity is decided by OC1A compare output mode control
1	0	-	OC1A disconnected, general IO port operation
1	1	-	reserved
1	2	2	OC1A signal is the same as OC1B polarity
		3	OC1A signal is opposite to OC1B polarity
1	3	2	OC1A signal is opposite to OC1B polarity
		3	OC1A signal is the same as OC1B polarity

Noted that:

OC1B output waveform polarity is controlled by OC1B compare output mode, this is the same when insert dead time is disable.

TCCR1D-TC1 Control Register D

TCCR1D-TC1 Control Register D								
Address: 0x83					Default value: 0x00			
Bit	7	6	5	4	3	2	1	0
	DSX17	DSX16	DSX15	DSX14	-	-	DSX11	DSX10
R/W	R/W	R/W	R/W	R/W	-	-	R/W	R/W

Bit	Name	Comment
7	DSX17	7 bit of TC1 trigger source selection control is enable When setting DSX17 to logic one, TC1 overflow, used as trigger source to shut down output compare signal waveform 0C0A/0C0B, is enable. When setting DOC1A/DOC1B to logic one, rising edge of interrupt flag register bit of selected trigger source will automatically shutdown 0C1A/0C1B waveform output. When setting DSX17 to logic zero, TC0 overflow, used as trigger source to shut down output compare signal waveform 0C1A/0C1B, is disable.
6	DSX16	6 bit of TC1 trigger source selection control is enable When setting DSX16 to logic one, TC2 overflow, used as trigger source to shut down output compare signal waveform 0C1A/0C2B, is enable. When setting DOC1A/DOC1B to logic one, rising edge of interrupt flag register bit of selected trigger source will automatically shutdown 0C1A/0C1B wave output When setting DSX16 to logic zero, TC2 overflow, used as trigger source to shut down output compare signal waveform 0C1A/0C1B, is disable.
5	DSX15	5 bit of TC1 trigger source selection control is enable When setting DSX15 to logic one, pin change level 1, used as trigger source to shut down output compare signal waveform 0C1A/0C1B, is enable. When setting DO1A/DOC1B to logic one, rising edge of interrupt flag register bit of selected trigger source will automatically shutdown 0C1A/0C1B waveform output When setting DSX15 to logic zero, pin change level 1, used as trigger source to shut down output compare signal waveform 0C1A/0C1B, is disable.
4	DSX14	4 bit of TC1 trigger source selection control is enable When setting DSX14 to logic one, external interrupt 1, used as trigger source to shut down output compare signal waveform 0C1A/0C1B, is enable. When setting DOC1A/DOC1B to logic one, rising edge of interrupt flag register bit of selected trigger source will automatically shutdown 0C1A/0C1B wave output When setting DSX14 to logic zero, external interrupt 1, used as trigger source to shut down output compare signal waveform 0C1A/0C1B, is disable.
3:2	-	reserved
1	DSX11	1 bit of TC1 trigger source selection control is enable When setting DSX11 to logic one, analog comparator 1, used as trigger source to shut down output compare signal waveform 0C1A/0C1B, is enable. When setting DOC1A/DOC1B to logic one, rising edge of interrupt flag register bit of selected trigger

		source will automatically shutdown 0C1A/0C1B waveform output When setting DSX11 to logic zero, analog comparator 1, used as trigger source to shut down output compare signal waveform 0C1A/0C1B, is disable.
0	DSX10	0 bit of TC1 trigger source selection control is enable When setting DSX10 to logic one, analog comparator 0, used as trigger source to shut down output compare signal waveform 0C1A/0C1B, is enable. When setting DOC1A/DOC1B to logic one, rising edge of interrupt flag register bit of selected trigger source will automatically shutdown 0C1A/0C1B wave output When setting DSX10 to logic zero, analog comparator 0, used as trigger source to shut down output compare signal waveform 0C1A/0C1B, is disable.

Below list shows selection control of waveform output trigger source
Trigger source selection control when shuttig down OC1A/OC1B waveform output

DOC1x	DSX1n=1	Trigger source	Comments
0	-	-	Settubg DOC1x to logic zero, function of trigger source shut-down waveform output is disable
1	0	Analog comparator 0	Rising edge of ACIF0 shuts down OC1x waveform output
1	1	Analog comparator 1	Rising edge of ACIF1 shuts down OC1x waveform output
1	4	External interrupt 1	Rising edge of INTF1 shuts down OC1x waveform output
1	5	Pin change level 1	Rising edge of PCIF1 shuts down OC1x waveform output
1	6	TC2 overflow	Rising edge of T0V2 shuts down OC1x waveform output
1	7	TC0 overflow	Rising edge of T0V0 shuts down OC1x waveform output

Note that:
DSX1n=1, it indicates that if the n bit of DSX1 register is one, each register can be set simutanously.

TCNT1L –TC1 Counting value register low byte

TCNT1L –TC1 Counting value register low byte								
Address: 0x84					Default Value: 0x00			
Bit	7	6	5	4	3	2	1	0
	TCNT1L7	TCNT1L6	TCNT1L5	TCNT1L4	TCNT1L3	TCNT1L2	TCNT1L1	TCNT1L0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	Name	Comments						
7:0	TCNT1	TC1 Counting value low byte TCNT1H and TCNT1L is combined to consist of TCNT1, than via TCNT1 register 16-bit counting value of counter can be access to for read-write directly. It is 2 times operations to read and write 16-bit register. If write to 16-bit TCNT1, must write TCNT1H firstly. If read 16-bit TCNT1, must read TCNT1L first. Write operation of CPU to TCNT1 register will prevent compare match from						

		happening in the next timer clock cycle, even timer stop working, this allows that value of initialized TCNT1 register is consistent with 0CR1x value, so interrupt is not triggered. If written value to TCNT1 equal to or override 0CR1x value, compare match will loss, which results in incorrect waveform. Timer stops counting if clock source is not selected, but CPU can be still access to TCNT1. CPU write counter has higher priority than clear or minus/plus operation.

TCNT1H –TC1 Counting value register high byte

TCNT1h –TC1 Counting value register high byte								
Address: 0x85					Default Value: 0x00			
Bit	7	6	5	4	3	2	1	0
	TCNT1H7	TCNT1H6	TCNT1H5	TCNT1H4	TCNT1H3	TCNT1H2	TCNT1H1	TCNT1H0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	Name	Comments						
7:0	TCNT1H	TC1 Counting value high byte TCNT1H and TCNT1L is combined to consist of TCNT1, than via TCNT1 register 16-bit counting value of counter can be access to for read-write directly. It is 2 times operations to read and write 16-bit register. If write 16-bit TCNT1, must write TCNT1H firstly. If read 16-bit TCNT1, must read TCNT1L first. Write operation of CPU to TCNT1 register will prevent compare match from happening in the next timer clock cycle, even timer stop working, this allows that value of initialized TCNT1 register is consistent with 0CR1x value, so interrupt is not triggered. If written value to TCNT1 equal to or override 0CR1x value, compare match will loss, which results in incorrect waveform. Timer stops counting if clock source is not selected, but CPU can be still access to TCNT1. CPU write counter has higher priority than clear or minus/plus operation.							

ICR1L –TC1 Input Capture Register Low byte

ICR1L –TC1 Input Capture Register Low byte								
Address: 0x86					Default Value: 0x00			
Bit	7	6	5	4	3	2	1	0
	ICR1L7	ICR1L6	ICR1L5	ICR1L4	ICR1L3	ICR1L2	ICR1L1	ICR1L0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	Name	Comments						
7:0	ICR1L	TC1 Counting value low byte ICR1H and ICR1L is combined to consist of ICR1 of 16-bit, It is 2 times operations to read and write 16-bit register. If write 16-bit ICR1, must write ICR1H firstly. If read 16-bit ICR1, must read ICR1L first. If input capture is triggered, counting value TCNT1 is copied to ICR1 register. ICR1 register is also used to define TOP of counting value.						

ICR1H –TC1 Input Capture Register High byte

ICR1H –TC1 Input Capture Register High byte								
Address: 0x87					Default Value: 0x00			
Bit	7	6	5	4	3	2	1	0
	ICR1H7	ICR1H6	ICR1H5	ICR1H4	ICR1H3	ICR1H2	ICR1H1	ICR1H0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	Name	Comments						
7:0	ICR1L	TC1 Counting value high byte ICR1H and ICR1L are combined to consist of ICR1 of 16-bit, It is 2 times operations to read and write 16-bit register. If write 16-bit ICR1, must write ICR1H firstly. If read 16-bit ICR1, must read ICR1L first. If input capture is triggered, counting value TCNT1 is copied to ICR1 register. ICR1 register is also used to define TOP of counting value.						

OCR1AL –TC1 Output Compare Register A low byte

OCR1AL –TC1 Output Compare Register A low byte								
Address: 0x88					Default Value: 0x00			
Bit	7	6	5	4	3	2	1	0
	ICR1AL7	ICR1AL6	ICR1AL5	ICR1AL4	ICR1AL3	ICR1AL2	ICR1AL1	ICR1AL0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	Name	Comments						
7:0	ICR1L	Output Compare Register A low byte ICR1AL and ICR1AH are combined to consist of ICR1A of 16-bit, It is 2 times operations to read and write 16-bit register. If write 16-bit ICR1A, must write ICR1AH firstly. If read 16-bit ICR1A, must read ICR1AL first. OCR1A compares with TCNT1 continuously. Compare match can generate output compare interrupt, or waveform on OC1A pin. In PWM mode, OCR1A register uses dual buffer register. But in normal and match clear mode, dual buffer function is disable. Updating OCR1A and moment of counter in maximum or minimum value can be syschronized by dual buffer, so as to prevent asymmetric PWM pulse and clear intervene pulse When using dual buffer, CPU is access to OCR1A buffer register. If dual buffer is disable, CPU is access to OCRIA itself.						

OCR1AH –TC1 output compare register A high byte

OCR1AH –TC1 output compare register A high byte								
Address: 0x89					Default Value: 0x00			
Bit	7	6	5	4	3	2	1	0
	OCR1AH7	OCR1AH6	OCR1AH5	OCR1AH4	OCR1AH3	OCR1AH2	OCR1AH1	OCR1AH0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	Name	Comments						
7:0	OCR1AH	Output Compare Register A high byte OCR1AL and OCR1AH are combined to consist of OCR1A of 16-bit, It is 2 times operations to read and write 16-bit register. If write 16-bit OCR1A, must write OCR1AH firstly. If read 16-bit OCR1A, must read						

		OCR1AL first. OCR1A compares with TCNT1 continuously. Compare match can generate output compare interrupt, or waveform on OC1A pin. In PWM mode, OCR1A register uses dual buffer register. But in normal and match clear mode, dual buffer function is disable. Updating OCR1A and moment of counter in maximum or minimum value are synchronized by dual buffer, so as to prevent asymmetric PWM pulse and clear intervene pulse When using dual buffer, CPU is access to OCR1A buffer register. If dual buffer is disable, CPU is access to OCRIA itself.

OCR1BL –TC1 output compare register B low byte

OCR1BL –TC1 output compare register B low byte								
Address: 0x8A					Default Value: 0x00			
Bit	7	6	5	4	3	2	1	0
	OCR1BL7	OCR1BL6	OCR1BL5	OCR1BL4	OCR1BL3	OCR1BL2	OCR1BL1	OCR1BL0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	Name	Comments						
7:0	OCR1BL	Output Compare Register B low byte OCR1BL and OCR1BH are combined to consist of OCR1B of 16-bit, It is 2 times operations to read and write 16-bit register. If write 16-bit OCR1B, must write OCR1BH firstly. If read 16-bit OCR1B, must read OCR1BL first. OCR1B compares with TCNT1 continuously. Compare match can generate output compare interrupt, or waveform on OC1B pin. In PWM mode, OCR1B register uses dual buffer register. But in normal and match clear mode, dual buffer function is disable. Updating OCR1B and moment of counter in maximum or minimum value are synchronized by dual buffer, so as to prevent asymmetric PWM pulse and clear intervene pulse When using dual buffer, CPU is access to OCR1B buffer register. If dual buffer is disable, CPU is access to OCRIB itself.							

TIMSK1 – TC1 interrupt mask register

TIMSK1 – TC1 interrupt mask register								
Address: 0x6F					Default value: 0x00			
Bit	7	6	5	4	3	2	1	0
	-	-	TICIE1	-	-	OCIE1A	OCIE1B	TOIE1
R/W	-	-	R/W	-	-	-	R/W	R/W
Initial	0	0	0	0	0	0	0	0

Bit	Name	Comment
7:6	-	reserved
5	TICIE1	TC1 input capture interrupt enable control bit If ICIE1 is logic one, and globe interrupt is set, TC1 input capture interrupt is enable. If input capture is triggered, also called that ICF1 flag of TIFR1 is set, so interrupt is generated.

		If ICIE1 is set to logic zero, TC1 input capture interrupt is disable.
4:3	-	reserve
2	OCIE1B	TC1 output compare B match interrupt enable bit If OCIE1B is set to logic one, and globe interrupt is set, TC1 output compare B match interrupt is enable. When compare match occurs, OCF1B of TIFR is set and interrupt occurs. If OCIE1B is set to logic zero, TC1 output compare B match interrupt is disable.
1	OCIE1A	TC1 output compare A match interrupt enable bit If OCIE1A is set to logic one, and globe interrupt is set, TC1 output compare A match interrupt is enable. When compare match occurs, OCF1A of TIFR is set and interrupt occurs. If OCIE1A is set to logic zero, TC1 output compare A match interrupt is disable.
0	T0IE1	TC1 overflow interrupt enable If T0IE1 is set to logic one, and globe interrupt is set, TC1 overflow interrupt is enable. If TC1 overflows, T0V1 of TIFR is set, interrupt occur. If T0IE1 is set to logic zero, TC1 overflow interrupt is disable.

TIFR1 – TC1 Interrupt Flag Register

TIFR1 – TC1 Interrupt Flag Register								
Address: 0x36					Default value: 0x00			
Bit	7	6	5	4	3	2	1	0
	-	-	ICF1	-	-	OCF1B	OCF1A	TOV1
R/W	-	-	R/W	-	-	-	R/W	R/W

Bit	Name	Comment
7:6	-	reserved
5	ICF1	Input capture flag bit If input capture event occurs, ICF1 flag is set. If ICR1 is used as TOP value of counter and counter reaches TOP, ICF1 flag is set. If ICIE1 is logic one and globe interrupt flag is set, input capture interrupt is generated. When executing this interrupt routine program, ICF1 is cleared automatically, or write logic one to ICF1 to clear this bit.
4:3	-	reserve
2	OCF1B	Output compare B match flag If TCNT1 equals to OCR1B, compare unit gives match signal and set compare bit OCF1B. At this moment if output cmpare interrupt sets OCIE1B to logic one, and globe interrupt is set, than output compare interrupt occurs. When executing this interrupt routine program, OCF1B is cleared automatically, or write logic one to OCF1B to clear this bit.
1	OCF1A	Output compare A match flag If TCNT1 equals to OCR1A, compare unit gives match signal and set compare bit OCF1A. At this moment if output cmpare interrupt sets OCIE1A to logic one, and globe interrupt is set, than output compare interrupt occurs. When executing this interrupt routine program, OCF1A is cleared automatically, or write logic one to OCF1A to clear this bit.
0	TOV1	overflow interrupt flag

		If counter overflows, overflow flag T0V1 is set. If at this moment overflow interrupt enable T0IE1 to logic one andglobe interrupt flag is set, overflow interrupt occurs. When executing this interrupt routine program, T0V1 is cleared automatically, or write logic one to T0V1 to clear this bit.

DTR1L –TC1 Dead time register low byte

DTR1L –TC1 Dead time register low byte								
Address: 0x8C					Default value: 0x00			
Bit	7	6	5	4	3	2	1	0
	DTR1L							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Comment
7:0	DTR1L	Dead time register high byte If DTEN1 is high, OC1A and OC1B are complimentary output, dead time inserted on OC1A output is decided by counter clock of DTR1L

DTR1H –TC1 Dead time register high byte

DTR1H –TC1 Dead time register high byte								
Address: 0x8D					Default value: 0x00			
Bit	7	6	5	4	3	2	1	0
	DTR1H							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Comment
7:0	DTR1H	Dead time register high byte If DTEN1 is high, OC1A and OC1B are complimentary output, dead time inserted on OC1B output is decided by counter clock of DTR1L

TCKCSR –TC Clock control status Register

TCKCSR –TC Clock control status Register								
Address: 0xEC					Default value: 0x00			
Bit	7	6	5	4	3	2	1	0
	-	F2XEN	TC2XF1	TC2XF0	-	AFCKS	TC2XS1	TC2XS0
R/W	-	R/W	R	R	-	R/W	R/W	R/W

Bit	Name	Comment
7	-	reserved
6	F2XEN	RC 32M frequency doubling output enable control bit If F2XEN is set to logic one, 32M RC frequency doubling output is enable, output 64M high speed clock If F2XEN is set to logic one, 32M RC frequency doubling output is disable, cannot output 64M high speed
5	TC2XF1	TC High Speed Clock mode flag 1 If TC2XF1 bit is read as logic one, it is indicated that timer counter 1 works in high speed clock mode.

		If it is logic zero, timer counter 1 works in system clock mode
4	TC2XF0	TC high speed clock mode flag bit 0, referring to section on timer/counter 0 register
3:2	-	reserved
1	TC2XS1	TC high speed clock mode selection control bit 1 If TC2XS1 bit is read as logic one, it is indicated that timer counter 1 works in high speed clock mode. If it is logic zero, timer counter 1 works in system clock mode
0	TC2XS0	TC high speed clock mode selection control bit 0, referring to section on timer/counter 0 register

TMR0/1/3 预分频器

3 个 10 位预分频器

复用模式下 TC0、TC1 和 TC3 复用预分频器 CPS310

独立模式下 TC0 独用预分频器 CPS310， TC1 独用预分频器 CPS1， TC3 独用预分频器 CPS3

支持软件复位

TMR0/1/3 Prescaler

- 3 pcs of 10-bit prescaler
- In multiplex mode, TC0,TC1 andTC3 multiplex prescaler CPS310
- In indepandant mode, TC0 inclusive prescaler CPS310, TC1 inclusive prescaler CPS1 TC3 inclusive prescaler CPS3
- Support software reset

概述

复用模式下（PSS1=0 且 PSS3=0）， TC0、 TC1 和 TC3 共用一个 10 位的预分频器 CPS310，但它们有不同的分频设置。

单用模式下（PSS1=1 且 PSS3=0）， TC1 独立使用一个 10 位的预分频器 CPS1， TC0 和 TC3 共用一个 10 位的预分频器 CPS310，但它们有不同的分频设置。

单用模式下（PSS1=0 且 PSS3=1）， TC3 独立使用一个 10 位的预分频器 CPS3， TC0 和 TC1 共用一个 10 位的预分频器 CPS310，但它们有不同的分频设置。

独立模式下（PSS1=1 且 PSS3=1），TC0 独立使用一个 10 位的预分频器 CPS310，TC1 独立使用一个 10 位的预分频器 CPS1，TC3 独立使用预分频器 CPS3。

Overview

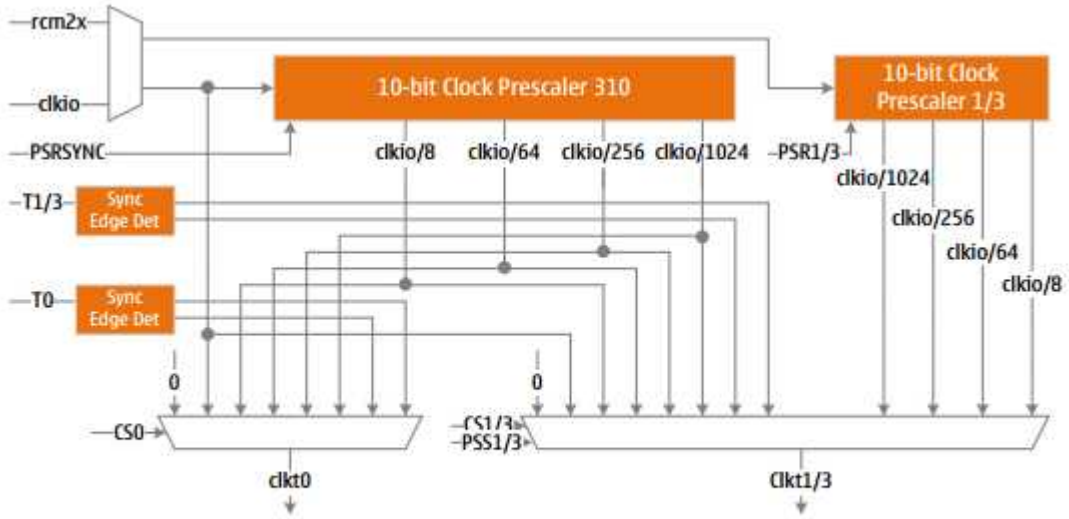
In multiplex mode (PSS1=0 and PSS3=0), TC0, TC1 and TC3 share one 10-bit prescaler CPS310, however they have different prescaler configuration.

In single mode (PSS1=0 and PSS3=0), TC1 inclusively uses a 10-bit prescaler CPS1, TC0 and TC3 share to use a 10-bit prescaler CPS310 but with their individual prescaler configuration.

In single mode (PSS1=0 and PSS3=0), TC3 inclusively uses ne 10-bit prescaler CPS3, TC0 and TC1 share to use a 10-bit prescaler CPS310 but with their individual prescaler configuration.

In independent mode (PSS1=1 and PSS3=1), TC0 inclusively uses a 10-bit prescaler CPS310, TC1 inclusively uses a 10-bit prescaler CPS1, TC3 inclusively uses CPS3.

Below description is used in TC0, TC1 and TC3, “n” indicates 0, 1 or 3.



TC0/TC1 /TC3 Prescaler 结构图

内部时钟源

当设置 CSn[2:0]=1 时， 定时器 3 只可由系统时钟 clkio 驱动， 定时计数器 0 或 1 可直接由系统时钟 clkio 或高速时钟 rcm2x （内部 32M RC 振荡器输出时钟的 2 倍频）驱动。预分频器可以输出 4 个不同的时钟频率，分别是 clkio/8，clkio/64，clkio/256 和 clkio/1024。

Internal Clock Source

If CS CSn[2:0]=1 is set, Timer 3 is only drived by system clock clkio, while Timer 0 and 1 can be directly drived by system clock clkio or high speed clock rcm2x (internal 32M RC oscillator outputs 2 times frequency of clock). Prescaler gives 4 different clock frequency: clkio/8，clkio/64，clkio/256 and clkio/1024

分频器复位

复用模式

当设置 PSS1 位为“0”且 PSS3 位为“0”时，TC0、TC1 和 TC3 共用一个预分频器 CPS310。

预分频器是独立运行的，其操作独立于 TC 的时钟选择逻辑，且它由 TC0、TC1 和 TC3 共享。由于不受时钟选择控制的影响，预分频器的状态对分频时钟的应用会有影响。当定时器使能并且选用预分频器的输出作为计数时钟源（6>CSn[2:0]>1）时，影响就会产生。从定时器使能到第一次计数可能要花费 1 到 N+1 个系统时钟，其中 N 为预分频因子（8，64，256 或 1024）。

通过复位预分频器来同步定时器和程序运行是可能的。但是必须注意，另一个定时器是否正在使用这个预分频器，复位预分频器会影响到所有与其连接的定时器。

Prescaler Reset

Multiplex mode

If setting PSS1 to logic zero meanwhile PSS3 to zero, TC0. TC1 and TC3 shares to use a prescaler CPS310. The prescaler is free running, i.e., operates independently of the Clock Select logic ofthe Timer/Counter, and it is shared by TC0, TC1 and TC3. Since the prescaler is not affected by the Timer/Counter’s clock select, the state of the prescaler will have implications for situations where a prescaled clock is used. Prescaling affect

occurs when the timer is enabled and prescaler is selected as counting clock source (6 > CSn2:0 > 1). The number of system clock cycles from when the timer isenabled to the first count occurs can be from 1 to N+1 system clock cycles, where N equals the prescaler divisor (8, 64, 256, or 1024). It is possible to use the prescaler reset for synchronizing the Timer/Counter to program execution. However, care must be taken if the other Timer/Counter that shares the same prescaler also uses prescaling. A prescaler reset will affect all Timer/Counters it is connected to.

单用模式

当设置 PSS1 位为“1”时，TC1 独立使用预分频器 CPS1，预分频器的复位由 PSR1 位来控制。各自的复位单独起作用，不会影响其它预分频器。

当设置 PSS3 位为“1”时，TC3 独立使用预分频器 CPS3，预分频器的复位由 PSR3 位来控制。各自的复位单独起作用，不会影响其它预分频器。

当设置 PSS1 位为“1”且 PSS3 位为“1”时，TC0 独立使用预分频器 CPS310，预分频器的复位由 PSRSYNC 位来控制， TC1 独立使用预分频器 CPS1， TC3 独立使用预分频器 CPS3，各自的复位单独起作用，不会影响其它预分频器。

In Single Mode

If setting PSS1 to logic one, TC1 uses prescaler CPS1 indepandantly, prescaler reset is decided by PSR1. Each reset works individually without affect other prescalers.

If setting PSS3 to logic one, TC3 uses prescaler CPS3 indepandantly, prescaler reset is decided by PSR3. Each reset works individually without affect other prescalers.

If setting PSS1 to logic one and PSS3 to logice one, TC0 uses prescaler CPS310 indepandantly, prescaler reset is decided by PSRSYNC, TC1 uses prescaler CPS31 indepandantly while TC3 uses prescaler CPS3 indepandantly. Each reset works individually without affect other prescalers.

外部时钟源

由 T0/T1/T3 引脚提供的外部时钟源可以用作计数时钟源。T0/T1/T3 引脚的信号经过同步逻辑和边沿检测器之后作为计数器的时钟源。每个上升沿（CSn[2:0]=7 ）或下降沿（CSn[2:0]=6）都会产生一个计数脉冲。外部时钟源不会送入预分频器。

由于引脚上同步与边沿检测电路的存在， T0/T1/T3 上电平的变化需要延迟 2.5 到 3.5 个系统时钟才能使计数器更新。

禁止或使能时钟输入必须在 T0/T1/T3 保持稳定至少需要一个系统时钟周期后才能进行，否则有产生错误计数时钟脉冲的可能。

为了保证正确的采样，外部时钟脉冲宽度必须大于一个系统时钟周期，在占空比为 50%时外部时钟频率必须小于系统时钟频率的一半。由于振荡器本身的误差带来的系统时钟频率及占空比的差异，建议外部时钟的最高频率不要大于 fsys/2.5。

An external clock source applied to the T0/T1/T3 pin can be used as Timer/Counter clock. The T0/T1/T3 pin signal, after went throught pin synchronization logic and the edge detector,will be used as clock source of counter. The edge detector generates one counting pulse on each rising edge (CSn[2:0] = 7) or falling edge (CSn[2:0] = 6). External clock source will not send to prescaler.

The synchronization and edge detector logic introduces a delay of 2.5 to 3.5 system clock cycles from an edge has been applied to the T1/T0 pin to the counter is updated

Enabling and disabling of the clock input must be done when T0/T1/T3 has been stable for at least one system

clock cycle, otherwise it is a risk that a false Timer/Counter clock pulse is generated.

Each half period of the external clock applied must be longer than one system clock cycle to ensure correct sampling. The external clock must be ensured to have less than half the system clock frequency given a 50/50% duty cycle. However, due to variation of the system clock frequency and duty cycle caused by Oscillator tolerances, it is recommended that maximum frequency of an external clock source is less than fclk_I/O/2.5.

Register Definition

GTCCR- General Timer Counter Register

GTCCR- General Timer Counter Register								
Address: 0x43					Default value: 0x00			
Bit	7	6	5	4	3	2	1	0
	TSM	-	-	-	-	-	PSRASY	PSRSYNC
R/W	R/W	-	-	-	-	-	W	W
Bit	Name	Comment						
7	TSM	Timer/Counter Synchronization Mode Writing the TSM bit to one activates the Timer/Counter Synchronization mode. In thismode, the value that is written to the PSRASY and PSRSYNC bits is kept, hence keeping the corresponding prescaler reset signals asserted. This ensures that the corresponding Timer/Counters are halted and can be configured to the same value. When the TSM bit is written to zero, the PSRASY and PSRSYNC bits are cleared by hardware, and the Timer/Counters start counting simultaneously.						
6:2	-	reserved						
1	PSRASY	Referring to section of TC2 register						
0	PSRSYNC	Prescaler CPS310 reset control bit When this bit is one, prescaler CPS310 will be Reset. This bit is normally cleared after reset immediately by hardware, except if the TSM bit is set. When this bit is zero, it is useless configuration. Note that in multiplex mode, TC0/TC1/TC3 share the same prescaler and a reset of this prescaler will affect all three timers In independant mode, reset will only affect TC0. It will always be zero if read value of this bit.						

PSSR- Prescaler Selection Register

PSSR- Prescaler Selection Register								
Address: 0XE2					Default value: 0x00			
Bit	7	6	5	4	3	2	1	0
	PSS1	PSS3	-	-	-	-	PSR3	PSR1
R/W	R/W	R/W	-	-	-	-	R/W	R/W
Bit	Name	Comment						
7	PSS1	Prescaler Selection Control bit Write PSS1 to logic one, TC1 uses prescaler CPS1 independantly.						

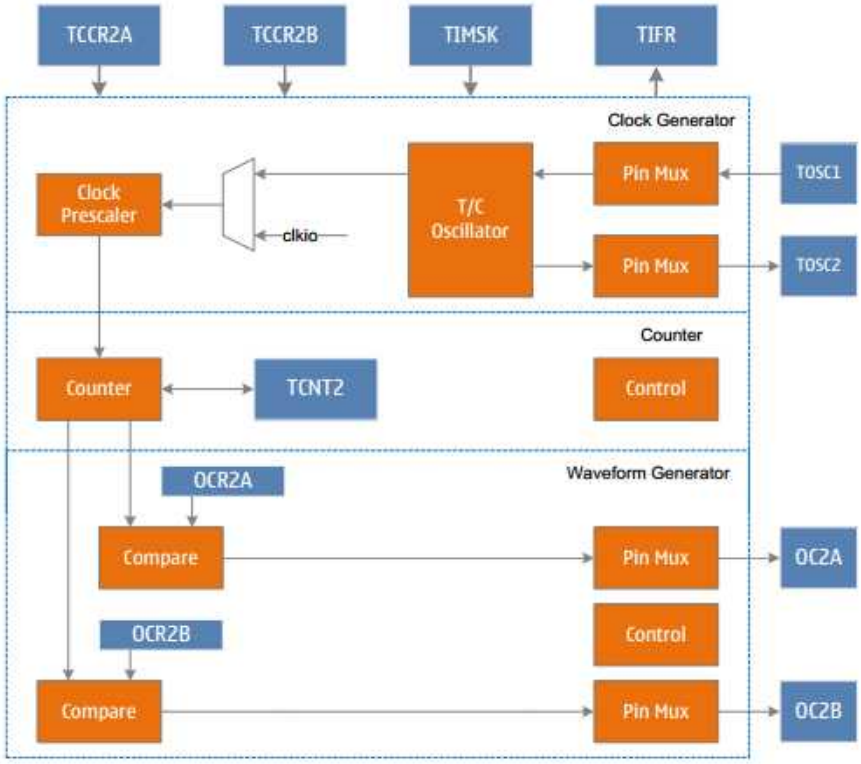
		Writing PSS1 to logic zero activates prescaler multiplex mode. TC0 andTC1 share prescaler CPS310. Prescaler CPS1 is useless and is reset all the time. If at the same time PSS3 is logic zero, TC3 and TC0, TC1 share prescaler CPS310. Prescaler CPS1 andCPS3 is usesless and is reset all the time.
6	PSS3	Prescaler Selection Control bit write PSS1 to logic one, TC3 uses prescaler CPS1 indepandantly. Writing PSS3 to logic zero activates prescaler multiplex mode. TC0 andTC3 share prescaler CPS310. Prescaler CPS3 is useless and is reset all the time. If at the same time PSS1 is logic zero, TC1 and TC0, TC3 share prescaler CPS310. Prescaler CPS1 andCPS3 is usesless and is reset all the time.
5:2	-	reserved
1	PSR3	Prescaler CPS3 Reset Control bit PCSR3 is only useful in TC3 single mode. Writing PSR3 to logic one will reset prescaler CPS3. Once reset hardware clears PSR3 bit. When setting PSR3 to logic zero, configuration is useless. It will be always zero when read this bit.
0	PSR1	Prescaler CPS1 Reset Control bit PCSR1 is only useful in TC1 single mode. Writing PSR1 to logic one will reset prescaler CPS1. Once reset hardware clears PSR1 bit. When setting PSR1 to logic zero, configuration is useless. It will be always zero when read this bit.

定时/计数器 2 (TMR2)

- 8 位计数器
- 两个独立的比较单元
- 比较匹配发生时自动清零计数器并自动加载
- 无干扰脉冲的相位修正的 PWM 输出
- 频率发生器
- 外部事件计数器
- 10 位的时钟预分频器
- 溢出和比较匹配中断
- 允许使用外部 32.768KHz 的 RTC 晶振计数

Timer/ Counter 2 (TMR2)

- 8-bit counter
- Two independent compare unit
- In compare match, clear counter and load automatically
- Non-intervene pulse correction PWM
- Frequency generator
- External event counter
- 10-bit clock prescaler
- Overflow and compare match interrupt
- External 32.768KHz of RTC oscillator counting is allowed.



TC2 结构图

TC2 是一个通用 8 位定时计数器模块，支持 PWM 输出，可以精确地产生波形。TC2 包含 1 个 8 位计数器，波形产生模式控制单元和 2 个输出比较单元。波形产生模式控制单元控制着计数器的工作模式和比较输出波形的产生。根据不同的工作模式，计数器对每一个计数时钟 Clkt2 实现清零、加一或减一操作。Clkt2 可以由内部时钟源或外部时钟源产生。当使用外部 32.768KHz 的晶振计数时，TC2 可用作 RTC 计数器。

当计数器的计数值 TCNT2 到达最大值（等于极大值 0xFF 或输出比较寄存器 OCR2A，定义为 TOP，定义极大值为 MAX 以示区别）时，计数器会进行清零或减一操作。当计数器的计数值 TCNT2 到达最小值（等于 0x00，定义为 BOTTOM）时，计数器会进行加一操作。当计数器的计数值 TCNT2 到达 OCR2A/OCR2B 时，也被称为发生比较匹配时，会清零或置位输出比较信号 OC2A/OCR2B，来产生 PWM 波形。

TC2 is a general purpose 8-bit Timer/Counter module, with PWM support. It allows accurate wave generation.

TC2 consists of 1 8-bit counter, waveform generation control unit and 2 output compare unit. TC0 and TC1 share 10-bit prescaler, In each different working mode, counter does operation of clear, plus one or minus one to each counting clock Clkt2, which is generated by internal or external clock source. When counting with external 32.768KHz of crystal oscillator, TC2 is used as RTC counter.

If counting number TCNT2 of counter is maximum (which is max. value OxFF, or define output compare register OCR2A as TOP, define max. value as MAX for better identify), counter will clear or make minus 1 operation.

If counting number TCNT2 of counter is minimum (which is Ox00, or define as BOTTOM), counter will operate PLUS 1.

If counting number TCNT2 is OCR2A/OCR2B, it signifies that compare match occurs, counter will clear or is set to output compare signal 0C2A/0C2B to generate PMW waveform.

工作模式

定时计数器 2 有四种不同的工作模式，包括普通模式（Normal），比较匹配时清零（CTC）模式，快速脉冲宽度调制（FPWM）模式和相位修正脉冲宽度调制（PCPWM）模式，由波形产生模式控制位 WGM2[2:0]来选择。下面具体来描述这四种模式。由于有两个独立的输出比较单元，分别用“A”

和“B”来表示，用小写的“x”来表示这两个输出比较单元通道。

Working Mode

Timer counter 2 has 4 different working modes, including normal mode (Normal), Compare Match on clear mode, (CTC), fast pulse width modulation mode (FPWM) and phase correction pulse width modulation mode (PCPWM), which are selected by waveform generation mode control bit WGM2[2:0]. In the below these 4 modes is written in details. Since there are 2 indepandent output compare unit, they are indicated separately with “A” and “B”. Small “x” is used to indicate channel of these two output compare unit.

普通模式

普通模式是定时计数器最简单的工作模式，此时波形产生模式控制位 WGM2[2:0]=0，计数的最大值 TOP 为 MAX（0xFF）。在这种模式下，计数方式为每一个计数时钟加一递增，当计数器到达 TOP 溢出后就回到 BOTTOM 重新开始累加。在计数值 TCNT2 变成零的同一个计数时钟里置位定时计数器溢出标志 TOV2。这种模式下 TOV2 标志就像是第 9 计数位，只是只会被置位不会被清零。

溢出中断服务程序会自动清除 TOV2 标志，软件可以用它来提高定时计数器的分辨率。普通模式下没有特殊情形需要考虑，可以随时写入新的计数值。设置 OC2x 引脚的数据方向寄存器为输出时才能得到输出比较信号 OC2x 的波形。当 COM2x=1 时，发生比较匹配时会翻转 OC2x 信号，这种情况下波形的频率可以用下面的公式来计算：

foc2xnormal = fsys/(2*N*256)

其中，N 表示的是预分频因子（1，8，64，256 或者 1024）。

输出比较单元可以用来产生中断，但是在普通模式下不推荐使用中断，这样会占用太多 CPU 的时间。

Normal Mode

The simplest mode of operation is the Normal mode (WGM02:0 = 0), its maximum TOP is MAX（0xFF). In this mode, counting increase by each counting clock, when it passes TOP and overflow, it will then restart to accumulate from the bottom (0x00). The Timer/Counter Overflow Flag (TOV2) will be set in the same timer clock cycle as the TCNT2 becomes zero. The TOV2 Flag in this case behaves like a ninth bit, except that it is only set, not cleared.

However, combined with the timer overflow interrupt that automatically clears the TOV2 Flag, the timer resolution can be increased by software. There are no special cases to consider in the Normal mode, a new counter value can be written anytime.

Waveform of output compare signal OC2x is generated only if data direction register of OC2x is set. If COM2x=1, OC2x singal will be toggled when compare match is happening, in this case waveform frequency can be calculated by below formula:

Foc2xnormal = fsys/(2*N*256),

N indicates prescaler factor (1，8，64，256 or 1024).

Using the Output Compare unit to generate waveforms in Normal mode is not recommended, since this will occupy too much of the CPU time.

CTC 模式

设置 WGM2[2:0]=2 时，定时计数器 2 进入 CTC 模式，计数的最大值 TOP 为 OCR2A。在这个模式下，计数方式为每一个计数时钟加一递增，当计数器的数值 TCNT2 等于 TOP 时计数器清零。 OCR2A 定义了计数的最大值，亦即计数器的分辨率。这个模式使得用户可以很容易的控制比较匹配输出的频率，也简化了外部事件计数的操作。

当计数器到达计数的最大值时，输出比较匹配标志 OCF2 被置位，相应的中断使能置位时将会产生中断。在中断服务程序里可以更新 OCR2A 寄存器即计数的最大值。在这个模式下 OCR2A 没有使用双缓冲，在计数器以无预分频器或很低的预分频器工作下将最大值更新为接近最小值的时候要小心。如果写入 OCR2A 的数值小于当时的 TCNT2 值时，计数器将丢失一次比较匹配。在下一次比较匹配发生之前，计数器不得不先计数到 TOP，然后再

从 BOTTOM 开始计数到 OCR2A 值。和普通模式一样，计数值回到 BOTTOM 的计数时钟里置位 TOV2 标志。设置 OC2x 引脚的数据方向寄存器为输出时才能得到输出比较信号 OC2x 的波形。当 COM2x=1 时，发生比较匹配时会翻转 OC2x 信号，这种情况下波形的频率可以用下面的公式来计算：

foc2xctc = fsys/(2*N*(1+OCR2A))

其中， N 表示的是预分频因子（1， 8， 64， 256 或者 1024）。

从公式可以看出，当设置 OCR2x 为 0x0 且无预分频器时，可以获得最大频率为 fsys/2 的输出波形。

CTC Mode

Set WGM0[2:0]=2, timer/ counter 2 is in CTC mode, max. Counting number TOP is OCR2A. In this mode, counting increase by each counting clock, counter is cleared when TCNT2 is TOP. The OCR2A defines the maximum value for the counter, hence also its resolution. This mode allows greater control of the compare match output frequency. It also simplifies the operation of counting external events. When counter value reaches TOP value, output compare match flag OCF2 is set, corresponding interrupt enable will generate interrupt. If the interrupt is enabled, the interrupt handler routine can be used for updating the TOP value of OCROA. However, changing TOP to a value close to BOTTOM when the counter is running with none or a low prescaler value must be done with care since the CTC mode does not have the double buffering feature. If the new value written to OCR2A is lower than the current value of TCNT2, the counter will miss the compare match. Before next compare match, the counter will then have to count to TOP and re-start from BOTTOM to count to OCROA value. As for the Normal mode of operation, the TOV2 Flag is set when counting value return to counting clock of BOTTOM. Waveform of output compare signal OC2x is generated only if data direction register of OC2x pin is set output.

When COM2x=1, compare match occures and OC2Cx signal is to be toggled. In this case waveform frequency can be calculated by below formula:

Foc2xctc = fsys/(2*N*(1+OCR2A))

Here, N indicates prescaler factor (1，8，64，256 or1024)

Seeing from above formular, set OCR2x to 0x0 meanwhile without prescaler, output waveform of max. fsys/2 can be gained.

快速 PWM 模式

设置 WGM2[2:0]=3 或 7 时，定时计数器 2 进入快速 PWM 模式，可以用来产生高频的 PWM 波形，计数最大值 TOP 分别为 MAX（0xFF）或 OCR2A。快速 PWM 模式和其他 PWM 模式不同在于它是单向操作。计数器从最小值 0x00 累加到 TOP 后又回到 BOTTOM 重新计数。当计数值 TCNT2 到达 OCR2x 或 BOTTOM 时，输出比较信号 OC2x 会被置位或清零，取决于比较输出模式 COM2x 的设置，详情见寄存器描述。由于采用单向操作，快速 PWM 模式的操作频率是采用双向操作的相位修正 PWM 模式的两倍。高频特性使得快速 PWM 模式适用于功率调节，整流以及 DAC 应用。高频信号可以减小外部元器件（电感电容等）的尺寸，从而降低系统成本。

当计数值到达最大值时，定时计数器溢出标志 TOV2 将会被置位，并把比较缓冲器的值更新到比较值。如果中断使能，在中断服务程序中可以更新比较缓冲器 OCR2x 寄存器。设置 OC2x 引脚的数据方向寄存器为输出时才能得到输出比较信号 OC2x 的波形。波形的频率可用下面的公式来计算：

foc2xfpwm = fsys/(N*(1+TOP))

其中，N 表示的是预分频因子（1，8，64，256 或者 1024）。

当 TCNT2 和 OCR2x 发生比较匹配时，波形产生器就置位（清零）OC2x 信号，当 TCNT2 被清零时，波形产生器就清零（置位）OC2x 信号，以此来产生 PWM 波。由此 OCR2x 的极值将会产生特殊的 PWM 波形。当 OCR2x 设置为 0x00 时，输出的 PWM 为每(1+TOP)个计数时钟里有一个窄的尖峰脉冲。当 OCR2x 设置为最大值时，输出的波形为持续的高电平或低电平。

Fast PWM Mode

Fast PWM mode of Timer/Counter 2 (WGM2[2:0]= 3 or 7) provides a high frequency PWM waveform generation option. TOP is defined as MAX(0xFF) or OCR2A. The fast PWM differs from the other PWM option by its single-slope operation.

The counter counts from min. Value 0x00 to TOP then restarts from BOTTOM. When counting value TCNT2 reaches OCR2x or BOTTOM, output compare OC2x is set or cleared, which depends on configuration of compare output COM2x, details referring to sections about register. Due to the single-slope operation, the operating frequency of the fast PWM mode can be twice as high as the phase correct PWM mode that uses dual-slope operation. This high frequency makes the fast PWM mode well suited for power regulation, rectification, and DAC applications. High frequency allows physically small sized external components (capacitors), and therefore reduces total system cost.

When counting value reaches maximum, timer counter overflow flag TOV2 will be set and value of compare buffer is updated to compare value. If interrupt disable, compare buffer OCR2x register can be updated in interrupt handler routine. Waveform of output compare signal OC2x is generated only if data direction register of OC2x pin is set output. Waveform frequency can be calculated by below formular:

Foc2xfpwm = fsys/(N*(1+TOP))

Here, N indicates prescaler factor (1，8，64，256 and 1024)

When compare match occurs between TCNT2 and OCR2x, waveform generator is set to OC0x (cleared). When TCT0 is cleared, waveform generator is (cleared) set to OC0x to generate PWM waveform, so extreme value of OCR0x will generate very special PWM waveform. If the OCR0A is set equal to 0x00 the output PWM will be a narrow spike for each

TOP+1 timer clock cycle. Setting the OCR0A equal to MAX will result in a constantly high or low output

相位修正 PWM 模式

当设置 WGM2[2:0]=1 或 5 时，定时计数器 2 进入相位修正 PWM 模式，计数的最大值 TOP 分别为 MAX（0xFF）或 OCR2A。计数器采用双向操作，由 BOTTOM 递增到 TOP，然后又 递减到 BOTTOM，再重复此操作。计数到达 TOP 和 BOTTOM 时均改变计数方向，计数值在 TOP 或 BOTTOM 上均只停留一个计数时钟。在递增或递减过程中，计数值 TCNT2 与 OCR2x 匹配 时，输出比较信号 OC2x 将会被清零或置位，取决于比较输出模式 COM2x 的设置。与单向操 作相比，双向操作可获得的最大频率要小，但其极好的对称性更适合于电机控制。 相位修正 PWM 模式下，当计数到达 BOTTOM 时置位 TOV2 标志，当计数到达 TOP 时把比较 缓冲器的值更新到比较值。如果中断使能，在中断服务程序中可以更新比较缓冲器 OCR2x 寄 存器。

设置 OC2x 引脚的数据方向寄存器为输出时才能得到输出比较信号 OC2x 的波形。波形的频 率可用下面的公式来计算：

foc2xpcpwm = fsys/(N*TOP*2)

其中，N 表示的是预分频因子（1，8，64，256 或者 1024）。

在递增计数过程中，当 TCNT2 与 OCR2x 匹配时，波形产生器就清零（置位）OC2x 信号。在 递减计数过程中，当 TCNT2 与 OCR2x 匹配时，波形产生器就置位（清零）OC2x 信号。由此 OCR2x 的极值会产生特殊的 PWM 波。当 OCR2x 设置为最大值或最小值时，OC2x 信号输出会 一直保持低电平或高电平。

为了保证输出 PWM 波在最小值两侧的对称性，在没有发生比较匹配时，有两种情况下也会 翻转 OC2x 信号。第一种情况是，当 OCR2x 的值由最大值 0xFF 改变为其他数据时。当 OCR2x 为最大值，计数值达到最大时，OC2x 的输出与前面降序计数时比较匹配的结果相同，即保 持 OC2x 不变。此时会更新比较值为新的 OCR2x 的值（非 0xFF）， OC2x 的值会一直保持，直到升序计数时发生比较匹配而翻转。此时 OC2x 信号并不以最小值为中心对称，因此需要在 TCNT2 到达最大值时翻转 OC2x 信号，此即没有发生比较匹配时翻转 OC2x 信号的第一种情 况。第二种情况是，当 TCNT2 从比 OCR2x 高的值开始计数时，因而会丢失一次比较匹配， 从而引起不对称情形的产生。同样需要翻转 OC2x 信号去实现最小值两侧的对称性。

Phase Correct PWM Mode

Setting WGM2[2:0] to 1 or 5, Timer/Counter 2 enter into Phase Correct PWM mode, each TOP value is MAX(0xFF) or OCR2A. The phase correct PWM mode is based on a dual-slope operation. The counter counts repeatedly TOP (increment) and then to BOTTOM (decrement).

When counter reaches TOP or BOTTOM, it will both change counting direction. TOP or BOTTOM value only stay for one counting clock.
On the compare match between TCNT1 and OCR1x while up-counting and down-counting, the Output Compare (OC2x) is cleared or set depending on the configuration of compare output mode COM2x. The dual-slope operation has lower maximum operation frequency than single slope operation. However, due to the symmetric feature of the dual-slope PWM modes, these modes are preferred for motor control applications.
In phase correct PWM mode, the TOV2 is set when the counter reaches BOTTOM. When the counter reaches TOP, value of compare buffer is updated to compare value. If interrupt enable, compare buffer OCR2x register can be updated in interrupt routine program.
Setting OC2x pin data direction register to output, it will generate waveform of output compare O21x.
Waveform frequency can be calculated via below formula:

foc1xcpcpwm = fsys/(N*TOP*2)

Here, N indicates prescaler factor (1，8，64，256 or 1024)
During up-counting, when match occur between TCNT2 and OCR2x, waveform generator clear (set) OC2x .
During down-counting, when match occur between TCNT2 and OCR2x, waveform generator set (clear) OC2x, so extreme value of OCR2x will generate very special PWM waveform. Setting the OCR2x equal to TOP or BOTTOM will result in a constantly high or low level change.

OC2x will be toggled in two cases when compare match does not occur while to ensure symmetry of both side of PMW waveform at its bottom value.
The first case : OCR2x value is changed from TOP to others. When the OCR2x value is TOP, counting value reach TOP, OC2x output is the same as the result of previous down-counting Compare Match, that is to say OC2x keep unchanged.

In this case Compare value is updated to a new OCR2x value (not 0xFF). OC2x keeps unchanged till it is toggled in up-counting compare match. At this moment OC2x is not symmetric with minimum value, so OC2x need to be toggled when TCNT2 reach maximum value.
The 2nd case: when TCNT2 starts to count from value higher than OCR2x, one compare match will loss, which will result in asymmetric. Same is that to achieve symmetric at both side of bottom value, OC2x is needed to be toggled.

TC2 的异步操作方式

当位于 ASSR 寄存器的 AS2 位为“1”时，TC2 工作在异步模式，计数器的时钟源来自于外 部定时计数器的振荡器。异步模式下 TC2 的操作要考虑如下几点

？ 在同步和异步模式之间的转换有可能造成 TCNT2、OCR2A、OCR2B、TCCR2A 和 TCCR2B 数 据的损坏。安全的操作步骤如下所示：

1. 清零 OCIE2A，TOIE2 和 OCIE2B 寄存器位来关闭 TC2 的中断；
2. 置位 AS2 位选择合适的时钟源；
3. 对 TCNT2、OCR2A、TCCR2A、OCR2B 和 TCCR2B 寄存器写入新的数据；
4. 切换到异步模式时，需等待 TCN2UB、OCR2AUB、TCR2AUB、OCR2BUB 和 TCR2BUB 位 清零；
5. 清零 TC2 的中断标志位；
6. 使能需要使用的中断。 ？

振荡器最好使用 32.768KHz 的手表晶振。系统时钟频率必须比晶振频率高 4 倍以上。 ？

CPU 写 TCNT2、OCR2A、TCCR2A、OCR2B 和 TCCR2B 时，硬件会将数据先放入暂存器，两 个 TOSC1 时钟上升沿后才锁存到对应的寄存器中。在数据从暂存器锁存到目的寄存器之前不能执行新的数据写入操作。各个寄存器都有各自独立的暂存器，因此写 TCNT2 并 不会干扰写 OCR2。异步状态

寄存器 ASSR 用来检查数据是否已经写入到目的寄存器。 ？

如果使用 TC2 作为 MCU 休眠模式的唤醒条件，则在各个寄存器更新结束之前不能进入 休眠模式，否则 MCU 可能会在 TC2 设置生效之前进入休眠模式，从而 TC2 无法唤醒系 统。 ？

如果使用 TC2 作为 MCU 休眠模式的唤醒条件，必须注意重新进入休眠模式的过程。中 断逻辑需要一个 TOSC1 时钟周期进行复位，如果从唤醒到重新进入休眠的时间小于一 个 TOSC1 时钟周期，中断将不再发生，器件也无法唤醒。推荐采用如下的操作方法：

1. 对各个寄存器写入合适的数据；

2. 等待 ASSR 相应的更新忙标志位清零；

3. 进入休眠模式。 ？

选择了异步工作模式，TC2 的振荡器将一直工作，除非进入掉电模式。用户必须注意， 此振荡器的稳定时间可能长达 1 秒钟，因此，建议用户在使能 TC2 的振荡器后至少等待 1 秒钟后再使用 TC2 的异步工作模式。 ？

异步工作模式时休眠模式下唤醒的过程：中断条件满足后，在下一个定时器时钟启动唤 醒过程。也就是说，在处理器可以读取计数器的数值之前计数器至少又累加了一个时钟。 唤醒后 MCU 执行中断服务程序，之后开始执行 SLEEP 语句之后的程序。 ？

从休眠模式唤醒之后短时间内读取 TCNT2 的值可能返回不正确的数据。因为 TCNT2 是 由异步的 TOSC1 时钟驱动的，而读取 TCNT2 必须通过一个内部系统时钟同步的寄存器 来完成，同步发生于每个 TOSC1 的上升沿。从休眠模式唤醒后系统时钟重新激活，读取 的 TCNT2 数值为进入休眠模式之前的值，直到下一个 TOSC1 上升沿的到来才会更新。从休眠模式唤醒时 TOSC1 的相位完全不可预测，而与唤醒时间有关。因此，读取 TCNT2 值的推荐序列为：

1. 写一个任意数值到 OCR2A 或 TCCR2A；

2. 等待相应的更新忙标志位被清零；

3. 读取 TCNT2。 ？

异步模式下，中断标志位的同步需要 3 个系统时钟周期加 1 个定时器周期。在 MCU 可 以读取引起中断标志置位的计数器数值之前计数器至少又累加了一个时钟。输出比较信 号的变化与定时器时钟同步，而不是系统时钟。

Asynchronous Operation of TC2

When AS2 located in ASSR register is set to logic one, TC2 works in asynchronous mode, at this moment counter clock sourcce is from oscillator of external timer counter. When Timer/Counter2 operates asynchronously, some considerations must be taken.

When switching between asynchronous and synchronous clocking of Timer/Counter2, data of TCNT2, OCR2A, OCRAB, TCCR2A and TCCR2B might be corrupted. A safe procedure for switching clock source is:

1. Disable the TC2 interrupts by clearing OCIE2A, TOIE2 and OCIE2B register bit.

2. Select clock source by setting AS2 as appropriate

3. Write new values to TCNT2, OCR2A, TCCR2A,OCR2B and TCCR2B.

4. To switch to asynchronous operation: Wait for TCN2UB, OCR2AUB, TCR2AUB, OCR2BUB and TCR2BUB to clear.

5. Clear TC2 interrupt flag.

6. Enable interrupt that is in need of.

Oscillator had better use clock oscillator of 32.768KHz. The system clock frequency must be more than four times the Oscillator frequency.

When CPU is writing to TCNT2, OCR2A, TCCRA2, OCR2BA and TCCR2B, the value is transferred to a temporary register by hardware, and latched into corresponding register after two rising edges on TOSC1. The user should not write a new value before the contents of the temporary register have been transferred to its destination. Each mentioned registers have their individual temporary register, which means that e.g. writing to TCNT2 does not disturb an OCR2 write in progress. To detect that a transfer to the destination register has taken place, the Asynchronous Status Register – ASSR has been implemented.

If using TC2 as condition to wake up MCU sleeping mode, sleeping mode would not be able to be entered into before each register finish its update, otherwise MCU would not enter into sleeping mode after TC2 configuration is active, which results that TC2 cannot wake up system.

If using TC2 as condition to wake up MCU sleeping mode, must note to re-enter into sleeping mode. Interrupt logic need a TOSC1 clock cyle to reset, otherwise interrupt will not occur and electronic parts will not be waken up. Below operations are recommended.
1. Write suitable data to each register.
2. Wait until the corresponding Update Busy Flag in ASSR returns to zero.
3. Enter into sleeping mode

When the asynchronous operation is selected, the TC2 is always running, except in Power-down modes. User should be aware of the fact that this Oscillator might take as long as one second to stabilize. So, after enable TC2 oscillator, the user is advised to wait for at least one second before using Timer/Counter2 asynchronous mode.

Description of wake up from sleeping mode when the timer is clocked asynchronously: When the interrupt condition is met, the wake up process is started on the following cycle of the timer clock, that is, the timer is always advanced by at least one before the processor can read the counter value. After wake-up, the MCU executes the interrupt routine, and resumes execution from the instruction following SLEEP.

Reading of the TCNT2 Register shortly after wake-up from sleeping mode may give an incorrect result. Since TCNT2 is clocked on the asynchronous TOSC1 clock, reading TCNT2 must be done through a register synchronized to the internal system clock . Synchronization takes place for every rising TOSC1 edge. When waking up from sleeping mode, and the system clock again becomes active, TCNT2 will read as the previous value (before entering sleep) until the next rising TOSC1 edge. The phase of the TOSC1 after waking up from sleeping mode is essentially unpredictable, as it depends on the wake-up time. The recommended procedure for reading TCNT2 is thus as follows:
1. Write any value to OCR2A or TCCR2A
2. Wait for the corresponding Update Busy Flag to be cleared
3. Read TCNT2
During asynchronous operation, the synchronization of the Interrupt Flags takes 3 processor cycles plus one timer cycle. The timer is therefore advanced by at least one before the MCU can read the timer value causing the setting of the Interrupt Flag. The Output Compare is changed on the timer clock and is not synchronized to the processor clock.

TC2 的预分频

TC2 预分频器的输入时钟称为 clkt2s，由位于 ASSR 寄存器的 AS2 位来选择内部系统时钟 clkio 或者外部 TOSC1 时钟源，缺省为与系统时钟 clkio 相连接。若 AS2 置位，TC2 将由 TOSC1 异步驱动。当 TOSC1 引脚和 TOSC2 引脚外接一个 32.768KHz 的钟表晶振，TC2 可用作 RTC 计数器。不推荐在 TOSC1 引脚上直接施加外部时钟信号

TC2 Prescaler

The clock source for TC2 is named clkT2S. By setting the AS2 bit in ASSR, to select internal system clock clkio or external TOSC1 clock source, otherwise it is connected with system clcok cklio by default. When AS2 is set, TC2 is driven asynchronized by TOSC1. When a crystal of 32.768kHz can then be connected between the TOSC1 and TOSC2 pins, TC2 can be used as RTC counter. It is not recommended to assign external clock signal directly on TOSC1 pin.

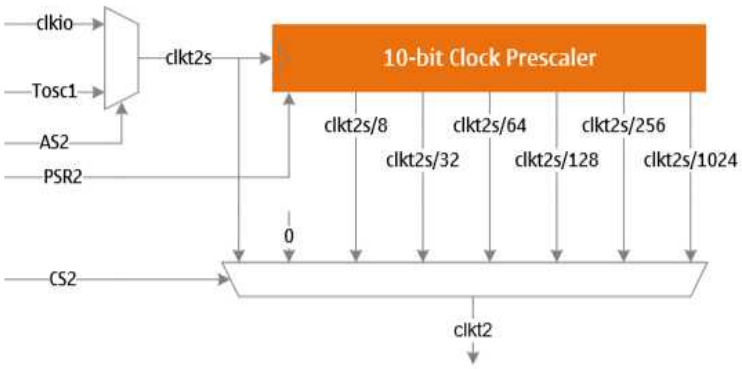


Figure 5 TC2 预分频器结构图

Above is picture of TC2 prescaler, possbile prescaler options areL： clkt2s/8, clkt2s/32, clkt2s/64, clkt2s/128, clkt2s/256 and clkt2s/1024, besides clkt2s and 0 (stop counting) are also for option. Setting PSR2 bit of SFIOR register to reset prescaler, than users are allowed to start working from predictable prescaler.

寄存器定义

Register Definition

TC2 Register List

Register	Address	Default value	Comments
TCCR2A	0xB0	0x00	TC2 contraol register A
TCCR2B	0xB1	0x00	TC2 contraol register B
TCNT2	0xB2	0x00	TC2 count value register
OCR2A	0xB3	0x00	TC2 output compare register A
OCR2B	0xB4	0x00	TC2 output compare register B
ASSR	0xB6	0x00	TC2 Asynchronous status register
TIMSK2	0x70	0x00	Timer counter interrupt mask register
TIFR2	0x37	0x00	Timer counter interrupt flag register

TCCR2A–TC2 control register A

TCCR2A–TC2 control register A								
Address: 0xB0					Default value: 0x00			
Bit	7	6	5	4	3	2	1	0
	COM2A1	COM2A0	COM2B1	COM2B0	-	-	WGM21	WGM20
7	WGM21	TC2 compare match output A mode control high COM2A1 and COM2A0 consist of output compare mode control COM2A[1:0] together to control output waveform of OC2A. When 1st bit or 2nd bit of COM2A are set, output compare waveform will take up OC2A pin, however data direction register of this pin must be set high in order to output waveform. In different working mode, COM2A controls output compare waveform differently, for details referring to “compare output mode control list”.						
6	COM2A0	TC2 compare match output A mode control low COM2A0 and COM2A1 consist of output compare mode control COM2A[1:0] together to control output waveform of OC2A. When 1st bit or 2nd bit of COM2A are set, output compare waveform will take up OC2A pin, however data direction register of this pin						

		must be set high in order to output waveform. In different working mode, COM2A controls output compare waveform differently, for details referring to “compare output mode control list”.
5	COM2B1	TC2 compare match output B mode control high COM2B1 and COM2B0 consist of output compare mode control COM2B[1:0] together to control output waveform of OC2B. When 1st bit or 2nd bit of COM2B are set, output compare waveform will take up OC2B pin, however data direction register of this pin must be set high in order to output waveform. In different working mode, COM2B controls output compare waveform differently, for details referring to “compare output mode control list”.
4	COM2B0	TC2 compare match output B mode control low COM2B0 and COM2B1 consist of output compare mode control COM2B[1:0] together to control output waveform of OC2B. When 1st bit or 2nd bit of COM2B are set, output compare waveform will take up OC2B pin, however data direction register of this pin must be set high in order to output waveform. In different working mode, COM2B controls output compare waveform differently, for details referring to “compare output mode control list”.
3:2	-	reserved
1	WGM21	TC2 waveform genreation mode control high WGM20 and WGM21, WGM22 consist of waveform genration mode control WGM2[2:0] together to control counting and waveform generation method of counter, for details referring to “compare output mode control list”.
0	WGM20	TC2 waveform genreation mode control low WGM21 and WGM20, WGM22 consist of waveform generation mode control WGM2[2:0] together to control counting and waveform generation method of counter, for details referring to “compare output mode control list”.

TCCR2B-TC2 control register B

TCCR2B-TC2 control register B								
Address: 0xB1						Default value: 0x00		
Bit	7	6	5	4	3	2	1	0
	FOC2A	FOC2B	-	-	WGM22	CS22	CS21	CS20
R/W	W	W	-	-	R/W	R/W	R/W	R/W
7	FOC2A	TC2 force output compare A control bit This bit works in non-PWM mode, by writing logic one to force output compare bit “FOC2A” to generate compare match. Force compare match does neither set OCF2A flag nor reload/clear timer, however output pin OC2A will be updated accordingly related to COM2A configuration, it seems that a real compare match does happen. Read return value of FOC2A till it is zero.						
6	FOC2B	TC2 force output compare B control bit This bit works in non-PWM mode, by writing logic one to force output compare bit “FOC2B” to generate compare match. Force compare match does neither set OCF2B flag nor reload/clear timer, however output pin OC2B will be updated accordingly related to COM2B configuration, it seems that a real compare match does happen. Read return value of FOC2B till it is zero.						

<table>
<tr><td>5:4</td><td>-</td><td>reserved</td></tr>
<tr><td>3</td><td>WGM22</td><td>TC2 waveform generation mode control high
WGM22 and WGM20, WGM21 consist of waveform generation mode control WGM2[2:0] together to control counting and waveform generation method of counter, for details referring to “compare output mode control list”.</td></tr>
<tr><td>2</td><td>CS22</td><td>TC2 clock selection control high
Used to select clock source of timer ccounter 2</td></tr>
<tr><td>1</td><td>CS21</td><td>TC2 clock selection control middle
Used to select clock source of timer ccounter 2</td></tr>
<tr><td>0</td><td>CS20</td><td>TC2 clock selection control low
Used to select clock source of timer ccounter 2
<table>
<tr><td>CS2[2:0]</td><td>comments</td></tr>
<tr><td>0</td><td>No clock source, stop countiing</td></tr>
<tr><td>1</td><td>clkt2s</td></tr>
<tr><td>2</td><td>clkt2s/8, from prescaler</td></tr>
<tr><td>3</td><td>clkt2s/32, from prescaler</td></tr>
<tr><td>4</td><td>clkt2s/64, from prescaler</td></tr>
<tr><td>5</td><td>clkt2s/128, from prescaler</td></tr>
<tr><td>6</td><td>clkt2s/256, from prescaler</td></tr>
<tr><td>7</td><td>clkt2s/1024, from prescaler</td></tr>
</table></td></tr>
</table>

Below list show how compare output mode controls output waveform in non-PWM mode (normal mode and CTC mode)

Table 1 OC2x compare output mode control in non -PWM mode

COM2x[1:0]	comments
0	OC2x disconnect, general IO port operation
1	Toggle OC2x singal under compare match
2	Clear OC2x singal under compare match
3	Set OC2x singal under compare match

Below list show how compare output mode controls output compare waveform in fast PWM mode

Table 2 OC2x compare output mode control in fast PWM mode

COM2x[1:0]	comments
0	OC2x disconnect, general IO port operation
1	Reserved
2	Clear OC2x singal under compare match, set OC2x signal upon maxinum value match
3	Set OC2x singal under compare match, clear OC2x signal upon maxinum value match

Below list show how compare output mode controls output compare waveform in phase correction mode

Table 3 OC2x compare output mode control in phase correction mode

COM2x[1:0]	comments
0	OC2x disconnect, general IO port operation
1	Reserved
2	Clear OC2x singal in up-counting, set OC2x signal in down-countiing
3	Set OC2x singal in up-counting, clear OC2x signal in down-countiing

Below list is about waveform generation control

Table 4 Waveform generation control

WGM2[2:0]	Working Mode	TOP Value	Update OCR2x time	Set TOV2 time
0	Normal	0xFF	immediately	MAX
1	PCPWM	0xFF	TOP	BOTTOM
2	CTC	OCR2A	immediately	MAX
3	FPWM	0xFF	TOP	MAX
4	reserved	-	-	-
5	PCPWM	OCR2A	TOP	BOTTOM
6	reserved	-	-	-
7	FPWM	OCR2A	TOP	TOP

TCNT2--TC2 counter register

TCNT2--TC2 counter register								
Address: 0xB2					Default value:0x00			
bit	7	6	5	4	3	2	1	0
	TCNT27	TCNT26	TCNT25	TCNT24	TCNT23	TCNT22	TCNT21	TCNT20
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	Name	Comments						
7:0	TCNT2	TC2 counter register Via TCNT2 register, to directly read/write 8-bit counter In the next timer clock cycle, compare match is prevented from happening by CPU write operation to TCNT2 register, even though timer have stopped, this allows that TCNT2 register value will be the same as value of OCR2, than interrupt will not occur. If a written TCNT2 value is the same as OCR2 or bypass OCR2, compare match will lose, which would result in a wrong waveform generation. Timer stops counting when clock source is not selected, but CPU can still access to TCNT2. Writing counter by CPU has higher priority than clear or plus/minus operation.							

OCR2A--TC2 output compare register A

OCR2A--TC2 output compare register A								
Address: 0xB3					Default value:0x00			
bit	7	6	5	4	3	2	1	0
	OCR2A7	OCR2A6	OCR2A5	OCR2A4	OCR2A3	OCR2A2	OCR2A1	OCR2A0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	Name	Comments						
7:0	OCR2A	TC2 output compare reigster A						

		OCR2A includes a 8-bit data and it compares with counter value TCNT2 continuously. Compare match is used to generate output compare interrupt, or generate waveform in OC2A pins. When in PWM mode, OCR2A register uses dual buffer register, while which is prohibited in normal mode and match clear mode. Via dual buffer register updating OCR2A register and counter maximum or mininum moment is synchronized, in order to generate asynmetric PWM pulse and eliminate pulse interrupt. If dual buffer is under use, CPU is access to OCR2A buffer register, however if it is prohibited CPU is access to OCR2A itself

OCR2B--TC2 output compare register B

OCR2B--TC2 output compare register B								
Address: 0xB4					Default value:0x00			
bit	7	6	5	4	3	2	1	0
	OCR2B7	OCR2B6	OCR2B5	OCR2B4	OCR2B3	OCR2B2	OCR2B1	OCR2B0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	Name	Comments						
7:0	OCR2B	TC2 output compare reigster B OCR2B includes a 8-bit data and it compares with counter value TCNT2 continuously. Compare match is used to generate output compare interrupt, or generate waveform in OC2B pins. When in PWM mode, OCR2B register uses dual buffer register, while which is prohibited in normal mode and match clear mode. Via dual buffer updating OCR2A register and counter maximum or mininum moment is synchronized, in order to generate asynmetric PWM pulse and eliminate pulse interrupt. If dual buffer function is under use, CPU is access to OCR2B buffer register, however if it is prohibited CPU is access to OCR2A itself							

TIMSK2--TC2 interrupt mask register

TIMSK2--TC2 interrupt mask register								
Address: 0x70					Default value:0x00			
bit	7	6	5	4	3	2	1	0
	-	-	-	-	-	OCIE2B	OCIE2A	TOIE2
R/W	-	-	-	-	-	R/W	R/W	R/W
Bit	Name	Comments						
7:3		reserved						
2	OCIE2B	TC2 output compare B match interrupt enable bit If OCIE2B bit is writtSen to logic one, and global interrupt is enable, TC2 output compare B match interrupt is enabled. If compare match occurs, that is OCF2B bit from TIFR2 is enabled, interrupt will happen. If OCIE2B is written to logic zero, TC2 output compare B match intterrupt is disabled.						
1	OCIE2A	TC2 output compare A match interrupt enable bit If OCIE2A bit is written to logic one, and global interrupt is enable, TC2 output compare A match interrupt is enabled.						

		If compare match occurs, that is OCF2A bit from TIFR2 is enabled, interrupt will happen. If OCIE2A is written to logic zero, TC2 output compare A match intterrupt is disabled.
0	TOIE2	TC2 overflow interrupt enable bit If TOIE2 is logic one, and globle interrupt enables , TC2 overflow interrupt is enabled. If TC2 overflows, that is TOV2 bit from TIFR2 is enabled, interrupt will happen. If TOIE2 is logic zero, TC2 overflow interrupt is disabled.

TIFR2--TC2 interrupt flag register

TIFR2--TC2 interrupt flag register								
Address: 0x37					Default value:0x00			
bit	7	6	5	4	3	2	1	0
	-	-	-	-	-	OCF2B	OCF2A	TOV2
R/W	-	-	-	-	-	R/W	R/W	R/W
Bit	Name	Comments						
7:3	-	reserved						
2	OCF2B	TC2 output compare B match flag bit When TCNT2 equals to OCR2B, compare unit gives match signal and enable compare flag OCF2B. At this moment if output compare B interrupt eanble OCIE2B to logic one while globe interrupt flag is enabled, output compare B interrupt occurs accordingly. When executing this interrupt routine program, OCF2B will clear automatically or is cleard by writting OCF2B to logic one.						
1	OCF2A	TC2 output compare A match flag bit When TCNT2 equals to OCR2A, compare unit gives match signal and enable compare flag OCF2A. At this moment if output compare A interrupt eanble OCIE2A to logic one while globe interrupt flag is enabled, output compare A interrupt occurs accordingly. When executing this interrupt routine program, OCF2A will clear automatically or is cleard by writting OCF2A to logic one.						
0	TOV2	TC2 overflow interrupt enable bit If counter overflows, overflow flag TOV2 is enabled. At this moment if overflow interrupt enables TOIE2 to logic one while globe interrupt flag is enabled, overflow interrupt happens. When executing this interrupt routine program, TOV2 will clear automatically or is cleard by writting TOV2 to logic one.						

ASSR-- Asynchronous Status Register

ASSR-- Asynchronous Status Register								
Address: 0xB6					Default value:0x00			
bit	7	6	5	4	3	2	1	0
	INTCK	-	AS2	TCN2UB	OCR2AUB	OCR2BUB	TCR2AUB	TCR2BUB
R/W	R/W	-	R/W	R/W	R/W	R/W	R/W	R/W
Bit	Name	Comments						

7	INTCK	Asynchronous clock selection control bit When writting INTCK bit to logic one, internal RC32K is selected as asynchronous clock source. When writting INTCK bit to logic zero, external crystal oscillator clock is selected as asynchronous clock source.
6	-	reserved
5	AS2	Timer 2 asynchronous mode selection control bit If AS2 is enabled to logic one, and timer 2 works in asynchronous mode, its clock source is selected by INTCK bit If AS2 is enabled to logic zero, and timer 2 works in asynchronous mode, its clock source is Clkio When AS2 value changes, TCNT2, OCR2A，OCR2B，TCCR2A and TCCR2B register value would be incorrect, they need to be re-configured.
4	TCN2UB	TCNT2 register update flag bit When timer 2 works in asynchronous mode and TCNT2 is under write operation, TCN2UB bit will be enabled. When TCNT2 value is finished to be update, hardware clears TCN2UB bit. TNCT2 can only be updated when TCN2UB is written to logic zero.
3	OCR2AUB	OCR2A register update flag bit When timer 2 works in asynchronous mode and OCR2A is under write operation, OCR2AUB bit will be enabled. When OCR2B value is finished to be update, hardware clears OCR2AUB bit. OCR2B can only be updated when OCR2AUB is written to logic zero.
2	OCR2BUB	OCR2B register update flag bit When timer 2 works in asynchronous mode and OCR2B is under write operation, OCR2BUB bit will be enabled. When OCR2B value is finished to be update, hardware clears OCR2BUB bit. OCR2B can only be updated when OCR2BUB is written to logic zero.
1	TCR2AUB	TCCR2A register update flag bit When timer 2 works in asynchronous mode and TCCR2A is under write operation, TCR2AUB bit will be enabled. When TCCR2A value is finished to be updated, hardware clears TCR2AUB bit. TCCR2A can only be updated when TCR2AUB is written to logic zero.
0	TCR2BUB	TCCR2B register update flag bit When timer 2 works in asynchronous mode and TCCR2B is under write operation, TCR2BUB bit will be enabled. When TCCR2B value is finished to be updated, hardware clears TCR2BUB bit. TCCR2B can only be updated when TCR2BUB is written to logic zero.

定时 / 计数器 3 (TMR3)

- **真正的 16 位设计，允许 16 位的 PWM**
- **3 个独立的输出比较单元**
- **双缓冲的输出比较寄存器**
- **1 个输入捕捉单元**
- **输入捕捉噪声抑制器**

- 比较匹配时自动清零计数器并自动加载
- 无干扰脉冲的相位修正的 PWM
- 可变的 PWM 周期
- 频率发生器
- 外部事件计数器
- 5 个独立的中断源
- 带死区时间控制
- 6 个可选触发源自动关闭 PWM 输出

Timer/ Counter 3 (TMR3)

- Real 16-bit design, support 16-bit PWM
- Three independent compare unit
- Dual buffered output compare register
- One input capture unit
- Input capture noise reducer
- In compare match, automate clearing counter and automate reload
- Non-intervene pulse phase correction PWM
- Variable PWM cycle
- Frequency generator
- External event counter
- 5 pcs of independant interrupt source
- With dead area time control
- 6 pcs of automate shut-down PWM output by trigger source for option

概 述

TC3 是一个通用 16 位定时计数器模块，支持 PWM 输出，可以精确地产生波形。TC3 包含 1 个 16 位计数器，波形产生模式控制单元，2 个独立的输出比较单元和 1 个输入捕 捉单元。波形产生模式控制单元控制着计数器的工作模式和比较输出波形的产生。根据 不同的工作模式，计数器对每一个计数时钟 Clkt3 实现清零、加一或减一操作。Clkt3 可 以由内部时钟源或外部时钟源产生。当计数器的计数值 TCNT3 到达最大值（等于极大值 0xFFFF 或固定值或输出比较寄存器 OCR3A 或输入捕捉寄存器 ICR3，定义为 TOP，定义极 大值为 MAX 以示区别）时，计数器会进行清零或减一操作。当计数器的计数值 TCNT3 到 达最小值（等于 0x0000，定义为 BOTTOM）时，计数器会进行加一操作。当计数器的计 数值 TCNT3 到达 OCR3A 或 OCR3B 或 OCR3C 时，也被称为发生比较匹配时，会清零或置 位输出比较信号 OC3A 或 OC3B 或 OC3C，来产生 PWM 波形。当开启输入捕捉功能时，计 数器被触发即开始或停止计数，ICR3 寄存器会记录捕捉信号触发周期内的计数值

Overview

TC3 is a general purpose 16-bit Timer/Counter module, with PWM support. It allows accurate wave generation. TC2 consists of 1 16-bit counter, waveform generation control unit and 2 independent output compare unit. Waveform generation mode control unit is used to control working mode of counter and generation of compare output waveform.

In each different working mode, counter does operation of clear, plus one or minus one to each counting clock Clkt3, which is generated by internal or external clock source.

If counting number TCNT3 of counter is maximum(max. value OxFFFF or a fixed value or output compare register OCR3A or input capture register ICR3, which are defined as TOP, define max. value as MAX for better identify), counter will clear or make minus 1 operation.

If counting number TCNT3 of counter is minimum (Ox0000, or define as BOTTOM), counter will operate PLUS 1.

If counting number TCNT3 arrives at OCR3A or OCR3B or OCR3C, it signifies that compare match occurs,

counter will clear or is set to output compare signal 0C3A or 0C3B or OC3C to generate PMW waveform.

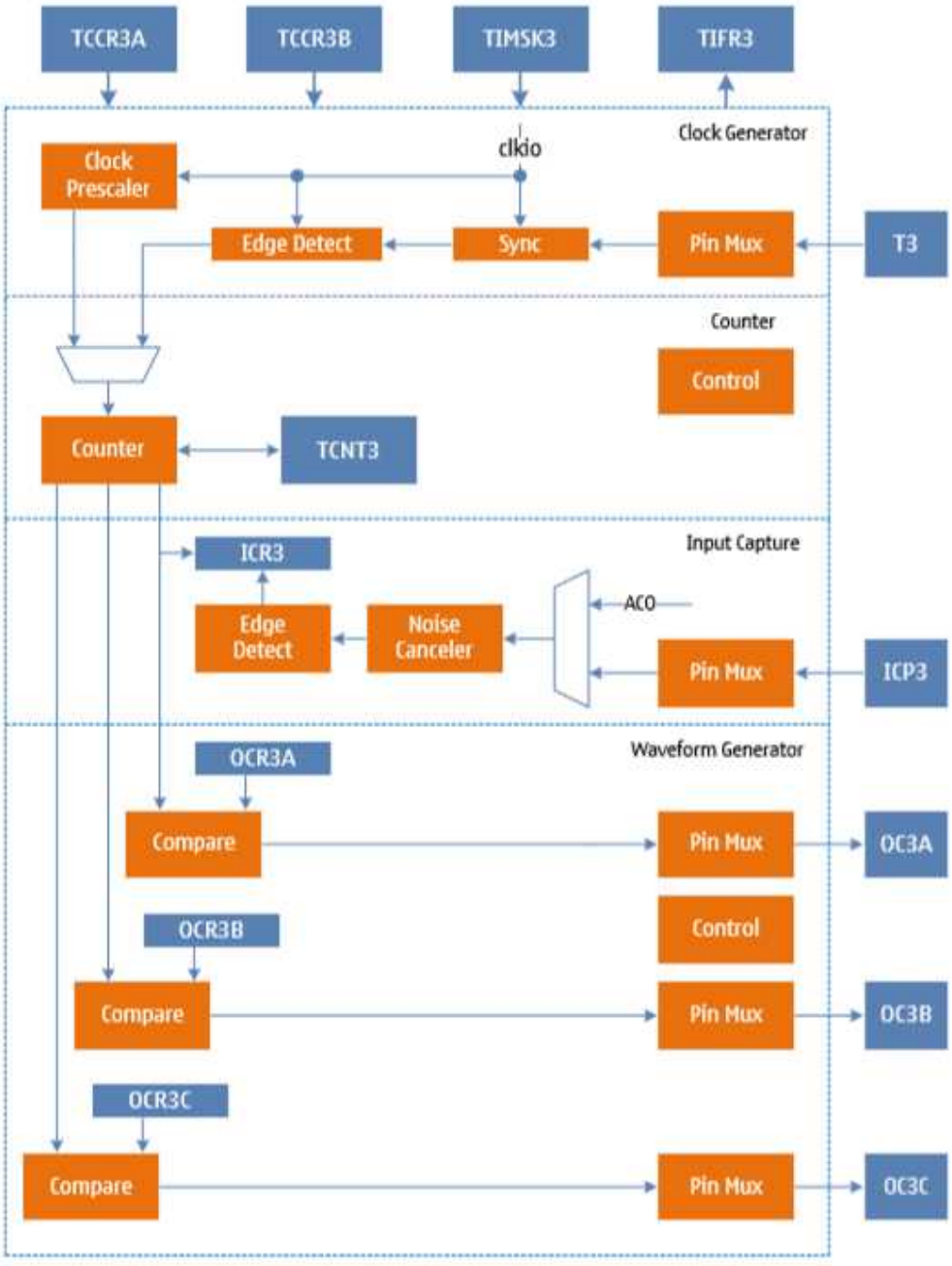


Figure 6 TC3 结构图

工作模式

定时计数器 1 有六种不同的工作模式，包括普通模式（Normal），比较匹配时清零（CTC） 模式，快速脉冲宽度调制（FPWM）模式，相位修正脉冲宽度调制（PCPWM）模式，相位频率修正脉冲宽度调制（PFCPWM）模式，和输入捕捉（ICP）模式。由波形产生模式控制位 WGM3[3:0]来选择。下面具体来描述这六种模式。由于有三个独立的输出比较单元，分别用 “A”、“B”和“C”来表示，用小写的“x”来表示这两个输出比较单元通道。

Working Mode

Timer counter 1 has 6 different working modes, including normal mode (Normal), Compare Match on clear mode, (CTC), fast pulse width modulation mode (FPWM) and phase correction pulse width modulation mode (PCPWM), phase frequency correction pulse width modulation mode(PFCPWM) and input capture mode (ICP), which are selected by waveform generation mode control bit WGM3[3:0]. In the below these 6 modes is

written in details. Since there are 3 independent output compare units, they are indicated separately with “A”, “B” and “C”. Small “x” is used to indicate channel of these two output compare unit.

普通模式

普通模式是定时计数器最简单的工作模式，此时波形产生模式控制位 WGM3[3:0]=0，计数 的最大值 TOP 为 MAX（0xFFFF）。在这种模式下，计数方式为每一个计数时钟加一递增，当 计数器到达 TOP 溢出后就回到 BOTTOM 重新开始累加。在计数值 TCNT3 变成零的同一个计 数时钟里置位定时计数器溢出标志 TOV3。这种模式下 TOV3 标志就像是第 17 计数位，只是 只会被置位不会被清零。溢出中断服务程序会自动清除 TOV3 标志，软件可以用它来提高定 时计数器的分辨率。普通模式下没有特殊情形需要考虑，可以随时写入新的计数值。

设置 OC3x 引脚的数据方向寄存器为输出时才能得到输出比较信号 OC3x 的波形。当 COM3x=1 时，发生比较匹配时会翻转 OC3x 信号，这种情况下波形的频率可以用下面的公式 来计算： fOC3xnormal = fsys/(2*N*65536) 其中，N 表示的是预分频因子（1，8，64，256 或者 1024）。

输出比较单元可以用来产生中断，但是在普通模式下不推荐使用中断，这样会占用太多 CPU 的时间。

Normal Mode

The simplest mode of operation is the Normal mode (WGM3[3:0] = 0), its maximum TOP is MAX（0xFFFF). In this mode, counting increase by each counting clock, when it passes TOP and overflow, it will then restart to accumulate from the bottom. The Timer/Counter Overflow Flag (TOV3) will be set in the same timer clock cycle as the TCNT3 becomes zero. The TOV3 Flag in this case behaves like a 17th bit, except that it is only set, not cleared.

However, combined with overflow interrupt that automatically clears the TOV3 Flag, the timer resolution can be increased by software. There are no special cases to consider in the Normal mode, a new counter value can be written anytime.

Waveform of output compare signal OC3x is generated only if data direction register of OC3x is set. If COM3x=1, OC3x signal will be toggled when compare match is happening, in this case waveform frequency can be calculated by below formula:

Foc3xnormal = fsys/(2*N*65536),

N indicates prescaler factor (1，8，64，256 or 1024).

Using the Output Compare unit to generate waveforms in Normal mode is not recommended, since this will occupy too much of the CPU time.

CTC 模式

设置 WGM3[3:0]=4 或 12 时，定时计数器 1 进入 CTC 模式。当 WGM3[3]=0 时，计数最大值 TOP 为 OCR3A，当 WGM3[3]=1 时，计数最大值 TOP 为 ICR3。下面以 WGM3[3:0]=4 为例来描述 CTC 模式在这个模式下，计数方式为每一个计数时钟加一递增，当计数器的数值 TCNT3 等于 TOP 时计数器清零。这个模式使得用户可以很容易的控制比较匹配输出的频率，也简化了外部事件计数的操作。

当计数器到达 TOP=OCR3A 时，输出比较匹配标志 OCF3A 被置位，当计数器到达 TOP=ICR3 时，输出比较匹配标志 ICF3 被置位，相应的中断使能置位时将会产生中断。在中断服务程序里可以更新 OCR3A 寄存器。在这个模式下 OCR3A 没有使用双缓冲，在计数器以无预分频器或很低的预分频器工作下将最大值更新为接近最小值的时候要小心。如果写入 OCR3A 的数值小于当时的 TCNT3 值时，计数器将丢失一次比较匹配。在下一次比较匹配发生之前，计数器不得不先计数到 MAX，然后再从 BOTTOM 开始计数到 OCR3A。和普通模式一样，计数值回到 0x0 的计数时钟里置位 TOV3 标志。

置 OC3x 引脚的数据方向寄存器为输出时才能得到输出比较信号 OC3x 的波形。波形的频率可以用下面的公式来计算：

fOC3xctc = fsys/(2*N*(1+OCR3A))

其中，N 表示的是预分频因子（1，8，64，256 或者 1024）。

从公式可以看出，当设置 OCR3A 为 0x0 且无预分频器时，可以获得最大频率为 fsys/2 的输出波形。

当 WGM3[3:0]=12 时与 WGM3[3:0]=4 类似，只是把与 OCR3A 相关的换成 ICR3 即可

CTC Mode

Set WGM3[3:0]=4 or 12, timer/ counter 1 is in CTC mode, max. When WGM3[3]=0, counting number TOP is OCR3A. When WGM3[3]=1, counting number TOP is ICR3. In the below we take WGM3[3:0] as an example to describe that in CTC mode: counting increase by each counting clock, counter is cleared when TCNT3 is TOP. This mode allows greater control of the compare match output frequency. It also simplifies the operation of counting external events.

When counter value reaches TOP(OCR3A) value, output compare match flag OCF3A is set, when counter value reaches TOP (ICR3), output compare match flag ICF3 is set, corresponding interrupt enable will generate interrupt. If the interrupt is enabled, the interrupt handler routine can be used for updating OCR3A register. However, this changing OCR3A does not use dual buffer. When the counter is running with none or a low prescaler, it must be done with care to update TOP value to Bottom. If the new value written to OCR3A is lower than the current value of TCNT3, the counter will miss one compare match. Before next compare match, the counter will then have to count to MAX and then re-start from BOTTOM to count to OCR3A value. As for the Normal mode of operation, the TOV3 Flag is set when counting value return to 0x00. Waveform of output compare signal OC2x is generated only if data direction register of OC2x pin is set output.
Waveform of output compare signal OC3x can only be gained when data direction register of OC3x is set to output. In this case waveform frequency can be calculated by below formula:
Foc3xctc = fsys/(2*N*(1+OCR3A))
Here, N indicates prescaler factor (1，8，64，256 or 1024)
Seeing from above formula, set OCR3A to 0x0 meanwhile without prescaler, output waveform of max. fsys/2 can be gained.
It is a similar case when WGM3[3:0]=12 and WGM3[3:0]=4, it is only necessary to change OCR3A related data to ICR3.

快速 PWM 模式

设置 WGM3[3:0]=5，6，7，14 或 15 时，定时计数器 1 进入快速 PWM 模式，计数最大值 TOP 分别为 0xFF， 0x1FF， 0x3FF， ICR3 或 OCR3A，可以用来产生高频的 PWM 波形。快速 PWM 模式和其他 PWM 模式不同在于它是单向操作。计数器从 BOTTOM 累加到 TOP 后又回到 BOTTOM 重新计数。当计数值 TCNT3 到达 TOP 或 BOTTOM 时，输出比较信号 OC3x 会被置位或清零，取决于比较输出模式 COM3 的设置，详情见寄存器描述。由于采用单向操作，快速 PWM 模式的操作频率是采用双向操作的相位修正 PWM 模式的两倍。高频特性使得快速 PWM 模式适用于功率调节，整流以及 DAC 应用。高频信号可以减小外部元器件（电感电容等）的尺寸，从而降低系统成本。

当计数值到达 TOP 时，定时计数器溢出标志 TOV3 将会被置位，并把比较缓冲器的值更新到比较值。如果中断使能，在中断服务程序中可以更新 OCR3A 寄存器。

设置 OC3x 引脚的数据方向寄存器为输出时才能得到输出比较信号 OC3x 的波形。波形的频率可用下面的公式来计算：
fOC3xfpwm = fsys/(N*(1+TOP))
其中，N 表示的是预分频因子（1，8，64，256 或者 1024）。
当 TCNT3 和 OCR3x 发生比较匹配时，波形产生器就置位（清零）OC3x 信号，当 TCNT3 被清零时，波形产生器就清零（置位）OC3x 信号，以此来产生 PWM 波。由此 OCR3x 的极值将会产生特殊的 PWM 波形。当 OCR3x 设置为 0x00 时，输出的 PWM 为每(1+TOP)个计数时钟里有一个窄的尖峰脉冲。当 OCR3x 设置为 TOP 时，输出的波形为持续的高电平或低电平。如果用 OCR3A 作为 TOP 并设置 COM3A=1，输出比较信号 OC3A 会产生占空比为 50%的 PWM 波。

Fast PWM Mode

Setting WGM3[3:0]=5,6,7,14 or 15, Timer/Counter 1 enters into fast PWM mode, their TOP are separately 0Xff, 0x1FF, 0x3FF, ICR3 or OCR3A, this mode provides a high frequency PWM waveform. The fast PWM differs

from the other PWM option by its single-slope operation.
The counter counts from Bottom to TOP then restarts from BOTTOM. When counting value TCNT3 reaches TOP or BOTTOM, output compare OC3x is set or cleared, which depends on configuration of compare output COM3, details referring to sections about register. Due to the single-slope operation, the operating frequency of the fast PWM mode can be twice as high as the phase correct PWM mode that uses dual-slope operation. This high frequency makes the fast PWM mode well suited for power regulation, rectification, and DAC applications. High frequency allows physically small sized external components (capacitors), and therefore reduces total system cost.

When counting value reaches TOP, timer counter overflow flag TOV3 will be set and value of compare buffer is updated to compare value. If interrupt is enabled, OCR3A register can be updated in interrupt handler routine. Waveform of output compare signal OC3x is generated only if data direction register of OC3x pin is set as output. Waveform frequency can be calculated by below formular:

Foc3xfpwm = fsys/(N*(1+TOP))

Here, N indicates prescaler factor (1，8，64，256 and 1024)
When compare match occurs between TCNT3 and OCR3x, waveform generator is set to OC3x (cleared). When TCNT3 is cleared, waveform generator will clear OC3x to generate PWM waveform, so extreme value of OCR3x will generate very special PWM waveform. If the OCR3x is set to 0x00 the output PWM will be a narrow spike for each TOP+1 timer clock cycle. Setting the OCR3x to TOP will result in a constantly high or low output. If OCR3A is set as TOP and COM3A is set to logic one, OC3A, output compare signal, will generate PWM waveform with 50% duty cycle.

相位修正 PWM 模式

当设置 WGM3[3:0]=1，2，3，10 或 11 时，定时计数器 1 进入相位修正 PWM 模式，计数的最大值 TOP 分别为 0xFF，0x1FF，0x3FF，ICR3 或 OCR3A。计数器采用双向操作，由 BOTTOM 递增到 TOP，然后又递减到 BOTTOM，再重复此操作。计数到达 TOP 和 BOTTOM 时均改变计数方向，计数值在 TOP 或 BOTTOM 上均只停留一个计数时钟。在递增或递减过程中，计数值 TCNT3 与 OCR3x 匹配时，输出比较信号 OC3x 将会被清零或置位，取决于比较输出模式 COM3 的设置。与单向操作相比，双向操作可获得的最大频率要小，但其极好的对称性更适合于电机控制。

相位修正 PWM 模式下，当计数到达 BOTTOM 时置位 TOV3 标志，当计数到达 TOP 时把比较缓冲器的值更新到比较值。如果中断使能，在中断服务程序中可以更新比较缓冲器 OCR3x 存器。

设置 OC3x 脚的数据方向寄存器为输出时才能得到输出比较信号 OC3x 波形。波形的频率可用下面的公式来计算：

fOC3xcpcpwm = fsys/(N*TOP*2)

其中，N 表示的是预分频因子（1，8，64，256 或者 1024）。

在递增计数过程中，当 TCNT3 与 OCR3x 匹配时，波形产生器就清零（置位） OC3x 信号。在递减计数过程中，当 TCNT3 与 OCR3x 匹配时，波形产生器就置位（清零）OC3x 信号。由此 OCR3x 的极值会产生特殊的 PWM 波。当 OCR3x 设置为 TOP 或 BOTTOM 时，OC3x 信号出会一直保持低电平或高电平。如果用 OCR3A 作为 TOP 并设置 COM3A=1，输出比较信号 OC3A 会产生占空比为 50%的 PWM 波。

为了保证输出 PWM 波在 BOTTOM 两侧的对称性，在没有发生比较匹配时，有两种情况下也会翻转 OC3x 信号。第一种情况是，当 OCR3x 的值由 TOP 改变为其他数据时。当 OCR3x 为 TOP，计数值达到 TOP 时，OC3x 的输出与前面降序计数时比较匹配的结果相同，即保持 OC3x 不变。此时会更新比较值为新的 OCR3x 的值（非 TOP）， OC3x 的值会一直保持，直到升序计数时发生比较匹配而翻转。此时 OC3x 信号并不以最小值为中心对称，因此需要在 TCNT3 到达最大值时翻转 OC3x 信号，此即没有发生比较匹配时翻转 OC3x 信号的第一种情况。第二种情况是，当 TCNT3 从比 OCR3x 高的值开始计数时，因而会丢失一次比较匹配，从而引起不对称情形的产生。同样需要翻转 OC3x 信号去实现最小值两侧的对称性。

Phase Correct PWM Mode

Setting WGM3[3:0] to 1, 2, 3, 10 or 11, Timer/Counter 1 enter into Phase Correct PWM mode, each TOP value is

0xFF，0x1FF，0x3FF，ICR3 or OCR3A. The phase correct PWM mode is based on a dual-slope operation. The counter counts repeatedly from BOTTOM to TOP than decrease to BOTTO. When counter reaches TOP or BOTTOM, it will both change counting direction. Counting value at TOP or BOTTOM only stays for one counting clock. On the compare match between TCNT3 and OCR3x while up-counting or down-counting, the Output Compare (OC3x) is cleared or set depending on the configuration of compare output mode COM3. The dual-slope operation has lower maximum operation frequency than single slope operation. However, due to the symmetric feature of the dual-slope PWM modes, these modes are preferred for motor control applications.

In phase correct PWM mode, the TOV3 is set when the counter reaches BOTTOM. When the counter reaches TOP, value of compare buffer is updated to compare value. If interrupt enable, compare buffer OCR3x register can be updated in interrupt routine program.

Setting OC2x pin data direction register to output, it will generate waveform of output compare OC3x. Waveform frequency can be calculated via below formula:

Foc3xcpcpwm = fsys/(N*TOP*2)

Here, N indicates prescaler factor (1，8，64，256 or 1024)

During up-counting, when match occurs between TCNT3 and OCR3x, waveform generator clear (set) OC3x . During down-counting, when match occurs between TCNT3 and OCR3x, waveform generator set (clear) OC3x, so extreme value of OCR3x will generate very special PWM waveform. Setting the OCR3x to TOP or BOTTOM will result in a constantly high or low level change. If OCR3A is set as TOP and COM3A is set to logic one, OC3A, output compare signal, will result in PWM waveform with 50% duty cycle.

OC3x will be toggled in two cases when compare match does not occur while to ensure symmetry of both side of PMW waveform at its bottom value.

The first case : OCR3x value is changed from TOP to others. When the OCR3x value is TOP, counting value reaches TOP, OC3x output is the same as compare match from previous down-counting, that is to say OC3x keeps unchanged.

In this case Compare value is updated to a new OCR3x value (not TOP). OC3x keeps unchanged till it is toggled in up-counting compare match. At this moment OC3x is not symmetric with minimum value, so OC3x need to be toggled when TCNT3 reaches maximum value.

The 2nd case: when TCNT3 starts to count from a value that is higher than OCR3x, one compare match will loss, which will result in asymmetric. It is same that to achieve symmetric at both side of bottom value, OC3x is needed to be toggled.

相位频率修正 PWM 模式

当设置 WGM3[3:0]=8 或 9 时，定时计数器 1 进入相位频率修正 PWM 模式，计数的最大值 TOP 分别为 ICR3 或 OCR3A。计数器采用双向操作，由 BOTTOM 递增到 TOP，然后又递减到 BOTTOM，再重复此操作。计数到达 TOP 和 BOTTOM 时均改变计数方向，计数值在 TOP 或 BOTTOM 上均只停留一个计数时钟。在递增或递减过程中，计数值 TCNT3 与 OCR3x 匹配时，输出比较信号 OC3x 将会被清零或置位，取决于比较输出模式 COM3 的设置。与单向操作相比，双向操作可获得的最大频率要小，但其极好的对称性更适合于电机控制。

相位频率修正 PWM 模式下，当计数到达 BOTTOM 时置位 TOV3 标志，并且把比较缓冲器的值更新到比较值，更新比较值的时间是相位频率修正 PWM 模式和相位修正 PWM 模式的最大不同点。如果中断使能，在中断服务程序中可以更新比较缓冲器 OCR3x 存器。当 CPU 改变 TOP 值即 OCR3A 或 ICR3 的值时，必须保证新的 TOP 值不小于已经在使用的 TOP 值，否则比较匹配将不会再发生。

设置 OC3x 脚的数据方向寄存器为输出时才能得到输出比较信号 OC3x 波形。波形的频率可用下面的公式来计算：

fOC3xcpfcpwm = fsys/(N*TOP*2)

其中，N 表示的是预分频因子（1，8，64，256 或者 1024）。

在递增计数过程中，当 TCNT3 与 OCR3x 匹配时，波形产生器就清零（置位） OC3x 信号。在递减计数过程中，当 TCNT3 与 OCR3x 匹配时，波形产生器就置位（清零）OC3x 信号。由此 OCR3x 的极值会产生特殊的 PWM 波。当 OCR3x 设置为 TOP 或 BOTTOM 时，OC3x 信号输出会一直保持低电平或高电平。如果用 OCR3A 作为 TOP 并设置 COM3A=1，输出比较信号 OC3A 会产生占空比为 50%的 PWM 波。
因为 OCR3x 寄存器是在 BOTTOM 时刻更新的，所以 TOP 值两边升序和降序的计数长度是一样的，也就产生了频率和相位都正确的对称波形。
当使用固定 TOP 值时，最好采用 ICR3 寄存器作为 TOP 值，即设置 WGM3[3:0]=8，此时 OCR3A 寄存器只需用来产生 PWM 输出。如果要产生频率变化的 PWM 波，必须通过改变 TOP 值， OCR3A 的双缓冲特性会更适合于这个应用。

Phase Correct PWM Mode

Setting WGM3[3:0] to 8 or 9, Timer/Counter 1 enters into Phase Correct PWM mode, each TOP value is ICR3 or OCR3A. The phase correct PWM mode is based on a dual-slope operation. The counter counts repeatedly from BOTTOM to TOP and than decrease to BOTTOM. When counter reaches TOP or BOTTOM, it will both change counting direction. Counting value at TOP or BOTTOM only stays for one counting clock.
On the compare match between TCNT3 and OCR3x during up-counting or down-counting, the Output Compare (OC3x) is cleared or set depending on the configuration of compare output mode COM3. The dual-slope operation has lower maximum operation frequency than single slope operation, due to its symmetric feature of the dual-slope PWM modes, these modes are preferred for motor control applications.
In phase correct PWM mode, the TOV3 is set when the counter reaches BOTTOM, meanwhile value of compare buffer is updated to compare value. The critical difference between phase frequency correction PWM mode and phase correction PWM mode is their timing of updating compare value. f interrupt is enabled, compare buffer OCR3x register can be updated in interrupt routine program. When CPU changes TOP value, OCR3A or ICR3, it is a must to make sure that the new TOP should never smaller than the current TOP value, otherwise compare match will not occur.
Setting OC3x pin data direction register to output, it will generate waveform of output compare OC3x. Waveform frequency can be calculated via below formula:

Foc3xcpfcpwm = fsys/(N*TOP*2)
Here, N indicates prescaler factor (1，8，64，256 or 1024)

During up-counting, when match occurs between TCNT3 and OCR3x, waveform generator will clear (set) OC3x . During down-counting, when match occurs between TCNT3 and OCR3x, waveform generator will set (clear) OC3x, so extreme value of OCR3x will generate very special PWM waveform. Setting the OCR3x to TOP or BOTTOM will result in a constantly high or low level change. If OCR3A is set as TOP while COM3A is set to logic one, output compare signal, OC3A, will result in PWM waveform with 50% duty cycle.
OCR3A register is updated at BOTTOM moment, so up-counting and down-counting length at both side of TOP value is the same, then symmetric waveform with correct frequency and phase is generated.
When using a fixed TOP value, it is best to take ICR3 register as TOP value, that is to say setting WGM3[3:0]=8, at this moment OCR3A register is only used to generate PWM output. If PWM waveform with variable frequency is in need, it is a must to change TOP value, so dual buffer of OCR3A is better suitable for such application.

输入捕捉模式
输入捕捉用来捕获外部事件，并为其赋予时间标记以说明此事件发生的时刻，可以在前面的计数模式下进行，不过要除去使用 ICR3 值作为计数 TOP 值的波形产生模式。
外部事件发生的触发信号由引脚 ICP3 输入，也可以通过模拟比较器单元来实现。当引脚 ICP3 上的逻辑电平发生变化，或模拟比较器的输出 ACO 电平发生变化，并且这个电平变化被输入捕捉单元所捕获，输入捕捉即被触发，此时 16 位的计数值 TCNT3 数据被复制到输入捕捉寄存器 ICR3，同时

输入捕捉标志 ICF3 置位，若 ICIE1 位为“1”，输入捕捉标志将产生输入捕捉中断。

通过设置模拟比较控制与状态寄存器 ACSR 的模拟比较输入捕捉控制位 ACIC 来选择输入捕捉触发源 ICP3 或 ACO。需注意的是，改变触发源有可能造成一次输入捕捉，因此在改变触发源后必须对 ICF3 进行一次清零操作来避免出现错误的结果。

输入捕捉信号经过一个可选的噪声抑制器之后送入边沿检测器，根据输入捕捉选择控制位 ICES1 的配置，看检测到的边沿是否满足触发条件。噪声抑制器是一个简单的数字滤波，对输入信号进行 4 次采样，只有当 4 次采样值都相等时其输出才会送入边沿检测器。噪声抑制器由 TCCR3B 寄存器的 ICNC1 位控制其使能或禁止。

使用输入捕捉功能时，当 ICF3 被置位后，应尽可能早的读取 ICR3 寄存器的值，因为下一次捕捉事件发生后 ICR3 的值将会被更新。推荐使能输入捕捉中断，在任何输入捕捉工作模式下，都不推荐在操作过程中改变计数 TOP 值。

输入捕捉到的时间标记可用来计算频率、占空比及信号的其它特征，以及为触发事件创建日志。测量外部信号的占空比时要求每次捕捉后都要改变触发沿，因此读取 ICR3 值以后须尽快改变触发的信号边沿。

Input Capture Mode

Input capture mode is used to capture external events, it gives external events a time flag as record. This mode is used in previous counter modes but not include in waveform generation mode when using ICR3 as counting TOP value.

Trigger signal of external events is input via pin ICP3, it can also be realized via analog comparator unit. If logic change level on pin ICP3 changes, or output AC0 level change of analog comparator changes, input capture is triggered when the just mentioned level change is captured by input capture unit. At this moment, 16-bit counting value TCNT3 data is copied to input capture register ICR3, meanwhile input capture flag ICF3 is set. If setting ICIE1 to logic one, input capture flag will generate input capture interrupt.

By setting analog compare input capture control bit（ACIC）of analog compare control and status register (ACSR), ICP3 or AC0, input capture trigger source, can be selected. Noted that one change of trigger source would result in one input capture, so after the change, must clear ICF3 to avoid a mistake.

Input capture signal is send to edge inspector after going through an optional noise suppression, to see if the inspected edge would satisfy trigger condition or not, based on configuration of input capture selection control bit ICES1. Noise surpression is a very simple digital filter; it samples the input signal for 4 times. Its output is sent to edge inspector only if all 4 sampling value are the same. Noice surpression is set to enable or disable via ICNC1 bit of TCCR3B register.

When using input capture, after ICF3 is set, ICR3 register value should be read ASAP, because ICR3 value will be updated after next capture event. It is recommend to enable input capture interrupt, and in whatever input capture mode, we suggest not to chang TOP value during operation.

Timing flag captured by input can be used to calculate frequency, duty cycle and other characteristic of signal, it can be also used to create log for trigger events. For measuring duty cycle of external signal, it is required to change trigger edge after each capture, so changing edge of triggered signal should be ASAP after reading ICR3 value.

PWM 输出的自动关闭与重启

当设置 TCCR3C 寄存器的 DOC3x 位为高时， PWM 输出的自动关闭功能会被使能，满足触发条件时，硬件会清零相应的 COM3x 位，将 PWM 输出信号 OC3x 与其输出引脚断开，切换成通用 IO 输出，实现 PWM 输出的自动关闭。此时，输出引脚的状态可由通用 IO 口的输出来控制。

PWM 输出的自动关闭被使能后，还需要设置其触发条件，由 TCCR3D 寄存器的 DSX3n 位来选择触发源。触发源有模拟比较器中断，外部中断，引

脚电平变化中断以及定时器溢出中断，具体情形请参考 TCCR3D 寄存器描述。当某个或某些触发源被选用作为触发条件后，在这些中断标志位被置位的同时，硬件会清零 COM3x 位来关闭 PWM 的输出。

当发生了触发事件关闭 PWM 输出后，定时器模块没有相应的中断标志位，软件需要通过读取触发源的中断标志位来得知触发条件和触发事件。

当 PWM 输出被自动关闭而需要再次重启输出时，软件只需要重新设置 COM3x 位，来切换 OC3x 信号输出到相应的引脚上。需要注意的是，发生自动关闭后，定时器并未停止工作，OC3x 信号的状态也一直在更新。软件可在定时器发生溢出或比较匹配后，再设置 COM3x 位来输出 OC3x 信号，这样可以获得明确的 PWM 输出状态。

Auto shut down and re-start of PWM output

If DOC3x of TCCR3C register is set high, PWM output is enabled to shut down automatically. Once trigger is satisfied, hardware will clear corresponding COM3x, PWM output OC3x will be apart from output pin and switch to general IO output, by this way automate shut-down of PWM output is done. Under this situation, status of output pin is controlled via output of general IO port.
When automate shut-down of PWM output is enabled, trigger condition must be set meanwhile. Trigger source is set by DSX3n of TCCR3D register. Trigger source has analog comparator interrupt, external interrupt, pin change interrupt and timer overflow interrupt, for details please refer to section of TCCR3D register. If one or some of these trigger sources is selected and these interrupt flag is set at the same time, hardware will clear COM3x to shut down PMW output.

When trigger events occur and PWM output is shut down, and there is no interrupt flag in timer module, trigger event and trigger condition can only be known through software reads interrupt flag of trigger source.

If PWM need to re-start output after automate shut-down, it is only necessary to use software to re-set COM3x and to switch OC3x output into related pins. Note that after automate shut-down occurs, timer does not stop working, OC3x status keeps updating. Software can set COM3x to output OC3x after timer overflow or compare match, by this way accurate PMW output status can be gotten.

死区时间控制

设置 DTEN3 位为“1”时，插入死区时间的功能被使能，OC3A 和 OC3B 的输出波形将在 B 通道比较输出所产生的波形基础上插入设定的死区时间，时间的长度为 DTR3 寄存器的计数时钟数所对应的时间值。如下图所示，OC3A 和 OC3B 的死区时间插入均是以通道 B 的比较输出波形为基准。当 COM3A 和 COM3B 同为“2”或“3”时，OC3A 的波形极性与 OC3B 的波形极性相同，当 COM3A 和 COM3B 分别为“2”或“3”时，OC3A 的波形与 OC3B 的波形极性相反。

Dead time control

Setting DTEN3 to logic one, it enables of inserting dead time, OC3A and OC3B will insert designed dead time based on waveform generated by B channel compare output, length of time is the timing value corresponded to counter clock number of DTR3 register. As shown below, OC3A and OC3B will insert designed dead time based on waveform generated by B channel compare output. If COM3A and COM3B are set to 2 or 3 at the same time, waveform polarity of OC3A and OC3B are the same, if they are set to 2 or 3 individually, waveform polarity of OC3A is opposite to that of OC3B.

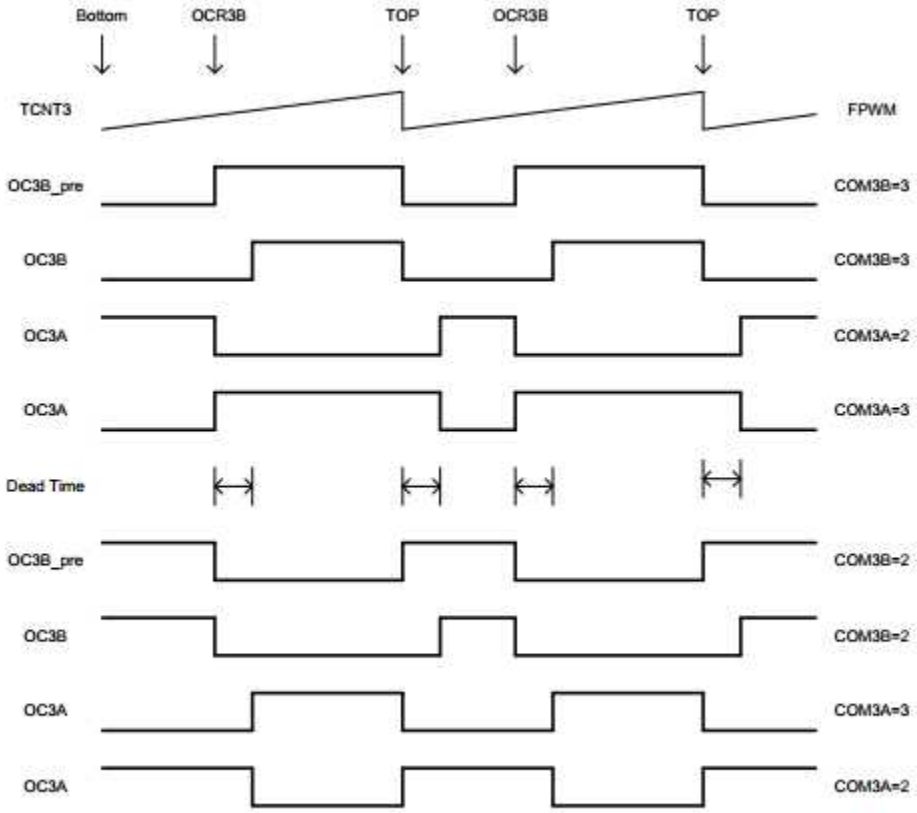


Figure 7　FPWM 模式下 TC3 死区时间控制

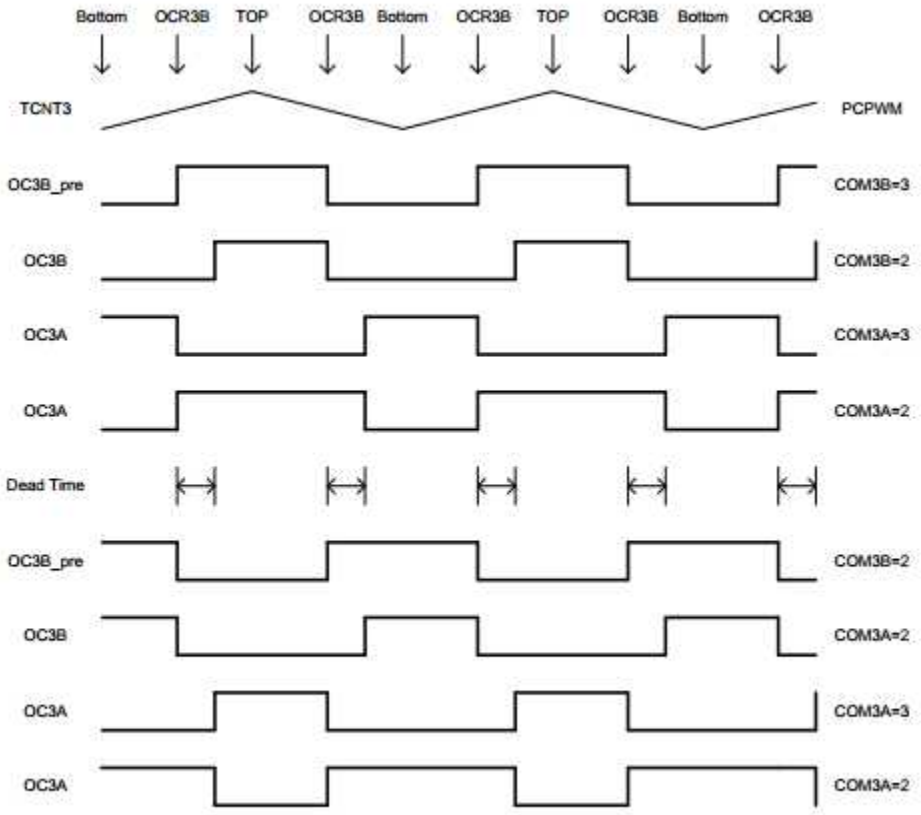


Figure 8　PCPWM 模式下 TC3 死区时间控制

设置 DTEN3 位为“0”时，插入死区时间的功能被禁止，OC3A 和 OC3B 的输出波形为各自比较输出所产生的波形。

Setting DTEN3 to logic zero, inserting dead time will be disabled. Output waveform of OC3A and OC3B are their individual compare output waveform.

Register Definition

TC3 Register List

Register	Location	Default Value	Comment
TCCR3A	0x90	0x00	TC3 control register A
TCCR3B	0x91	0x00	TC3 control register B
TCCR3C	0x92	0x00	TC3 control register C
TCCR3D	0x93	0x00	TC3 control register D
TCNT3L	0x94	0x00	TC3 counter register low byte
TCNT3H	0x95	0x00	TC3 counter register high byte
ICR3L	0x96	0x00	TC3 input capture register low byte
ICR3H	0x97	0x00	TC3 input capture register high byte
OCR3AL	0x98	0x00	TC3 output compare register A low byte
OCR3AH	0x99	0x00	TC3 output compare register A high byte
OCR3BL	0x9A	0x00	TC3 output compare register B low byte
OCR3BH	0x9B	0x00	TC3 output compare register B high byte
DTR3L	0x9C	0x00	TC3 deadtime register low byte

DTR3H	0x9D	0x00	TC3 deadtime register high byte
OCR3CL	0x9E	0x00	TC3 output compare register C low byte
OCR3CH	0x9F	0x00	TC3 output compare register C high byte
TIMSK3	0x71	0x00	Timer counter interrupt mask register
TIFR3	0x38	0x00	Timer counter interrupt flag register

TCCR3A—TC3 control register A

TCCR3A—TC3 control register A								
Address: 0x99					Default value: 0x00			
Bit	7	6	5	4	3	2	1	0
	COM3A1	COM3A0	COM3B1	COM3B0	COM3C1	COM3C0	WGM31	WGM30
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Comment
7	COM3A1	Compare match output A mode control high COM3A1 and COM3A0 combine to consist of COM3A[1:0] to control 0C3A output waveform. If 1st or 2nd bit of COM3A are set, output compare waveform takes up 0C3A pin, however whose data direction regester must be set to high. In different mode, C0M3A control compare waveform differently, for details referring to list on compare output control.
6	COM3A0	Compare match output A mode control low COM3A1 and COM3A0 combine to consist of COM3A[1:0] to control 0C3A output waveform. If the 1st or 2nd bit of COM3A is set, compare waveform takes up 0C3A pin, however whose data direction regester must be set to high. In different mode, C0M3A controls compare waveform differently, for details referring to list on compare output control.
5	COM3B1	Compare match output B mode control high COM3B1 and COM3B0 combine to consist of COM3B[1:0] to control 0C3B output waveform. If 1st bit or 2nd bit of COM3B bits are set, compare waveform takes up 0C3B pin, however whose data direction regester must be set to high. In different mode, C0M3B control compare waveform differently, for details referring to list on compare output control.
4	COM3B0	Compare match output B mode control low COM3B1 and COM3B0 combine to consist of COM3B[1:0] to control 0C3B output waveform. If 1st bit or 2nd bit of COM3B bits are set, compare waveform takes up 0C3B pin, however whose data direction regester must be set to high. In different mode, C0M3B control compare waveform differently, for details referring to list on compare output control.
3	COM3C1	Compare match output C mode control high COM3C1 and COM3C0 combine to consist of COM3C[1:0] to control 0C3C output waveform. If 1st bit or 2nd bit of COM3C bits are set, compare waveform takes up 0C3C pin, however whose data direction regester must be set to high. In different mode, C0M3C control compare waveform differently, for details referring to list on compare output control.
2	COM3C0	Compare match output C mode control low

		COM3C1 and COM3C0 combine to consist of COM3C[1:0] to control 0C3C output waveform. If 1st bit or 2nd bit of COM3C bits are set, compare waveform takes up 0C3C pin, however whose data direction regester must be set to high. In different mode, C0M3C control compare waveform differently, for details referring to list on compare output control.
1	WGM31	Waveform generation control sub-low WGM31 and WGM33, WGM32, WGM30 generate waveform generation control WGM3[3:0] together, couting method and waveform generation method of control counter is referred to list on waveform generation mode.
0	WGM30	Waveform generation control lowest WGM30 and WGM33, WGM32, WGM31 generate waveform generation control WGM3[3:0] together, couting method and waveform generation method of control counter is referred to list on waveform generation mode.

Below list show how compare output mode controls output compare waveform in non- PWM mode (normal mode and CTC mode)

COM3x[1:0]	Comments
0	0C3x disconnect, general IO port operation
1	Toggle 0C3x signal at compare match
2	Clear 0C3x signal at compare match
3	Set 0C3x signal at compare match

Below list show how compare output mode control output compare waveform in fast PWM mode

COM3x[1:0]	Comments
0	0C3x disconnect, general IO port operation
1	When WGM3 is set to 15, OC3A signal is toggled at compare match, OC3B is disconnected. When WGM3 is other value, 0C3x disconnect, general IO port operation
2	Clear 0C3x signal at compare match, set 0C3x at max. value match
3	Set 0C3x signal at compare match, clear 0C3x at max. value match

Below list show how compare output mode controls output compare waveform in Phase Correct Mode

COM3x[1:0]	Comments
0	0C3x disconnect, general IO port operation
1	When WGM3 is set to 9 or 11, OC3A signal is toggled at compare match, OC3B is disconnected. When WGM3 is other value, 0C3x disconnect, general IO port operation
2	0C3x is cleared at compare match in up-counting; 0C3x is set at compare match in down-counting.
3	0C3x is set at compare match in up-counting; 0C3x is cleared at compare match in down-counting.

TCCR3B --TC3 control register B

TCCR3B --TC3 control register B								
Address: 0x91					Address: 0x91			
Bit	7	6	5	4	3	2	1	0
	ICNC3	ICES3	-	WGM33	WGM32	CS32	CS31	CS30

R/W	R/W	R/W	-	R/W	R/W	R/W	R/W	R/W
Bit	Name	Comment						
7	ICNC3	Input capture noise supression enable control If setting ICNC3 to logic one, input capture noise surpression is enabled, than input of external pin ICP3 is filtered, input signal is only valid if 4 sampling value in a row are the same, becaues of this funtion input capture is delayed for 4 clock cycles. If setting ICNC3 to logic zero, input capture noise supression is disabled, than input of external pin ICP3 is activated directly.						
6	ICES3	Input capture trigger edge selection control If setting ICES3 to logic one, rising edge of selected pin change level will trigger input capture If setting ICES3 to logic zero, falling edge of selected pin change level will trigger input capture. When an event is captured, counter value will be copied to ICR3 register, meanwhile input capture flag ICF3 is set. If interrupt is enabled, input capture interrupt is generated.						
5	-	reserved						
4	WGM33	Waveform generation mode control high WGM33 and WGM32, WGM31,WGM30 together consist of waveform generation control WGM3[3:0], counting method and waveform generation method of control counter is referred to list on waveform generation mode.						
3	WGM32	Waveform generation mode control sub-high WGM32 and WGM33, WGM31, WGM30 together consist of waveform generation control WGM3[3:0], counting method and waveform generation method of control counter is referred to list on waveform generation mode.						
3	CS32	Clock selection control high Used to select clock source of timer/counter 3						
1	CS31	Clock selection control middle Used to select clock source of timer/counter 3						
0	CS30	Clock selection control low Used to select clock source of timer/counter 3						
		CS3[2:0]	comments					
		0	No clock source, stop couting					
		1	clksys					
		2	Clksys/8, from frequency prescaler					
		3	Clksys/64, from frequency prescaler					
		4	Clksys/256, from frequency prescaler					
		5	Clksys/1024, from frequency prescaler					
		6	External clock T3 pin, falling edge trigger					
		7	External clock T3 pin, rising edge trigger					

Below list shows waveform generaton mode control

Table 5 Waveform genearation mode control

WGM3[3:0]	Working Mode	TOP Value	Updating OCR1A time	Set TOV3 time
0	Normal	0xFFFF	Immediately	MAX

1	8-bit PCPWM	0x00FF	TOP	BOTTOM
2	9-bit PCPWM	0x01FF	TOP	BOTTOM
3	10-bit PCPWM	0x03FF	TOP	BOTTOM
4	CTC	OCR3A	Immediately	MAX
5	8-bit FPWM	0x00FF	BOTTOM	TOP
6	9-bit FPWM	0x01FF	BOTTOM	TOP
7	10-bit FPWM	0x03FF	BOTTOM	TOP
8	PFCPWM	ICR3	BOTTOM	BOTTOM
9	PFCPWM	ICR3A	BOTTOM	BOTTOM
10	PCPWM	ICR3	TOP	BOTTOM
11	PCPWM	OCR3A	TOP	BOTTOM
12	CTC	ICR3	Immediately	MAX
13	reserved	-	-	-
14	FPWM	ICR3	TOP	TOP
15	FPWM	OCR3A	TOP	TOP

TCCR3C—TC3 control register C

TCCR3C—TC3 control register C								
Address: 0x92					Default value: 0x00			
Bit	7	6	5	4	3	2	1	0
	FOC3A	FOC3B	DOC3B	DOC3A	DTEN3	-	DOC3C	FOC3C
R/W	R/W	R/W	-	-	-	-	-	-
Bit	Name	Comment						
7	FOC3A	Force output compare A If not in PWM mode, writing logic one to force output compare bit F0C3A can result in compare match. Force compare match neither sets 0CF3A flag, nor reload or clear timer, but will result in output pin 0C3A to be updated according to C0M3A configuration, which seems a real compare match occurs. In PWM mode, TCCR3A register should be cleared when writing it. Read return value of F0C3A till it is zero.						
6	FOC3B	Force output compare B If not in PWM mode, writing logic one to force output compare F0C3B can result in compare match. Force compare match neither sets 0CF3B flag, nor reload or clear timer, but will result in output pin 0C3B to be updated according to C0M3B configuration, which seems a real compare match occur. In PWM mode, TCCR3A register should be cleared when writing it. Read return value of F0C3B till it is zero.						
5	DOC3B	prohibit output compare B enable control bit When DOC3B is set to high, prohibit output compare B of hardware is enabled, COM3B bit is cleared and output pin OC3B is disconnected once conditions of prohibit output is satisfied, then this pin is changed to general IO operation. When DOC3B is set to low, prohibit output compare B of hardware is invalid.						
4	DOC3A	prohibit output compare A enable control bit When DOC3A is set to high, prohibit output compare A of hardware is enabled, COM3A bit is cleared and output pin OC3A is disconnected once conditions of prohibit output is						

		satisfied, then this pin is changed to general IO operation. When DOC3A is set to low, prohibit output compare A of hardware is invalid.
3	DTEN3	Dead time enable control bit When DTEN3 bit is set to high, dead time is enabled, OC3A and OC3B become complementary output, dead time insert is designed according to DTR3L and DTR3H. When DTEN3 bit is set to low, dead time is disabled. Both OC3A and OC3B are single output.
2	-	
1	DOC3C	prohibit output compare C enable control bit When DOC3C is set to high, prohibit output compare C of hardware is enabled, COM3C bit is cleared and output pin OC3C is disconnected once conditions of prohibit output is satisfied, then this pin is changed to general IO operation. When DOC3C is set to low, prohibit output compare C of hardware is invalid.
0	FOC3C	Force output compare C If not in PWM mode, writing logic one to force output compare F0C3C can result in compare match. Force compare match neither sets 0CF3C flag, nor reload or clear timer, but will result in output pin 0C3C to be updated according to C0M3C configuration, which seems a real compare match occur. In PWM mode, TCCR3A register should be cleared when writing it. Read return value of F0C3C till it is zero.

TCCR3D—TC3 control register D

TCCR3D—TC3 control register D								
Address: 0x93					Default value: 0x00			
Bit	7	6	5	4	3	2	1	0
	DSX37	DSX36	DSX35	DSX34	-	-	DSX31	DSX30
R/W	R/W	R/W	R/W	R/W	-	-	R/W	R/W

Bit	Name	Comment
7	DSX37	TC3 trigger source selection control enable 7th bit When setting DSX37 to logic one, TC0 overflow, used as trigger source to shut down output compare signal waveform 0C3x, is enable. When setting DOC3x to logic one, rising edge of interrupt flag register bit of selected trigger source will automatically shut down 0C3x wave output When setting DSX37 to logic zero, TC0 overflow, used as trigger source to shut down output compare signal waveform 0C3x, is disabled.
6	DSX36	TC3 trigger source selection control enable 6th bit When setting DSX36 to logic one, TC2 overflow, used as trigger source to shut down output compare signal waveform 0C3x, is enable. When setting DOC3x to logic one, rising edge of interrupt flag register bit of selected trigger source will automatically shut down 0C3x wave output When setting DSX36 to logic zero, TC2 overflow, used as trigger source to shut down output compare signal waveform 0C3x, is disabled.
5	DSX35	TC3 trigger source selection control enable 5th bit

		When setting DSX35 to logic one, pin change level 1, used as trigger source to shut down output compare signal waveform 0C3x, is enable. When setting DOC3x to logic one, rising edge of interrupt flag register bit of selected trigger source will automatically shut down 0C3x wave output When setting DSX35 to logic zero, pin change level 1, used as trigger source to shut down output compare signal waveform 0C3x, is disabled.
4	DSX34	TC3 trigger source selection control enable 4th bit When setting DSX34 to logic one, external interrupt 1, used as trigger source to shut down output compare signal waveform 0C3x, is enabled. When setting DOC3x to logic one, rising edge of interrupt flag register bit of selected trigger source will automatically shut down 0C3c wave output When setting DSX34 to logic zero, external interrupt 1, used as trigger source to shut down output compare signal waveform 0C3c, is disabled.
3:2	-	reserved
1	DSX31	TC3 trigger source selection control enable 1st bit When setting DSX31 to logic one, analog comparator 1, used as trigger source to shut down output compare signal waveform 0C3x, is enabled. When setting DOC3x to logic one, rising edge of interrupt flag register bit of selected trigger source will automatically shut down 0C3x wave output When setting DSX31 to logic zero, analog comparator 1, used as trigger source to shut down output compare signal waveform 0C3x, is disabled.
0	DSX30	TC3 trigger source selection control enable 0 bit When setting DSX30 to logic one, analog comparator 0, used as trigger source to shut down output compare signal waveform 0C3x, is enabled. When setting DOC3x to logic one, rising edge of interrupt flag register bit of selected trigger source will automatically shut down 0C3x wave output When setting DSX30 to logic zero, analog comparator 0, used as trigger source to shut down output compare signal waveform 0C3x, is disabled.

Below list shows selection control of waveform output trigger source

Shut-down OC3x selection control of waveform output trigger source

DOC3x	DSX3n=1	Trigger Source	Comment
0	-	-	DOC3x bit is logic zero, trigger source to shut down waveform output is disabled.
1	0	Analog comparator 0	Rising edge of ACIF0 will shut down OC3x waveform output
1	1	Analog comparator 1	Rising edge of ACIF1 will shut down OC3x waveform output
1	4	External interrupt 1	Rising edge of INTF1 will shut down OC3x waveform output
1	5	Pin change level 1	Rising edge of PCIF1 will shut down OC3x waveform output
1	6	TC2 overflow	Rising edge of TOV2 will shut down OC3x waveform output

1	7	TC0 overflow	Rising edge of TOV0 will shut down OC3x waveform output

Note that:

2) DSX3n=1, indicate that the N bit of TCCR1D is logic one, each register can be set at the same time.

TCNT3L—TC3 counter register low byte

TCNT3L—TC3 counter register low byte								
Address: 0x44					Default Value: 0x00			
Bit	7	6	5	4	3	2	1	0
	TCNT3L7	TCNT3L6	TCNT3L5	TCNT3L4	TCNT3L3	TCNT3L2	TCNT3L1	TCNT3L0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	Name	Comments						
7:0	TCNT3L	TC3 Counting value low byte TCNT3H and TCNT3L is combined to consist of TCNT3, than via TCNT3 register 16-bit counting value of counter can be access to for read-write directly. It is 2 times operations to read and write 16-bit register. If write to 16-bit TCNT3, must write TCNT3H firstly. If read 16-bit TCNT3, must read TCNT3L first. Write operation of CPU to TCNT3 register will prevent compare match from happening in the next timer clock cycle, even timer stop working, this allows that value of initialized TCNT3 register is consistent with 0CR3x value, so interrupt is not triggered. If written value to TCNT3 equals or overrides OCR3x, compare match will loss, which results in incorrect waveform generation. Timer stops couting if no clock source is selected, however CPU can still be access to TCNT3.CPU writing to counter has high priority than clear or minus/plus operation.							

TCNT3H—TC3 counter register high byte

TCNT3H—TC3 counter register high byte								
Address: 0x95					Default Value: 0x00			
Bit	7	6	5	4	3	2	1	0
	TCNT3H7	TCNT3H6	TCNT3H5	TCNT3H4	TCNT3H3	TCNT3H2	TCNT3H1	TCNT3H0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	Name	Comments						
7:0	TCNT3H	TC3 Counting value high byte TCNT3H and TCNT3L is combined to consist of TCNT3. It is 2 times operations to read and write 16-bit register. If write to 16-bit TCNT3, must write TCNT3H firstly. If read 16-bit TCNT3, must read TCNT3L first. Write operation of CPU to TCNT3 register will prevent compare match from happening in the next timer clock cycle, even timer stop working, this allows that value of initialized TCNT3 register is consistent with 0CR3x value, so interrupt is not triggered. If written value to TCNT3 equals or overrides OCR3x, compare match will loss, which							

		results in incorrect waveform generation. Timer stops couting if no clock source is selected, however CPU can still be access to TCNT3.CPU writing to counter has high priority than clear or minus/plus operation.

ICR3L—TC3 capture register low byte

ICR3L—TC3 capture register low byte								
Address: 0x96					Default Value: 0x00			
Bit	7	6	5	4	3	2	1	0
	ICR3L7	ICR3L6	ICR3L5	ICR3L4	ICR3L3	ICR3L2	ICR3L1	ICR3L0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	Name	Comments						
7:0	ICR3L	TC3 input capture value low byte ICR3H and ICR3L are combined to consist of 16-bit ICR3. It is 2 times operations to read and write 16-bit register. If write to 16-bit ICR3, must write ICR3H firstly. If read 16-bit ICR3, must read ICR3L first. When input capture is triggered, counter value TCNT3 is updated and copied to ICR3 register. ICR3 register is also used to define TOP value of counter						

ICR3H—TC3 capture register high byte

ICR3H—TC3 capture register high byte								
Address: 0x97					Default Value: 0x00			
Bit	7	6	5	4	3	2	1	0
	ICR3H7	ICR3H6	ICR3H5	ICR3H4	ICR3H3	ICR3H2	ICR3H1	ICR3H0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	Name	Comments						
7:0	ICR3H	TC3 input capture value high byte ICR3H and ICR3L are combined to consist of 16-bit ICR3. It is 2 times operations to read and write 16-bit register. If write to 16-bit ICR3, must write ICR3L firstly. If read 16-bit ICR3, must read ICR3L first. When input capture is triggered, counter value TCNT3 is updated and copied to ICR3 register. ICR3 register is also used to define TOP value of counter						

OCR3AL—TC3 output compare register A low byte

OCR3AL—TC3 output compare register A low bytee								
Address: 0x98					Default Value: 0x00			
Bit	7	6	5	4	3	2	1	0
	OCR3AL7	OCR3AL6	OCR3AL5	OCR3AL4	OCR3AL3	OCR3AL2	OCR3AL1	OCR3AL0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	Name	Comments						
7:0	OCR3AL	Output compare register A low byte OCR3AL and OCR3AH are combined to consist of 16-bit OCR3A. It is 2 times operations to read and write 16-bit register. If write to 16-bit OCR3A, must write OCR3AH firstly. If read 16-bit OCR3A, must read OCR3AL first.						

		When in PWM mode, OCR3A register will use dual buffer register. While in normal mode and match on clear mode, dual buffer is disabled. Updating OCR3A register and counting TOP or BOTTOM time are synchronized by dual buffer to prevent asysmetric PWM pulse and eliminate intervene pulse. When using dual buffer, CPU is access to COR3A buffer register. If dual buffer is disabled, CPU is acces to OCR3A itself.

OCR3AH—TC3 output compare register A high byte

OCR3AH—TC3 output compare register A high byte								
Address: 0x99					Default Value: 0x00			
Bit	7	6	5	4	3	2	1	0
	OCR3AH7	OCR3AH6	OCR3AH5	OCR3AH4	OCR3AH3	OCR3AH2	OCR3AH1	OCR3AH0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	Name	Comments						
7:0	OCR3AL	Output compare register A high byte OCR3AL and OCR3AH are combined to consist of 16-bit OCR3A. It is 2 times operations to read and write 16-bit register. If write to 16-bit OCR3A, must write OCR3AH firstly. If read 16-bit OCR3A, must read OCR3AL first. OCR3A compares with counter value TCNT3 continuously. Compare match is used to generate output compare interrupt, or generate waveform in OC3A pin. When in PWM mode, OCR3A register will use dual buffer register. While in normal mode and match on clear mode, dual buffer is disabled. Updating OCR3A register and counting TOP or BOTTOM time are synchronized by dual buffer to prevent asysmetric PWM pulse and eliminate intervene pulse. When using dual buffer, CPU is access to OCR3A buffer register. If dual buffer is disabled, CPU is acces to OCR3A itself.							

OCR3BL—TC3 output compare register B low byte

OCR3BL—TC3 output compare register B low byte								
Address: 0x9A					Default Value: 0x00			
Bit	7	6	5	4	3	2	1	0
	OCR3BL7	OCR3BL6	OCR3BL5	OCR3BL4	OCR3BL3	OCR3BL2	OCR3BL1	OCR3BL0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	Name	Comments						
7:0	OCR3AL	Output compare register B low byte OCR3BL and OCR3BH are combined to consist of 16-bit OCR3B. It is 2 times operations to read and write 16-bit register. If write to 16-bit OCR3B, must write OCR3BH firstly. If read 16-bit OCR3B, must read OCR3BL first. OCR3B compares with counter value TCNT3 continuously. Compare match is used to generate output compare interrupt, or generate waveform in OC3B pin. When in PWM mode, OCR3B register will use dual buffer register. While in normal mode and match on clear mode, dual buffer is disabled. Updating OCR3B register and counting TOP or BOTTOM time are synchronized by dual buffer to prevent							

		asysmetric PWM pulse and eliminate intervene pulse. When using dual buffer, CPU is access to OCR3B buffer register. If dual buffer is disabled, CPU is acces to OCR3B itself.

OCR3BH—TC3 output compare register B high byte

OCR3BH—TC3 output compare register B high byte								
Address: 0x9B					Default Value: 0x00			
Bit	7	6	5	4	3	2	1	0
	OCR3BH7	OCR3BH6	OCR3BH5	OCR3BH4	OCR3BH3	OCR3BH2	OCR3BH1	OCR3BH0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	Name	Comments						
7:0	OCR3BH	Output compare register B high byte OCR3BL and OCR3BH are combined to consist of 16-bit OCR3B. It is 2 times operations to read and write 16-bit register. If write to 16-bit OCR3B, must write OCR3BH firstly. If read 16-bit OCR3B, must read OCR3BL first. OCR3B compares with counter value TCNT3 continuously. Compare match is used to generate output compare interrupt, or generate waveform in OC3B pin. When in PWM mode, OCR3B register will use dual buffer register. While in normal mode and match on clear mode, dual buffer is disabled. Updating OCR3B register and counting TOP or BOTTOM time are synchronized by dual buffer to prevent asysmetric PWM pulse and eliminate intervene pulse. When using dual buffer, CPU is access to OCR3B buffer register. If dual buffer is disabled, CPU is acces to OCR3B itself.							

OCR3CL--TC3 output compare register C low byte

OCR3CL--TC3 output compare register C low byte								
Address: 0x9E					Default Value: 0x00			
Bit	7	6	5	4	3	2	1	0
	OCR3CL7	OCR3CL6	OCR3CL5	OCR3CL4	OCR3CL3	OCR3CL2	OCR3CL1	OCR3CL0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	Name	Comments						
7:0	OCR3CL	Output compare register C low byte OCR3CL and OCR3CH are combined to consist of 16-bit OCR3C. It is 2 times operations to read and write 16-bit register. If write to 16-bit OCR3C, must write OCR3CH firstly. If read 16-bit OCR3C, must read OCR3CL first. OCR3C compares with counter value TCNT3 continuously. Compare match is used to generate output compare interrupt, or generate waveform in OC3C pin. When in PWM mode, OCR3C register will use dual buffer register. While in normal mode and match on clear mode, dual buffer is disabled. Updating OCR3C register and counting TOP or BOTTOM time are synchronized by dual buffer to prevent asysmetric PWM pulse and eliminate intervene pulse. When using dual buffer, CPU is access to OCR3C buffer register. If dual buffer is disabled, CPU is acces to OCR3C itself.							

OCR3CH--TC3 output compare register C high byte

OCR3CH--TC3 output compare register C high byte								
Address: 0x9F					Default Value: 0x00			
Bit	7	6	5	4	3	2	1	0
	OCR3CH7	OCR3CH6	OCR3CH5	OCR3CH4	OCR3CH3	OCR3CH2	OCR3CH1	OCR3CH0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	Name	Comments						
7:0	OCR3CH	Output compare register C high byte OCR3CL and OCR3CH are combined to consist of 16-bit OCR3C. It is 2 times operations to read and write 16-bit register. If write to 16-bit OCR3C, must write OCR3CH firstly. If read 16-bit OCR3C, must read OCR3CL first. OCR3C compares with counter value TCNT3 continuously. Compare match is used to generate output compare interrupt, or generate waveform in OC3C pin. When in PWM mode, OCR3C register will use dual buffer register. While in normal mode and match on clear mode, dual buffer is disabled. Updating OCR3C register and counting TOP or BOTTOM time are synchronized by dual buffer to prevent asysmetric PWM pulse and eliminate intervene pulse. When using dual buffer, CPU is access to OCR3C buffer register. If dual buffer is disabled, CPU is acces to OCR3C itself.							

DTR3L--TC3 dead time register low byte

DTR3L--TC3 dead time register low byte								
Address: 0x9C					Default Value: 0x00			
Bit	7	6	5	4	3	2	1	0
	DTR3L7	DTR3L6	DTR3L5	DTR3L4	DTR3L3	DTR3L2	DTR3L1	DTR3L0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	Name	Comments						
7:0	DTR3L	Dead time register low byte When DTEN3 bit is set to high, OC3A and OC3B are complementary output, dead time inserted into OC3A output is decided by DTR3L counter clocks						

DTR3H--TC3 dead time register high byte

DTR3L--TC3 dead time register high byte								
Address: 0x9D					Default Value: 0x00			
Bit	7	6	5	4	3	2	1	0
	DTR3H7	DTR3H6	DTR3H5	DTR3H4	DTR3H3	DTR3H2	DTR3H1	DTR3H0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	Name	Comments						
7:0	DTR3H	Dead time register high byte When DTEN3 bit is set to high, OC3A and OC3B are complementary output, dead time inserted into OC3A output is decided by DTR3H counter clocks						

TIMSK3-TC3 interrupt mask register

DTR3L--TC3 dead time register high byte								
Address: 0x71					Default Value: 0x00			
Bit	7	6	5	4	3	2	1	0
	-	-	ICIE3	-	OCIE3C	OCIE3B	OCIE3A	TOIE3
R/W	-	-	R/W	-	R/W	R/W	R/W	R/W
Bit	Name	Comments						
7:6	-	reserved						
5	ICIE3	TC3 input capture interrupt enable control bit When ICIE3 is logic one, and globe interrupt is set, TC3 input capture interrupt is enabled. When input capture is triggered, that is to say ICF3 flag of TIFR3 is set, interrupt happens. When ICIE3 bit is logic zero, TC3 input capture interrupt is disabled.						
4	-	reserved						
3	OCIE3C	TC3 output compare C match interrupt enable control bit When ICIE3C is logic one, and globe interrupt is set, TC3 output compare C match interrupt is enabled. If compare match is actived, that is to say OCF3C flag of TIFR3 is set, interrupt happens. When OCIE3C bit is logic zero, TC3 output compare C match interrupt is disabled.						
2	OCIE3B	TC3 output compare B match interrupt enable control bit When ICIE3B is logic one, and globe interrupt is set, TC3 output compare B match interrupt is enabled. If compare match is actived, that is to say OCF3B flag of TIFR3 is set, interrupt happens. When OCIE3B bit is logic zero, TC3 output compare B match interrupt is disabled.						
1	OCIE3A	TC3 output compare A match interrupt enable control bit When ICIE3A is logic one, and globe interrupt is set, TC3 output compare A match interrupt is enabled. If compare match is actived, that is to say OCF3A flag of TIFR3 is set, interrupt happens. When OCIE3A bit is logic zero, TC3 output compare A match interrupt is disabled.						
0	TOIE3	TC3 overflow interrupt enable bit When TOIE3 is logic one, and globe interrupt is set, TC3 overflow interrupt is enabled. When TC3 overflows, that is to day TOV3 bit of TIFR3 is set, interrupt is happening. When TOIE3 bit is logic zero, TC3 overflow interrupt is disabled.						

TIFR3--TC3 interrupt flag register

TIFR3--TC3 interrupt flag register								
Address: 0x38					Default Value: 0x00			
Bit	7	6	5	4	3	2	1	0
	-	-	ICF3	-	-	OCF3B	OCF3A	TOV3
R/W	-	-	R/W	-	-	R/W	R/W	R/W
Bit	Name	Comments						
7:6	-	reserved						
5	ICF3	Input capture flag bit When input capture event happens, ICF3 flag is set. When ICR3 is considered as TOP value, and counter value reaches TOP, ICF3 flag is set.						

		If ICIE1 is logic one and globe interrupt flag is set, input capture interrupt is generated. ICF3 flag is not cleared automatically, software has to write logic one to ICF3 bit to clear.
4	-	reserved
3	OCF3C	Output compare C match flag bit When TCNT3 is the same as OCR3C, compare unit sends match singal and set compare flag OCF3C. At this moment if output compare interrupt enable OCIE3C is logic one and globe interrupt flag is set, than output compare interrupt is generated. OCF3 flag bit is not cleared automatically, software has to write logic one to OCF3C bit to clear.
2	OCF3B	Output compare B match flag bit When TCNT3 is the same as OCR3B, compare unit sends match singal and set compare flag OCF3B. At this moment if output compare interrupt enable OCIE3B is logic one and globe interrupt flag is set, than output compare interrupt is generated. OCF3B flag bit is not cleared automatically, software has to write logic one to OCF3B bit to clear.
1	OCF3A	Output compare A match flag bit When TCNT3 is the same as OCR3A, compare unit sends match singal and set compare flag OCF3A. At this moment if output compare interrupt enable OCIE3A is logic one and globe interrupt flag is set, than output compare interrupt is generated. OCF3A flag bit is not cleared automatically, software has to write logic one to OCF3A bit to clear.
0	TOV3	Overflow flag bit When counter overflows, overflow flag TOV3 is set. At this moment if overflow interrupt enable TOIE3 is logic one and globe interrupt flag is set, overflow interrupt is happening. TOV3 flag bit is not cleared automatically, software has to write logic one to TOV3 bit to clear.

同步串行外设接口 (SPI)

- 全双工，三线同步数据传输
- 主机或从机操作
- 最低位或最高位优先传输
- 7 种可编程的比特率
- 发送结束中断标志
- 写入冲突标志保护机制
- 可从闲置模式唤醒
- 主机操作时具有倍速模式
- 支持主机双线输入模式
- 输入/输出均有 4 个缓存寄存器

Synchronous Serial Peripheral Interface(SPI)

- Full-duplex, Three-wire Synchronous Data Transfer
- Master or Slave Operation

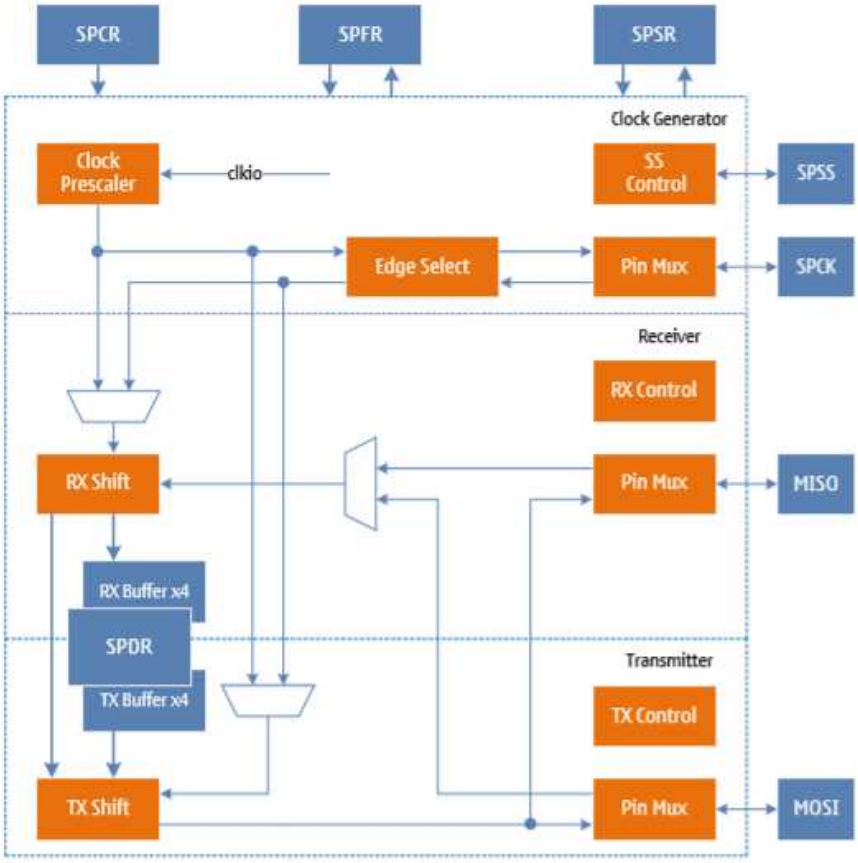
- LSB First or MSB First Data Transfer
- Seven Programmable Bit Rates
- End of Transmission Interrupt Flag
- Write Collision Flag Protection
- Wake-up from Idle Mode
- Double Speed on Master SPI Mode
- Supporting master 2-wire intput mode
- Input/output both have 4 buffer register.

综述

SPI 主要包括三个部分：时钟预分频器，时钟检测器，从机选择检测器，发送器和接收器。

Overview

SPI mainly has 3 parts: clock prescaler, clock inspector, slave selection inspector, sender and receiver.



SPI 结构图

控制和状态寄存器由这三个部分共享。时钟预分频器只工作在主机操作模式下，由比特 率控制位来选择分频系数，从而产生相应的分频时钟，输出到 SPCK 引脚上。时钟检测器只 工作在从机操作模式下，检测从 SPCK 引脚上输入的时钟沿，根据 SPI 的数据传输模式对发 送和接收移位寄存器进行移位操作。从机选择检测器对从机选择信号 SPSS 进行检测，得到传输的状态来控制发送器和接收器的操作。发送器由一个移位寄存器和发送控制逻辑组成。 接收器由一个移位寄存器，四个接收缓冲器和接收控制逻辑组成。

Control and status registers are shared by these 3 parts. Timer prescaler works only in master SPI mode, prescaler factor is selected by byte rate control bit to generate prescaler clock and output to SPCK pin; Clock inspector works only in slave SPI mode and inspects clock edge input from SPCK pin, as per SPI data transmission mode to make shift operation to send and receive shift registers; Slave selection inspector

make inspection on slave selection signal SPSS, transmission status is used to control operations of receiver and sender; Sender consists of a shift register and send control logic; Receiver consists of a shift register, 4 receiver buffer and receive control logic.

时钟产生

时钟产生逻辑分为主机时钟预分频器和从机时钟检测器，分别工作在主机操作和从机操 作模式下。时钟预分频器由比特率控制位和倍速控制位来选择分频系数，产生相应的分频时 钟（共有 7 种可选的分频系数，详细信息见寄存器描述），输出到 SPCK 引脚为通信提供时 钟，同时为内部发送和接收移位寄存器提供移位时钟。时钟检测器对输入时钟 SPCK 进行边 沿检测，根据 SPI 的数据传输模式对发送器和接收器进行移位操作。为保证对时钟信号的正 确采样，SPCK 时钟的高电平和低电平的宽度均须大于 2 个系统时钟周期。

Clock Generation

Clock generation logic consists of master timer prescaler and slave clock inspector, which works individually in master and slave operation mode. Via byte rate control bit and double-speed control bit, clock prescaler decides prescaler factor to generate prescaler (in total 7 prescaler factors for option, for details referring to register definition) which are output to SPCK pin and supply clock for communiction and supply shift clock for internal send and receive shift register; Clock inspector tests edge of input clock SPCK. As per SPI data transmision mode to make shift operations to sender and receiver. To ensure correct sampling of clock signal, both high and low SPCK clock have to 2 system clock cycle wider.

发送和接收

SPI 模块在单线模式下支持同时发送和接收，在双线模式下只支持主机双线接收。

Send and Receive

In single-wire, SPI modular support sending and receiving synchronously, while in 2-wire only mater 2-wire receive is supported.

单线发送和接收

SPI 的主机将需要通信的从机选择信号 SPSS 拉低，即可启动一次传输过程。主机和从机 将需要传输的数据准备好，主机在时钟信号 SPCK 上产生时钟脉冲以交换数据，主机的数据 从 MOSI 移出，从 MISO 移入，从机的数据从 MISO 移出，从 MOSI 移入，交换完数据后主机 拉高 SPSS 信号即可完成通信

当配置为主机时，SPI 模块并不控制 SPSS 引脚，必须由用户软件来处理。软件拉低 SPSS 引脚，选择要通信的从机，启动传输。软件将需要传输的数据写入 SPDR 寄存器即会启动时 钟发生器，硬件产生通信的时钟，并把 8 位数据移出给从机，同时把从机的数据移入。移位 一个字节的数据后，停止时钟发生器，并置位传输完成标志 SPIF。软件可再次写入数据到 SPDR 寄存器来继续传输下一个字节，也可以拉高 SPSS 信号来结束当前传输。最后进来的数 据将保存在接收缓冲器中

当配置为从机时，只要 SPSS 信号一直为高，SPI 模块将保持睡眠状态，并保持 MISO 引 脚为三态。这时软件可更新 SPDR 寄存器的内容。即使此时 SPCK 引脚上有输入时钟脉冲， SPDR 的数据也不会被移出，直至 SPSS 信号被拉低。当一个字节的数据传输完成之后，硬件 置位传输完成标志 SPIF。此时软件在读取移入的数据之前可继续往 SPDR 寄存器写入数据， 最后进来的数据将保存在接收缓冲器中。

SPI 模块在发送方向只有四个缓冲器，在接收方向也有四个缓冲器。在发送数据时，当发 送缓冲器处于非满状态（即发送缓冲器满标志位 WRFULL 位为低）时，可对 SPDR 寄存器进 行写操作。而在接收数据时，当接收缓冲器属于非空状态（即接收缓冲器空标志位 RDEMPT 位为低）时，可通过访问 SPDR 寄存器读取已经接收到的字符。

Single-Wire send and receive

The SPI Master initiates the communication cycle when pulling low the Slave Select signal SPSS. Master and Slave prepare the data to be sent , and the Master generates the required clock pulses on the SPCK line to interchange data. Master data is always shifted from MOSI to MISO – While Slave data from MISO to MOSI. After each data packet, the Master will pull SPSS to finish communiction.

When configured as a Master, the SPI interface has no automatic control of the SS line. This must be handled

by user software before communication can start. When this is done, writing data that need to be transmitted, to SPDR register
starts the SPI clock generator, the hardware generates communication clock and shifts the eight bits into the Slave, at the same time slave data are shifted in.
After shifting one byte, the clock generator stops, setting the end of Transmission Flag (SPIF).The Master may continue to shift the next byte by writing it into SPDR, or signal the end of packet by pulling high SPSS. The last incoming byte will be kept in the Buffer Register for later use.
When configured as a Slave, the SPI interface will remain sleeping with MISO tri-stated as long as the SPSS is driven high. In this state, software may update the contents of the SPDR, but the data will not be shifted out by incoming clock pulses on the SCK pin until the SPSS is driven low.
As one byte has been completely shifted, the end of Transmission Flag, SPIF is set. The Slave may continue to place new data to be sent into SPDR before reading the incoming data. The last incoming byte will be kept in the Buffer Register for later use.

主机双线接收

SPI 模块的双线模式只在主机操作模式下有效，与单线模式的不同在于 MOSI 和 MISO 都 用于主机接收数据，每一个 SPCK 时钟脉冲同时接收 2 个比特的数据（MISO 线上的数据在前，MOSI 线上的数据在后），接收完两个字节的数据之后硬件置位传输完成标志 SPIF，数据 保存到接收缓冲器和移位寄存器中。此时软件须读取 SPDR 寄存器两次来得到所接收的两个 字节的数据。需要注意的是，虽然双线模式下主机不向从机发送数据，软件仍需要往 SPDR 寄存器写入数据来启动时钟发生器产生通信时钟，写入一次 SPDR 寄存器即可接收两个字节 的数据。

2-Wire receive

SPI 2-Wire receive mode works only in master operation. Different from 1-wire mode, both its MOSI and MISO are used for master to receive data. Each SPCK clock pulse receives 2 byte data synchronously (data in MISO line is in the front, in MOSI is behind), after data receiving, hardware sets SPIF, data will be saved in receive buffer and shift register. At this moment software has to read SPDR register two times to get 2 byte data. Note that even tough master never sends data to slave in 2-wire mode, software has to write data to SPDR register to start clock generator, which generates communication clock. Writing data to SPDR register one time can receive 2 byte data.

数据模式

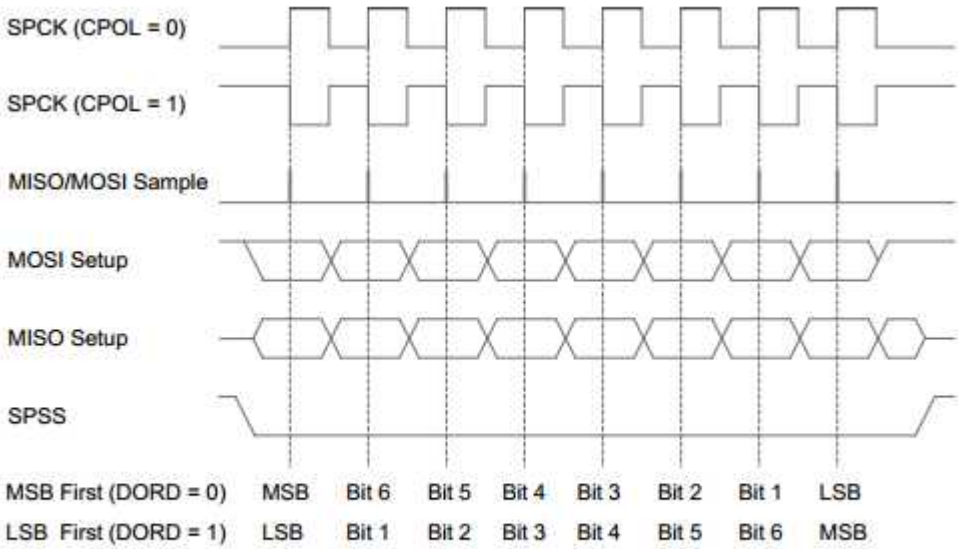
单线模式下，相对于串行数据，SPI 有 4 种 SPCK 相位和极性的组合方式，由 CPHA 和 CPOL 来控制，如下表所示

Data Mode

There are four combinations of SPCK phase and polarity with respect to serial data in single-wire mode, which are determined by CPHA and CPOL. The SPI data transfer formats are shown as below

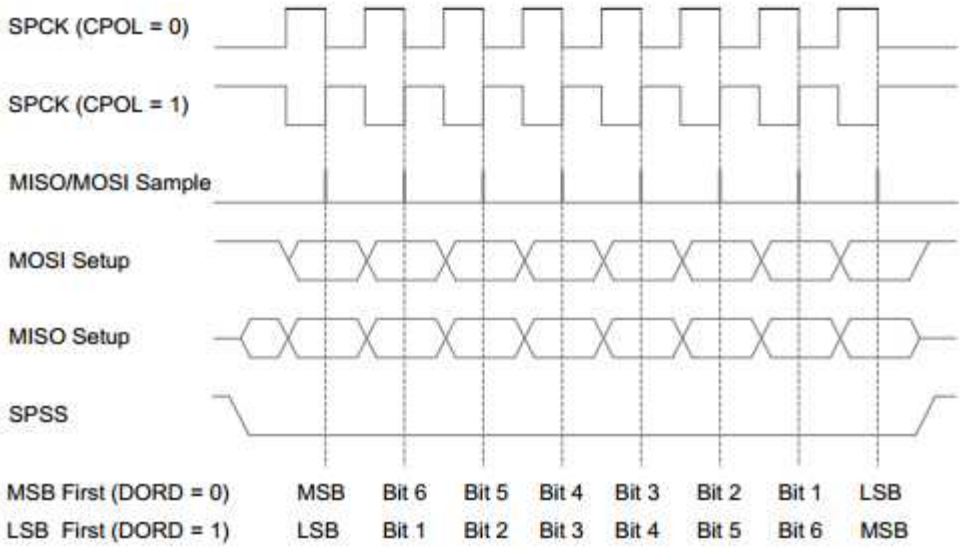
CPOL	CPHA	Leading Edge	Trail Edge	SPI mode0
0	0	Sample (Rising)	Setup (Falling)	0
0	1	Setup (Rising)	Sample (Falling)	1
1	0	Sample (Falling)	Setup (Rising)	2
1	1	Setup (Falling)	Sample (Rising)	3

When CPHA=0, Data sampling and clock edge setup are shown as below:

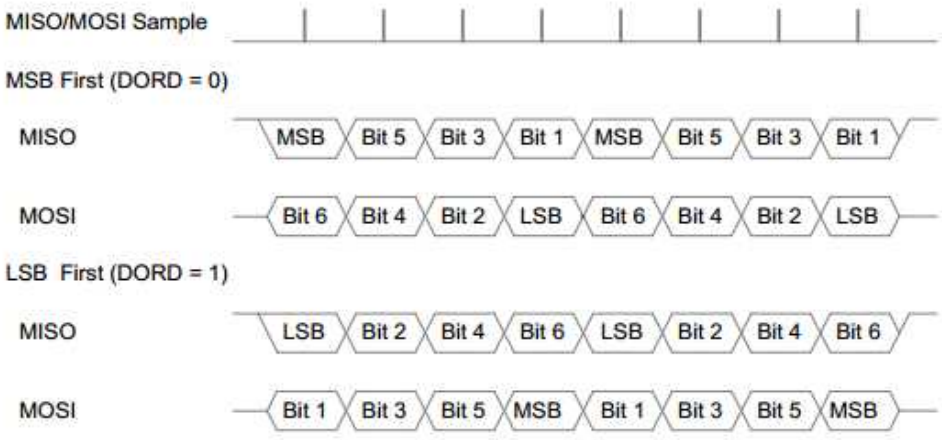


CPHA 为“0”时 SPI 数据传输模式

When CPHA=1, Data sampling and clock edge setup are shown as below:



In 2-Wire mode, MISO and MOSI both are used as master input, data sampling time is still decided by data transmission mode, sampling mode is shown as below:
In master mode, sample mode when DUAL is “1”



主机模式下 DUAL 为“1”时 SPI 数据采样模式

SPSS 引脚功能

当配置为从机时，从机选择信号 SPSS 引脚总是作为输入。当 SPSS 引脚保持为低时，SPI 接口被激活， MISO 引脚成为输出引脚（软件进行相应的端口配置），其它引脚均为输入。当 SPSS 引脚保持为高时， SPI 模块被复位，且不再接收数据。SPSS 引脚对于数据包/字节的同步非常有用，可以使从机的位计数器和主机的时钟发生器同步。当 SPSS 拉高时， SPI 从机立即复位接收和发送逻辑，并丢弃移位寄存器里不完整的数据。

当配置为主机时，用户软件可以决定 SPSS 引脚的方向。

若 SPSS 配置为输出，则它可以用来驱动从机的 SPSS 引脚。若 SPSS 配置为输入，必须保持为高以保证主机的正常工作。当配置为主机且 SPSS 引脚为输入，外部电路拉低 SPSS 引脚时， SPI 模块会认为是另外一个主机选择自己作为从机并开始传输数据。为了防止总线冲突，SPI 模块将进行如下动作：

1. 清零位于 SPCR 寄存器的 MSTR 位，转换为从机，从而 MOSI 和 SPCK 变为输入；
2. 置位位于 SPSR 寄存器的 SPIF 位，若中断使能则产生 SPI 中断。

因此，使用中断方式处理 SPI 主机的数据传输，并且存在 SPSS 被拉低的可能性时，中断服务程序应该检查 MSTR 位是否为“1”。若被清零，软件须将其置位，以重新使能 SPI 主机模式。

SPSS Pin Function

When salve is configured, SPSS pin, slave selection signal, is always used as input. When SPSS pin keeps low, SPI interface is activated, MISO pin becomes output pin (interface configuration are done by software), while other pins are all input. When SPSS pin keeps high, SPI modular is re-set and does not receive data anymore. SPSS pin is very useful to synchronize data package and byte, it can synchronize slave bit counter and master clock generator. If SPSS is driven high, SPI slave reset immediately to receive and send logic, and dismiss incomplete data in shift register.

When master is configured, SPSS pin direction is decided by user software.

If SPSS is configured as output, it can drive slave SPSS pin. If SPSS is configured as input, must keep it high to ensure master normal operations. When master is configured and SPSS is set to input, while SPSS pin is drive low by external circuit, SPI modular will consider that another master choose to be slave and start to transmiss data. To avoid bus conflict, SPI modular makes below actions:

1. Clear MSTR bit of SPCR register, switch to slave, MOSI and SPCK change to input

2. Set SPIF bit of SPSR register, if interrupt is enabled, will generate SPI interrupt.

So, interrupt routine handle program should check if MSTR bit is logic one when interrupt is used for SPI master data transmission and SPSS is possibly drived low. If MSTR is clear, must set by software to re-start SPI master mode.

SPI 初始化

进行通信之前首先要对 SPI 进行初始化。初始化过程通常包括主机从机操作的选择，数据传输模式的设定，比特率的选择，以及各个引脚的方向控制

等。其中主机和从机操作下引脚方向的控制各不相同，如下表所示：

SPI Initiation

Before communication, SPI must be initianized, this process includes master/slave operation selection, data transmission mode setup, byte rate select, direction control of each pin, etc. Pin direction controls are different in mast /slave mode, details are shown in the below:

Pin Direction Control

Pin	Direction in master mode	Direction in slave mode
MOSI	Defined by user software	Input
MISO	Input	Defined by user software
SPCK	Defined by user software	Input
SPSS	Defined by user software	Input

SPI 主机初始化

SPI 主机模式的初始化过程如下：

1. 置位 MSTR 位，设置比特率选择控制位，数据传输模式，数据传输次序，中断使能与否，以及双线使能与否；
2. 设置 MOSI 和 SPCK 引脚为输出；
3. 置位 SPE 位。

主机模式下，当不希望 SPI 模块被别的主机选择作为从机使用时，可设置 SPSS 引脚为输出。

SPI Master Initiation

SP Initiation process in master mode is as below:

1. Set MSTR bit, set up byte rate control bit, set up data transmission mode and order, check if interrupt is enabled or not and 2-wire is enabled or not;

2. Set MOSI and SPCK pin as output;

3. Set SPE bit

In master mode, SPSS has to configure as output if SPI modular is not wished to use as slave choosen by other master.

SPI 从机初始化

SPI 从机模式初始化过程如下：

1. 清零 MSTR 位，设置数据传输模式，数据传输次序，中断使能与否；
2. 设置 MISO 引脚为输出；
3. 置位 SPE 位。

SPI Slave Initiation

SP Initiation process in slave mode is as below:

1. **Clear MSTR bit, set up data transmission mode and order, check if interrupt is enabled or not.**
2. **Set MISO pin as output**
3. **Set SPE bit**

SPI 中断

当发生下列事件之一或多个时，SPI 的中断标志位 SPIF 将会被置位：

1. 当配置为主机且 SPSS 引脚为输入，外部电路拉低 SPSS 引脚；
2. 当发送缓冲器状态为满，软件继续往 SPDR 寄存器写入数据；
3. 当接收缓冲器状态为满；

4. 当写入发送缓冲器中的数据均已发送出去，发送缓冲器状态为空。

SPI Interrupt:

SPI interrupt flag bit, SPIF, will be set when one or more than on of the below events happen

1. **Master is configured and SPSS pin is input, SPSS pin is driven low by external circuit**
2. **Status of send buffer is full, software keeps writing data to SPDR register**
3. **Status of receive buffer is full**
4. **Data written to send buffer have been send out, send buffer is empty.**

当 SPIF 位被置位，且 SPI 中断使能位 SPIE 和全局中断使能位都为高时，会产生 SPI 中断。进入中断服务程序后，硬件会对 SPIF 进行清零。若 SPIF 位是由上述事件中的 1 和 2 来置位的， SPIF 会被清零；若 SPIF 位是由上述事件中的 3 和 4 来置位的， SPIF 并不会被清零，因为接收或发送缓冲器状态未发生改变时，仍会置位 SPIF 位，此时需要通过软件操作来清零。

When SPIF is set, while SPI interrupt enable bit SPIE and globe interrupt enable bit are high, SPI interrupt is generated. After entering into interrupt program, hardware will clear SPIF. If SPIF is set by above event 1 and 2, SPIF will be cleared. If SPIF is set by above event 3 and 4, SPIF will not be cleared, because when receive or send buffer status has no change, SPIF bit is set still, in this case clear must be done by software.

SPI 中断服务程序中，软件清零 SPIF 位的操作顺序如下：

1） 读取 SPIF 位的状态，若为低，表明 SPIF 位已被硬件清零，无需软件再次清零；若为高，继续一下操作；

2） 读取 SPFR 寄存器，若 RDFULL 位为高，表明当前接收缓冲器状态为满，读取 SPDR 寄存器获得接收数据，RDFULL 位会变为低，软件可继续读取 SPDR 寄存器获得接收数据，直到 RDEMPT 位为高；

3） 读取 SPFR 寄存器，若 RDFULL 位为低，而 WREMPT 位为高，表明当前接收缓冲器状态为非满，而发送缓冲器状态为空，软件可读取 SPDR 寄存器获得接收数据，直到 RDEMPT 位为高；

4） 软件获取所接收到的数据后，再执行清零 SPIF 位。因 SPIF 位为只读位，不能直接对 SPIF 位进行清零，而需要先读取 SPSR 寄存器，再访问 SPDR（读或写 SPDR 寄存器）的方式来清零 SPIF 位。

In SPI interrupt program, software clears SPIF per below operation order:

1) **Read SPIF bit status, if low it indicates that SPIF bit has ben cleared by hardware, software does no need to clear; If high, continue to go for next operation;**
2) **Read SPFR register, if RDFULL bit is high it indicates that current receive buffer is full, read SPDR register to gain receive data, RDFULL bit becomes low, software can continue to read SPDR register to gain receive data, till RDEMPT bit is high;**
3) **Read SPFR register, if RDFULL bit is low while WREMPT bit is high, it indicates that current receive buffer is not full but send buffer is empty, software can read SPDR register to get receive data till RDEMPT bit is high**
4) **Software clears SPIF bit after received data. As SPIF bit is read-only, it is impossible to clear SPIF directly, to clear SPIF bit, it has to read SPSR register firstly and than access to SPDR(read or write SPDR register)**

Register Definition

SPI Register List

Register	Address	Default Value	Comment
SPCR	0x4C	0x00	SPI control register
SPSR	0x4D	0x00	SPI status register
SPDR	0x4E	0x00	SPI data register
SDFR	0x39	0x00	SPI buffer register

SPCR-SPI Control Register

SPCR-SPI Control Register								
Address: 0x4C				Default Value:0x00				
Bit	7	6	5	4	3	2	1	0
Name	SPIE	SPE	DORD	MSTR	CPOL	CPHA	SPR1	SPR0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	Name	Comment						
7	PIE	SPI interrupt enable bit If set SPIE bit to logic one, SPI interrupt is enabled. When SPIFin SPSR register is set and globe interrupt is enabled, SPI interrupt is geneated						
6	SPE	SPI Enable bit IF set SPE to logic one, SPI modular is enabled. Before any SPI operation must set SPE If set SPE to logic zero, SPI modular is disabled.						
5	DORD	Data order control bit If set DORD bit to logic one, LSB of data will be send firstly If set DORD bit to logic zero, MSB of data will be send firstly						
4	MSTR	Master/Slave selection control bit If set MSTR to logic one, master operation mode is selected If set MSTR to logic zero, slave operation mode is selected In master mode, when SPSS pin is configured to input and pulled low, MSTR is to be cleared; SPIF in SPSR register will be set. User need to re-configure MSTR to enter into master mode.						
3	CPOL	Clock polarity control bit When set CPOL to logic one, SPCK is high in idle status When set CPOL to logic zero, SPCK is low in idle status						
		CPOL	Leading Edge	Trail Edge				
		0	Rising Edge	Falling Edge				
		1	Falling Edge	Rising Edge				
2	CPHA	Clock phase control bit When CPHA is configured to logic one, rising edge sets data, trail edge samples data When CPHA is configured to logic zero, rising edge samples data, trail edge sets data						
		CPHA	Leading Edge	Trail Edge				
		0	Sampling	Set up				
		1	Set up	Sampling				
1	SPR1	Clock speed rate selection bit 1 SPR1 and SPR0 are used to select clock speed rate of SPI transmission. For details referring to list about relation between SPCK and system clock						
0	SPR0	Clock speed rate selection bit 0 SPR1 and SPR0 are used to select clock speed rate of SPI transmission. For details referring to list about relation between SPCK and system clock						

SPSR—SPI Status Register

SPSR—SPI Status Register								
Address: 0x4D				Default Value:0x00				
Bit	7	6	5	4	3	2	1	0
Name	SPIE	WCOL	-	-	-	DUAL	-	SPI2X
R/W	R	R	R	R	R	R/W	R	R/W
Initial	0	0	0	0	0	0	0	0
Bit	Name	Comment						
7	SPIF	SPI interrupt flag bit SPIF flag is set after serial transmission. In master mode, SPIF is set also when SPSS pin is configured as input and pulled down. At this moment if both SPIE of SPCR register and globe interrupt enable bit are set, SPI interrupt is generated. After entered into interrupt program, SPIF bit will clear automatically, or by reading SPSR register and than access to SPDR register, SPIF bit is cleared.						
6	WCOL	Write conflict flag bit During data transmission, writing SPDR register will set WCOL bit. Via reading SPSR register and than access to SPDR register, WCOL is to be cleared.						
5	-	reserved						
4	-	reserved						
3	-	reserved						
2	DUAL	2-Wire control bit When set DUAL bit to logic one, enable SPI 2-wire transmission mode When set DUAL bit to logic zero, disable SPI 2-wire transmission mode 2-wire transmission mode is valid only in SPI master mode, MISO and MOSI are used as master data input, regarding data transmission mode, refer to sections of master 2-wire receive and data mode						
1	-	reserved						
0	SPI2X	SPI speed double control bit When set SPI2X to logic one, SPI transmission speed will double When set SPI2X to logic zero, SPI transmission speed will not double For details about control mode, referring to list about relation between SPCK and system clock						

Below shows relation between SPCK and system clock

Relation between SPCK and system clock

SPI2X	SPR1	SPR0	SPCK frequency
0	0	0	fsys/4
0	0	1	fsys/16
0	1	0	fsys/64
0	1	1	fsys/128
1	0	0	fsys/2
1	0	1	fsys/8
1	1	0	fsys/32
1	1	1	fsys/64

SPDR—SPI data register

SPDR—SPI data register								
Address: 0x4E				Default value: 0x00				
Bit	7	6	5	4	3	2	1	0
Name	SPDR7	SPDR6	SPDR5	SPDR4	SPDR3	SPDR2	SPDR1	SPDR0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	Name	Comments						
7:0	SPDR	SPI sends and receives data SPI send data and receive data share SPI data register SPDR. Writing data to SPDR means writing send data to shift register. Reading SPDR data means reading receive data buffer.						

SPFR—SPI buffer register

SPFR—SPI buffer register								
Address: 0x4E				Default value: 0x00				
Bit	7	6	5	4	3	2	1	0
Name	RDFULI	RDEMPT	RDPTR1	RDPER0	WRFULL	WREMPT	WRPTR1	WRPTR0
R/W	R	R/W	R	R	R	R/W	R	R
Bit	Name	Comments						
7	RDFULL	Receive buffer full flag bit When receive buffer data is 4 byte, RDFULL bit is high, it indicates that receive buffer is full and interrupt flag bit is set. If software does not read and get away receive buffer data in time, when to receive data again, receive buffer will overflow, previous data will be covered by new data. When receive buffer data is less than 4 byte, RDFULL bit is low, it indicates that receive buffer is not full yet, it can continue to receive data. When setting RDEMPT and WREMPT simutanously, receive and send buffer address and SPI shift register hand will both clear to zero, RDFULL is low.						
6	RDEMPT	Receive buffer empty flag bit If it does not receive data, RDEMPT bit is high, which indicates receiver buffer is empty If it receives data, data will be saved to receive buffer, RDEMPT bit is low, which indicates that receive buffer is not empty, at this moment via access to SPDR register, MCU can read data of receive buffer. To ensure received data will not loss, software can read and get away receiver buffer data when it is not empty and RDEMPT bit is low. When set RDEMPT to logic one, receive buffer address is cleared to zero. When setting RDEMPT and WREMPT simutanously, receive and send buffer address and SPI shift register hand will both clear to zero, RDFULL is high.						
5	RDPTR1	Receive buffer address high						
4	RDPTR0	Receive buffer address low When writing SPDR register, MCU can read all received data from receive buffer,						

		meanwhile receive buffer address will accumulate. When setting RDEMPT bit to logic one, receive buffer address is cleared to zero.
3	WRFULL	Send buffer full flag bit When receive buffer data is 4 byte, WRFULL bit is high, it indicates that send buffer is full. When send buffer data is less than 4 byte, WRFULL bit is low, it indicates that send buffer is not full yet, if want to increase transmission speed, when send buffer is not full and WRFULL is low, software can write data, SPI control will send these data out by order.
2	WREMPT	Send buffer empty flag bit If data written to send buffer is finished for sending, WREMPT bit is high, which indicates send buffer is empty and interrupt flag bit SPIF is set. After writing SPDR register, send buffer address will accumulate, if data written to send buffer are not all send out, receive buffer has at least 1 byte data, WREMPT bit is low, which indicate send buffer is not empty. When setting WREMPT bit to logic one, send buffer address is cleared to zero. When setting RDEMPT and WREMPT bit simutanously, receive and send buffer address, as well as SPI shift register hand will all clear to zero, WREMPT bit is high
1	WRPTR1	Send buffer address high
0	WRPTR0	Send buffer address low Wen writing to SPDR register, Data in SPDR will be written to send buffer, meanwhile send buffer address will accumulate When setting WREMPTto logic one, send buffer address will be clared to zero.

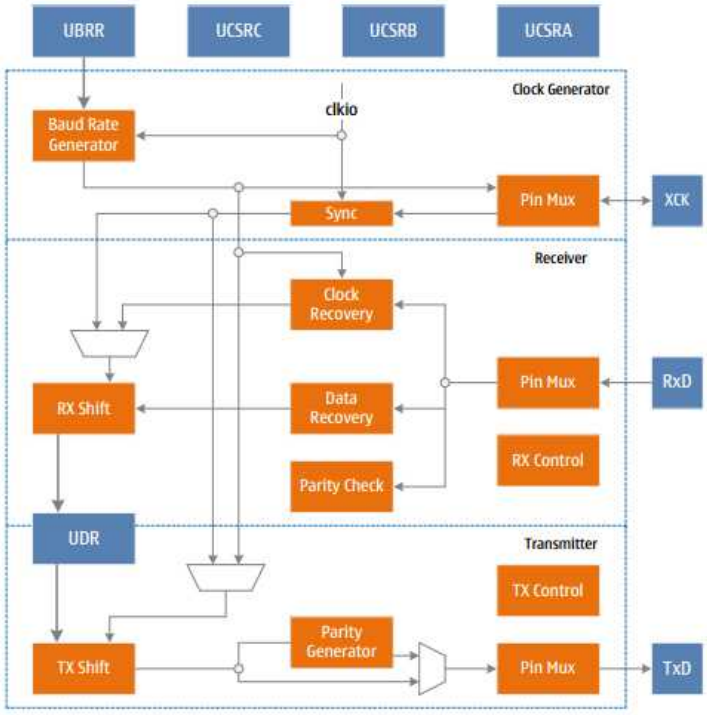
USART0 - 通用同步/异步串行收发器

- 全双工操作 (独立的串行接收和发送寄存器)
- 异步或同步操作
- 主机或从机操作
- 高精度的波特率发生器
- 支持 5，6，7，8，或 9 个数据位和 1，或 2 个停止位
- 硬件支持的奇偶产生和校验机制
- 数据过速检测
- 帧错误检测
- 噪声滤波，包括错误的起始位检测以及数字低通滤波器
- 三个独立的中断：发送结束中断，发送数据寄存器空中断以及接收结束中断
- 多处理器通信模式
- 倍速异步通信模式

USART0- General Synchronous/Asynchronous serial receiver

- Full Duplex Operation (Independent Serial Receive and Transmit Registers
- Asynchronous or Synchronous Operation
- Master or Slave Clocked Synchronous Operation
- High Resolution Baud Rate Generator
- Supports Serial Frames with 5, 6, 7, 8, or 9 Data Bits and 1 or 2 Stop Bits
- Odd or Even Parity Generation and Parity Check Supported by Hardware

- Data OverRun Detection
- Framing Error Detection
- Noise Filtering Includes False Start Bit Detection and Digital Low Pass Filter
- Three Separate Interrupts on TX Complete, TX Data Register Empty and RX Complete
- Multi-processor Communication Mode
- Double Speed Asynchronous Communication Mode



USART 结构图

USART 主要包括三个部分：时钟发生器，发送器和接收器。控制和状态寄存器由这三个部分共享。时钟发生器由波特率发生器和同步从机操作模式下外部输入时钟的同步逻辑组成。XCK 引脚只用于同步传输模式。发送器包括一个写数据缓冲器，串行移位寄存器，奇偶发生器以及处理不同帧格式所需的控制逻辑。写数据缓冲器允许连续发送数据而不会在数据帧之间引入延迟。接收器具有时钟和数据恢复单元，用于异步数据的接收。除了恢复单元，接收
器还包括奇偶校验，控制逻辑，串行移位寄存器和一个两级接收缓冲器 UDR。接收器支持与发送器相同的帧格式，而且可以检测帧错误，数据过速和奇偶校验错误。

USART mainly includes 3 units: Clock Generator, Transmitter and Receiver. Control status Registers are shared by all units. The Clock Generation logic consists of synchronization logic for external clock input used by synchronous slave operation, and the baud rate generator. The XCK pin is only used by synchronous transfer mode. The Transmitter consists of a single write buffer, a serial Shift Register, Even and Odd Parity Generator and Control logic for handling different serial frame formats. The write buffer allows a continuous transfer of data without any delay between frames.
The Receiver is the most complex part of the USART module due to its clock and data recovery units. The recovery units are used for asynchronous data reception. In addition to the recovery units, the Receiver includes a Parity Checker, Control logic, a serial Shift Register and a two level receive buffer (UDR). The Receiver supports the same frame formats as the Transmitter, and can detect Frame Error, Data OverRun and Parity Errors.

时钟产生

时钟产生逻辑为发送器和接收器产生基础时钟。 USART 支持 4 种模式的时钟：正常的异步模式，倍速的异步模式，主机同步模式，以及从机同步模式。USCRC 的 UMSEL 位用于选择同步或异步模式。USCRA 的 U2X 位控制异步模式下的倍速使能。仅在同步模式下有效的 XCK 引脚的数据方向寄存器(与 IO 复用)决定了时钟源是由内部产生(主机模式)还是外部产生(从机模式)。

Clock Generation

The Clock Generation logic generates the base clock for the Transmitter and Receiver. The USART supports four modes of clock operation: Normal asynchronous, Double Speed asynchronous, Master synchronous and Slave synchronous mode. The UMSEL bit in USART Control and Status Register C (UCSRC) selects between asynchronous and synchronous operation. Double Speed (asynchronous mode only) is controlled by the U2X found in the UCSRA Register. When using synchronous mode (UMSELn = 1), the Data Direction Register for the XCK pin controls whether the clock source is internal (Master mode) or external (Slave mode). The XCK pin is only active when using synchronous mode.

波特率发生器

波特率寄存器 UBRR 和降序计数器连接在一起作为 USART 的可编程的预分频器或波特率发生器。降序计数器工作在系统时钟(fsys)下，当其计数到零或 UBRRL 寄存器被写时，会自动加载 UBRR 寄存器的值。当计数到零时产生一个时钟，该时钟作为波特率发生器的输出时钟，频率为 fsys/(UBRR+1)。

Baud Rate Generator

The USART Baud Rate Register (UBRR) and the down-counter connected to it function as a programmable prescaler or baud rate generator. The down-counter, running at system clock (fosc), is loaded with the UBRR register value automatically each time when the counter has counted down to zero or the UBRRL Register is written. A clock is generated each time the counter reaches zero. This clock is the baud rate generator clock output (= fosc/(UBRRn+1)).

Below list gives equations for baud rate(bit/second) and UBRR value in each working mode:

Working mode	Equation for Calculating Baud(1)	Equation for Calculating UBRR Valu
Asynchronous Normal mode	BAUD = fsys/(16*(UBRR+1))	UBRR = fsys/(16*BAUD) - 1
Asynchronous Double Speed mode	BAUD = fsys/(8*(UBRR+1))	UBRR = fsys/(8*BAUD) -1
Synchronous Master mode	BAUD = fsys/(2*(UBRR+1))	UBRR = fsys/(2*BAUD) - 1

Noted:

1. The baud rate is defined to be the transfer rate in bit per second (bps)
2. BAUD=Baud rate (in bits per second, bps), fsys=system clock. UBRR is combined value of baud register UBRRH and UBRRL.

倍速工作模式

通过设定 UCSRA 寄存器的 U2X 位可以是传输速率加倍，该位只在异步工作模式下有效，同步工作模式下置该位为“0”。

设置该位将会把波特率分频器的分频值减半，有效地加倍异步通信的传输速率。在这种情况下，接收器只使用一半的采样数来对数据进行采样及时钟恢复，因此需要更精准的波特率设置和系统时钟。发送器则没有变化。

Double Speed Operation

The transfer rate can be doubled by setting the U2Xnbit in UCSRA. Setting this bit only has effect for the asynchronous operation. Set this bit to zero when using synchronous operation.

Setting this bit will reduce the divisor of the baud rate divider from by half, effectively doubling the transfer

rate for asynchronous communication. Note however that the Receiver will in this case only use half the number of samples for data sampling and clock recovery, and therefore a more accurate baud rate setting and system clock are required when this mode is used. For the Transmitter, there are no downsides.

外部时钟

同步从机操作模式由外部时钟驱动。外部时钟经过同步寄存器和边沿检测器之后才被发送器和接收器使用，这一过程会引入两个系统时钟的延时，因此外部 XCK 的最大时钟频率由以下公式限制：

fXCK < fsys/4

要注意 fsys 有系统时钟的稳定性决定，为了防止因频率漂移而丢失数据，建议保留足够的裕量。

External Clock

External clocking is driven by the synchronous slave modes of operation. External clock must then pass through synchronization register and an edge detector before it can be used by the Transmitter and Receiver. This process introduces a two CPU clock period delay and therefore the maximum external XCK clock frequency is limited by the following equation:

fXCK < fsys/4

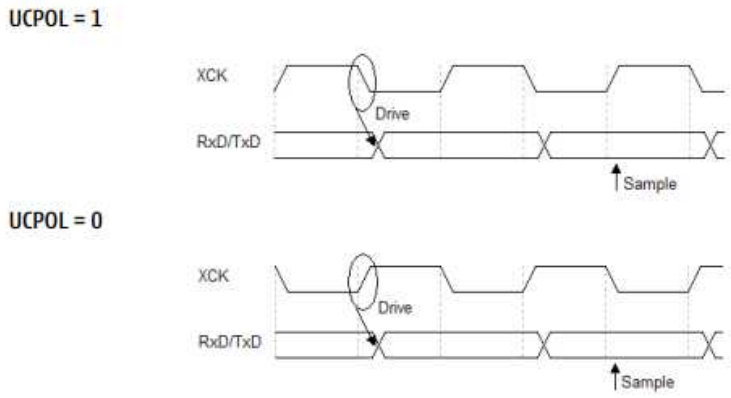
Note that fsys depends on the stability of the system clock . It is therefore recommended to add some margin to avoid possible loss of data due to frequency variations.

同步时钟操作

同步模式下， XCK 引脚被用于时钟输入（从机模式）或时钟输出（主机模式）。时钟的边沿与数据采样和数据变化关系的基本规律是：对数据输入端（RxD）采样所使用的时钟沿与数据输出端变化所使用的时钟沿是相反的。

Synchronous Clock Operation

When synchronous mode is used , the XCK pin will be used as either clock input (Slave) or clock output (Master). The rule between the clock edges and data sampling or data change is that: Data input (on RxD) is sampled at the opposite clock edge of the edge the data output is changed.



同步模式下的 XCK 时序

如上图所示，当 UCPOL 值为“1”时，在 XCK 的下降沿改变数据输出，在 XCK 的上升沿进行数据采样；当 UCPOL 值为“0”时，在 XCK 的上升沿改变数据输出，在 XCK 的下降沿进行数据采样。

As shown above, If UCPOLn is set to logic one, the data will be changed at falling XCK edge and sampled at rising XCK edge. When UCPOL is zero the data will be changed at rising XCK edge and sampled at falling XCK edge.

帧格式

一个串行数据帧由数据字加上同步位（起始位和停止位）以及用于纠错的奇偶校验位构成。USART 接受以下 30 种组合的数据帧格式：

- ？ 1 个起始位
- ？ 5、6、7、8 或 9 个数据位
- ？ 无校验位、奇校验位或偶校验位
- ？ 1 或 2 个停止位

数据帧以起始位开始，紧接着是数据字的最低位，接着是其它数据位，以数据字的最高位结束，最多成功传输 9 位数据。如果使能了校验，校验位将紧接着数据字，最后是停止位。当一个完整的数据帧传输后，可以立即传输下一个新的数据帧，或者使传输线处于空闲（高电平）状态。下图为可能的数据帧结构，方括号中的位是可选的。

Frame Formats

A serial frame is defined to be one character of data bits with synchronization bits (start and stop bits), and a parity bit for error checking. The USART accepts all 30 combinations of the following as valid frame formats:

- 1 start bit
- 5, 6, 7, 8, or 9 data bits
- no, even or odd parity bit
- 1 or 2 stop bits

A frame starts with the start bit followed by the least significant data bit. Then the next data bits, up to a total of nine, are succeeding, ending with the most significant bit. Ifenabled, the parity bit is inserted after the data bits, before the stop bits. When a complete frame is transmitted, it can be directly followed by a new frame, or the communication line can be set to an idle (high) state. Below picture shows the possible combinations of the frame formats. Bits inside brackets are optional.

说明：

1）IDLE 通信线（RxD 或 TxD）上没有数据传输，线路空闲时必须为高电平

2）St 起始位，总是为低电平

3）0-8 数据位

4）P 校验位，奇校验或偶校验

5）Sp 停止位，总是为高电平

数据帧的结构由 UCSRB 和 UCSRC 寄存器中的 UCSZ[2:0]、UPM[1:0]和 USBS 设定。接收与发送使用相同的设置。设置的任何改变都可能破坏正在进行的数据传输。其中， UCSZ[2:0]确定了数据帧的数据位数，UPM[1:0]用于使能和确定校验的类型，USBS 设置帧有一位或两位结束位。接收器会忽略第二个停止位，因此帧错误只在第一个结束位为“0”时被检测到。

Note:

1) IDLE: No transfers on the communication line (RxD or TxD). An IDLE line must be high.

2) St: Start bit, always low.

3) Data bits (0 to 8).

4) P Parity bit. Can be odd or even

5) Sp: Stop bit, always high

The frame format is set by the UCSZ[2:0], UPM[1:0] and USBS bits in UCSRB and UCSRC. The Receiver and Transmitter use the same setting. Note that changing the setting of any of these bits will corrupt all ongoing communication for both the Receiver and Transmitter. The USART Character SiZe (UCSZ[2:0]) bits select the number of data bits in the frame. The USART Parity mode (UPM[1:]0) bits enable and set the type of parity bit. The selection between one or two stop bits is done by the USART Stop Bit Select (USBS) bit. The Receiver ignores the second stop bit. An FE (Frame Error) will therefore only be detected in the cases where the first stop bit is zero.

校验位计算

校验位的计算是对数据的各个位进行异或运算。如果选择了奇校验，则异或结果还需要取反。 校验位与数据位的关系如下:

Peven = dn-1⊕…⊕d3⊕d2⊕d1⊕d0⊕0

Podd = dn-1⊕…⊕d3⊕d2⊕d1⊕d0⊕1

说明： 1）Peven 偶校验结果 2）Podd 奇校验结果 3）dn 第 n 个数据位

Parity Bit Calculation

The parity bit is calculated by doing an exclusive-or of all the data bits. If odd parity is used, the result of the exclusive or is inverted. The relation between the parity bit and data bits is as follows:

Peven = dn-1⊕…⊕d3⊕d2⊕d1⊕d0⊕0

Podd = dn-1⊕…⊕d3⊕d2⊕d1⊕d0⊕1

Note that:

1) Peven: Parity bit using even parity

2) Podd Parity bit using odd parity

3) Dn: Data bit n of the character If used, the parity bit is located between the last data bit and first stop bit of a serial frame

USART 初始化

进行通信之前首先要对 USART 进行初始化。初始化过程通常包括波特率的设定，帧结构的 设定，以及根据需要使能接收器或发送器。对于中断驱动的 USART 操作，在初始化时要清零 全局中断标志并禁止 USART 的所有中断。

在进行重新初始化比如改变波特率或帧结构时，必须确保没有数据传输。TXC 标志位可以用 来检测发送器是否完成了所有传输，RXC 标志位可以用来检测接收缓冲器中是否还有数据未 被读出。如果 TXC 标志位用作此用途，在每次发送数据之前（写 UDR 寄存器之前）必须清 零 TXC 标志位。

USART Initialization

The USART has to be initialized before any communication can take place. The initialization process normally consists of setting the baud rate, setting frame format and enabling the Transmitter or the Receiver depending on the usage. For interrupt driven USART operation, the Global Interrupt Flag should be cleared (and interrupts globally disabled) when doing the initialization.

Before doing a re-initialization with changed baud rate or frame format, be sure that there are no ongoing transmissions during the period the registers are changed. The TXC Flag can be used to check that the Transmitter has completed all transfers, and the RXC Flag can be used to check that there are no unread data in the receive buffer. Note that the TXC Flag must be cleared before each transmission (before UDRn is written) if it is used for this purpose.

发送器

置位 UCSRB 寄存器的 TXEN 位将使能 USART 的数据发送。使能后 TxD 引脚的通用 IO 功能即 被 USART 功能所取代，成为发送器的串行输出。发送数据之前要设置好波特率、工作模式与 帧格式。如果使用同步发送模式，施加于 XCK 引脚上的时钟信号即为数据发送的时钟。

Transmitter

The USART Transmitter is enabled to transmit data by setting the Transmit Enable (TXEN) bit in the UCSRB Register. When the Transmitter is enabled, the general IO port operation of the TxD pin is overridden by the USART and given the function as the Transmitter's serial output. The baud rate, mode of operation and frame format must be set up once before doing any transmissions. If synchronous operation is used, the clock on the XCK pin will be overridden and used as transmission clock

发送 5 到 8 为数据的帧

将需要发送的数据加载到发送缓冲器中来启动数据发送。CPU 通过写 UDR 寄存器来加载数 据。当发送移位寄存器可以发送新一帧数据的时候，缓冲器中的数据将转移到移位寄存器中。 当移位寄存器处于空闲状态（没有正在进行的数据传输），或者前一帧数据的最后一个停止 位发送完毕，它

将加载新的数据。一旦移位寄存器加载了新的数据，它将按照既定的设置传输一个完整的帧.

Sending Frames with 5 to 8 Data Bit

A data transmission is initiated by loading the transmit buffer with the data to be transmitted. The CPU can load data by writing to the UDR register. The buffered data in the transmit buffer will be moved to the Shift Register when the Shift Register is ready to send a new frame. The Shift Register is loaded with new data if it is in idle state (no ongoing transmission) or immediately after the last stop bit of the previous frame is transmitted. When the Shift Register is loaded with new data, it will transfer one complete frame at the designed rate.

发送 9 位数据的帧

如果发送 9 位数据的帧，应先将数据的第 9 位写入寄存器 UCSRB 的 TXB8 位，然后再将低 8 位数据写入发送数据寄存器 UDR。第 9 位数据在多机通信中用于表示地址帧，在同步通信中 可以用于协议处理。

Sending Frames with 9 Data Bit

If 9-bit characters are used (UCSZn = 7), the ninth bit must be written to the TXB8 bit in UCSRB before the low eighth byte of the character is written to UDR. The ninth bit can be used to indicatr an address frame in multi processor communication mode or for other protocol handling in synchronization communication.

发送奇偶校验位

奇偶校验产生电路为串行数据帧生成相应的校验位。当校验位使能时（UPM1 = 1），发送控制逻辑电路会在数据字的最后一位与第一个停止位之间插入奇偶校验位。

Parity Generator

The Parity Generator calculates the parity bit for the serial frame data. When parity bit is enabled (UPM1 = 1), the transmitter control logic inserts the parity bit between the last data bit and the first stop bit of the frame that is sent.

发送标志位与中断处理

USART 发送器有两个标志位：USART 数据寄存器空标志 UDRE 和传输结束标志 TXC，两个标 志位都可以产生中断。

数据寄存器空标志 UDRE 用来表示发送缓冲器是否可以写入一个新的数据。该位在发送缓冲 器空时被置“1”，满时被置“0”。当 UDRE 位为“1”时，CPU 可以往数据寄存器 UDR 写入新的数 据，反之则不能

当 UCSRB 寄存器中的数据寄存器空中断使能位 UDRIE 为“1”时，只要 UDRE 被置位（且全局 中断使能），就将产生 USART 数据寄存器空中断请求。对寄存器 UDR 执行写操作将清零 UDRE。 当采用中断方式传输数据时，在数据寄存器空中断服务程序中必须写入一个新的数据到 UDR 以清零 UDRE，或者是禁止数据寄存器空中断。否则一旦该中断服务程序结束，一个新的中 断将再次产生。

当整个数据帧被移出发送移位寄存器，同时发送寄存器中又没有新的数据时，发送结束标志 TXC 将被置位。当 UCSRB 上的发送结束中断使能位 TXCIE（且全局中断使能）置“1”时，随着 TXC 标志位被置位，USART 发送结束中断将被执行。一旦进入中断服务程序，TXC 标志位即 被自动清零，CPU 也可以对该位写“1”来清零

Transmitter Flags and Interrupts

The USART Transmitter has two flags that indicate its state: USART Data Register Empty (UDRE) and Transmit Complete (TXC). Both flags can be used for generating interrupts.

The Data Register Empty (UDRE) Flag indicates whether the transmit buffer is ready to receive new data. This bit is set to logic one when the transmit buffer is empty, and cleared when the transmit buffer is empty. When UDRE is logic one, CPU can write new data to data register UDR, otherwise cannot write new data. When the Data Register Empty Interrupt Enable (UDRIE) bit is written to one, the USART Data Register Empty Interrupt will be executed as long as UDRE is set (provided that global interrupts are enabled). UDREn is cleared by writing UDR.

When interrupt-driven data transmission is used, the Data Register Empty interrupt routine must either write new data to UDR in order to clear UDRE or disable the Data Register Empty interrupt, otherwise a new

interrupt will occur once the interrupt routine terminates.
The Transmit Complete (TXCn) Flag bit is set to logic one when the entire frame in the Transmit Shift Register has been shifted out and there are no new data currently present in the transmit register.
When transmit complete interrupt enable bit TXCIE in UCSRB is set to logicone, and TXC flag bit is set, USART transit end interrupt is to be executed. Once enterring into interrup routine, TXC flag is automatically cleared, or to be cleared by CPU writing logice one to this bit.

禁止发送器

当 TXEN 清零后，只有等所有的数据都发送完成以后发送器才能够真正禁止，即发送移位寄 存器与发送缓冲寄存器中都没有要传送的数据。发送器禁止以后，TxD 引脚恢复其通用 IO 功 能。

Disabling the Transmitter

The disabling of the Transmitter (setting the TXEN to zero) will not become effective until ongoing and pending transmissions are completed after TXEN is cleared, i.e., when the Transmit Shift Register and Transmit Buffer Register do not contain data to be transmitted. When disabled, the TxD pin will resume to general IO.

接收器

置位 UCSRB 寄存器的接收允许位（RXEN）即可启动 USART 接收器。使能后 RxD 引脚的通用 IO 功能被 USART 功能所取代，成为接收器的串行输入口。进行数据接收之前首先要设置好 波特率、操作模式及帧格式。如果使用同步接收模式，XCK 引脚上的时钟被用为传输时钟。

Receiver

The USART Receiver is enabled by setting the Receive Enable (RXEN) bit in the UCSRB Register. When the Receiver is enabled, the normal pin operation of the RxD pin is overridden by the USART and given the function as the Receiver's serial input. The baud rate, mode of operation and frame format must be set up once before any data reception. If synchronous operation is used, the clock on the XCK pin will be used as transfer clock.

接收 5 到 8 位数据的帧

一旦接收器检测到一个有效的起始位，便开始接受数据。起始位后的每一位数据都将以所设 定的波特率或 XCK 时钟来进行接收，直到收到一帧数据的第一个停止位，第二个停止位会被接收器忽略。接收到的每一位数据被送入接收移位寄存器，收到第一个停止位以后，接收器 置位位于 UCSRA 寄存器的接收数据完成标志 RXC 位，并把移位寄存器中完整的数据帧转移 到接收缓冲器中，CPU 通过读取 UDR 寄存器就可以获得接收到的数据。

Receiving Frames with 5 to 8 Data Bits

The Receiver starts data reception when it detects a valid start bit. Each bit that follows the start bit will be sampled at the designed baud rate or XCK clock, and shifted into the Receive Shift Register until the first stop bit of a frame data is received. A second stop bit will be ignored by the Receiver.
After receiving the first stop bit, receiver sets receive data complete flag RXC bit loacted in UCSRA register and complete data frames in shift register will be moved into receive buffer. By reading UDR register, CPU can get received data.

接收 9 位数据的帧

如果设定了 9 位数据的数据帧，在从 UDR 读取低 8 位数据之前必须首先读取寄存器 UCSRB 的 RXB8 位来获得第 9 位数据。这个规则同样适用于状态标志位 FE、DOR 以及 PE。读取 UDR 存储单元会改变接收缓冲器的状态，进而改变同样存储于缓冲器中的 TXB8、FE、DOR 及 PE 位。

Receiving Frames with 9 Data Bits

If 9-bit characters are used, the ninth bit must be read from the RXB8 bit in UCSRB before reading the low bits from the UDR. This rule applies to the FE, DOR and PE as well. Reading the UDR I/O location will change the state of the receive buffer and consequently the TXB8, FE, DOR and PE bits, which all are stored in the

bufffer, will change.

接收结束标志及中断处理

USART 接收器有一个标志位：接收结束标志 RXC，用来表明接收缓冲器中是否有未被读出的 数据。当接收缓冲器中有未被读出的数据时，此位为“1”，反之为“0”。如果接收器被禁止， 接收缓冲器会被刷新，RXC 也会被清零。 置位 UCSRB 的接收结束中断使能位 RXCIE 后，只要 RXC 标志被置位（且全局中断被使能）， 就会产生 USART 接收结束中断。使用中断方式进行数据接收时，数据接收结束中断服务程 序必须从 UDR 读取数据来清零 RXC 标志，否则只要中断处理程序一结束，一个新的中断就 会产生。

Receive Compete Flag and Interrupt

The USART Receiver has one flag: Receive Complete (RXC) Flag. It indicates if there are unread data present in the receive buffer. This flag is one when unread data exist in the receive buffer, and otherwise it is zero. If the Receiver is disabled, the receive buffer will be flushed and consequently the RXC bit will become zero. When the Receive Complete Interrupt Enable (RXCIE) in UCSRB is set, the USART Receive Complete interrupt will be executed as long as the RXC Flag is set (provided that global interrupts are enabled). When interrupt-driven data reception is used, the receive complete routine must read the received data from UDR in order to clear the RXC Flag, otherwise a new interrupt will occur once the interrupt routine terminates.

接收错误标志

USART 接收器有三个错误标志：帧错误 FE、数据溢出 DOR 及奇偶校验错误 PE。它们都位于 UCSRA 寄存器。错误标志与数据帧一起保存在接收缓冲器当中。所有的错误标志都不能产生 中断。

帧错误标志 FE 表明存储在接收缓冲器中的下一个可读帧的第一个停止位的状态。停止位正 确（值为“1”）则 FE 标志为“0”，否则 FE 标志为“1”。这个标志可用来检测同步丢失、传输中 断，也可用于协议处理。

数据溢出标志 DOR 表明由于接收缓冲器满造成了数据丢失。当接收缓冲器为满，接收移位 寄存器中已有数据，若此时检测到一个新的起始位，数据溢出就产生了。DOR 标志被置位即 表明在最近一次读取 UDR 和下一次读取 UDR 之间丢失了一个或多个数据帧。当数据帧成功 地从移位寄存器转入接收缓冲器后，DOR 标志被清零。

奇偶校验错标志 PE 表明接收缓冲器中的下一帧数据在接收时有奇偶错误。如果不使能奇偶 校验，PE 被清零。

Receiver Error Flags

The USART Receiver has three Error Flags: Frame Error (FE), Data OverRun (DOR) and Parity Error (PE). All can be accessed by reading UCSRA.

Common for the Error Flags is that they are located in the receive buffer together with the data frame. All Error Flags cannot generate interrupt.

The Frame Error (FE) Flag indicates the state of the first stop bit of the next readable frame stored in the receive buffer. The FE Flag is zero when the stop bit was correctly read (as one), otherwise the FE Flag will be one. This flag can be used for detecting out-of-sync conditions, detecting break conditions and protocol handling.

Data Overrun flag(DOR) indicates that data will be lost due to receive buffer full. When receive buffer is full, data has existed in receive shift register, at this moment if a new start bit is detected, data overrun occurs. DOR flag is set, that is to say one or more frame have lost between a recently read UDR and the next one. DOR flag is cleared when frame is transferred successfully from shift register to receive buffer. Even Odd Parity Flag (PE) indicates next frame located in receive buffer has even odd errors during receive. PE will be cleared if even odd parity is not enabled.

奇偶校验器

置位奇偶校验模式位 UPM1 将启动奇偶校验器。校验的模式（偶校验或奇校验）由 UPM0 决定。奇偶校验使能后，校验器将计算输入数据的奇偶并把结果与数据帧的奇偶位进行比较。校验结果将与数据和停止位一起存储在接收缓冲器中。CPU 通过读取 PE 位来检查接收的帧当中是否有奇偶错误。如果下一个从接收缓冲器中读出的数据有奇偶错误，并且奇偶校验使能，则 UPE 被置位，一直有效到接收缓冲器 UDR 被读取。

Parity Checker

The Parity Checker is actived when Parity mode (UPM1) bit is set. Type of Parity Check to be performed (odd or even) is selected by the UPM0 bit. When enabled, the Parity Checker calculates the parity of the data bits that is input and compares the result with the parity bit from the data frame. The result of the check is stored in the receive buffer together with the received data and stop bits. The Parity Error Flag can then be read by software to check if the frame had a Parity Error. The UPE bit is set if the next character that can be read from the receive buffer had a Parity Error and the Parity Checking was enabled. This bit is valid until the receive buffer (UDR) is read.

禁止接收器

与发送器相比， 禁止接收器即刻起作用。正在接收的数据将丢失。禁止接收器（RXEN 清零）后，接收器将不再占用 RxD 引脚，接收缓冲器也会被刷新。

Disabling the Receiver

In contrast to the Transmitter, disabling of the Receiver will become effective immediately. Data from ongoing receptions will therefore be lost. When disabled (i.e., the RXEN is set to zero) the Receiver will no longer override the normal function of the RxD port pin. The Receiver buffer will be flushed when the Receiver is disabled.

异步数据接收

USART 有一个时钟恢复单元和数据恢复单元来处理异步数据接收。时钟恢复逻辑用于同步从 RxD 引脚输入的异步串行数据和内部的波特率时钟。数据恢复逻辑用于采集数据，并通过低通滤波器过滤所输入的每一位数据，从而提高接收器的抗干扰性能。异步接收的工作范围依赖于内部波特率时钟的精度、帧输入的速率及一帧所包含的数据位数。

Asynchronous Data Reception

The USART includes a clock recovery and a data recovery unit for handling asynchronous data reception. The clock recovery logic is used for synchronizing the internally generated baud rate clock and the incoming asynchronous serial data at the RxD pin. The data recovery logic samples and low pass filters each incoming bit, thereby improving the noise immunity of the Receiver. The asynchronous reception operational range depends on the accuracy of the internal baud rate clock, the rate of the incoming frames, and the frame size in number of bits.

异步工作范围

接收器的工作范围依赖于接收到的数据速率与内部波特率之间的不匹配程度。如果发送器以过快或过慢的比特率传输数据，或者接收器内部产生的波特率没有相同的频率，那么接收器就无法与起始位同步。为了确保接收器不会错过下一帧起始位的采样，数据输入速率和内部接收器波特率不能相差太大，用它们之间的比值来描述波特率的误差范围。下面两个表格分别给出了普通模式下和倍速模式下容许的最大波特率误差范围。

Asynchronous Operational Range

The operational range of the Receiver is dependent on the mismatch between the received bit rate and the internally generated baud rate. If the Transmitter is sending frames at too fast or too slow bit rates, or the internally generated baud rate of the Receiver does not have the same base frequency, the Receiver will not be able to synchronize with the start bit.

To ensure that receiver will not miss sampling of next frame start bit, there must no big difference between incoming bit rate and internal transmit baud rate, ratio between them is used to descript tolerance range of baud rate. Below 2 lists show allowed max. baud rate tolerance range in normal mode and speed doubling mode.

Max baud rate tolerance range of receiver in normal mode

Data bit+ total parity bit	Max Total Error (%	Recommended Max

length		Receiver Error (%)
5	+6.7/-6.8	±3.0
6	+5.8/-5.9	±2.5
7	+5.1/-5.2	±2.0
8	+4.6/-4.5	±3.0
9	+4.1/-4.2	±1.5
10	+3.8/-3.8	±1.5

Max baud rate tolerance range of receiver in speed doubling mode

Data bit+ total parity bit length	Max Total Error (%	Recommended Max Receiver Error (%)
5	+5.7/-5.9	±2.5
6	+4.9/-5.1	±2.0
7	+4.4/-4.5	±1.5
8	+3.9/-4.0	±1.5
9	+3.5/-3.6	±1.0
10	+3.2/-3.3	±1.0

从表中可以看出，普通模式下波特率允许有更大的变化范围。上述推荐的波特率误差范围是假定接收器和发送器对最大总误差具有同等贡献的前提下得出的。产生接收器波特率误差的可能原因有两个。首先，接收器系统时钟的稳定性与工作电压和温度有关。使用晶振来产生系统时钟时一般不会有此问题，但使用内部振荡器时，系统时钟可能会有偏差。第二个原因是波特率发生器不一定能通过对系统时钟的分频来得到恰好想要的波特率。此时可以调整 UBRR 的值，使得误差低至可以接受。

Seeing from above list, in normal mode baud rate allows wider error range. The above recommended baud rate error range is based on assumption that receiver and transmitter makes same level contribution to max. total error. There are 2 reasons to generate receive baud rate error. Firstly, stability of receiver system clock is related to working voltage and temperature. Normally there is no error if system clock is generated by crystal oscillator, while system clock will have error if internal oscillator is used. Secondly, baud rate generator could not get preferred rate via prescale of system clock. In this case, error can decrease to acceptable level by adjusting UBRR value.

波特率设置及引入误差

对于标准晶振及谐振器频率来说，异步模式下的实际通信的波特率可通过波特率计算公式来获得，它与常用通信波特率之间的误差可用如下公式来计算：

$$Error[\%] = (Baudreal/Baud - 1)*100\%$$

其中，Baud 为常用的通信波特率，Baudreal 为通过计算公式算出来的波特率，带入波特率计算公式即可得到波特率误差与系统时钟 fsys 和波特率寄存器 UBRR 值之间的关系如下：

普通模式：

$$Error[\%] = (fsys/(16*(UBRR+1))/Baud - 1)*100\%$$

倍速模式：

$$Error[\%] = (fsys/(8*(UBRR+1))/Baud - 1)*100\%$$

当不考虑通信两边的时钟误差，即系统时钟 fsys 为标准时钟时，即可得到波特率误差 UBRR 值之间的关系。下表即为 16MHz 系统时钟下不同 UBRR 值设置下的波特率误差。

Baud Rate Setting and Incoming Error

For standard crystal and resonator frequencies, the actual used baud rates for asynchronous operation can be calculated by baud rate equation, tolerance between actual baud rate and normal used

baud rate can be calculated by as below:

Error[%] = (Baudreal/Baud – 1)*100%

In which, Baud indicates normal used communication baud rat, Baudreal is baud rate calculated by the above equation. Relation between baud rate error and system clock fsys as well as baud rate register UBRR is as below

Normal Mode:

Error[%] = (fsys/(16*(UBRR+1))/Baud – 1)*100%

Double Speed Mode:

Error[%] = (fsys/(8*(UBRR+1))/Baud – 1)*100%

If communication clock error is not considered, and if system clock fsys is standard clock, relationship in between each baud rate error UBRR value can be gotten. Below list shows baud rate error with different UBRR value in 16MHz system clock.

Error generated by setting UBRR in 16MHz system clock

Baud Rate（bps）	fsys = 16.000MHz			
	Normal Mode ((U2X = 0)		Double Speed Mode(U2X = 1)	
	UBRR	Error	UBRR	Error
2400	416	-0.1%	832	0.0%
4800	207	0.2%	416	-0.1%
9600	103	0.2%	207	0.2%
14.4K	68	0.6%	138	-0.1%
19.2K	51	0.2%	103	0.2%
28.8K	34	-0.8%	68	0.6%
38.4K	25	2.1%	34	-0.8%
57.6K	16	0.2%	51	0.2%
76.8K	12	0.2%	25	0.2%
115.2K	8	-3.5%	16	2.1%
230.4K	3	8.5%	8	-3.5%
250K	3	0%	7	0%
0.5M	1	0%	3	0%
1M	0	0%	1	0%

多处理器通信模式

置位 UCSRA 的多处理器通信模式（MPCM）位可以对 USART 接收器接收到的数据帧进行过滤。那些没有地址信息的帧将被忽略，也不会存入接收缓冲器。在一个多处理器系统中，各处理器通过相同的串行总线进行通信，这种过滤有效的减少了需要 CPU 处理的数据帧的数量。MPCM 位的设置不影响发送器的工作，但在多处理器通信的系统中，它的使用方法会有所不同。

如果接收器所接收的数据帧长度为 5 到 8 位，那么第一个停止位会用来表示当前帧包含的是数据还是地址信息。如果接收器所接收的数据帧长度是 9 位，那么由第 9 位来确定是数据还是地址信息。如果帧类型标志位为“1”，那么这是地址帧，否则为数据帧

在多处理器通信模式下，允许多个从处理器从一个主处理器接收数据。首先要通过解码地址帧来确定所寻址的是哪一个从处理器。被寻址的从处理器将正常接收后续的数据，而其他的从处理器则会忽略这些数据帧直到接收到下一个地址帧。

Multi-processor Communication Mode

Setting the Multi-processor Communication mode (MPCM) bit in UCSRA enables a filtering function of incoming frames received by the USART Receiver. Frames that do not contain address information will be ignored and not put into the receive buffer. This effectively reduces the number of incoming frames that has

to be handled by the CPU, in a system with multiple MCUs that communicate via the same serial bus. The Transmitter is unaffected by the MPCM setting, but has to be used differently when it is a part of a system utilizing the Multi-processor Communication mode.

If the Receiver is set up to receive frames that contain 5 to 8 data bits, then the first stop bit indicates if the current frame contains data or address information. If the Receiver is set up for frames with nine data bits, then the ninth bit

is used for identifying address and data frames. When the frame type bit is one, the frame contains an address. When the frame type bit is zero the frame is a data frame.

The Multi-processor Communication mode enables several slave MCUs to receive data from a master MCU. This is done by first decoding an address frame to find out which MCU has been addressed. If a particular slave

MCU has been addressed, it will receive the following data frames as normal, while the other slave MCUs will ignore the received frames until another address frame is received.

对于一个作为主机的处理器来说，它可以使用 9 位数据帧格式，并用第 9 位数据来标识帧格式。在这种通信模式下，从处理器也必须工作于 9 位数据帧格式。

下面即为多处理器通信模式下进行数据交换的步骤：

1. 所有从处理器都工作在多处理器通信模式（置位 MPCM）；
2. 主处理器发送地址帧，所有从处理器都接收此帧。从处理器 UCSRA 寄存器的 RXC 位正常置位；
3. 每个从处理器都读取 UDR 寄存器的内容，解码地址帧来确定是否被选中。如果选中，就清零 UCSRA 寄存器的 MPCM 位，未被选中，则保持 MPCM 为“1”并等待下一个地址帧的到来；
4. 被寻址的从处理器接收所有的数据帧，直到收到一个新的地址帧。未被寻址的从处理器忽略这些数据帧；
5. 被寻址的从处理器收到最后一个数据帧后，置位 MPCM 位，并等待下一个地址帧的到来。然后从第二步骤重复进行。

使用 5 到 8 位数据的帧格式是可以的，但是不切实际，因为接收器必须在使用 n 和 n+1 帧格式之间进行切换。由于接收器和发送器使用相同的字符长度设置，这种设置使得全双工操作变得很困难。如果使用 5 到 8 位数据的帧格式，发送器应该设置两个停止位，其中第一个停止位被用于判断帧类型。

For an MCU to act as a master MCU, it can use a 9-bit character frame format. The ninth bit is used as frame format flag. In this mode, slave MCU must also work with 9-bit character frame format.

Below is the procedures of data switch in Multi-processor Communication Mode:

1. All Slave MCUs are in Multi-processor Communication mode (MPCM is set).
2. The Master MCU sends an address frame, and all slaves receive this frame. In the Slave MCUs, the RXC bit in UCSRA will be set as normal.
3. Each Slave MCU reads the UDR Register, decoding address frame determines if it has been selected. If so, it clears the

MPCM bit in UCSRA, otherwise it waits for the next address frame and keeps the MPCM setting to logic one

4. The addressed slave MCU will receive all data frames until a new address frame is received. The other Slave will ignore the data frames.
5. When the last data frame is received by the addressed slave MCU, the addressed MCU sets the MPCM bit and waits for a new address frame from master. The process then repeats from 2

Using any of the 5- to 8-bit character frame formats is possible, but impractical since the Receiver must change between using n and n+1 character frame formats. This makes full-duplex operation difficult since the Transmitter and Receiver uses the same character size setting.

If 5- to 8-bit character frames are used, the Transmitter must be set to use two stop bit since the first stop bit is used for indicating the frame type.

Register Definition

UCSRA—USART control and status register A

UCSRA—USART control and status register A								
Address:0xC0				Default value: 0x20				
Bit	7	6	5	4	3	2	1	0
Name	RXC	TXC	UDRE	FE	DOR	PE	U2X	MPME
R/W	R	R/W	R	R	R	R	R/W	R/W

Bit	Name	Comment
7	RXC	Receive Complete Flag When RXC is logic one, it indicates that there is unread data in receive buffer. When RXC is logic zero, it indicates that there is no unread data in receive buffer. If receiver is disabled, receive buffer is flushed which makes RXC cleared. If receive complete interrupt enable bit RXCIE is logic one, RXC can be used to generate receive complete interrupt.
6	TXC	Transmit complete flag bit TXC is set when data are send out from transmit shift register and transmit buffer is empty. TXC is cleared automatically when transmit complete interrupt is executed, or by writing TXC to logic one. When transmit complete interrupt enable bit TXCIE is logic one, TXC is used to generate transmit complete interrupt.
5	UDRE	Data register empty flag bit When UDRE is logic one, it indicates that USART transmit data buffer is empty and can write data into When UDRE is logic zero, it indicates that USART transmit data buffer is full and can not write data into When data register empty interrupt enable bit UDRIE is logicone, UDRE can be used to generate data register empty interrupt
4	FE	Frame Error Flag bit When FE is logic one, it indicates that receive data buffer is receiving frame with error, that is today the first stop bit is logic zero. If FE is logic zero, it indicates that receive data buffer is receiving frame without any error, saying the first stop bit is logic one. Once set, FE is effective till UDR is read. If write to UCSRA, FE bit must be zero.
3	DOR	Data Overrun bit If receive buffer is full (include 2 data), there is data in receive shift register. If a new start bit is detected, data overrun occurs, DOR is set till UDR is read. f write to UCSRA, DOR bit must be zero.
2	PE	Parity Error flag When parity error check is enable (UPM1 is set to logic one), and frame in receive buffer has parity error, PE is set till UDR is read. If write to UCSRA, PE bit must be zero.
1	U2X	Double speed transmit enable bit

		When U2X is logic one, transmit speed rate is double in asynchronous communication mode. When U2X is logic zero, transmit speed rate is normal in asynchronous communication mode. This bit is only valid in asynchronous mode, it is cleared to zero in synchronous mode.
0	MPCM	Multi processor communication mode enable bit If MPCM is set, multi processor communication mode is active. After MPCM is set, incoming frame in USART receiver that does not contain address info will be ignored. Transmitter is not affected by MPCM setting.

UCSRB—USART control and status register B

UCSRA—USART control and status register B								
Address:0Xc1				Default value: 0x00				
Bit	7	6	5	4	3	2	1	0
Name	RXCIE	TXCIE	UDRIE	RXEN	TXEN	UCSZ2	RXB8	TXB8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W
Bit	Name	Comment						
7	RXCIE	Receive complete interrupt enable bit If enabled, RXC interrupt is enabled, if clear, RXC interrupt is disabled. When RXCIE is logic one, globe interrupt is enabled. When RXC in UCSRA register is logic one, USART receive complete interrupt occurs.						
6	TXCIE	Transmit complete interrupt enable bit If enabled, TXC interrupt is enabled, if clear, TXC interrupt is disabled. When RTXCIE is logic one, globe interrupt is enabled. When TXC in UCSRA register is logic one, USART receive complete interrupt occurs.						
5	UDRIE	Data register empty interrupt enable bit If enabled, UDRE interrupt is enabled, if cleared, UDRE interrupt is disabled. When UDRIE is logic one, globe interrupt is enabled. When UDRE in UCSRA register is logic one, USART data register empty interrupt occurs.						
4	RXEN	Receive enable bit If enabled, USART receiver is active. USART receive will override general IO port of RxD pin. If disabling receiver, receive buffer will be flushed, and FE, DOR and PE flag becomes not effective.						
3	TXEN	Transmit enable bit If enabled, USART transmitter is active. USART transmit will override general IO port of TxD pin. If disabling receiver, USART can only really disabled till all data transmit is complete.						
2	UCSZ2	Character length control 2nd bit Data bit number contained by frames are set by UCSZ2 and UCSZ1:0 in UCSRC register together.						
1	RXB8	Receive data 8th bit If frame is the 9th bit, RXB8 is the highest bit of receiving data. Before reading low 8th bit in UDR, must read RXB8 firstly.						
0	TXB8	Transmit data 8th bit If frame is the 9th bit, TXB8 is the highest bit of transmitting data. Before reading low 8th bit in UDR, must read TXB8 firstly.						

UCSRC– USART Control and Status Register C

UCSRA—USART control and status register C								
Address:0XC2				Default value: 0x06				
Bit	7	6	5	4	3	2	1	0
Name	UMSEL1	UMSEL0	UPM1	UPM0	USBS	UCSZ1	UCSZ0	UCPOL
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
7:6	UMSEL1:0	USART mode selection bit UMSEL select synchronous or asynchronous mode operation						
		UMSEL	Mode					
		0	USART Asynchronous operation mode					
		1	USART Synchronous operation mode					
		2	SPI Slave operation mode					
		3	SPI Master operation mode					
5:4	UPM1:0	Parity check mode selection bit High UPM1 to enable or disable parity check, low UPM0 to even check or odd check						
		UPM1:0	Comment					
		0	Disable parity check					
		1	reserved					
		2	Enable odd check					
		3	Enable even check					
3	USBS	Stop bit selection, to select bit number of stop bit						
		USBS	bit number of stop bit					
		0	1					
		1	2					
2:1	UCSZ1:0	Frame character length selection bit Data bit number contained by frames is set by UCSZ2 in UCSRB and UCSZ1:0 together.						
		UCSZ2:0	Frame length					
		0	5 bit					
		1	6 bit					
		2	7 bit					
		3	8 bit					
		4	reserved					
		5	reserved					
		6	reserved					
		7	9 bit					
0	UCPOL	Clock polarity selection bit In USART synchronous mode, UCPOL determines the relationship between output data change & input data sampling and synchronous clock XCK. In asynchronous mode, it has nothing to do with UCPOL, this bit will be cleared.						
		UCP01	Transmit data change	Receive data sampling				
		0	Rising edge of XCK	Falling edge of XCK				

		1	Falling edge of XCK	Rising edge of XCK

UBRRL – USART baud rate register high

UBRRL – USART baud rate register high								
Address:0XC5				Default value: 0x00				
Bit	7	6	5	4	3	2	1	0
Name	-	-	-	-	UBRR11	UBRR10	UBRR9	UBRR8
R/W	-	-	-	-	R/W	R/W	R/W	R/W
Bit	Name	Comment						
7:4	-	reserved						
3:0	UBRR[11:8]	USART baud rate register high USART baud register includes UBRRL and UBRRH, they together are used to set up communication baud rate. UBRR = {UBRR[11:8], UBRRL}						
		Working Mode			Baud Rate Equation			
		Asynchronous normal mode			BAUD = fsys/(16*(UBRR+1))			
		Asynchronous speed doubling mode			BAUD = fsys/(8*(UBRR+1))			
		Synchronous master mode			BAUD = fsys/(2*(UBRR+1))			

UDR—USART Data Register

UDR—USART Data Register								
Address:0XC6				Default value: 0x00				
Bit	7	6	5	4	3	2	1	0
Name	UDR7	UDR6	UDR5	UDR4	UDR3	UDR2	UDR1	UDR0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	Name	Comment						
7:0	UDR	USART transmitted and received date USART transmit data buffer and receive data buffer share USART data register UDR. Writing data to UDR is to write transmit data buffer, while reading data from UDR is to read receive data buffer. In frame of 5 to 8 bit mode, transmitter will ignore unused 9th bit while receiver set them to logic zero. Only when UDRE flag in UCSRA register is logic one, it is allowed of write operation to transmit buffer, otherwise operations to transmitter will be wrong. When transmit shift register is empty, transmitter will load transmit buffer data to transmit shift register and then output data by serial from TxD pin. Receive buffer includes a 2 level FIFO. Once receive buffer is read, FIFO will change its state.							

- 全双工操作，三线同步数据传输
- 主机或从机操作
- 支持全部四种工作模式（模式 0、1、2 和 3）
- 低位或高位首先传输（可配置的数据传输顺序）
- 队列操作（双缓冲器）
- 高分辨率波特率产生器

USART0 – SPI Working Mode

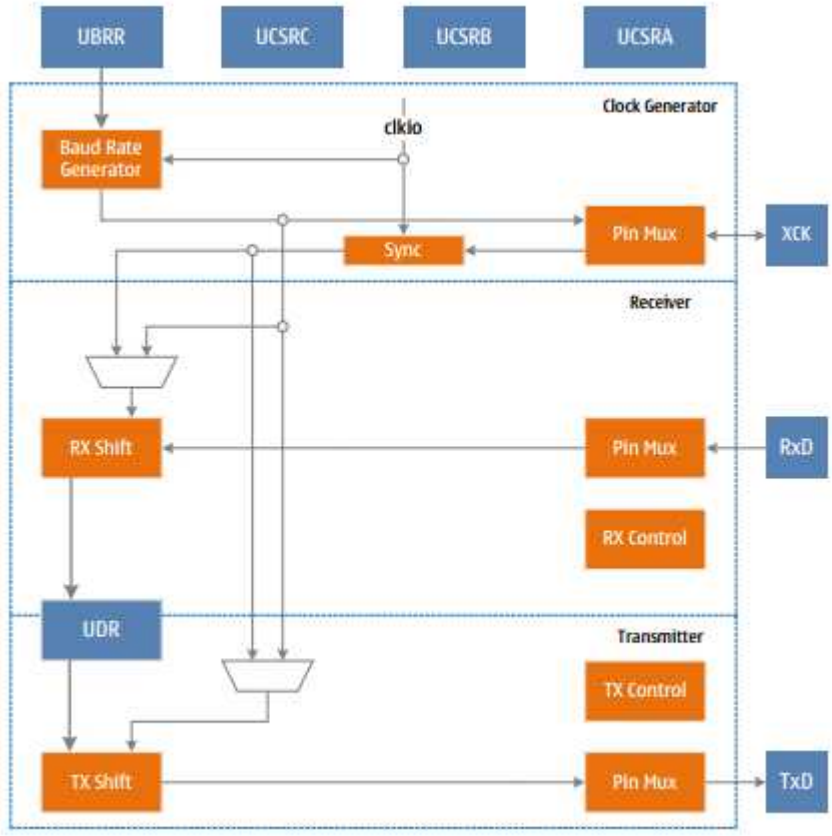
- Full Duplex, Three-wire Synchronous Data Transfer
- Master or Slave Operation
- Supports all four SPI Modes of Operation (Mode 0, 1, 2, and 3)
- LSB First or MSB First Data Transfer (Configurable Data Order)
- Queued Operation (Double Buffered)
- High Resolution Baud Rate Generator

综述

当设置 USCRC 的 UMSEL1 位为“1”时，使能 SPI 工作模式，用 USPI 来表示。此 SPI 模块为三线 SPI 工作模式，与四线 SPI 模式相比，缺少从机选择线，其它三根线均一致。USPI 占用 USART 的资源，包括发送和接收移位寄存器和缓冲器，以及波特率发生器。奇偶校验产生和检查逻辑，数据和时钟恢复逻辑均无效。控制和状态寄存器的地址是一样的，不过寄存器位的功能会随着 SPI 工作模式的需要而发生改变。

Overview

Setting both UMSEL1 in USCRC to one enables SPI working mode, which is indicated by USPI. In this case SPI is 3-wire without slave selection wire compared with 4-wire SPI mode, the other 3 wires are consistent. USPI takes up USART resource including transmit/receive shift register and buffer as well as baud rate generator. The parity generator and checker, the data and clock recovery logic are all invalid. Control and status register address are the same but function of register bit will change as per SPI modes.



USART in SPI 结构图

时钟产生

当 SPI 工作在主机模式时，需要提供通信用的时钟，复用 USART 的波特率发生器来产生这个时钟。该时钟从 XCK 引脚输出，因此 XCK 引脚的数据方向寄存器（DDR_XCK）必须设置为“1”。

时钟频率有以下计算公式决定：

BAUD = fsys/(2*(UBRR+1))

当 SPI 工作在从机模式时，通信时钟由外部主机提供，从 XCK 引脚输入，因此 XCK 引脚的数据方向寄存器（DDR_XCK）必须设置为“0”。

Clock Generation

When SPI is in master mode, clock for communication is in need of, alternative USART baud rate generator is used to generate this clock, which is output from XCK pin, so data direction register (DDR_XCK) of XCK pin must be set to logic one.

Clock frequency is determined by below formular:

BAUD = fsys/(2*(UBRR+1))

When SPI is in slave mode, clock for communication is given by master and input from XCK pin, so data direction register (DDR_XCK) of XCK pin must be set to logic zero.

SPI 数据模式和时序

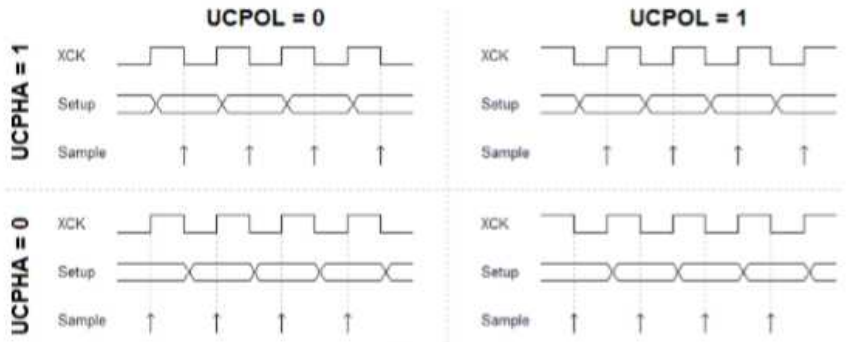
SPI 有四种时钟相位和极性的组合方式，有控制位 UCPHA 和 UCPOL 来决定，具体的控制如 下表和下图所示：

SPI data mode dand Timing

There are four combinations of phase and polarity , which are determined by control bits UCPHA and UCPOL, detailed control is shown as below:

SPI Working Mode

SPI Mode	UCPOL	UCPHA	Leading Edge	Trailing Edge
0	0	0	Sample (Rising)	Setup (Falling)
1	0	1	Setup (Rising)	Sample (Falling)
2	1	0	Sample (Falling)	Setup (Rising)
3	1	1	Setup (Falling)	Sample (Rising)



SPI 工作模式图示

帧格式

SPI 的一个串行帧可以由最低位或最高位开始，到最高位或最低位结束，总共 8 位数据。一 帧结束以后，可以紧接着传输新的一帧，传输结束即可拉高数据线为空闲状态。

Frame Formats

A frame starts with the least or most significant data bit. Then the next data bits, up to a total of eight, are succeeding, ending with the most or least significant bit accordingly. When a complete frame is transmitted, a new frame can directly follow it, or the communication line can be set to an idle (high) state.

数据传输

SPI 置 UCSRB 寄存器的 TXEN 位为“1”来使能发送器，TxD 引脚被发送器占用来发送串行输出 数据。此时接收器可以不使能.

SPI 置 UCSRB 寄存器的 RXEN 位为“1”来使能接收器，RxD 引脚被接收器占用来接收串行输入 数据。此时发送器须使能。

SPI 发送和接收都使用 XCK 来当作传输时钟。

进行通信之前首先要对 SPI 进行初始化。初始化过程通常包括波特率的设定，帧数据位传 输顺序的设定，以及根据需要使能接收器或发送器。对于中断驱动的 SPI 操作，在初始化时要清零全局中断标志并禁止 SPI 的所有中断。

在进行重新初始化比如改变波特率或帧结构时，必须确保没有数据传输。TXC 标志位可以 用来检测发送器是否完成了所有传输，RXC 标志位可以用来检测接收缓冲器中是否还有数 据未被读出。如果 TXC 标志位用作此用途，在每次发送数据之前（写 UDR 寄存器之前）必 须清零 TXC 标志位。初始化 SPI 以后，往 UDR 寄存器写入数据即可开始数据传输。由于发送器控制着传输时钟， 发送和接收数据均是如此操作。当发送移位寄存器准备好发送新一帧数据的时候，发送器就 会把写入到 UDR 寄存器的数据从发送缓冲器移到发送移位寄存器里并发送出去。为了保证 输入缓冲器和发送数据同步，每发送一个字节的数据后都必须读取一次 UDR 寄存器。当发 生数据溢出时，最近收到的数据将会丢失，而不是最早收到的数据。

Data Transfer

Transmitter is enable by setting TXEN in UCSRB register to logic one by SPI, TxD pin is taken up by transmitter to transmit serial output data. In this case receiver is not necessary to be enabled.

Receiver is enable by setting RXEN in UCSRB register to logic one by SPI, RxD pin is taken up by receiver to transmit serial input data. In this case receiver must be enabled.

Both SPI receive and transmit are using XCK as transfer clock.

After initialization the USART is ready for doing data transfers. Normally initialization process includes baud rate setting, frame transmit timing setting and enabling receiver or transmitter as per demands. For interrupt-driven SPI operation during initialization, globe interrupt flag must be cleared and disable all interrupt in SPI.

Before re-initilization, i.e. change baud rate or frame construction, must make sure that there is no ongoing data under transmission. TXC flag can be used to detect if transmitter has completed all transfer or not, while RXC flag is used to detect if receive buffer cotains unread data or not. If TXC is used for this purpose, TXC flag bit must be cleared before each data transmit (before writing UDR register).
After SPI initialization, data transfer starts once after writing data to UDR register. Due to that transmitter controls transfer clock, transit and receive data are the same operation. When transmit shift register is ready to send a new frame, transmitte will move data written to UDR register from transmit buffer to transmit shift register and send it out. To ensure of data of input buffer and transmit is synchronized, must read UDR reigster after each transmit of one byte data. If data overruns, the latest receive data instead of data receive earlier would loss.

发送标志位与中断

SPI 发送器有两个标志位：SPI 数据寄存器空标志 UDRE 和传输结束标志 TXC，两个标志位都 可以产生中断。

数据寄存器空标志 UDRE 用来表示发送缓冲器是否可以写入一个新的数据。该位在发送缓 冲器空时被置“1”，满时被置“0”。当 UDRE 位为“1”时，CPU 可以往数据寄存器 UDR 写入新 的数据，反之则不能。

当 UCSRB 寄存器中的数据寄存器空中断使能位 UDRIE 为“1”时，只要 UDRE 被置位（且全局 中断使能），就将产生 SPI 数据寄存器空中断请求。对寄存器 UDR 执行写操作将清零 UDRE。当采用中断方式传输数据时，在数据寄存器空中断服务程序中必须写入一个新的数 据到 UDR 以清零 UDRE，或者是禁止数据寄存器空中断。否则一旦该中断服务程序结束， 一个新的中断将再次产生。

当整个数据帧被移出发送移位寄存器，同时发送寄存器中又没有新的数据时，发送结束标 志 TXC 将被置位。当 UCSRB 上的发送结束中断使能位 TXCIE（且全局中断使能）置“1”时， 随着 TXC 标志位被置位，SPI 发送结束中断将被执行。一旦进入中断服务程序，TXC 标志位 即被自动清零，CPU 也可以对该位写“1”来清零。

Transmitter and Receiver Flags and Interrupts

SPI transmitter has two flag bit: SPI data register empty bit UDRE and transmit complete falg TXC, both of them can generate interrupt.
Data register empty bit UDRE indicates if new data can be written to transmit buffer or not. This bit is set to one when transmit buffer is empty while zero when it is full. If UDRE is set to logic one, CPU can write new data to data register UDR, otherwise cannot.
When Data register empty interrupt enable bit UDRIE located in UCSRB register is set to logic one, SPI data register empty interrupt request is generated only if UDRE is set and globe interrupt is enabled. UDRE will be cleared if write operation to register UDR is exceucted. When data transfer is driven by interrupt, must write new data to UDR in register empty interrupt routine in order to clear UDRE, or disable data register empty inteerrupt, otherwise a new interrupt will occur if this interrupt routine is complete.
When a complete frame is moved out of transmit shift register, meanwhile there is no new data in transmit register, transmit complete flag TXC will be set. When transmit complete interrupt eanble bit TXCIE in UCSRB is set to one (meanwhile globe interrupt is eanbled), SPI transmit complete interrupt will be executed once TXCC flag bit is set. Once in interrupt routine, TXC flag bit will be cleared automatically, CPU can write logic one to this bit to clear.

禁止发送器

当 TXEN 清零后，只有等所有的数据都发送完成以后发送器才能够真正禁止，即发送移位 寄存器与发送缓冲寄存器中都没有要传送的数据。发送器禁止以后，TxD 引脚恢复其通用 IO 功能。

Disable transmitter

After TXEN is cleared, transmitter can only be disabled really when all data have been transmitted, that is to say there is no data for transmision in transmit shift register and transmit buffer register. When transmitter is disabled, TxD pin will resume its genearal IO function.

接收结束标志及中断

SPI 接收器有一个标志位：接收结束标志 RXC，用来表明接收缓冲器中是否有未被读出的数 据。当接收缓冲器中有未被读出的数据时，此位为“1”，反之为“0”。如果接收器被禁止， 接收缓冲器会被刷新，RXC 也会被清零。置位 UCSRB 的接收结束中断使能位 RXCIE 后，只 要 RXC 标志被置位（且全局中断被使能），就会产生 SPI 接收结束中断。使用中断方式进行 数据接收时，数据接收结束中断服务程序必须从 UDR 读取数据来清零 RXC 标志，否则只要中断处理程序一结束，一个新的中断就会产生。

Receive complete flag and interrupt

SPI receiver has one flag: receive complete flag which indicates if there is unread data in receive buffer, when there is, this flag is set to one, otherwise it is set to zero. If receiver is disabled, receive buffer will be flushed and RXC will also be cleared.

When receive complete interrupt enable bit RXCIE in UCSRB is set, SPI receive complete interrupt will be generated if RXC flag is set and globe interrupt is set.

When data transfer is driven by interrupt, data receive complete interrupt routine must read data from UDR to clear RXC flag. Otherwise a new interrupt will occur once if interrupt handle rountine is complete.

禁止接收器

与发送器相比，禁止接收器即刻起作用。正在接收的数据将丢失。禁止接收器（RXEN 清 零）后，接收器将不再占用 RxD 引脚，接收缓冲器也会被刷新。

Disable receiver

Compared with transmitter, disabling receiver becomes effective immediately, ongoing received data will be lost. Afer receiver is disabled (RXEN is cleared), receiver will not take up RxD pin anymore, receive buffer will be flushed also.

Register Definition

USART Register List

Register	Address	Default Value	Comment
UCSRA	0xC0	0x20	USPI control and status register A
UCSRB	0xC1	0x00	USPI control and status register B
UCSRC	0xC2	0x06	USPI control and status register C
UBRRL	0xC4	0x0	USPI baud rate register low
UBRRH	0xC5	0x0	USPI baud rate register high
UDR	0xC6	0x0	USPI data register

UCSRA--USPI control and status register A

UCSRA--USPI control and status register A								
Address: 0xC0				Default value: 0x20				
Bit	7	6	5	4	3	2	1	0
Name	RXC	TXC	UDRE	-	-	-	-	-
R/W	R	R/W	R	-	-	-	-	-
Bit	Name	Comment						
7	RXC	Receive complete flag When RXC is logic one, it indicates there is unread data in receive buffer. When RXC is logic zero, it indicates there is no unread data in receive buffer. If receiver is disabled, receive buffer will be flushed which results in clear of RXC. If receive complete interrupt enable bit RXCIE is one, RXC can be used to generate receive complete interrupt.						

6	TXC	Transmit complete flag TXC is set when data in transmit shift register is moved out and transmit buffer is empty. TXC will be cleared automatically once transmit complete interrupt is executed, or it can be done by writing logic one to TXC. When transmit complete interrupt enable bit TXCIE is one, TXC can be used to generate transmit complete interrupt.
5	UDRE	Data register flag When UDRE is one, it indicates transmit data buffer is empty and it is possible to write data. When UDRE is zero, it indicates transmit data buffer is full and it is mpossible to write data. When data register interrupt enable it UDRIE is one, UDRE is used to generate data register empty interrupt.
4:0	-	Reserved in USPI mode.

UCSRB--USPI control and status register B

UCSRB--USPI control and status register B								
Address: 0xC1				Default value: 0x00				
Bit	7	6	5	4	3	2	1	0
Name	RXCIE	TXCIE	UDRIE	RXEN	TXEN	-	-	-
R/W	R/W	R/W	R/W	R/W	R/W	-	-	-

Bit	Name	Comment
7	RXCIE	Receive complete interrupt enable After setting, RXC interrupt is enabled, after clear, RXC interrupt is Disabled. When RXCIE is one, globe interrupt is enabled. USPI receive complete interrupt can be generated when RXC in UCSRA register is logic one.
6	TXCIE	Transmit complete interrupt enable After setting, TXC interrupt is enabled, after clear, TXC interrupt is Disabled. When TXCIE is one, globe interrupt is enabled. USPI transmit complete interrupt can be generated when TXC in UCSRA register is logic one.
5	UDRIE	Data register empty interrupt enable After setting, UDRE interrupt is enabled, after clear, UDRE interrupt is Disabled. When UDRE is one, globe interrupt is enabled. USPI data register empty interrupt can be generated when UDRE in UCSRA register is logic one.
4	RXEN	Receive enable bit After setting, USPI receiver is actived. General IO port of RxD pin will be overrided by USPI receive. Disabling receiver will flush receive buffer.
3	TXEN	Transmit enable bit After setting, USPI receiver will start. General IO port of RxD pin will be overrided by USPI transmit. When TXEN is cleared, USART is disabled really only when all ongoing data under transmision is complete.
2:0	-	Reserved in USPI

UCSRC--USPI control and status register C

UCSRC--USPI control and status register C

Address: 0xC2				Default value: 0x86				
Bit	7	6	5	4	3	2	1	0
Name	UMSEL1	UMSEL0	-	-	-	DORO	UCPHA	UCPOL
R/W	R/W	R/W	-	-	-	R/W	R/W	R/W
Bit	Name	Comment						
7:6	UMSEL1:0	USART mode selection bit UMSEL select sysnchronous or asynchronous operation mode						
		UMSEL	Mode					
		0 1 2 3	USART Asynchronous operation mode USART Synchronous operation mode SPI slave operation mode SPI master operation mode					
5:3	-	Reserved in USPI						
2	DORO	Data transmit order selection bit						
		DORD	Order					
		0	Hight bit to transfer first					
		1	Low bit to transfer first					
1	UCPHA	Clock phase selection UCPHA determinsif data sample happens in leading edge or trailing edge.						
		UCPHA	Order					
		0	Hight bit to transfer first					
		1	Low bit to transfer first					
0	UCPOL	Clock polarity selection UCPOL determins if data change and sample happens in leading edge or trailing edge.						
		UCPOL	Transmitted data change	Received data samle				
		0	Leading edge of XCK	Trailing edge of XCK				
		1	Trailing edge of XCK	Leading edge of XCK				

UBRRL--USPI baud rate register low

UBRRL--USPI baud rate register low								
Address: 0xC4				Default value: 0x00				
Bit	7	6	5	4	3	2	1	0
Name	UBRR7	UBRR6	UBRR5	UBRR4	UBRR3	UBRR2	UBRR1	UBRR0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	Name	Comment						
7:0	UBRR[7:0]	Low byte of USPI baud rate register, it includes UBRRL and UBRRH, they are combined to set baud rate of communication.						

UBRRH--USPI baud rate register high

UBRRH--USPI baud rate register high								
Address: 0xC5				Default value: 0x00				
Bit	7	6	5	4	3	2	1	0
Name	-	-	-	-	UBRR11	UBRR10	UBRR9	UBRR8

R/W	-	-	-	-	R/W	R/W	R/W	R/W
Bit	Name	Comment						
7:4	-	Reserved in USPI						
3:0	UBRR[11: 8]	High byte of USPI baud rate register, it includes UBRRL and UBRRH, they are combined to set baud rate of communication. UBRR = {UBRR[11:8], UBRRL}						
		Working Mode	Baud rate equation					
		Slave mode	Baud reate determined by external master configuration					
		Master Mode	BAUD = fsys/(2*(UBRR+1))					

UDR--USPI data register

UDR--USPI data register								
Address: 0xC6				Default value: 0x00				
Bit	7	6	5	4	3	2	1	0
Name	UDR7	UDR6	UDR5	UDR4	UDR3	UDR2	UDR1	UDR0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	Name	Comment						
7:0	UDR	Data received and transmitted by USPI USPI transmit data buffer and receive data buffer share USPI data register UDR. Writing data to UDR is to writing to transmit data buffer, whle read data from UDR is to read from receive data buffer. In frame of 5--8 bit mode, transmitter ignores unused 9th bit, while receiver set the to zero. Write operation to transmit buffer is done only if UDRE flag in UCSRA register is logic one, otherwise operations of transitter will go wrong. When transmit shift register is empty, transitter will load data of transmit buffer to transmit shift register, then output data by serial from TxD pin. Receive buffer includes a 2 level FIFO. Once receive buffer is read, FIFO will change its state.							

TWI – 双线串行总线(I2C)

- 简单且强大而灵活的通讯接口，只需要 2 线
- 支持主机和从机操作
- 器件可以工作于发送器模式或接收器模式
- 7 位地址空间允许有 128 个从机
- 支持多主机仲裁
- 高达 400Kbps 的数据传输率
- 完全可编程的从机地址以及公共地址
- 睡眠模式下地址匹配时可以唤醒

2-wire Serial Interface （I2C）

- Simple Yet Powerful and Flexible Communication Interface, only two Bus Lines Needed
- Both Master and Slave Operation Supported
- Device can Operate as Transmitter or Receiver
- 7-bit Address Space Allows up to 128 Different Slave Addresses
- Multi-master Arbitration Support

- Up to 400kHz Data Transfer Speed
- Fully Programmable Slave Address with General Address
- Address Recognition Causes Wake-up When AVR is in Sleep Mode

TWI 总线介绍

两线串行接口 TWI 很适合于典型的处理器应用。TWI 协议允许系统设计者只用两根双向的传输线就可以将 128 个不同的设备互连到一起。这两根线是时钟 SCL 和数据 SDA。外部硬件只需要在每根线上接两个上拉电阻。所有连接到总线上的设备都有自己的地址。 TWI 协议解决了总线仲裁的问题。

2-wire Serial Interface Bus Definition

The 2-wire Serial Interface (TWI) is ideally suited for typical microcontroller applications. The TWI protocol allows the systems designer to interconnect up to 128 different devices using only two bi-directional bus lines, one for clock (SCL) and one for data (SDA). The only external hardware needed to implement the bus is two pull-up resistors for each of the TWI bus lines. All devices connected to the bus have individual addresses, and mechanisms for resolving bus contention are inherent in the TWI protocol.

TWI 术语

TWI Terminology

The following definitions are frequently encountered in this section.

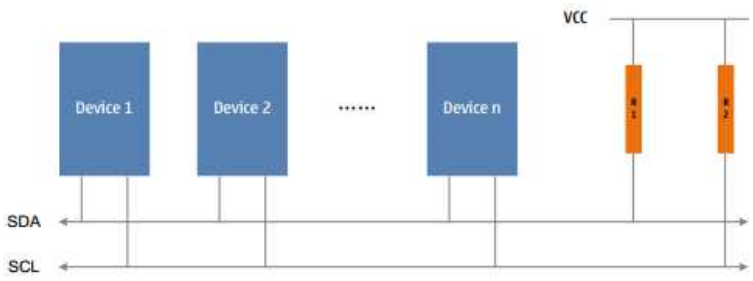
Term	Description
Master	The device that initiates and terminates a transmission. The Master also generates the SCL clock.
Slave	The device addressed by a Master.
Transmitter	The device placing data on the bus.
Receiver	The device reading data from the bus.

电气连接

如下图所示，TWI 接口的两根线都通过上拉电阻与正电源连接。所有 TWI 兼容器件的总线驱动都是漏极开路或集电极开路的，这样就实现了对接口操作的线与功能。当 TWI 器件输出为“0”时，TWI 总线会产生低电平。当所有的 TWI 器件输出为三态时，总线允许上拉电阻将电压拉高。为保证所有的总线操作，凡是与 TWI 总线连接的器件都必须上电。

Electrical Interconnection

As depicted below, both bus lines are connected to the positive supply voltage through pull-up resistors. The bus drivers of all TWI-compliant devices are open-drain or open-collector. This implements a wired-AND function which is essential to the operation of the interface. A low level on a TWI bus line is generated when TWI devices output a zero. A high level is output when all TWI devices tri-state their outputs, allowing the pull-up resistors to pull the line high. Note that all AVR devices connected to the TWI bus must be powered in order to allow any bus operation.



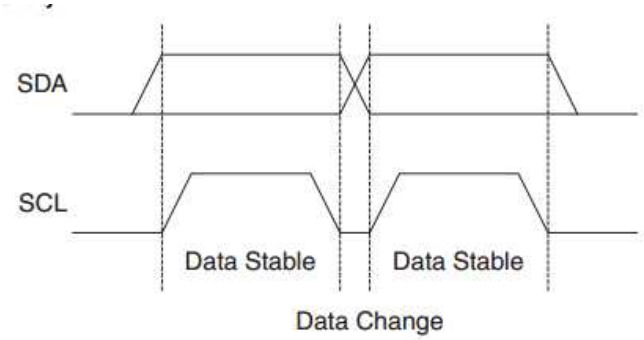
TWI 总线互连图

数据传输和帧结构

TWI 总线上的每一位数据传输都是和时钟同步的。当时钟线为高时，数据线上的电平必须保持稳定，除非是为了产生开始或停止状态。

Data Transfer and Frame Format

Each data bit transferred on the TWI bus is accompanied by pulse on the clock line. The level of the data line must be stable when the clock line is high. The only exception to this rule is for generating start and stop conditions.



开始和停止状态

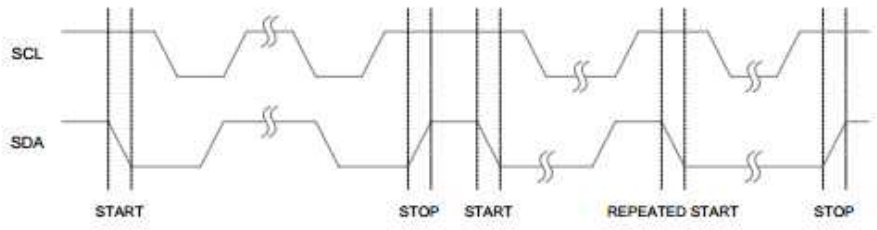
TWI 的传输由主机来启动和停止。主机在总线上发出 START 状态以启动数据传输，发出 STOP 状态以停止数据传输。在 START 和 STOP 状态之间，总线被认为是忙碌的，不允许其它主机试图占用总线的控制权。有一种特殊情况只允许发生在 START 和 STOP 状态之间产生一个新的 START 状态，这被称为 REPEATED START 状态，适用于当前主机在不放弃总线控制的情况下启动新的传输。REPEATED START 之后直到下一个 STOP 之前，总线仍然被认为是忙碌的。这与 START 是一致的，因此在本文档中，如果没有特殊说明，均采用 START 来表述 START 和 REPEATED START。如下图所示，START 和 STOP 条件是在 SCL 线为高时，改变 SDA 线的电平状态。

START and STOP Conditions

The Master initiates and terminates a data transmission of TWI. The transmission is initiated when the Master issues a

START condition on the bus, and it is terminated when the Master issues a STOP condition. Between a START and a STOP condition, the bus is considered busy, and no other master should try to seize control of the bus. A special case occurs when a new START condition is issued between a START and STOP condition. This is referred to as a REPEATED START condition, and is used when the Master wishes to initiate a new transfer without relinquishing control of the bus. After a REPEATED START, the bus is considered busy still until the next STOP. This is identical to the START behavior, and therefore START is used to describe both START and REPEATED START in the remainder of this datasheet, unless otherwise noted.

As depicted below, START and STOP conditions are signaled by changing the level of the SDA line when the SCL line is high.



START、REPEATED START 和 STOP 状态图

地址包格式

所有 TWI 总线上传输的地址包都是 9 位数据长度，由 7 位地址，1 位 READ/WRITE 控制位和 1 位应答位组成。当 READ/WRITE 位为“1”，则执行读操作；当 READ/WRITE 位为“0”时，执行写操作。从机被寻址后，必须在第 9 个 SCL（ACK）周期通过拉低 SDA 线做出应答。若该从机忙或有其它原因无法响应主机，则应在 ACK 周期保持 SDA 线为高。然后主机可以发出 STOP 状态或 REPEATED START 状态重新开始发送。

地址包包括一个从机地址和一个读或写控制位，分别用 SLA+R 或 SLA+W 来表示。

地址字节的 MSB 位首先发生。除了保留地址“00000000”被留用作广播呼叫以及所有形如“1111xxxx”格式的地址需要保留作将来使用外，其它从机地址可由设计者自由分配。

当发生广播呼叫时，所有的从机应在 ACK 周期通过拉低 SDA 线来做出应答。当主机需要发送相同的信息给多个从机时可以使用广播功能。当广播呼叫地址加上 WRITE 位被发送到总线上以后，所有需要响应该广播呼叫的从机将在 ACK 周期拉低 SDA 线。所有这些响应了广播呼叫的从机将会接收紧跟的数据包。需要注意的是，发送广播呼叫地址加上 READ 位是没有意义的，因为如果几个从机同时发送不同的数据会带来总线冲突。

Address Packet Format

All address packets transmitted on the TWI bus are 9 bits long, consisting of 7 address bits, one READ/WRITE control bit and an acknowledge bit. If the READ/WRITE bit is set, a read operation is to be performed; otherwise a write operation should be performed. When a Slave recognizes that it is being addressed, it should acknowledge by pulling SDA low in the ninth SCL (ACK) cycle. If the addressed Slave is busy, or for some other reason cannot service the Master’s request, the SDA line should be left high in the ACK clock cycle.

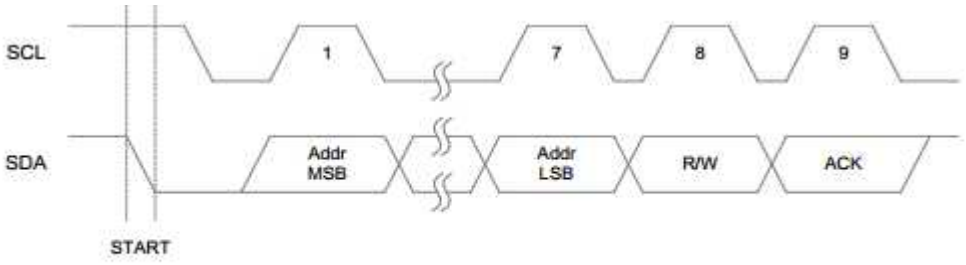
The Master can then transmit a STOP condition, or a REPEATED START condition to initiate a new transmission.

An address packet consisting of a slave address and a READ or a WRITE bit is called SLA+R or SLA+W, respectively.

The MSB of the address byte is transmitted first. Slave addresses can freely be allocated by the designer, except for address 0000000 reserved for a general call and address in the format of “1111xxxx” kept for future purpose.

When a general call is issued, all slaves should respond by pulling the SDA line low in the ACK cycle. A general call is used when a Master wishes to transmit the same message to several slaves in the system. When the general call address followed by a Write bit is transmitted on the bus, all slaves set up to acknowledge the general call will pull the SDA line low in the ACK cycle. The following data packets will then be received by all the slaves that acknowledged the general call. Note that transmitting the general call

address followed by a Read bit is meaningless, as this would cause conflict on bus if several slaves started transmitting different data.



TWI 地址包格式图

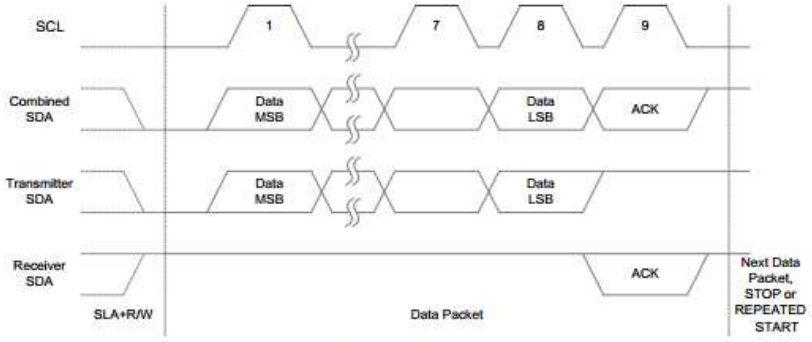
数据包格式

所有 TWI 总线上传输的数据包都是 9 位数据长度，由 1 个数据字节和 1 位应答位组成。在数据传输期间，主机负责产生传输时钟 SCL 和 START 及 STOP 状态，发送器发送要传输的字节数据，接收器产生接收响应。确认信号 ACK 是接收器在第 9 个 SCL（ACK）周期通过拉低 SDA 线来产生的。如果接收器在 ACK 周期保持 SDA 线为高，则发出的是未确认信号 NACK。当接收器已经接收到了最后一个字节，或者由于某些原因不能再接收任何数据，则应该在收到最后字节后通过发送 NACK 来告知发送器。数据字节的 MSB 位先传输。

Data Packet Format

All data packets transmitted on the TWI bus are nine bits long, consisting of one data byte and an acknowledge bit. During a data transfer, the Master generates the clock and the START and STOP conditions, while the transmitter need to transmit byte data, and the Receiver is responsible for acknowledging the reception. An Acknowledge (ACK) is signaled by the Receiver pulling the SDA line low during the ninth SCL cycle. If the Receiver leaves the SDA line high in ACK cycle, an unconfirmed NACK is signaled.
When the Receiver has received the last byte, or for some reason cannot receive any more bytes, it should inform the Transmitter by sending a NACK after the final byte. The MSB of the data byte is transmitted first.

Data Packet Format is as follows:



TWI 数据包格式图

组合地址和数据包的传输

一次传输基本上由 1 个 START，1 个 SLA+R/W，1 个或多个数据包以及 1 个 STOP 组成。只有 START 和 STOP 的空信息是非法的。可以使用 SCL 线的线与功能来实现主机与从机的握手。从机可以通过拉低 SCL 线来延长 SCL 的地电平周期。当主机设定的时钟速度远远快于从机，或从机需要额外的时间来处理数据时，这个特性就非常有用。从机延长 SCL 的低电平周期并不会影响 SCL 的高电平周期，它仍然是由主机决定的。由此可，从机可以通过改变 SCL 的占空比来降低 TWI 的数据传输速度。

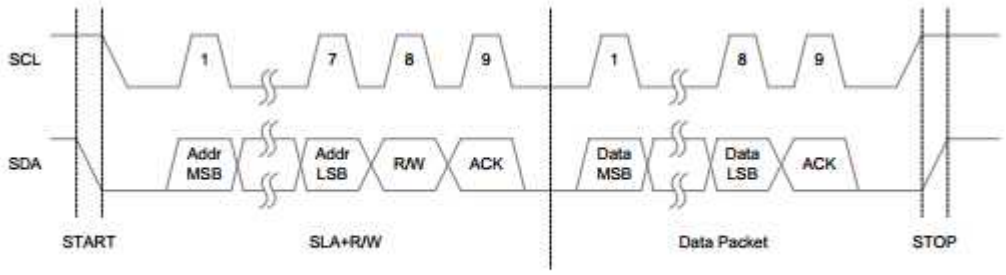
下图所示的是一个典型的数据传输。注意 SLA+R/W 与 STOP 之间可以传送多个字节，取

下图所示的是一个典型的数据传输。注意 SLA+R/W 与 STOP 之间可以传送多个字节，取决于应用软件的实现协议。

Combining Address and Data Packets into a Transmission

A transmission basically consists of a START condition, a SLA+R/W, one or more data packets and a STOP condition. An empty message, consisting of a START followed by a STOP condition, is illegal. Note that the wired and Function of the SCL line can be used to implement handshaking between the Master and the Slave. The Slave can extend the SCL low period by pulling the SCL line low. This is useful if the clock speed set up by the Master is too fast for the Slave, or the Slave needs extra time for processing between the data transmissions. The Slave extending the SCL low period will not affect the SCL high period, which is determinedby the Master. As a consequence, the Slave can reduce the TWI data transfer speed by changing the SCL duty cycle.

Below shows a typical data transmission. Note that several data bytes can be transmitted between the SLA+R/W and the STOP condition, depending on the software protocol implemented by the application software.



典型的 TWI 传输

多主机系统及其仲裁和同步

TWI 协议允许总线上有多个主机，并采用了特殊的措施来保证即使两个或多个主机同时启动传输也能够像普通传输一样处理。多主机系统会出现两个问题：

1. 实现的算法只允许多主机中的一个主机完成传输。当其它主机发现它们失去选择权后必须停止它们的传输。这个选择的过程就叫做仲裁。当竞争中的主机发现其仲裁失败后，应立即切换到从机模式来检测自己是否被获得总线控制权的主机寻址。事实上多主机同时开始传输时不应该被从机检测到，即不允许破坏正在总线上传送的数据。
2. 不同的主机可能使用不同的 SCL 频率。为保证传送的一致性，必须设计一种同步主机串行时钟的方案。这会简化仲裁过程。

总线的线与功能就是用来解决上述问题的。所有主机的串行时钟都会线与到一起产生一个组合时钟，其高电平时间等于所有主机时钟中最短的一个，其低电平则等于所有主机时钟中最长的一个。所有主机都监听 SCL，当组合 SCL 时钟变高或变低时，它们可以有效地分别开始计算各自 SCL 高电平和低电平溢出周期。

多主机的 SCL 时钟同步机制如下图所示：

Multi-master Bus Systems, Arbitration and Synchronization

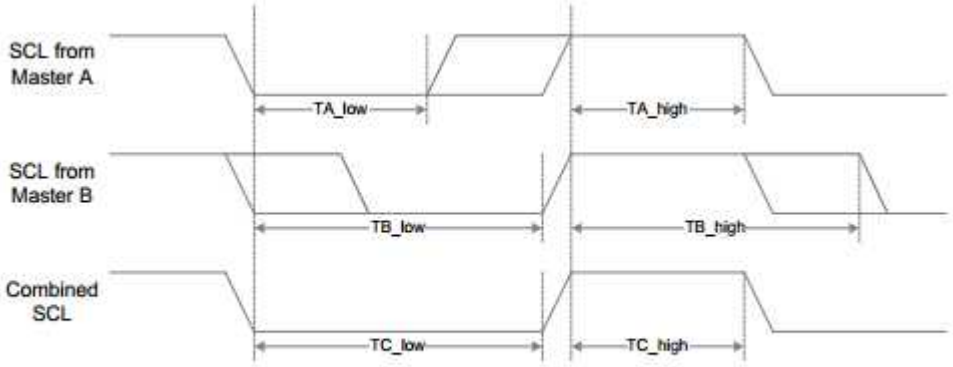
The TWI protocol allows bus systems with several masters. Special concerns have been taken in order to ensure that transmissions will proceed as normal, even if two or more masters initiate a transmission at the same time. Two problems arise in multi-master systems:

1. An algorithm must be implemented allowing only one of the masters to complete the transmission. All other masters should cease transmission when they discover that they have lost the selection process. This selection process is called arbitration. When a contending master discovers that it has lost the arbitration process, it should immediately switch to Slave mode to check whether it is being addressed by the winning master. The fact that multiple masters have started transmission at the same time should not be detectable to the slaves, i.e.the data being transferred on the bus must not be corrupted.

2. Different masters may use different SCL frequencies. A scheme must be devised to synchronize the serial clocks from all masters, in order to let the transmission proceed in a lockstep fashion. This will facilitate the arbitration process.

The wired-ANDing of the bus lines is used to solve boththese problems. The serial clocks from all masters will be wired-ANDed, yielding a combined clock with a high period equal to the one from the Master with the shortest high period. Note that all masters listen to the SCL line, effectively starting to count their SCL high and low time-out periods when the combined SCL clock goes high or low, respectively.

SCL Synchronization Between Multiple Masters:



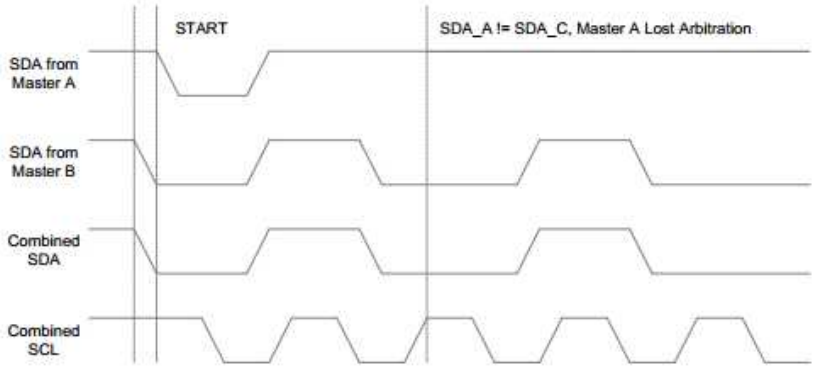
多主机 SCL 时钟同步时序图

输出数据之后所有的主机都持续监听 SDA 线来实现仲裁。如果从 SDA 读回的数值与主机输出的数值不匹配，该主机即失去仲裁。要注意的是，主机输出高电平的 SDA，而另一个主机输出低电平的 SDA 时才会失去仲裁。失去仲裁的主机应立即转换为从机模式，并检测是否被寻址。失去仲裁的主机必须将 SDA 线置高，但在当前的数据或地址包结束之前还可以产生时钟信号。仲裁将会持续到系统只剩下一个主机，这可能会占用多个比特。如果多个

主机对相同的从机寻址，仲裁将会持续到数据包。

Arbitration is carried out by all masters continuously monitoring the SDA line after outputting data. If the value read from the SDA line does not match the value the Master had output, it has lost the arbitration. Note that a Master can only lose arbitration when it outputs a highSDA value while another Master outputs a low value. The losing Master should immediately go to Slave mode, checking if it is being addressed by the winning Master. The SDA line should be left high, but losing masters are allowed to generate a clock signal until the end of the current data or address packet.

Arbitration will continue until only one Master remains, and this may take many bits. If several masters are trying to address the same Slave, arbitration will continue into the data packet.



两个主机之间的仲裁

注意不允许在以下情形进行仲裁：

- ？ 一个 REPEATED START 状态与一个数据位之间；
- ？ 一个 STOP 状态与一个数据位之间；
- ？ 一个 REPEATED START 状态与 STOP 状态之间；

应用软件必须考虑上述情况，保证不会出现这些非法仲裁情形。这意味着在多主机系统中，所有的数据传输必须由相同的 SLA+R/W 与数据包组成。换句话说，所有的传送必须包含相同数目的数据包，否则仲裁结果无法定义。

Note that arbitration is not allowed between:

- A REPEATED START condition and a data bit.
- A STOP condition and a data bit.
- A REPEATED START and a STOP condition

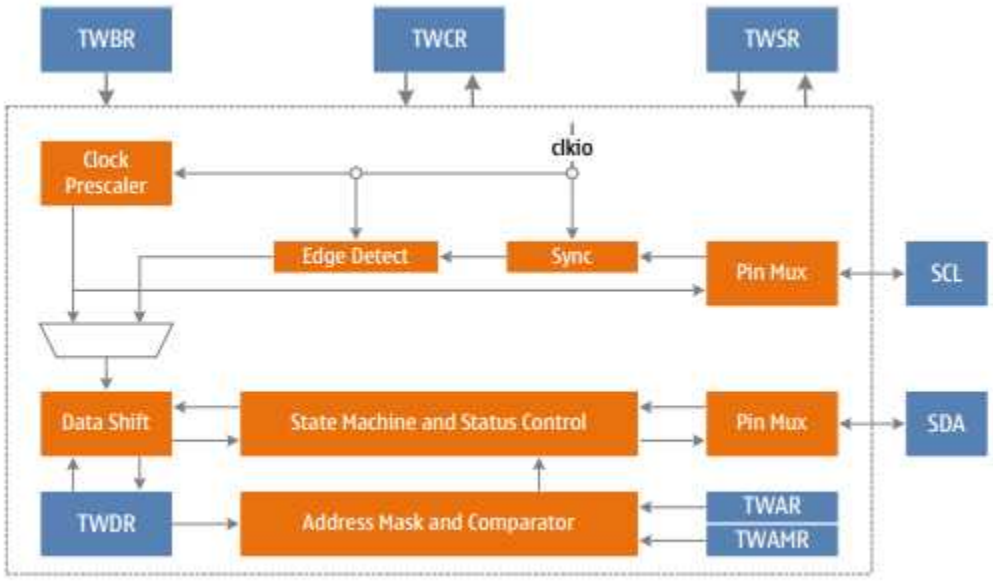
It is the user software's responsibility to ensure that these illegal arbitration conditions never occur. This implies that in multi-master systems, all data transfers must use the same composition of SLA+R/W and data packets. In other words: All transmissions must contain the same number of data packets, otherwise the result of the arbitration is undefined.

TWI 模块综述

TWI 模块的结构图如下图所示。

TWI Modular Overview

TWI modular construction drawing is as below:



TWI Block 结构图

TWI 模块主要包括比特率发生器，总线接口单元，地址比较器和控制单元等。具体见下列详细描述。

The TWI module is comprised of baud rate generator, bus interface unit, address comparator and control unit. For details refer to the below.

比特率发生器单元

比特率发生器单元主要控制主机模式下的 SCL 时钟周期。SCL 时钟周期由 TWI 比特率寄存器 TWBR 和 TWI 状态寄存器 TWSR 中的预分频控制位共同决定。从机操作不受比特率或预分频设置的影响，但要保证从机的工作时钟至少是 SCL 频率的 16 倍。注意，从机可能会延长 SCL 的低电平周期，从而降低 TWI 总线的平均时钟频率。SCL 时钟频率有以下的计算公式产生：

fscl = fsys/(16 + 2*TWBR*4TWPS)

其中，TWBR 为 TWI 比特率寄存器的数值，TWPS 为 TWI 状态寄存器中的预分频控制位。

Bit Rate Generator Unit

This unit controls the period of SCL when operating in a Master mode. The SCL period is controlled by settings in the TWI Bit Rate Register (TWBR) and the Prescaler bits in the TWI Status Register (TWSR). Slave operation does not depend on Bit Rate or Prescaler settings, but the CPU clock frequency in the Slave must be at least 16 times higher than the SCL frequency. Note that slaves may prolong the SCL low period, thereby reducing the average TWI bus clock period. The SCL frequency is generated according to the following equation:

fscl = fsys/(16 + 2*TWBR*4TWPS)

TWBR = Value of the TWI Bit Rate Register.

TWPS = Value of the prescaler in TWI status register

总线接口单元

总线接口单元包括数据和地址移位寄存器 TWDR，START/STOP 控制器和仲裁判定硬件电路。

TWDR 包含要发送的地址或数据字节，或者已接收的地址或数据字节。除了包含 8 位的 TWDR，总线接口单元还包括发送或接收的 ACK/NACK 寄存器。这个 ACK/NACK 寄存器不能直接被应用软件访问。当接收数据时，它可以通过 TWI 控制寄存器 TWCR 来置位或清零。当发送数据时，接收到的 ACK/NACK 值由 TWI 状态寄存器 TWSR 中的 TWS 值来反映。

START/STOP 控制器负责产生和检测 START，REPEATED START 和 STOP 状态。当 MCU 处于某些休眠模式时， START/STOP 控制器仍可以检测 START 和 STOP 状态，当被 TWI 总线上的主机寻址时将 MCU 从休眠模式唤醒。

如果 TWI 以主机模式启动了数据传输，仲裁检测电路将持续监听总线，以确定是否仍拥有总线控制权。当 TWI 模块丢失总线控制权后，控制单元将会执行正确的动作并产生合适的状态码来通知 MCU。

Bus Interface Unit

This unit contains the Data and Address Shift Register (TWDR), a START/STOP Controller and Arbitration detection hardware.

The TWDR contains the address or data bytes to be transmitted, or the address or data bytes received. In addition to the 8-bit TWDR, the Bus Interface Unit also contains a register containing the (N)ACK bit to be transmitted or received.

This (N)ACK Register is not directly accessible by the application software. However, when receiving, it can be set or cleared by manipulating the TWI Control Register (TWCR). When in Transmitter mode, the value of the received (N)ACK bit can be determined by the TWS value in the TWSR.

The START/STOP Controller is responsible for generation and detection of START, REPEATED START, and STOP conditions. The START/STOP controller is able to detect START and STOP conditions even when the AVR MCU is in one of the sleep modes, enabling the MCU to wake up if addressed by a Master.

If the TWI has initiated a transmission as Master, the Arbitration Detection hardware continuously monitors the bus trying to determine if bus control is in process. If the TWI has lost the control of bus, the Control Unit is informed. Correct action can then be taken and appropriate status codes generated.

地址匹配单元

地址匹配单元用来检查接收到的地址字节是否与 TWI 地址寄存器中的 7 位地址相匹配。当 TWAR 寄存器中的 TWI 广播呼叫识别使能位（TWGCE）置位，从总线接收到的地址也会与广播地址比较。一旦地址匹配成功，控制单元将执行正确的动作。 TWI 模块可以响应或不响应主机的寻址，这取决于 TWCR 寄存器的设置。即使在休眠模式下，地址匹配单元也可以比较地址，若被总线上的主机寻址，则将 MCU 从休眠模式唤醒。

Address Match Unit

The Address Match unit checks if received address bytes match the seven-bit address in the TWI Address Register (TWAR). If the TWI General Call Recognition Enable (TWGCE) bit inthe TWAR is written to one, all incoming address bits will also be compared against the General Call address. Uponan address match, the Control Unit is informed, allowing correct action to be taken. The TWI may or may not acknowledge its address, depending on settings in the TWCR. The Address Match unit is able to compare addresses even when the AVR MCU is in sleep mode, enabling the MCU to wake up if addressed by a Master.

控制单元

控制单元负责监听总线并根据 TWCR 的设置产生相应的响应。当 TWI 总线上发生需要应用软件参与的事件时， TWI 中断标志位 TWINT 将会被置位。在接下来的一个时钟周期， TWI 状态寄存器 TWSR 将会被更新为表明该事件的状态码。在 TWINT 被置位时， TWSR 包含确切的状态信息。在其它时间里， TWSR 为一个特殊的状态码，表示没有确切的状态信息。一旦 TWINT 标志位被置位，SCL 线就一直保持低电平，暂停总线上的 TWI 传输，让应用软件处理事件。

下列情形下，TWINT 标志位将置位：

- ？ TWI 传送完 START/REPEATED START 状态后
- ？ TWI 传送完 SLA+R/W 后
- ？ TWI 传送完一个地址字节后
- ？ TWI 总线仲裁失败后
- ？ TWI 被主机寻址后（从机地址匹配或广播方式）
- ？ 被寻址作为从机工作时，收到 STOP 或 REPEATED START 后
- ？ 由非法的 START 或 STOP 状态所引起的总线错误时

Control Unit

The Control unit monitors the TWI bus and generates responses corresponding to settings in the TWI Control Register (TWCR). When an event requiring the attention of the application occurs on the TWI bus, the TWI Interrupt Flag (TWINT) is set. In the next clock cycle, the TWI Status Register (TWSR) is updated with a status code identifying the event. The TWSRcontains specific status information when the TWI Interrupt Flag is set. At all other times, the TWSR is a special status code indicating that no relevant status information is available. As long as the TWINT Flag is set, the SCL line is held low. This allows the application software to complete its tasks before allowing the TWI transmission to continue.

The TWINT Flag is set in the following situations:

- ◠ **After the TWI has transmitted a START/REPEATED START condition.**
- ◠ **After the TWI has transmitted SLA+R/W.**
- ◠ **After the TWI has transmitted an address byte.**
- ◠ **After the TWI bus has lost arbitration.**
- ◠ **After the TWI has been addressed by master (or by slave address match or general call)**
- ◠ **After a STOP or REPEATED START has been received while still addressed as a Slave.**
- ◠ **When a bus error has occurred due to an illegal START or STOP condition**

TWI 的使用

TWI 接口是面向字节和基于中断的。所有的总线事件，如接收到一个字节或发送了一个 START 信号等，都会产生一个 TWI 中断。由于 TWI 是基于中断的，因此在 TWI 字节传送的过程中，应用软件可以自如的进行其它操作。TWCR 寄存器中的 TWI 中断使能位 TWIE 和全局中断使能位一起来控制在 TWINT 标志位置位时是否产生 TWI 中断。如果 TWIE 位被清零，应用软件必须采用查询 TWINT 标志位的方式来检测 TWI 总线上的动作。

当 TWINT 标志位被置位时，表示 TWI 接口完成了当前的操作，等待应用软件的响应。在这种情况下， TWI 状态寄存器 TWSR 中包含了反映当前总线状态的状态码。应用软件可以通过设置 TWCR 和 TWDR 寄存器，来决定在接下来的 TWI 总线周期 TWI 接口该如何工作。

下图给出的是应用程序与 TWI 接口连接的例子。该例中，主机期望发送一个字节的数据给从机。这里的描述很简单，接下来的章节会有更详细的展示。

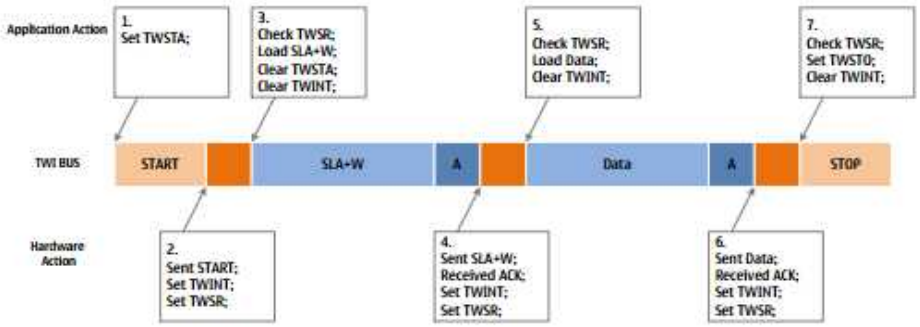
Using the TWI

The AVR TWI is byte-oriented and interrupt based. TWI Interrupts are issued after all bus events, like reception of a byte or transmission of a START condition. Because the TWI is interrupt-based, the application software is free to carry on other operations during a TWI byte transfer. Note that the TWI Interrupt Enable (TWIE) bit in TWCR together with the Global Interrupt Enable bit allow the application to decide whether or not assertion of the TWINT Flag should generate an interrupt request. If the TWIE bit is cleared, the application must poll the TWINT Flag in order to detect actions on the TWI bus. When the TWINT Flag is set, the TWI has finished anoperation and awaits application response. In this case, the TWI Status Register (TWSR) contains a value indicating the current state of the TWI bus.

The application software can then decide how the TWI should behave in the next TWI bus cycle by setting the TWCR and TWDR Registers.

Below is a simple example of how the application can interface to the TWI hardware. In this example, a Master wishes to transmit a single data byte to a Slave. This description is quite simple; a more detailed explanation follows later in this section.

Typitical TWI transmit process flow chart



图中所示的 TWI 传输过程为：

1. TWI 传输的第一步是发送 START。通过往 TWCR 寄存器写入特定值，指示 TWI 硬件发送 START 信号。写入的值将在随后详细说明。在写入的值中要置位 TWINT，这非常重要，往 TWINT 位写“1”会清零该位。TWCR 寄存器的 TWINT 置位期间 TWI 不会启动任何操作。一旦软件清零 TWINT 位，TWI 模块立即启动 START 信号的传送。

2. 当 START 状态发送完毕， TWCR 的 TWINT 标志位会被置位， TWSR 更新为的新的状态码，表示 START 信号成功发送。

3. 应用程序查看 TWSR 的值，确定 START 状态已经成功发送。如果 TWSR 显示为其它值，应用程序可以执行一些特殊操作，比如调用错误处理程序。当确定状态码与预期一致后，程序将 SLA+W 的值载入到 TWDR 寄存器中。TWDR 寄存器可同时在地址和数据中使用。随后软件往 TWCR 寄存器写入特定值，指示 TWI 硬件发送 TWDR 中的 SLA+W 的值。写入的值将在随后详细说明。在写入的值中要置位 TWINT，来清零 TWINT 标志位 。TWCR

寄存器的 TWINT 置位期间 TWI 不会启动任何操作。一旦软件清零 TWINT 位，TWI 模块立即启动地址包的传送。

4. 当地址包发送完毕后，TWCR 的 TWINT 标志位会被置位，TWSR 更新为新的状态码，表示地址包成功发送。状态码同样会反映从机是否响应该地址包。

5. 应用程序查看 TWSR 的值，确定地址包已成功发送，收到的 ACK 为期望值。如果 TWSR 显示为其它值，应用程序可以执行一些特殊操作，比如调用错误处理程序。当确定状态码与预期一致后，程序将 Data 的值载入到 TWDR 寄存器中。随后软件往 TWCR 寄存器写入特定值，指示 TWI 硬件发

送 TWDR 中的 Data 的值。写入的值将在随后详细说明。在写入的值中要置位 TWINT，来清零 TWINT 标志位。TWCR 寄存器的 TWINT 置位期间 TWI 不会启动任何操作。一旦软件清零 TWINT 位，TWI 模块立即启动数据包的传送。

6. 当数据包发送完毕后，TWCR 的 TWINT 标志位会被置位，TWSR 更新为新的状态码，表示数据包成功发送。状态码同样会反映从机是否响应该数据包。

7. 应用程序查看 TWSR 的值，确定数据包已成功发送，收到的 ACK 为期望值。如果 TWSR 显示为其它值，应用程序可以执行一些特殊操作，比如调用错误处理程序。当确定状态码与预期一致后，软件往 TWCR 寄存器写入特定值，指示 TWI 硬件发送 STOP 信号。写入的值将在随后详细说明。在写入的值中要置位 TWINT，来清零 TWINT 标志位。TWCR 寄存器的 TWINT 置位期间 TWI 不会启动任何操作。一旦软件清零 TWINT 位，TWI 模块立即启动 STOP 信号的传送。需要注意的是，在 STOP 信号发送完毕之后 TWINT 不会被置位。

尽管示例比较简单，但它包含了 TWI 数据传输过程中的所有规则。总结如下：

？ 当 TWI 完成一次操作并等待应用程序的反馈时， TWINT 标志置位。 SCL 时钟线会被一直拉低直到 TWINT 被清零；

？ 当 TWINT 标志置位，用户必须更新所有 TWI 寄存器的值为与下一个 TWI 总线周期相关

的值。例如，TWDR 寄存器必须载入下一个总线周期要发送的值。

？ 当更新完所有的寄存器，同时完成其它必要的操作之后，应用程序写 TWCR 寄存器。在写 TWCR 时， TWINT 位必须被置位，用来清零 TWINT 标志。 TWINT 被清零之后， TWI 开始执行由 TWCR 设定的操作。

1. The first step in a TWI transmission is to transmit a START condition. This is done by writing a specific value into TWCR, instructing the TWI hardware to transmit a START condition. Which value to write will be described later on. However, it is important that the TWINT bit should be set in the value written. Writing a one to TWINT clears the flag.Immediately after the application has cleared TWINT, the TWI will initiate transmission of the START condition.

2. When the START condition has been transmitted, the TWINT Flag in TWCR is set, and TWSR is updated with a status code indicating that the START condition has successfully been sent.

3. The application software should now examine the value of TWSR, to make sure that the START condition was successfully transmitted. If TWSR indicates otherwise, the application software might take some special action, like calling an error routine. Assuming that the status code is as expected, the application must load SLA+W into TWDR.

Remember that TWDR is used both for address and data. After TWDR has been loaded with the desired SLA+W, a specific value must be written to TWCR, instructing the TWI hardware to transmit the SLA+W present in TWDR.

Which value to write is described later on. However, it is important that the TWINT bit must be set in the value written in order to clear TWINT flag. The TWI will not start any operation as long as the TWINT bit in TWCR is set.Immediately after the application has cleared TWINT, the TWI will initiate transmission of the address packet.

4. When the address packet has been transmitted, the TWINT Flag in TWCR is set, and TWSR is updated with a status code indicating that the address packet has successfully been sent. The status code will also reflect whether a Slave has acknowledged the packet or not.

5. The application software should now examine the value of TWSR, to make sure that the address packet was successfully transmitted, and that the value of the ACK bit was as expected. If TWSR indicates otherwise, the application software might take some special action, like calling an error routine. Assuming that the status code is as expected, the application must load a data packet into TWDR. Subsequently, a specific value must be written to TWCR, instructing the TWI hardware totransmit the data packet present in TWDR. Which value to write is described later on. However, it is important that the TWINT bit is set in the value written. Writing a one to TWINT clears the flag. The TWI will not start any operation as long as the TWINT bit in TWCR is set. Immediately after the application has cleared TWINT, the TWI will initiate transmission of the data packet.

6. When the data packet has been transmitted, the TWINT Flag in TWCR is set, and TWSR is updated with a status code indicating that the data packet has successfully been sent. The status code will also reflect

whether a Slave has acknowledged the packet or not.

7. The application software should now examine the value of TWSR, to make sure that the data packet was successfully transmitted, and that the value of received ACK bit was as expected. If TWSR indicates otherwise, the application software might take some special action, like calling an error routine. Assuming that the status code is as expected, the application must write a specific value to TWCR, instructing the TWI hardware to transmit a STOP condition. Which value to write is described later on. However, it is important that the TWINT bit is set in the value written. Writing a one to TWINT clears the flag. The TWI will not start any operation as long as the TWINT bit in TWCR isset. Immediately after the application has cleared TWINT, the TWI will initiate transmission of the STOP condition. Note that TWINT is NOT set after a STOP condition has been sent.

Even though this example is simple, it shows the principles involved in all TWI transmissions. These can be summarized as follows:

- ◠ When the TWI has finished an operation and expects application response, the TWINT Flag is set. The SCL line is pulled low until TWINT is cleared.
- ◠ When the TWINT Flag is set, the user must update all TWI Registers with the value relevant for the next TWI bus cycle. As an example, TWDR must be loaded with the value to be transmitted in the next bus cycle. I.E., TWDR register must be loaded the value to be transmitted in the next bus cycle.
- After all TWI Register updates and other pending application software tasks have been completed, TWCR is written. When writing TWCR, the TWINT bit should be set. Writing a one to TWINT clears the flag. The TWI will then commence executing whatever operation was specified by the TWCR setting.

传输模式

TWI 可以工作在下面 4 种主要的模式：主机发送器（MT），主机接收器（MR），从机发送器（ST）和从机接收器（SR）。同一应用下可以使用多种模式。例如， TWI 可以使用 MT 模式往 TWI EEPROM 写入数据，用 MR 模式从 EEPROM 读取数据。如果该系统上还有其它主机，有些也可能往 TWI 发送数据，则会使用 SR 模式。这是由应用软件来决定采用何种模式。

下面会对这些模式进行详细说明。在每种模式下的数据传输中，会结合图片来描述可能的状态码。这些图片包含了如下的缩写：

S： Start 状态

Rs： REPEATED START 状态

R： 读操作标志位（SDA 为高电平）

W： 写操作标志位（SDA 为低电平）

A: 应答位（SDA 为低电平）

NA： 无应答位（SDA 为高电平）

Data：8 位数据字节

P： STOP 状态

SLA： 从机地址

图片中的圆圈用来表示 TWINT 标志置位，圆圈中的数字表示 TWSR 寄存器中的状态码，其中预分频控制位被屏蔽为“0”。在这些地方，应用程序必须执行相应的操作来继续或完成 TWI 传输。TWI 传输会被挂起，直到 TWINT 标志位被清零。

当 TWINT 标志被置位， TWSR 中的状态码用来决定适当的软件操作。各表格中给出了每个状态码下所需的软件操作和后续串行传输的细节。注意表格里 TWSR 中的预分频控制位被屏蔽为“0”。

Transmission Modes

The TWI can operate in one of four major modes. These are named Master Transmitter (MT), Master Receiver (MR), Slave Transmitter (ST) and Slave Receiver (SR). Several of these modes can be used in the same

application. As an example, the TWI can use MT mode to write data into a TWI EEPROM, and use MR mode to read the data back from the EEPROM. If other masters are present inthe system, some of these might transmit data to the TWI, and then SR mode would be used. It is the application software that decides which modes are legal.

The following sections describe each of these modes. Possible status codes are described along with figures detailing data transmission in each of the modes. These figures contain the following abbreviations:
S: START condition
Rs: REPEATED START condition
R: Read bit (high level at SDA)
W: Write bit (low level at SDA)
A: Acknowledge bit (low level at SDA)
NA: Not acknowledge bit (high level at SDA)
Data: 8-bit data byte
P: STOP condition
SLA: Slave Address
ircles are used to indicate that the TWINT Flag is set. The numbers in the
circles in the figures show TWINT flag enable bit, while number inside of circles show the status code held in TWSR, with the prescaler bits masked to zero. At these points, actions must be taken by the application to continue or complete the TWI transfer. The TWI transfer is suspended until the TWINT Flag is cleared by software.
When the TWINT Flag is set, the status code in TWSR is used to determine the appropriate software action. For each status code, the required software action and details of the following serial transfer are given in below lists. Note that the prescaler bits are masked to zero in these tables.

主机发送模式

在主机发送模式中， TWI 会发送一定数量的数据字节到从机接收器。为了进入主机模式，必须发送 START 信号。接下来的地址包格式决定 TWI 是进入主机发送器模式还是主机接收器模式。如果发送 SLA+W，则进入主机发送模式。如果发送 SLA+R，则进入主机接收模式。这一章节所提到的状态码均假设预分频控制位为“0”。

Master Transmitter Mode

In the Master Transmitter mode, a number of data bytes are transmitted to a Slave Receiver by TWI. In order to enter a Master mode, a START condition must be transmitted. The format of the following address packet determines whether Master Transmitter or Master Receiver mode is to be entered. If SLA+W is transmitted, MT mode is entered, if SLA+R is transmitted, MR mode is entered. All the status codes mentioned in this section assume thatthe prescaler bits are zero or are masked to zero.

通过往 TWCR 寄存器写入下列数值来发出 START 信号：

A START condition is sent by writing the following value to TWCR:

TWCR	TWINT	TWEA	TWSTA	TWSTO	TWWC	TWEN	–	TWIE
value	1	X	1	0	X	1	0	X

TWEN 位必须置“1”来使能 TWI 接口，TWSTA 置“1”来发送 START 信号，TWINT 置“1”来清零 TWINT 标志位。 TWI 模块检测总线状态，在总线空闲时立即发送 START 信号。当发送完 START 后，硬件置位 TWINT 标志位，同时更新 TWSR 的状态码为 0x08。

为了进入主机发送模式，必须发送 SLA+W。这可通过下面操作来完成。先往 TWDR 寄存器写入 SLA+W，然后往 TWINT 位写“1”清零 TWINT 标志位来继续传输，即往 TWCR 寄存器写入下列数值来发送 SLA+W：

TWEN must be set to enable TWI Interface, TWSTA must be written to one to transmit a START condition and TWINT must be written to one to clear the TWINT Flag. The TWI will then test the Bus status and generate a START condition as soon as the bus becomes free. After a START condition has been transmitted, the TWINT Flag is set by hardware, and the status code in TWSR will be 0x08.

In order to enter MT mode, SLA+W must be transmitted. This is done by below operations. Writing SLA+W to TWDR register first, than write one to TWINT to clear TWINT flag bit in order to continue transmission, i.e. writing below value to TWCR register to transmit SLA+W:

TWINT	TWEA	TWSTA	TWSTO	TWWC	TWEN	-	TWIE
1	x	0	0	x	1	0	x

当 SLA+W 发送完成且收到应答信号后，TWINT 又被置位，同时 TWSR 的状态码更新。可能的状态码为 0x18、0x20 或 0x38。各个状态码下合适的响应会在状态码表格中详细描述。

当 SLA+W 发送成功后，可以开始发送数据包。这可通过往 TWDR 寄存器写入数据来完成。TWDR 只有在 TWINT 标志位为高时才可以写入。否则，访问被忽略，同时写冲突标志位 TWWC 会被置位。更新完 TWDR 后，往 TWINT 位写“1”清零 TWINT 标志位来继续传输。即往 TWCR 寄存器写入下列数值来发送数据：

When SLA+W have been transmitted and an acknowledgement bit has been received, TWINT is set again and a number of status codes in TWSR update. Possible status codes are 0x18, 0x20, or 0x38. The appropriate action to be taken for each of these status codes is detailed in below list. When SLA+W has been successfully transmitted, a data packet should be transmitted. This is done by writing the data byte to TWDR. TWDR must only be written when TWINT flag is high. If not, the access will be discarded, meanwhile the Write Collision flag bit (TWWC) will be set. After updating TWDR, the TWINT bit should be cleared (by writing it to one) to continue the transfer. Thisis accomplished by writing the following value to TWCR.

TWINT	TWEA	TWSTA	TWSTO	TWWC	TWEN	-	TWIE
1	x	0	0	x	1	0	x

当数据包发送完成且收到应答信号后， TWINT 又被置位，同时 TWSR 的状态码更新。可能的状态码为 0x28 或 0x30。各个状态码下合适的响应会在状态码表格中详细描述。

当数据发送成功后，可以继续发送数据包。这个过程一直重复，直到最后一个字节发送完毕。主机产生 STOP 信号或 REPEATED START 信号整个传输才结束。

通过往 TWCR 寄存器写入下列数值来发出 STOP 信号：

When data package have been transmitted and an acknowledgement bit has been received, TWINT is set again and

a number of status codes in TWSR update. Possible status codes are 0x28 or 0x30. The appropriate action to be taken for each of these status codes is detailed in below list.

When a data is successfully transmitted, following data package can be transmitted; such process will repead till the last byte data is complete. The whole transmit is complete only when master generates STOP condition or REPEATED START condition.

By writing below value to TWCR register to transmit STOP condition:

TWINT	TWEA	TWSTA	TWSTO	TWWC	TWEN	-	TWIE
1	x	0	1	x	1	0	x

通过往 TWCR 寄存器写入下列数值来发出 REPEATED START 信号：

By writing below value to TWCR register to transmit REPEATED START condition:

TWINT	TWEA	TWSTA	TWSTO	TWWC	TWEN	-	TWIE
1	x	1	0	x	1	0	x

在发送 REPEATED START（状态码为 0x10）之后，TWI 接口可以再次访问相同的从机，或访问新的从机而不用发送 STOP 信号。REPEATED START 使得主机可以在不丢失总线控制权的情况下在不同从机之间，主机发送器和主机接收器模式之间进行切换。

主机发送模式下的状态码及相应的操作如下表所示：

主机发送模式的状态码表

After a repeated START condition (state 0x10), TWI Interface can access the same Slave again, or a new Slave without transmitting a STOP condition. Repeated START enables the Master to switch between Slaves, Master Transmitter mode and Master Receiver mode without losing control over the bus.

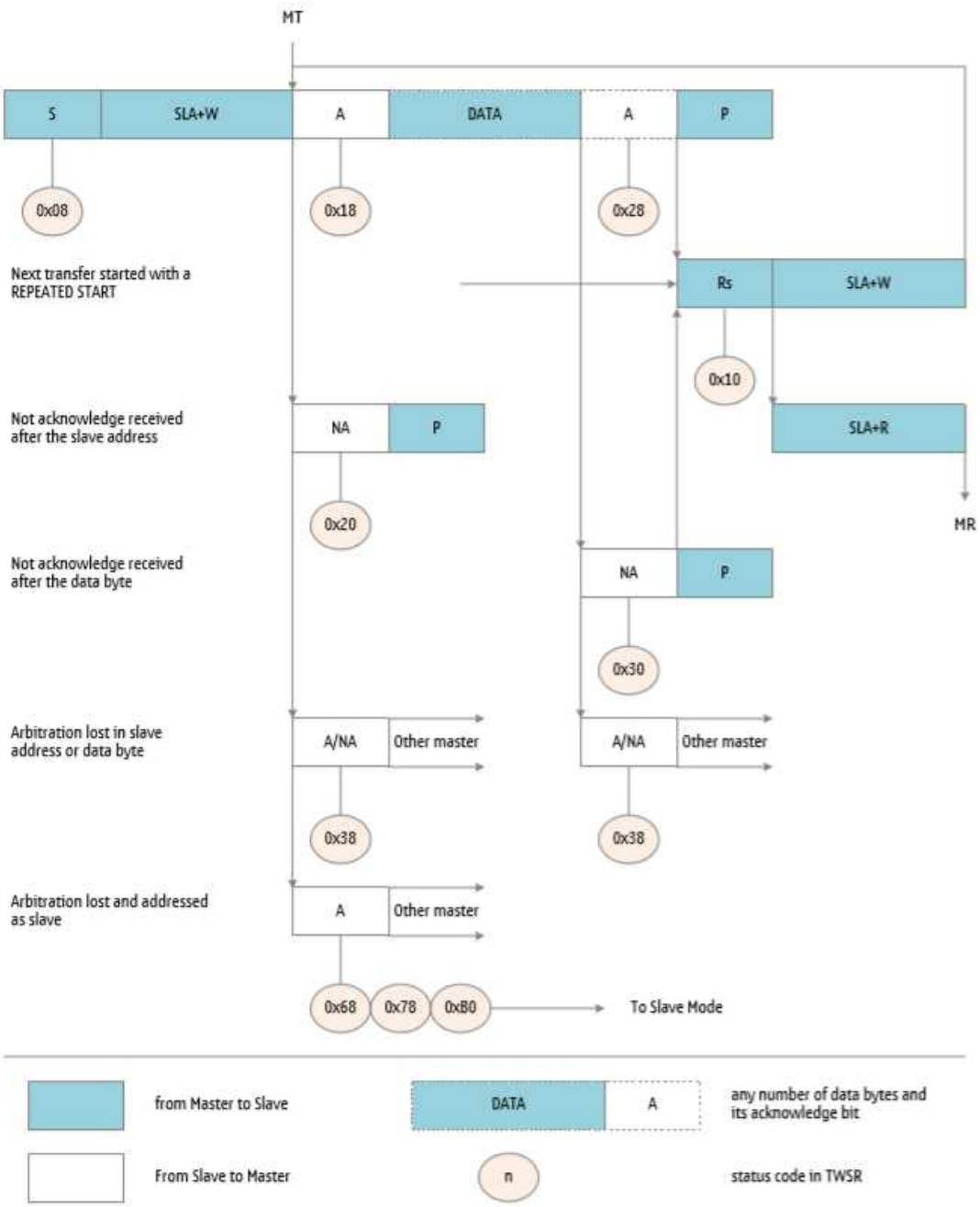
Status codes and relevant operations in master transmit mode is as below:

Status Codes in Master Transmit Mode

Status Code	Bus and Hardware status	Application software response					Next action of hardware
		Read/ Write TWDR	Operations to TWCR				
			STA	STO	TWINT	TWEA	
0x08	STRART transmitted	Load SLA+W	0	0	1	x	To transmit SLA+W To receive ACK or NACK
0x10	REPEATED START traansmitted	Load SLA+W	0	0	1	x	To transmit SLA+W To receive ACK or NACK
		Load SLA+R	0	0	1	x	To transit SLA+R To receive ACK or NACK Will switch to MR mode
0x18	SLA+W transmitted Received ACK	Load data	0	0	1	x	Will transmit data Will receive ACK or NACK
		No operation	1	0	1	x	Will transmit REPEATED START
		No operation	0	1	1	X	Will transmit STOP Will reset TWSTO flag
		No operation	1	1	1	x	Will transmit STOP Will reset TWSTO flag Will transmit START
0x20	SLA+W transmitted NACK received	Load data	0	0	1	x	Will transmit data Will receive ACK or NACK
		No operation	1	0	1	x	Will receive REPEATED START
		No operation	0	1	1	x	Will transmit STOP Will reset TWSTO flag

		No operation	1	1	1	x	Will transmit STOP Will reset TWSTO flag Will transmit START
0x28	Date byte transmitted ACK received	Load data	0	0	1	x	Will transmit data Will receive ACK or NACK
		No operation	1	0	1	X	Will transmit REPEATED START
		No operation	0	1	1	X	Will transmit STOP Will reset TWSTO flag
		No operation	1	1	1	x	Will transmit STOP Will reset TWSTO flag Will transmit START
0x30	Date byte transmitted ACK received	Load data	0	0	1	x	Will transmit data Will receive ACK or NACK
		No operation	1	0	1	x	Will transmit REPEATED START
		No operation	0	1	1	x	Will transmit STOP Will reset TWSTO flag
		No operation	1	1	1	x	Will transmit STOP Will reset TWSTO flag Will transmit START
0x38	SLA+W or data arbitration failure	No operation	0	0	1	x	Will release bus Will enter to slave mode unaddressed
		No operation	1	0	1	x	Will transmit START in idle state

Format and Status of master transmit mode is as shown below:



主机接收模式

在主机接收模式中，TWI 会从从机发送器接收一定数量的数据字节。为了进入主机模式，必 须发送 START 信号。接下来的地址包格式决定 TWI 是进入主机发送器模式还是主机接收器 模式。如果发送 SLA+W，则进入主机发送模式。如果发送 SLA+R，则进入主机接收模式。这 一章节所提到的状态码均假设预分频控制位为“0”。

通过往 TWCR 寄存器写入下列数值来发出 START 信号

Master Receiver mode

In the Master Receiver mode, a number of data bytes are received from a Slave Transmitter. In order to enter a Master mode, a START condition must be transmitted. The format of the following address packet determines whether Master Transmitter or Master Receiver mode is to be entered. If SLA+W is transmitted, MT mode is entered, if SLA+R is transmitted, MR mode is entered. All the status codes mentioned in this section assume that the prescaler bits are zero or are masked to zero.

A START condition is sent by writing the following value to TWCR:

TWINT	TWEA	TWSTA	TWSTO	TWWC	TWEN	-	TWIE
1	x	1	0	x	1	0	x

TWEN 位必须置“1”来使能 TWI 接口，TWSTA 置“1”来发送 START 信号，TWINT 置“1”来清零 TWINT 标志位。TWI 模块检测总线状态，在总线空闲时立即发送 START 信号。当发送完 START 后，硬件置位 TWINT 标志位，同时更新 TWSR 的状态码为 0x08。

为了进入主机接收模式，必须发送 SLA+R。这可通过下面操作来完成。先往 TWDR 寄存器写 入 SLA+R，然后往 TWINT 位写“1”清零 TWINT 标志位来继续传输，即往 TWCR 寄存器写入下 列数值来发送 SLA+R：

TWEN must be written to one to enable TWI Interface, TWSTA must be written to one to transmit a START condition and TWINT must be set to one to clear the TWINT Flag. The TWI will then test Bus state and generate a START condition as soon as the bus becomes free. After a START condition has been transmitted, the TWINT Flag is set by hardware, and the status code in TWSR will be 0x08 . In order to enter MR mode, SLA+R must be transmitted. This is done by writing below operations. First write SLA+R to TWDR register, than the TWINT bit should be cleared (by writing it to one) to continue the transfer. This is accomplished by writing the following value to TWCR to transmit SLA+R:

TWINT	TWEA	TWSTA	TWSTO	TWWC	TWEN	-	TWIE
1	x	0	0	x	1	0	x

当 SLA+R 发送完成且收到应答信号后，TWINT 又被置位，同时 TWSR 的状态码更新。可能的 状态码为 0x38、0x40 或 0x48。各个状态码下合适的响应会在状态码表格中详细描述。

当 SLA+R 发送成功后，可以开始接收数据包。通过往 TWINT 位写“1”清零 TWINT 标志位来继 续接收。即往 TWCR 寄存器写入下列数值来启动接收：

When SLA+R have been transmitted and an acknowledgement bit has been received, TWINT is set again and a number of status codes in TWSR update. Possible status codes are 0x38, 0x40, or 0x48. The appropriate action to be taken for each of these status codes is detailed in below list.

When SLA+R have been transmitted successfully, it is ready to receive data package. Writing one to TWINT bit to clear TWINT flag bit in order to continue receive, i.e. to start receive by writing below value to TWCR register:

TWINT	TWEA	TWSTA	TWSTO	TWWC	TWEN	-	TWIE
1	x	0	0	x	1	0	x

当数据包接收完成且发送应答信号后，TWINT 又被置位，同时 TWSR 的状态码更新。可能的 状态码为 0x50 或 0x58。各个状态码下合适的响应会在状态码表格中详细描述。

当数据接收成功后，可以继续接收数据包。这个过程一直重复，直到最后一个字节接收完毕。主机接收到最后一个字节后，必须发送 NACK 应答信号给从机发送器。主机产生 STOP 信号 或 REPEATED START 信号整个接收才结束。

通过往 TWCR 寄存器写入下列数值来发出 STOP 信号

We data package has received and acknowledgment bit has transmitted, TWINT is set again and TWSR status code update, Possible status code is 0x50, 0x58. The appropriate action to be taken for each of these status codes is detailed in below list.

When data is received successfully, following data package can be received, this process will repead until receive of last byte is complete. After master receive the last byte, must send NACK acknowledgment condition to slave transmitter. The whole recdeive is complete only when master has generated STOP or REPEATED START condition.

Writing the below value to TWCR regiter to transmit STOP condition:

TWINT	TWEA	TWSTA	TWSTO	TWWC	TWEN	-	TWIE
1	x	0	1	x	1	0	x

Writing the below value to TWCR regiter to transmit REPEATED START condition:

TWINT	TWEA	TWSTA	TWSTO	TWWC	TWEN	-	TWIE
1	x	1	0	x	1	0	x

在发送 REPEATED START（状态码为 0x10）之后，TWI 接口可以再次访问相同的主机，或访 问新的主机而不用发送 STOP 信号。REPEATED START 使得主机可以在不丢失总线控制权的情 况下在不同从机之间，主机发送器和主机接收器模式之间进行切换。

主机接收模式下的状态码及相应的操作如下表所示：

主机接收模式的状态码表

After a REPEATED START condition (state 0x10), TWI Interface can access the same Master again, or a new Master without transmitting a STOP condition. Repeated START enables the Master to switch between Slaves, Master Transmitter mode and Master Receiver mode without losing control over the bus.

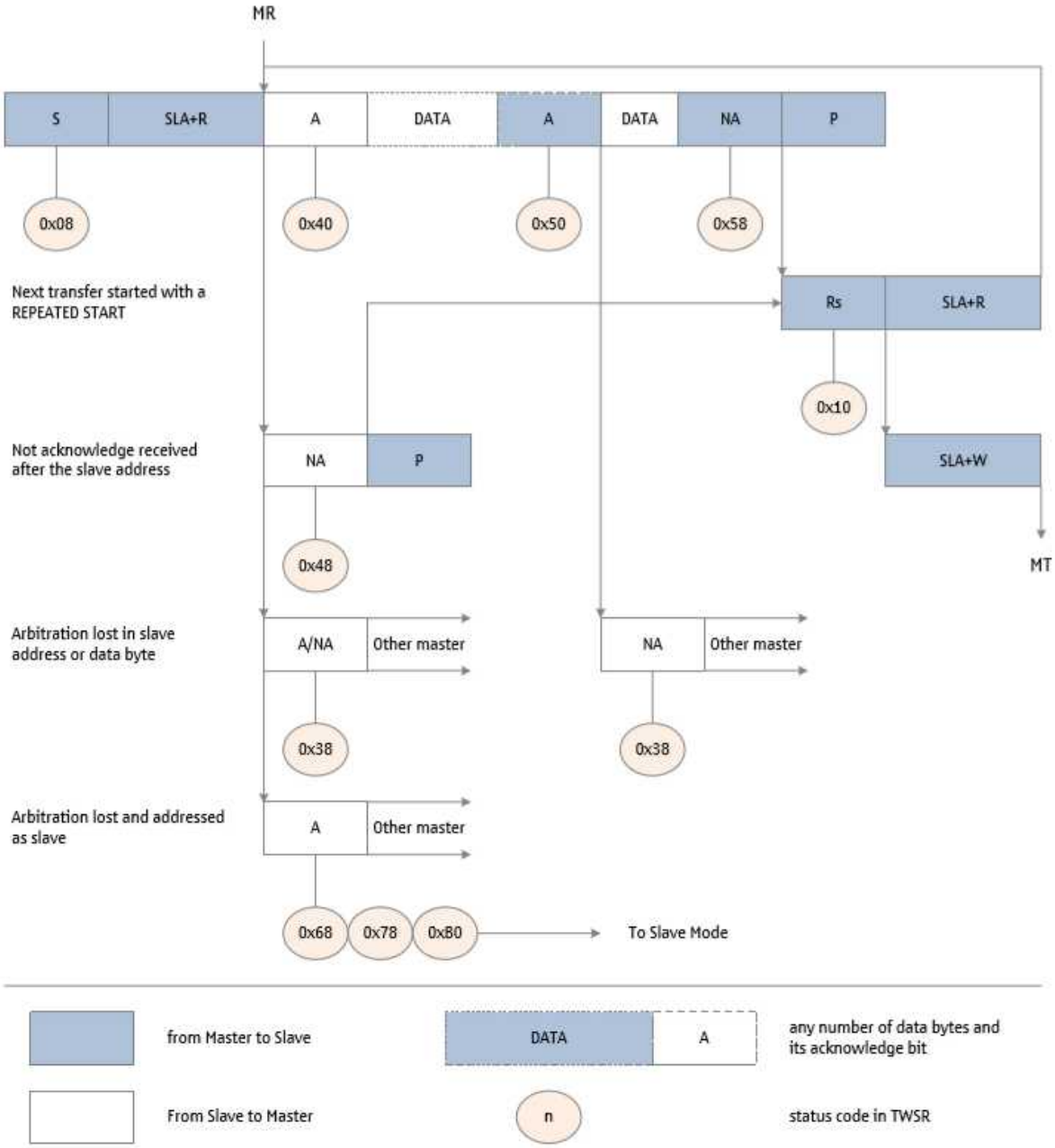
Status codes and relevant operations in master transmit mode is as below:

Status Codes in Master Transmit Mode

Status Code	Bus and Hardware status	Application software response					Next action of hardware
		Read/ Write TWDR	Operations to TWCR				
			STA	STO	TWINT	TWEA	
0x08	STRART transmitted	Load SLA+R	0	0	1	x	To transmit SLA+R To receive ACK or NACK
0x10	REPEATED START traansmitted	Load SLA+R	0	0	1	x	To transmit SLA+R To receive ACK or NACK
		Load SLA+W	0	0	1	x	To transit SLA+W To receive ACK or NACK Will switch to MR mode
0x38	SLA+R or data arbitration failure	No operation	0	0	1	x	Will release bus Will enter to slave mode unaddressed
		No operation	1	0	1	x	Will transmit START in idle state
0x40	SLA+R has transmitted ACK has received	No operation	0	0	1	0	Will receive data Will transmit NACK
		No operation	0	0	1	x	Will receive data Will transmit ACK
0x48	SLA+R has transmitted NACK has	No operation	1	0	1	x	Will transmit REPEATED START
		No	0	1	1	x	Will transmit STOP

	received	operation					Will reset TWSTO flag
		No operation	1	1	1	x	Will transmit STOP Will reset TWSTO flag Will transmit START
0x50	Data byte has received ACK has transmitted	Read data	0	0	1	0	Will receive data Will transmit NACK
		Read data	0	0	1	1	Will receive data Will transmit ACK
0x58	Data byte received NACK transmitted	Read data	1	0	1	x	Will send REPEATED START
		Read data	0	1	1	x	Will transmit STOP Will reset TWSTO flag
		Read data	1	1	1	x	Will transmit STOP Will reset TWSTO flag Will transmit START

Format and Status of Master Receive Mode is as shown below



从机接收模式

在从机接收模式中，可以从主机发送器接收一定数量的数据字节。这一章节所提到的状态码 均假设预分频控制位为“0”。

为启动从机接收模式，要设置 TWAR 和 TWCR 寄存器。

TWAR 需设置如下： TWA6

Slave Receiver Mode

In the Slave Receiver mode, a number of data bytes are received from a Master Transmitter. All the status codes mentioned in this section assume that the prescaler bits are zero or are masked to zero.

To initiate the Slave Receiver mode, TWAR and TWCR must be set as follows:

TWAR setting as below:

TWAR	TWA6	TWA5	TWA4	TWA3	TWA2	TWA1	TWA0	TWGCE
value	Device's Own Slave Address							

TWAR 的高 7 位是主机寻址时 TWI 接口会响应的从机地址。若 LSB 置位，TWI 会响应广播呼 叫地址（0x00），否则忽略广播呼叫地址。

TWCR 需设置如下

The upper 7 bits are the address to which the TWI Interface will respond when addressed by a Master. If the LSB is set, the TWI will respond to the general call address (0x00), otherwise it will ignore the general call address.

TWCR Setting is as below:

TWCR	TWINT	TWEA	TWSTA	TWSTO	TWWC	TWEN	–	TWIE
value	0	1	0	0	0	1	0	X

TWEN 必须置位以使能 TWI 接口，TWEA 必须置位以使主机寻址（从机地址或广播呼叫）到 自己时返回确认信息 ACK。TWSTA 和 TWSTO 必须清零。

初始化 TWAR 和 TWCR 之后，TWI 接口开始等待，直到自己的从机地址（或广播地址）被寻 址。当紧跟着从机地址的数据方向位为“0”（表示写操作）时，TWI 进入从机接收模式。当数 据方向位为“1”（表示读操作）时，TWI 进入从机发送模式。接收到自己的从机地址和写操 作标志位后，TWINT 标志位被置位，有效的状态码也更新到 TWSR 中。各个状态码下合适的 响应会在状态码表格中详细描述。需要注意的是，当主机模式下的 TWI 仲裁失败后也可以 进入从机接收模式（见状态码 0x68 和 0x78）。

如果在传输过程中 TWEA 位被复位，TWI 将在接收到一个字节后返回 NACK（高电平）到 SDA 线上。这可用来表示从机不能接收更多的数据。当 TWEA 位为“0”时，TWI 也不会响应自己 的从机地址。不过 TWI 仍会监听总线，一旦 TWEA 被置位，就可以恢复地址识别并响应。也 就是说，可以利用 TWEA 暂时将 TWI 接口从总线中隔离出来。

在除空闲模式外的其它休眠模式时，TWI 接口的时钟可以被关闭。若是能了从机接收模式， 接口将利用总线时钟继续响应从机地址或广播地址。地址匹配将唤醒 MCU。在唤醒期间， TWI 接口将保持 SCL 为低电平，直到 TWINT 标志被清零。当 TWI 接口时钟恢复正常后可以 接收更多的数据。

从机接收模式的状态码如下表所示：

从机接收模式的状态码表

When TWAR and TWCR have been initialized, the TWI waits until it is addressed by its own slave address (or the general call address if enabled) followed by the data direction bit. If the direction bit is “0” (write), the TWI will operate in SR mode, otherwise ST mode is entered. After its own slave address and the write bit have been received, the TWINT Flag is set and a valid status code can update to TWSR.

The status code is used to determine the appropriate software action. The appropriate action to be taken for each status code is detailed in below table.

The Slave Receiver mode may also be entered if arbitration is lost while the TWI is in the Master mode (see states 0x68 and 0x78).

If the TWEA bit is reset during a transfer, the TWI will return a “Not Acknowledge” to SDA after TWI received data byte. This can be used to indicate that the Slave is not able to receive any more bytes. While TWEA is zero, the TWI does not acknowledge its own slave address. However, TWI still monitored the BUS and address response and recognition may resume at any time by setting TWEA. This implies that the TWEA bit may be used to temporarily isolate the TWI from the Bus.

In all sleep modes other than Idle mode, the clock system to the TWI is turned off. If the slave receive mode is set, the

interface can still acknowledge its own slave address or the general call address by using the Bus clock. Address match will then wake up from sleep and the TWI will hold the SCL clock low during the wake up and until the TWINT Flag is cleared. Further data reception will be carried out as normal, with the TWI interface clocks running as normal.

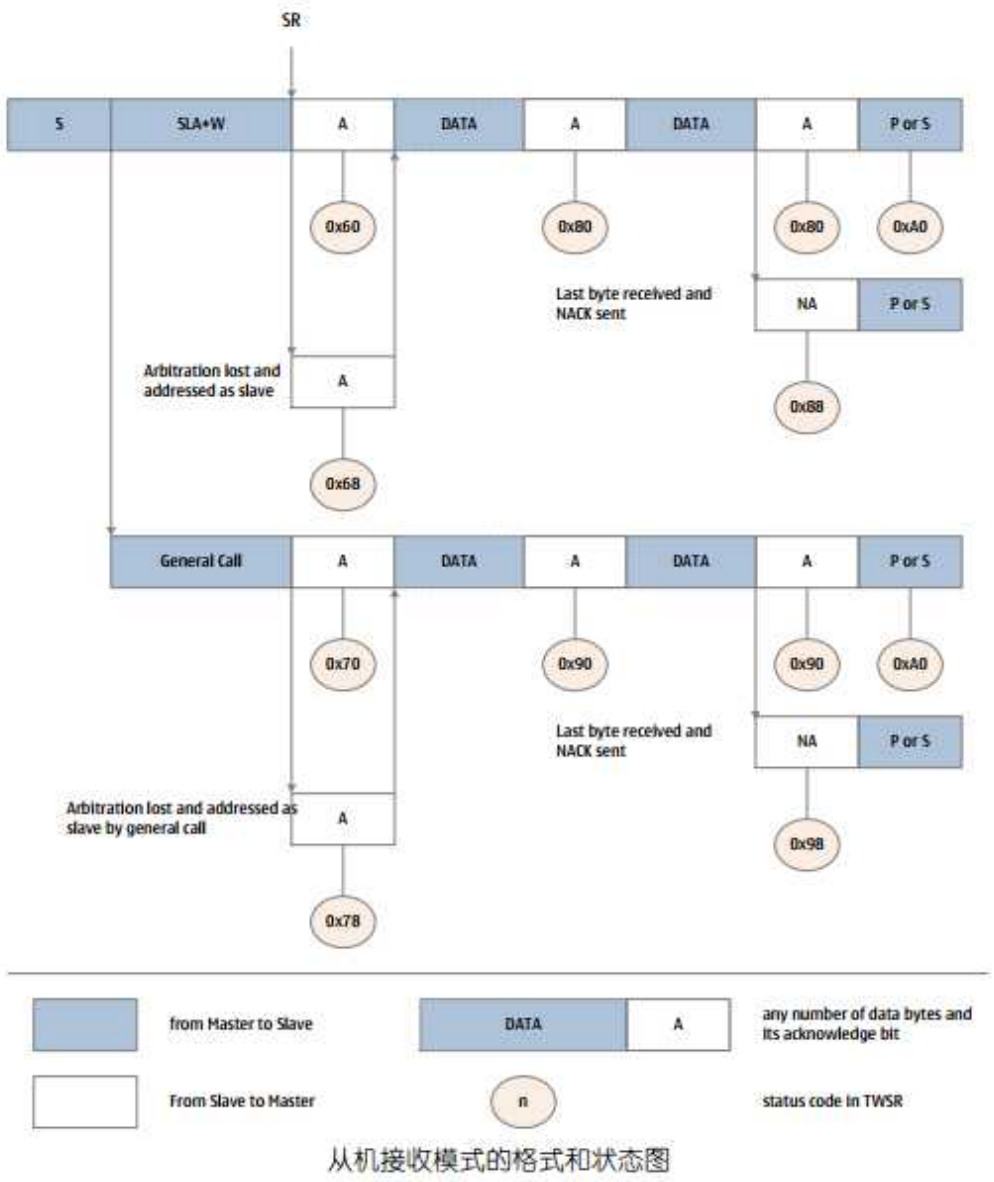
Status Code in Slave Receive Mode is shown as below:

Status Code	Bus and Hardware status	Application software response					Next action of hardware
		Read/Write TWDR	Operations to TWCR				
			STA	STO	TWINT	TWEA	
0x60	SLA+W has received ACK has transmitted	No operation	x	0	1	0	Will receive data Will transmit NACK
		No operation	x	0	1	1	Will receive data Will transmit ACK
0x68	Arbitration failure during SLA+R/W transmit SLA+W received ACK transmitted	No operation	x	0	1	0	Will receive data Will transmit NACK
		No operation	x	0	1	1	Will receive data Will transmit ACK
0x70	General call address received ACK transmitted	No operation	x	0	1	0	Will receive data Will transmit NACK
		No operation	x	0	1	1	Will receive data Will transmit ACK
0x78	Arbitration failure during SLA+R/W transmit SLA+W received ACK transmitted	No operation	x	0	1	0	Will receive data Will transmit NACK
		No operation	x	0	1	1	Will receive data Will transmit ACK
0x80	Own data received ACK transmitted	Read data	x	0	1	0	Will receive data Will transmit NACK
		Read data	x	0	1	1	Will receive data Will transmit ACK
0x88	Own data received NACK transmitted	Read data	0	0	1	0	Will switch to slave mode unaddressed Will not acknowledge slave address and general call
		Read data	0	0	1	1	Will switch to slave mode unaddressed Will acknowledge slave address; Acknowledge general call when TWGCE=1
		Read data	1	0	1	0	Will switch to slave mode unaddressed Will not acknowledge slave address and general call; Will transmit START when BUS is idle
		Read data	1	0	1	1	Will switch to slave mode unaddressed

							Will acknowledge slave address; Acknowledge general call when TWGCE=1 Transmit START when BUS is idle.
0x90	General call received ACK transmitted	Read data	x	0	1	0	Will receive data Will transmit NACK
		Read data	x	0	1	1	Will receive data Will transmit NACK
0x98	General call received NACK transmitted	Read data	0	0	1	0	Will switch to slave mode unaddressed Will not acknowledge slave address and general call;
		Read data	0	0	1	1	Will switch to slave mode unaddressed Will acknowledge slave address; Acknowledge general call when TWGCE=1
		Read data	1	0	1	0	Will switch to slave mode unaddressed Will not acknowledge slave address and general call; Transmit START when BUS is idle.
		Read data	1	0	1	1	Will switch to slave mode unaddressed Will acknowledge slave address ; Acknowledge general call when TWGCE=1 Transmit START when BUS is idle.
0XA0	Receive STOP or REPEATED START in slave mode	No operation	0	0	1	0	Will switch to slave mode unaddressed Will not acknowledge slave address and general call;
		No operation	0	0	1	1	Will switch to slave mode unaddressed Will acknowledge slave address ; Acknowledge general call when TWGCE=1
		No operation	1	0	1	0	Will switch to slave mode unaddressed Will not acknowledge slave address and general call; Transmit START when BUS is idle.
		No operation	1	0	1	1	Will switch to slave mode unaddressed Will acknowledge slave address ;

							Acknowledge general call when TWGCE=1 Transmit START when BUS is idle.

Format and Status figure in slave receive mode is shown as below:



从机接收模式的格式和状态图

从机发送模式

在从机发送模式中，可以往主机接收器发送一定数量的数据字节。这一章节所提到的状态码均假设预分频控制位为“0”。为启动从机接收模式，要设置 TWAR 和 TWCR 寄存器。

TWAR 需设置如下：

Slave Transmitter Mode

In the Slave Transmitter mode, a number of data bytes are transmitted to a Master Receiver . All the status codes mentioned in this section assume that the prescaler bits are zero or are masked to zero.

To initiate slave receive mode, TWAR and TWCR registr shold be set.

TWAR should be set as below:

TWAR	TWA6	TWA5	TWA4	TWA3	TWA2	TWA1	TWA0	TWGCE
value	Device's Own Slave Address							

TWAR 的高 7 位是主机寻址时 TWI 接口会响应的从机地址。若 LSB 置位， TWI 会响应广播呼叫地址（0x00），否则忽略广播呼叫地址。

TWCR 需设置如下

The upper seven bits are the address to which the TWI Interface will respond when addressed by a Master. If the LSB is set, the TWI will respond to the general call address (0x00), otherwise it will ignore the general call address.

TWINT	TWEA	TWSTA	TWSTO	TWWC	TWEN	-	TWIE
0	1	0	0	0	1	0	x

TWEN 必须置位以使能 TWI 接口，TWEA 必须置位以使主机寻址（从机地址或广播呼叫）到自己时返回确认信息 ACK。TWSTA 和 TWSTO 必须清零。

初始化 TWAR 和 TWCR 之后，TWI 接口开始等待，直到自己的从机地址（或广播地址）被寻址。当紧跟着从机地址的数据方向位为“0”（表示写操作）时， TWI 进入从机接收模式。当数据方向位为“1”（表示读操作）时，TWI 进入从机发送模式。接收到自己的从机地址和读操作标志位后，TWINT 标志位被置位，有效的状态码也更新到 TWSR 中。各个状态码下合适的响应会在状态码表格中详细描述。需要注意的是，当主机模式下的 TWI 仲裁失败后也可以进入从机发送模式（见状态码 0xB0）。

如果在传输过程中 TWEA 位被复位，TWI 将在发送最后一个字节后切换到未寻址从机模式。主机接收器为最后一个字节的传输给出 NACK 或 ACK 后，TWSR 寄存器中的状态码将会更新为 0xC0 或 0xC8。如果主机接收器继续传输操作，从机发送器不会响应，主机将会接收到全“1”的数据（即 0xFF）。当从机发送完最后一个字节的数据（TWEA 被清零）并期望得到 NACK 响应，而主机想要接收更多的数据而发送 ACK 作为响应时，TWSR 会更新为 0xC8。

当 TWEA 位为“0”时， TWI 也不会响应自己的从机地址。不过 TWI 仍会监听总线，一旦 TWEA 被置位，就可以恢复地址识别并响应。也就是说，可以利用 TWEA 暂时将 TWI 接口从总线中隔离出来。

在除空闲模式外的其它休眠模式时，TWI 接口的时钟可以被关闭。若是能了从机接收模式，接口将利用总线时钟继续响应从机地址或广播地址。地址匹配将唤醒 MCU。在唤醒期间，TWI 接口将保持 SCL 为低电平，直到 TWINT 标志被清零。当 TWI 接口时钟恢复正常后可以接收更多的数据。

TWEN must be written to one to enable the TWI. The TWEA bit must be written to one to enable the acknowledgement of the device's own slave address or the general call address. TWSTA and TWSTO must be written to zero.

When TWAR and TWCR have been initialized, the TWI waits until it is addressed by its own slave address (or the general call address if enabled) followed by the data direction bit. If the direction bit is “1” (read), the TWI will operate in ST mode, otherwise SR mode is entered. After its own slave address and the write operation have been set, the TWINT Flag is set and a valid status code can be updated to TWSR. The status code is used to determine the appropriate software action. The appropriate action to be taken for each status code is detailed in below table. Note that the Slave Transmitter mode may also be entered if arbitration is lost while the TWI is in the Master mode (see state 0xB0).

If the TWEA bit is reset during a transfer, the TWI will enter into unaddressed slave mode after the last byte of has been transferred. Status code in TWSR will be updated to State 0xC0 or state 0xC8, depending on whether the Master Receiver transmits a NACK or ACK after the final byte. If master receiver keeps on transmition while slave transmit has no response, Master Receiver receives all “1” data（0xFF. State 0xC8 is entered if the Master demands additional data bytes (by transmitting ACK), even though the Slave has

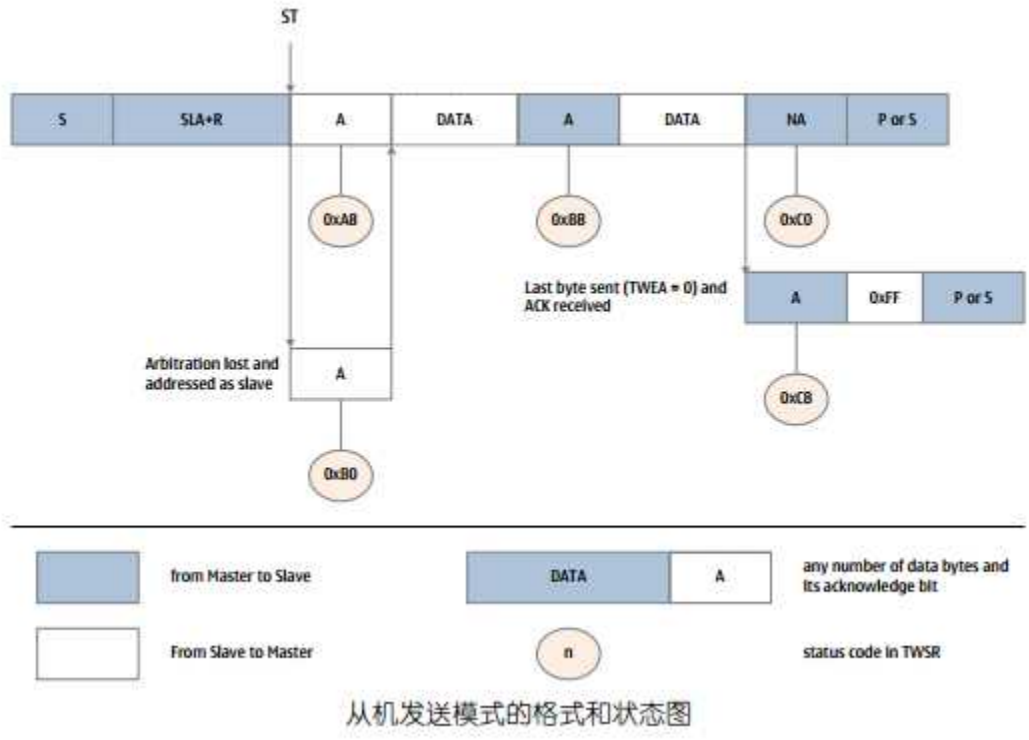
transmitted the last byte (TWEA zero and expecting NACK from the Master).
While TWEA is zero, the TWI does not respond to its own slave address. However, bus is still monitored by TWI and address recognition may resume at any time by setting TWEA. This implies that the TWEA bit may be used to temporarily isolate the TWI from the Bus.
In all sleep modes other than idle mode, the clock system to the TWI is turned off. If slave receive mode is set, the interface can still acknowledge slave address or the general call address by using the Bus clock. Address match will then wake up MCU and the TWI will hold the SCL clock low during the wake up and until the TWINT Flag is cleared.
Further data transmission will be received when TWI interface clock resume to normal state.

Status Codes in Slave Transmit Mode is as below:

Status Code	Bus and Hardware status	Application software response					Next action of hardware
		Read/Write TWDR	Operations to TWCR				
			STA	STO	TWINT	TWEA	
0XA8	SLA+R has received ACK has transmitted	Load data	x	0	1	0	Will transmit the last data Expect to receive NACK
		Load data	x	0	1	1	Will receive data Will transmit ACK
0XB0	Arbitration lost during SLA+R/W transmit SLA+R received ACK transmitted	Load data	x	0	1	0	Will transmit the last data Expect to receive NACK
		Load data	x	0	1	1	Will transmit data Will receive ACK
0XB8	Data transmitted ACK received	Load data	x	0	1	0	Will receive data Will transmit NACK
		Load data	x	0	1	1	Will receive data Will transmit ACK
0XC0	data transmitted NACK received	No operation	0	0	1	0	Will switch to slave mode unaddressed Will not acknowledge slave address and general call
		No operation	0	0	1	1	Will switch to slave mode unaddressed Will acknowledge slave address Acknowledge general call when TWGCE=1
		No operation	1	0	1	0	Will switch to slave mode unaddressed Will not acknowledge slave address and general call; Will transmit START when BUS is idle
		No operation	1	0	1	1	Will switch to slave mode unaddressed Will acknowledge slave address; Acknowledge general call when

							TWGCE=1 Transmit START when BUS is idle.
0XC8	Last data transmitted ACK received	No operation	0	0	1	0	Will switch to slave mode unaddressed Will not acknowledge slave address and general call
		No operation	0	0	1	1	Will switch to slave mode unaddressed Will acknowledge slave address Acknowledge general call when TWGCE=1
		No operation	1	0	1	0	Will switch to slave mode unaddressed Will not acknowledge slave address and general call; Will transmit START when BUS is idle
		No operation	1	0	1	1	Will switch to slave mode unaddressed Will acknowledge slave address; Acknowledge general call when TWGCE=1 Transmit START when BUS is idle.

Format and status in slave transmit mode is as below:



从机发送模式的格式和状态图

其他状态

有两个状态码没有相应的 TWI 状态定义，如下表所示：

其他状态码表

Miscellaneous States

There are 2 status code without corresponding TWI status definition, details as per below:

Status Code (TWSR) Prescaler Bits are 0	Status of the 2-wire Serial Bus and 2-wire Serial Interface Hardware	Application Software Response					Next Action Taken by TWI Hardware
		To/from TWDR	To TWCR				
			STA	STO	TWINT	TWEA	
0xF8	No relevant state information available; TWINT = "0"	No TWDR action	No TWCR action				Wait or proceed current transfer
0x00	Bus error due to an illegal START or STOP condition	No TWDR action	0	1	1	X	Only the internal hardware is affected, no STOP condition is sent on the bus. In all cases, the bus is released and TWSTO is cleared.

状态码 0xF8 表示当前没有相关信息，因为 TWINT 标志为“0”。这种状态可能发生在 TWI 接口没有参与串行传输或当前传输还没有完成.

状态 0x00 表示串行传输过程中发生了总线错误。当非法的 START 或 STOP 出现时总线错误就会发生。比如说在地址和数据、地址和 ACK 之间出现了 START 或 STOP。总线错误将置位 TWINT。为了从错误中恢复，必须置位 TWSTO，并通过写“1”以清零 TWINT。这将使 TWI 接口进入未寻址从机模式而不会产生 STOP，以及释放 SCL 和 SDA，并清零 TWSTO 位。

Status code 0XF8 indicates that there is no relevant information, as TWINT flag is zero, this could happen in case that TWI interface does not transfer serial data or current ongong transmission is not complete.

Status code 0x00 indicates BUS error during serial transmission when there is illegal START or STOP, i.e. there is START or STOP between address and data, data and ACK. TWINT will be set if there is bus error. To correct this erro, must set TWSTO, and clear TWINT by writing one, by this way TWI interfacecan enter into unaddressed slave mode without STOP, and release SCL and SDA, meanwile TWSTO is cleared.

组合模式

在某些情况下，为了完成期望的工作，必须将几种 TWI 模式组合起来。 例如，从串行 EEPROM

读取数据，典型的传输包括以下步骤：

1. 传输必须启动；
2. 必须告诉 EEPROM 应该读取数据的位置；
3. 必须完成读操作；
4. 传输必须结束。

注意数据可以从主机传送到从机，反之亦然。主机告诉从机要读取数据的位置，采用的是主机发送模式。接下来，从从机读取数据，采用的是主机接收模式。传输的方向会改变。主机必须保持各个阶段的总线控制权，所有的步骤是不间断的操作。如果在多主机系统中，在步骤 2 和 3 之间另有主机改变了读取数据的位置，则打破了这一原则，主机读取数据的位置会是错误的。改变数据传输的方向是通过在传送地址字节和接收数据之间发送 REPEATED START 来实现的。发送 REPEATED START 之后，主机仍拥有总线控制权。

下图描述了这个传输过程：

Combining Several TWI Modes

In some cases, several TWI modes must be combined in order to complete the desired action. For example reading data from a serial EEPROM. Typically, such a transfer involves the following steps:

1. The transfer must be initiated.

2. The EEPROM must be instructed what location should be read.

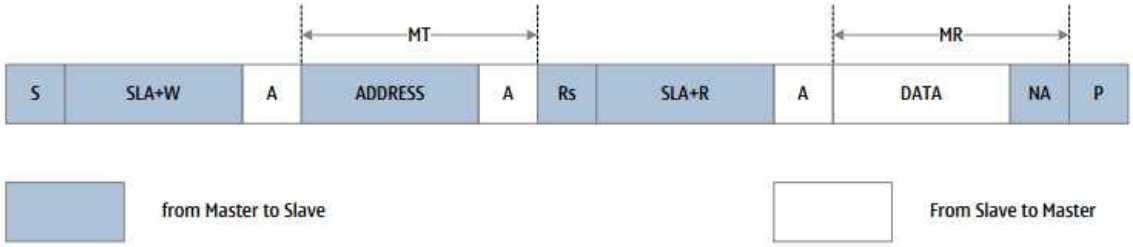
3. The reading must be performed.

4. The transfer must be finished.

Note that data is transmitted both from Master to Slave and vice versa. The Master must instruct the Slave what location it wants to read, requiring the use of the MT mode. Subsequently, data must be read from the Slave, implying the use of the MR mode. Thus, the transfer direction must be changed. The Master must keep

control over the bus during all these steps, and all steps should be carried out continuously. This principle will be violated in a multi mastersystem, if another Master can alter the data pointer between steps 2 and 3, and the Master will read the wrong data location. Such a change in transfer direction is accomplished by transmitting a REPEATED START between the transmission of the address byte and reception of the data. After a REPEATED START, the Master keeps ownership of the bus.

The following figure shows the flow in this transfer:



组合多种 TWI 模式来访问串行 EEPROM 图

多主机系统及仲裁

如果有多个主机连接在同一 TWI 总线上，它们中的一个或多个也许会同时开始数据传输。TWI 协议确保在这种情况下，通过一个仲裁过程，允许其中的一个主机进行传送也不会丢失数据。下面以两个主机试图向从机发送数据为例来描述总线仲裁的过程。

有几种不同的情况会产生总线仲裁过程：

- ？ 两个或更多的主机同时与一个从机进行通信。在这种情况下，无论主机还是从机都不知道总线上有竞争；
- ？ 两个或更多的主机同时对同一个从机进行不同的数据或操作方向访问。这种情况下就会发生仲裁，在 READ/WRITE 位或数据位。当有其它主机往 SDA 线上发送“0”时，往 SDA 线
- 上发送“1”的主机就会仲裁失败。失败的主机将会切换到未被寻址的从机模式，或者等
- 待总线空闲时发送一个新的 START 信号，这都取决于应用软件的操作。
- ？ 两个或更多的主机访问不同的从机。在这种情况下，总线仲裁发生在 SLA 阶段。当有其它主机往 SDA 线上发送“0”时，往 SDA 线上发送“1”的主机就会仲裁失败。在 SLA 总线仲裁时失败的主机将切换到从机模式，并检查自己是否被获得总线控制权的主机寻址。如果被寻址，它将进入 SR 或 ST 模式，这取决于 SLA 后面的 READ/WRITE 位。如果未被寻址，它将切换到未被寻址的从机模式，或者等待总线空闲时发送一个新的 START 信号，这取决于应用软件的操作。

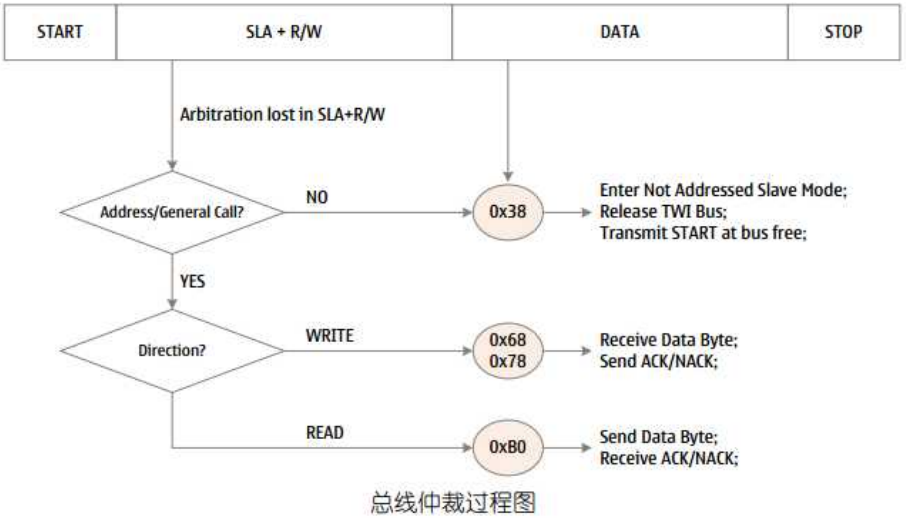
Multi-master Systems and Arbitration

If multiple masters are connected to the same TWI bus, transmissions may be initiated simultaneously by one or more of them. The TWI standard ensures that such situations are handled in such a way that one of the masters will be allowed to proceed with the transfer, while no data will be lost. An example of a BUS arbitration situation is depicted below, where two masters are trying to transmit data to a Slave Receiver. Several different scenarios may arise during arbitration, as described below:

- ◠ Two or more masters are performing identical communication with the same Slave. In this case, neither the Slave nor any of the masters will know about the bus contention.
- ◠ Two or more masters are accessing the same Slave with different data or direction bit. In this case, arbitration will occur, either in the READ/WRITE bit or in the data bits. The masters trying to transmit a one to SDA while another Master transmits a zero will lose the arbitration. Losing masters will switch to not addressed Slave mode or wait until the bus is free and transmit a new START condition, depending on application software action.

- Two or more masters are accessing different slaves. In this case, arbitration will occur in the SLA bits. Masters trying to output a one on SDA while another Masteroutputs a zero will lose the arbitration. Masters losing arbitration in SLA will switch to Slave mode to check if they are being addressed by the winning Master. If addressed, they will switch to SR or ST mode, depending on the value of the READ/WRITE bit after SLA. If they are not being addressed, they will switch to not addressed Slave mode or wait until the bus is free and transmit a new START condition, depending on application software action.

Below picture shows bus arbitration process:



总线仲裁过程图

Register Definition

TWI Register List

Register	Address	Default Value	Comment
TWBR	0x B8	0x00	TWI bite rate register
TWSR	0x B9	0x00	TWI status register
TWAR	0x BA	0x00	TWI address register
TWDR	0x BB	0x00	TWI data register
TWCR	0x BC	0x00	TWI control register
TWAMR	0x BD	0x00	TWI address mask register

TWBR--TWI bite rate register

TWBR--TWI bite rate register								
Address: 0x B8				Default value:0x00				
Bit	7	6	5	4	3	2	1	0
	TWBR7	TWBR6	TWBR5	TWBR4	TWBR3	TWBR2	TWBR1	TWBR0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	Name	Comment						
7:0	TWBR[7:0]	TWI bite rate selection bit TWBR is prescaler factor of bite rate generator, which is a prescaler used for generating SCL clock in master mode. Bit rate equation is as below:						

		fscl = fsys/(16 + 2*TWBR*4TWPS)

TWSR--TWI status register

TWSR--TWI status register								
Address: 0x B9				Default value:0xF8				
Bit	7	6	5	4	3	2	1	0
	TWS7	TWS6	TWS5	TWS4	TWS3	-	TWPS1	TWPS0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Comment
7:3	TWBR[7:3]	TWI status flag bit 5 bit TWS indicates state of TWI logic and Bus. Each status value has its own definition, details referring to TWI working mode. During detection, values which read from TWSR including 5 bit status value and 2 bit prescaler control bit, should mask prescaler bit to zero. Such prescaler configuration is indepandant from status detection.
2	-	reserved
1	TWPS1	TWI presclaer control high TWPS[1:0] consisted of TWPS1 and TWPS0 is used to control bite rate presclaer factory, it can control bite rate together with TWBR
0	TWPS0	TWI presclaer control low TWPS[1:0] consisted of TWPS0 and TWPS1 is used to control bite rate presclaer factory, it can control bite rate together with TWBR

TWPS[1:0]	Prescaler factor
0	1
1	4
2	16
3	64

TWAR--TWI adress register

TWSR--TWI adress register								
Address: 0x BA				Default value:0x00				
Bit	7	6	5	4	3	2	1	0
	TWAR6	TWAR5	TWAR4	TWAR3	TWAR2	TWAR1	TWAR0	TWGCE
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Comment
7:0	TWA[7:0]	TWI slave address bit TWA is TWI slave address. When TWI is in slave mode, TWI will acknowledge according to this address, which is not in need in master mode. However in multi-master system, slave address must be set in order other masters can access.
0	TWGCE	TWI general call enable control bit When setting TWGCE to one, enable TWI bus geneeral call acknowlege When setting TWGCE to zero, disable TWI bus geneeral call acknowlege TWI acknowledges bus general call when TWGCE is set and received address frame is 0x00.

TWDR--TWI Data Register

TWDR--TWI Data Register								
Address: 0x BB				Default value:0xFF				
Bit	7	6	5	4	3	2	1	0
	TWD7	TWD5	TWD5	TWD4	TWD3	TWD2	TWD1	TWD0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	Name	Comment						
7:0	TWD[7:0]	TWI data register TWD is going to transmit a following byte in BUS or the last byte just received from BUS.						

TWCR--TWI control register

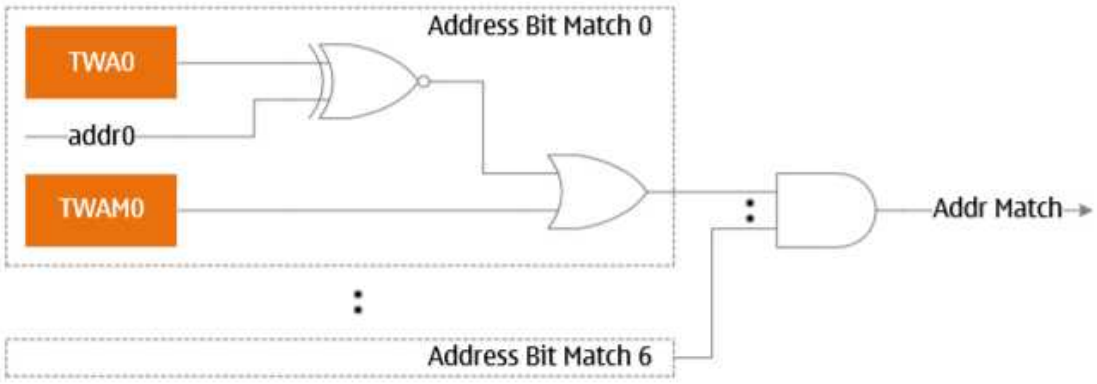
TWCR--TWI control register								
Address: 0x BC				Default value:0x00				
Bit	7	6	5	4	3	2	1	0
	TWINT	TWEA	TWSTA	TWSTO	TWWC	TWEN	-	TWIE
	R/W	R/W	R/W	R/W	R/W	R/W	-	R/W
Bit	Name	Comment						
7	TWINT	TWI interrupt flag bit Hardware will be set when TWI completed current worka nd wishes to introduce application software. If globe interrupt is bit and TWIE is set, TWI interrupt will generate, MCU will execute TWI interrupt routine. When TWINT flag is set, SCL condition low will be prolonged. TWINT can only be cleared by writing one, it can not be cleared by hardware automatically even interrupt routine is executed. Meanwhile noted that TWI is activated once this bit is cleared. So before clearing TWINT, need to access to TWAR, TWAMR, TWSR and TWDR register first.						
6	TWEA	TWI enable acknowledge control bit TWEA is used to control generation of acknowledgment pulse. When setting TWEA to one and one of the below conditions will be satisfied, acknowledgment pulse will be generated in TWI BUS: 1) Receive device's slave address; 2) Receive general call when TWGCE is set 3) Receive one byte data in master receive or slave receive mode When TWEA is set to zero, device will disconnect from TWI bus temperarily. After TWEA is set to one, address recognition resume again.						
5	TWSTA	TWI start status contro When CPU wishes itself to become master in TWI bus, TWSTA is set. Hardware will detect if bus can be used or not. When bus is free, start status will be generated in BUS, otherwise TWI will wait untill stop status is detected and than claim its wish to become master by generating start status. After transmitting start status, softwar must clear TWSTA bit.						
4	TWSTO	TWI Stop status control In master mode TWI will generate stop status in BUS when TWSTO is one, than to clear TWSTO automatically. In slave mode, TWI can resume from error state by						

		setting TWSTO. In this case stop status will not appear, but only TWI will return to defined but not addressed slave mode, meanwhile to release SCL and SDA condition line to high resistance state.
3	TWWC	TWI write collison flag bit When TWINT flag is low, wrting TWDR register to set TWWC flag bit. When TWINT flag is high, wrting TWDR register to clear TWWC flag bit.
2	TWEN	TWI enable control bit TWEN enables TWI operations and activate TWI interface. When TWEN is one, TWI control IO pin and link IO pin to SCL and SDA pin. When TWEN is zero, TWI interface modula is closed, all transmission including ongong operation will terminate.
1	-	reserved
0	TWIN	When setting TWIE to one, and globe interrupt is set, TWI interrupt request will be activated only if TWINT flag is high.

TWAMR--TWI address mask register

TWAMR--TWI address mask register								
Address: 0x BD				Default value:0x00				
Bit	7	6	5	4	3	2	1	0
	TWAR7	TWAR5	TWAR5	TWAR4	TWAR3	TWAR2	TWAR1	TWGCE
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	Name	Comment						
7:0	TWAM[6:0]	TWI address mask control bit TWAM is slave address mask control of 7 bit. Each bit is used to mask(disable) relevant address bit in TWAR. When address bit is masked, address match logic will ignore the result compared between received address bit and relevant TWA bit. Below figure shows details about address match logic						
0	-	reserved						

TWI address match logic:



模拟 比较器 0 (AC0)

ㄏ 10mV 的比较精度 ㄏ

出厂失调校准

支持 3 路片外模拟输入

支持 ADC 的多路复用输入(ADMUX)

支持内部差分放大器输入(DFFO)

支持内部 8 位 DAC 输入(DAO)

可编程输出数字滤波控制

Analog Comparator

- **10mv comparision accurancy**
- **Support 3 channel off-chip analog input**
- **Support ADC multi-channel alternate input(ADMUX)**
- **Support internal differential amplifier input (DFFO)**
- **Support internal 8-bit DAC input(DAO)**
- **Programmable output digital filter control**

综述

模拟比较器对输入比较器正端与负极的电平进行比较，当正端电压比负端电压高时，模拟 比较器的输出 ACO 被置位。当 ACO 的电平发生变化时，信号的边沿可用来触发中断。输出 信号 ACO 还可用来触发定时计数器 1 的输入捕捉以及对定时器产生的 PWM 输出进行控制。

LGT8FX8P 集成模拟比较器 AC0，包括一个多路模拟输入选择器，比较器正、负端输入源可 以选择来自外部端口或者来自多种内部产生的参考源。模拟比较器本身支持失调校准，可以 保证比较器工作的一致性。比较器支持一个可选的硬件迟滞功能，用于改善比较器输出的稳 定性。

同时比较器输出端集成一个硬件可以编程数字滤波器，可以根据应用需求，选择合适 的滤波设置，以获得更加稳定的比较输出。 比较器输出状态可以直接通过寄存器读取，也可以产生中断请求，实现更高效的实时事件 俘获功能。比较器的输出也可以直接输出到外部 IO 端口

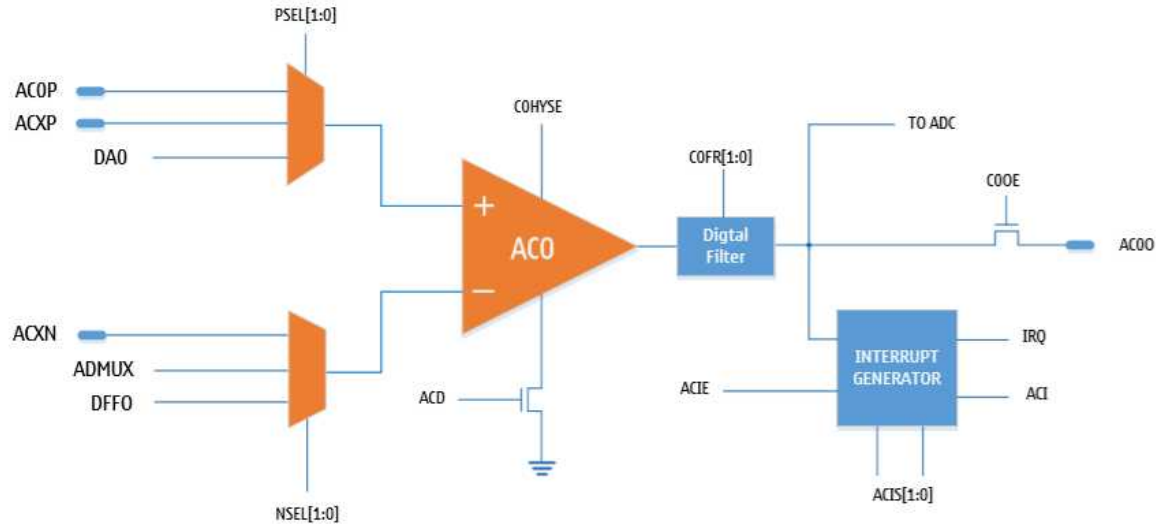
Overview

The Analog Comparator compares the input values on the positive pin and negative pin. When the voltage on the positive pin is higher than the voltage on the negative pin , the Analog Comparator output, ACO, is set. When pin level of ACO changes, edge of signal can be used to trigger interrupt. The comparator's output can be set to trigger the Timer/Counter1 Input Capture function and control PWM output from Timer.

LGT8FX8P ACO includes a multi-channel analog input controller, input source of comparator negative and positive pin can choose either externa port or serveral internal reference source. ACO itself supports detunuing calibration to ensure of consistency. Comparator supports hardware delay as option to improve output stability.

A digital filter programmable by hardware is intergrated in comparator output, as per application demand, filter should be configured correctly to get more stable comparator output. Comparator output status can directly be read via register, also can generate interrupt request, to achieve more effective real-time event capture. Output of comparator can also output to external IO port.

Operational Amplifier/Analog Comparator 0 Construction Chart is as below:



模拟比较器 0 功能示意图

模拟比较器的输入

模拟比较器的两个输入端都支持多种可选输入源。正端的输入三路可选：

1. 外部独立模拟输入 AC0P

2. 2. 模拟比较器 0/1 公用模拟输入 ACXP

3. 3. 内部 8 位 DAC 的输出 DAO

Analog Comparator Input

Both input of analog comparator support multiplexed intput source. Positive pin has 3 for option:

1. External indepandant analog input ACOP

2. Analog comparator 0/1 public analog input ACXP

3. Output DAO of internal 8-bit DAC

输入源的选择由控制状态寄存器 C0SR 中的 C0BG 位以及 C0XR 寄存器的 C0PS0 位共同控 制，具体请参考本章节寄存器描述部分。

AC0P 为 AC0 专用正端模式输入通道。注意在不同封装片 AC0P 的脚位略有区别。QFP48 封装的 AC0P 为独立端口。QFP32 封装此 AC0P 端口与 PD6 并联到一个端口上。

ACXP 为比较器 0/1 公用正端输入。LGT8FX8P 内部有两个模拟比较器，ACXP 同时连接到 两个比较器的正端多路复用选择器，便于实现两个比较器的协同工作。

DAO 来自内部 8 位 DAC 的输出。DAC 的参考源可以选择来自系统电源，内部参考或者来 自外部参考的输入。DAC 的配置请参考 DAC 相关章节。

Input selection is controlled together by COBG in control status register COSR and COPS0 in COXR register. For details please refer to register definition of this chapter.

ACOP is the delicated input channel in positive mode of ACO. Note that for different package, pins locatioon of ACOP might slightly different. For QFP48 package, ACOP is indepandant port. While for QFP32 ACOP port is in parallel with PD6 to one port.

ACXP is 0/1 public positive input of comparator. There are 2 analog comparator in LGT8FX8P. ACXP is connected with positive multiplexed alternate controller of both comparator to acheive coopeation of both comparator.

DAO is from output of internal 8-bit DAC. Reference source of DAC can be choose from system power, internal reference or external reference input. DAC configuration refers to DAC related section.

COBG	COPSO	ACO positive input source
0	0	ACOP
0	1	ACXP

1	0	DAO
1	1	close comparator positive input channel

负端输入也可以选择三种不同的模拟输入:

1. 比较器 0/1 公用模拟输入 ACXN

2. ADC 多路器的输出 ADMUX

3. 内部差分放大器输出 DFFO

比较器负端输入通道选择由来自 ADC 模块的 ADCSRB 寄存器中的 CME00/01 位控制。

当比较器负端输入选择为 ADMUX 时，需要通过 ADC 模块的 ADMUX 寄存器 CHMUX 位选择模 拟输入通道，这种模式下，比较器的输入可以实现更加灵活的扩展。

ACXN 为比较器 0/1 公用的负端输入，便于实现比较器 0/1 的协同工作；

DFFO 来自内部的差分放大器输出。差分放大器可选 x1/x8/x16/x32 增益控制，可实现小 信号的检测与测量。

Negative input can select 3 analog input:

1. Comparator 0/1 public analage input ACXN
2. ADC multiplexed output ADMUX
3. Internal differential amplifier output DFFO

Comparator negative input is controlled by CEM00/01 in ADCSRB register of ADC.

When comparator negative pin is set to ADMUX, analog input channel should be selected by CHMUX bit in ADMUX register of ADC. In this mode, input of comparator can be extended more flexiblly.

ACXN is the public negative input of comparator 0/1, to acheive cooperation of comparator 0/1.

DFFO is from internal differential amplifier output. DA can choose gain control of x1/x8/x16/x32 to acheive detection and measurement on minor singal.

COBG	COPSO	ACO nagative input source
0	0	ACXN
0	1	ACMUX
1	0	DFFO
1	1	close comparator negative input channel

比较器输出滤波

比较器输出端内部支持一个可控的迟滞电。用户可以通过 C0XR 寄存器的 C0HYSE 位使能迟滞电路。迟滞电路可以消除比较器状态变化过程的不稳定状态，达到输出滤波功能。

建议用户在使用比较器时，打开迟滞电路，获得一个稳定的比较器输出。

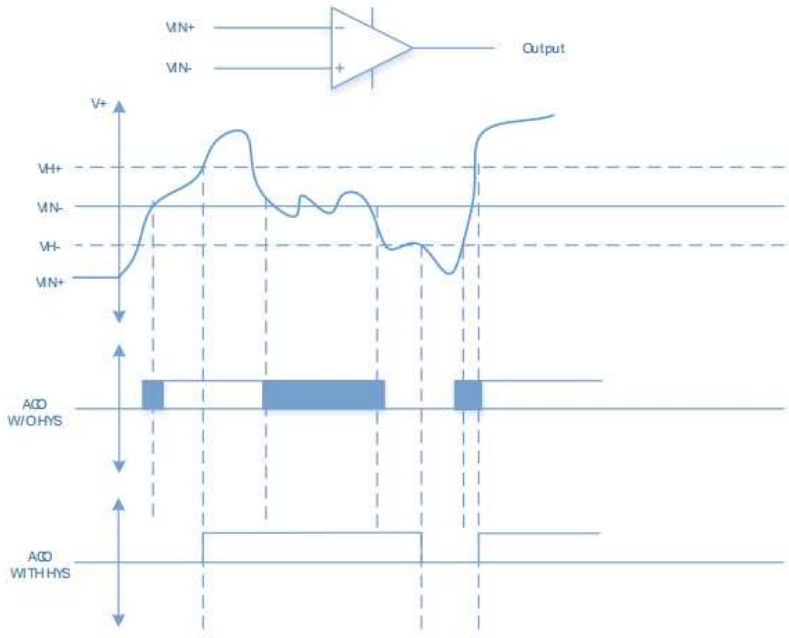
如下图所示，迟滞电路位于比较器模拟输出与数字输出之间。当比较器正端的输入电压 VIN+大于(VIN- + VH+)时，比较器 COUT 输出为高；当 VIN+电压小于(VIN- - VH-)时，比较器输出低。迟滞电路避免了当比较器正端电压接近负端电压时，电路本身带来的抖动。比较器迟滞电压与比较器输出关系图：

Comparator Output Filter

There is a controllable delay circuit in comparator output port, which can be enabled by setting C0HYSE in C0XR register. The delay circuit can eliminate unstable state when comparator state is changing, so that output filter function can be achieved.

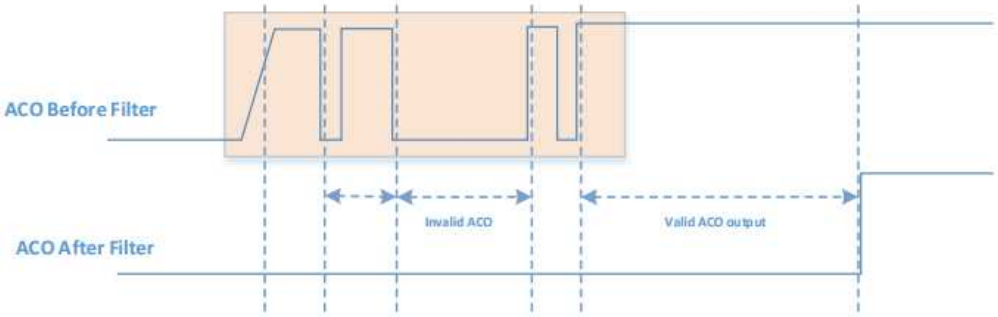
It is recommended for users to open delay circuit when using comparator to get a stable comparator output. As shown below, delay circuit is located between comparator analog output and digital output. When input voltage VIN+ in comparator positive pin is higher than (VIN- + VH+), comparator COUT output is high. Otherwise comparator output is low. Delay circuit can avoid vibration caused by circuit itself when comparator positive voltage is close to negative voltage. Relationship chart between comparator delay circuit

and comparator output is as below:



尽管迟滞电路对于抑制接近比较器阀值的电压纹波非常有效，但实际应用环境中，输入信号会受到不同强度的干扰。 较强的干扰可能会导致输入电平瞬间抬高， 超出迟滞电路的阀值范围，无法被有效抑制。**LGT8FX8P** 在比较器输出端集成了一个可编程的数字滤波器，可以滤除瞬时干扰对比较器输出产生的影响。数字滤波器可以根据应用需求，选择合适的滤波时间宽度，只有当比较器的输出稳定持续满足滤波时间限制，滤波电路才更新比较器的输出。而达到一个更加稳定的输出结果。

Delay circuit is very useful and effective to restrain voltage ripple which is close to comparator threshold value, however in real application, input signal will be intervened of some degree. A strong intervene will make input pin level pull up instantly, if it is over delay circuit threshold value, voltage ripple cannot be restrained effectively. In LGT8FX8P there is a programmable digital filter integrated in comparator output to filter instant intervenes to comparator output. As per actual demand, digital filter can choose suitable filter time width. Filter circuit will only update comparator output when comparator output can stably and continuously satisfy filter time limitation to achieve a more stable output result.



比较器输出滤波时序

AC0 的数字滤波通过 C0XR 寄存器的 C0FEN 以及 C0FS 位控制，具体设置方式请参考本章寄存器定义部分。

ACO digital filter is set by C0FEN and C0FS bit in C0XR register, for detailed setting refer to register definition in this chapter.

比较器输出与 PWM 控制

LGT8FX8P 支持多通道 PWM 输出，PWM 信号可以与比较器模块配合使用。比较器的输出，可用于直接关断 PWM 信号，从而实现比较灵活的 PWM 保护方案。

与 PWM 输出相关的控制，请参考定时器章节的相关部分

Comparator output and PWM control

LGT8FX8P supports multi-channel PWM output, PWM signal can be used together with comparator modular. Comparator output can be used to shut down PWM signal directly to achieve flexible PWM protection solution.

Related control about PWM output, refer to relevant section about Timer.

Register Definition

C0SR—AC0 control and status register

C0SR—AC0 control and status register								
Address:0x50				Default value：0x80				
Bit	7	6	5	4	3	2	1	0
Name	C0D	C0BG	C0O	C0I	C0IE	C0IC	C0IS1	C0IS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	Name	Comment						
7	C0D	Analog comparator disable When setting C0D to logic one, analog comparator is shut down. When setting C0D to logic zero, analog comparator is open.						
6	C0BG	Analog comparator 0 positive input source selection bit. AC0 positive input source is set together by C0PS0 in C0BG and C0XR register, {C0BG, C0PS0} = 00= AC0P is positive input 01= ACXP is negative input 10= output of internal DAC is positive input 11= shut down positive input source of AC0						
5	C0O	Output status bit of analog comparator After synchronization, output of analog comparator will connect with C0O bit directly. Output value of analog comparator can be gotten by software reading C0O bit value.						
4	C0I	Analog comparator interrupt flag When output event of analog comparator triggers interrupt defined by C0IS, C0I is set. When interrupt enable bit C0IE is one and globe interrupt is set, interrupt occurs. When executing analog comparator interrupt program, C0I will be cleared automatically or by writing one.						
3	C0IE	Analog comparator interrupt enable bit When setting C0IE to logic one, and globe interrupt is enabled, AC0 interrupt is enabled. When setting C0IE to logic zero, AC0 interrupt is disabled.						
2	C0IC	Analog comparator input capture enable bit C0IC=1, Timer/Counter 1 input capture source is from output of analog comparator. C0IC=0, Timer/Counter 1 input capture source is from external pin ICP1.						

1	C0IS1	Analog comparator interrupt mode control high	
0	C0IS0	Analog comparator interrupt mode control low. C0IS0 and C0IS1 consist of C0IS[1:0] which is used to control interrupt trigger method of analog comparator.	
		C0IS[1;0]	Interrupt mode
		00	Triggered by rising edge or falling edge of AC0
		01	Reserved
		10	Triggered by falling edge of AC0
		11	Triggered by rising edge of AC0

ADCSRB—ADC control and status register B

ADCSRB—ADC control and status register B								
Address:0x7B				Default value：0x00				
Bit	7	6	5	4	3	2	1	0
Name	CEM01	CEM00	CME11	CME10	ACTS	ADTS2	ADTS1	ADTS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	Name	Comment						
7	CEM01	AC0 negative input selection, CME0 = {CME01, CME00}						
6	CME00	00: external port ACXN is used as AC0 negative input 01: ADC multi-channel alternate output is used as AC0 negative input 10: Differential amplifier output is used as AC0 negative input 11: Shut down AC0 negative input source						
5	CME11	AC1 negative input selection, CME1 = {CME11, CME10}						
4	CME10	00: external port ACXN is used as AC1 negative input 01: external port AC1N is used as AC1 negative input 10: internal 1/5 bypass voltage of ADC is used as AC1 negative input 11: output of differential operation is used as AC1 negative input						
3	ACHS	AC trigger source channel selection 0- AC0 output is used as ADC auto-swtich trigger source 1- AC1 output is used as ADC auto-swtich trigger source						
2:0	ADTS	Refer to ADC register section						

C0XR—AC0 Auxiliary control register

C0XR—AC0 Auxiliary control register								
Address:0x51				Default value：0x00				
Bit	7	6	5	4	3	2	1	0
Name	-	C0OE	C0HYSE	C0PS0	C0WKE	C0FEN	C0FS1	C0FS0
R/W	-	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	Name	Comment						
7	-	reserved						
6	C0OE	AC0 comparator output to external port enable control C0OE=1, AC0 comparator is output to external port PD2 C0OE=0, comparator output to external port is disabled.						
5	C0HYSE	AC0 output delay enable control 1= enable output delay 2= disable output delay						

4	C0PS0	AC0 negative input source selection LOW Negative input source is control together by C0PS0 and C0BG, for detail refer to C0SR register definition.
3	COWKE	AC0 is used to enable sleep wake up 1= enable wake-up of comparator output 0= shut down wake-up of comparator output
2	C0FEN	Comparator digital filter enable control 1= enable digital filter 2= disable digital filter
1:0	C0FS[1:0]	Comparator digital filter width configuration 00= shut down 01= 32us 10= 64us 11= 96us

模拟比较器 1 (AC1)

- 10mV 的比较精度
- 出厂失调校准
- 支持 4 路片外模拟输入
- 支持内部 1/5 分压器输入(VDO)
- 支持内部差分放大器输入(DFFO)
- 支持内部 8 位 DAC 输入(DAO)
- 可编程输出滤波控制

Analog Comparator 1 AC1

- 10mv comparision accurancy
- Factory distuning calibration
- Support 4 channel off-chip analog input
- Support internal 1/5 voltage divider input(VDO)
- Support internal differential amplifier input (DFFO)
- Support internal 8-bit DAC input(DAO)
- Programmable output digital filter control

综述

模拟比较器对输入比较器正端与负极的电平进行比较，当正端电压比负端电压高时，模拟比较器的输出 ACO 被置位。当 ACO 的电平发生变化时，信号的边沿可用来触发中断。输出信号 ACO 还可用来触发定时计数器 1 的输入捕捉以及对定时器产生的 PWM 输出进行控制。

LGT8FX8P 集成模拟比较器 AC1， 包括一个多路模拟输入选择器， 比较器正、负端输入源可以选择来自外部端口或者来自多种内部产生的参考源。

模拟比较器本身支持失调校准，可以保证比较器工作的一致性。 比较器支持一个可选的硬件迟滞功能， 用于改善比较器输出的稳定性。同时比较器输出端集成一个硬件可以编程数字滤波器， 可以根据应用需求，选择合适的滤波设置，以获得更加稳定的比较输出。

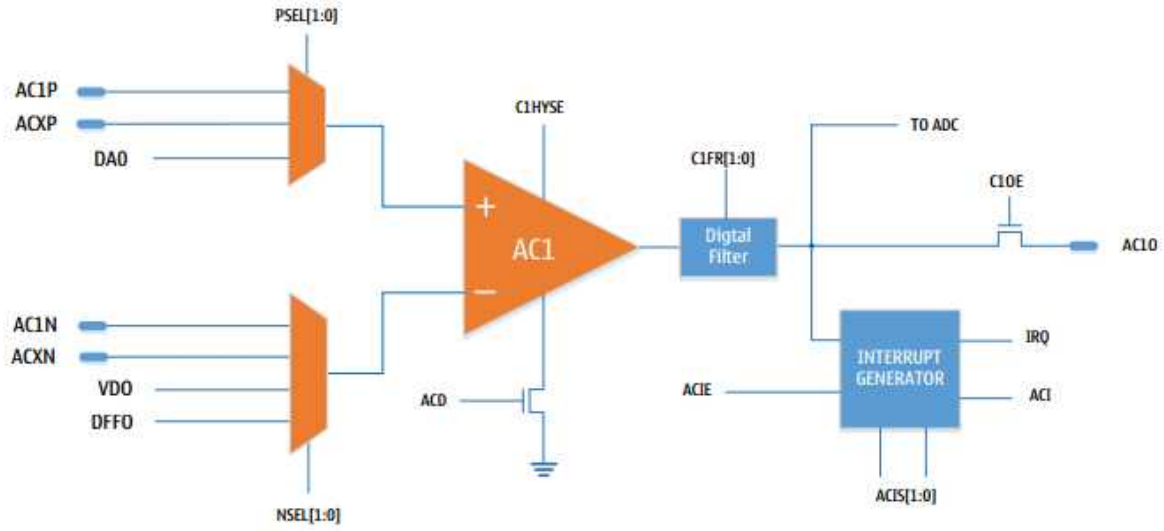
比较器输出状态可以直接通过寄存器读取， 也可以产生中断请求， 实现更高效的实时事件俘获功能。比较器的输出也可以直接输出到外部 IO 端口。

Overview

The Analog Comparator compares the the positive pin level and negative pin level of input comparator. When

the voltage on the positive pin is higher than the voltage on the negative pin , the Analog Comparator output, ACO, is set. When pin level of ACO changes, edge of signal can be used to trigger interrupt.
Output signal AC0 can be set to trigger the Timer/Counter1 Input Capture function and control PWM output from Timer. LGT8FX8P AC1 includes a multi-channel analog input controller, input source of comparator negative and positive pin can choose either externa port or serveral internal reference sources. ACO itself supports detunuing calibration to ensure of consistency. Comparator supports hardware delay as option to improve output stability. A digital filter programmable by hardware is intergrated in comparator output, as per application demand; filter should be configured correctly to get more stable comparator output.
Comparator output status can directly be read via register, also can generate interrupt request, to achieve more effective real-time event capture. Output of comparator can also output to external IO port.

Operational Amplifier/Analog Comparator 1 Construction Chart is as below:



模拟比较器 1 模块结构示意图

模拟比较器的输入

模拟比较器的两个输入端都支持多种可选输入源。正端的输入三路可选：

1. 外部独立模拟输入 AC1P
2. 模拟比较器 0/1 公用模拟输入 ACXP
3. 内部 8 位 DAC 的输出 DAO

Analog Comparator Input

Both input of analog comparator support multiplexed input source. Positive pin has 3 for option:

4. External indepandant analog input AC1P
5. Analog comparator 0/1 public analog input ACXP
6. Output DAO of internal 8-bit DAC

输入源的选择由控制状态寄存器 C1SR 中的 C1BG 位以及 C1XR 寄存器的 C1PS0 位共同控制，具体请参考本章节寄存器描述部分。

AC1P 为 AC1 专用正端模式输入通道。

ACXP 为比较器 0/1 公用正端输入。 LGT8FX8P 内部有两个模拟比较器， ACXP 同时连接到两个比较器的正端多路复用选择器，便于实现两个比较器的协同工作。

DAO 来自内部 8 位 DAC 的输出。 DAC 的参考源可以选择来自系统电源，内部参考或者来自外部参考的输入。DAC 的配置请参考 DAC 相关章节。

Input selection is controlled together by C1BG in control status register C1SR and C1PS0 in C1XR register. For details please refer to register definition of this chapter.
AC1P is the delicated input channel in positive mode of AC1.
ACXP is 0/1 public positive input of comparator. There are 2 analog comparator in LGT8FX8P. ACXP is connected with positive multiplexed alternate controller of both comparators to acheive coopeation of both comparator.
DAO is from output of internal 8-bit DAC. Reference source of DAC can be choose from system power, internal reference or external reference input. DAC configuration refers to DAC related section

C1BG	C1PSO	AC1 positive input source
0	0	ACXP
0	1	AC1P
1	0	DAO
1	1	Shut down comparator positive input channel

负端输入也可以选择 4 种不同的模拟输入:

1. 外部模拟输入 AC1N 作为 AC1 负端输入
2. 比较器 0/1 公用负端输入 ACXN
3. ADC 内部 1/5 分压器输出作为 AC1 的负端输入
4. 内部差分放大器输出 DFFO 作为 AC1 的负端输入

比较器负端输入通道选择由来自 ADC 模块的 ADCSRB 寄存器中的 CME11/10 位控制。当比较器负端输入选择为 ADC 内部多路分压器输出时，需要通过 ADC 模块的 ADCSRC 寄存器 VDS 位选择多路分压的输入参考源。

ACXN 为比较器 0/1 公用的负端输入，便于实现比较器 0/1 的协同工作；

DFFO 来自内部的差分放大器输出。 差分放大器可选 x1/x8/x16/x32 增益控制， 可实现小信号的检测与测量。

Negative input can select 4 analog input:

1. External analog input AC1N is used as AC1 negative input
2. Comparator 0/1 public negative input ACXN
3. ADC internal 1/5 voltage divider as AC1 negative input
4. Internal differential amplifier output DFFO as AC1 negative input

Comparator negative input channel is controlled by CEM11/10 in ADCSRB register of ADC.
When comparator negative input chooses ADC internal multi-channel voltage divider output, input reference source of multi-channel voltage is set by VDS in ADCSRC register of ADC.
ACXN is the public negative input of comparator 0/1, to acheive cooperation of comparator 0/1.
DFFO is from internal differential amplifier output. DA can choose gain control of x1/x8/x16/x32 to acheive detection and measurement on minor singal.

CME11	CME10	AC1 negative input
0	0	ACXN
0	1	AC1N
1	0	VD0
1	1	DFF0

比较器输出滤波

比较器输出端内部支持一个可控的迟滞电。用户可以通过 C1XR 寄存器的 C1HYSE 位使能迟滞电路。迟滞电路可以消除比较器状态变化过程的不稳定

状态，达到输出滤波功能。

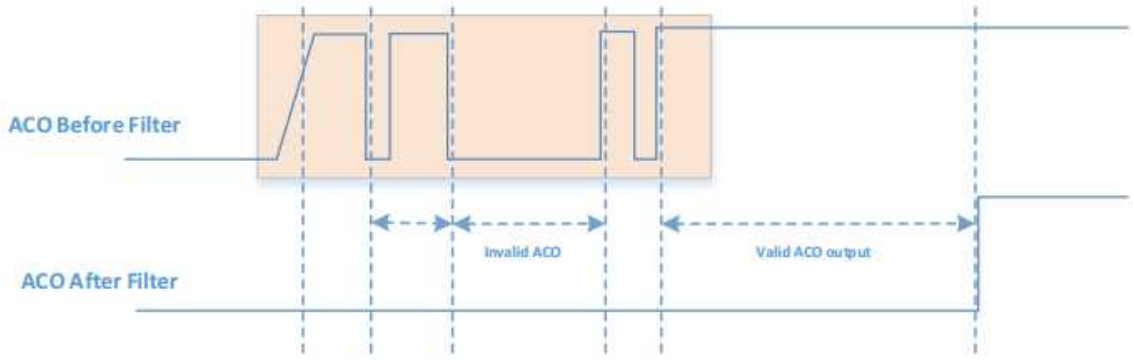
建议用户在使用比较器时，打开迟滞电路，获得一个稳定的比较器输出。

如下图所示，迟滞电路位于比较器模拟输出与数字输出之间。当比较器正端的输入电压 VIN+大于(VIN- + VH+)时，比较器 COUT 输出为高；当 VIN+电压小于(VIN- - VH-)时，比较器输出低。迟滞电路避免了当比较器正端电压接近负端电压时，电路本身带来的抖动。比较器迟滞电压与比较器输出关系图：

Comparator Output Filter

There is a controllable delay circuit in comparator output port, which can be enabled by setting C1HYSE in C1XR register. The delay circuit can eliminate unstable state when comparator state is changing, so that output filter function can be achieved.

It is recommended for users to open delay circuit when using comparator to get a stable comparator output. As shown below, delay circuit is located between comparator analog output and digital output. When input voltage VIN+ in comparator positive pin is higher than (VIN- + VH+), comparator COUT output is high. Otherwise comparator output is low. Delay circuit can avoid vibration caused by circuit itself when comparator positive voltage is close to negative voltage. Relationship chart between comparator delay circuit and comparator output is as below:



比较器输出滤波时序

AC1 的数字滤波通过 C1XR 寄存器的 C0FEN 以及 C1FS 位控制，具体设置方式请参考本章寄存器定义部分。

AC1 digital filter is set by C0FEN and C1FS bit in C1XR register, for detailed setting refer to register definition in this chapter.

比较器输出与 PWM 控制

LGT8FX8P 支持多通道 PWM 输出，PWM 信号可以与比较器模块配合使用。比较器的输出，可用于直接关断 PWM 信号，从而实现比较灵活的 PWM 保护方案。与 PWM 输出相关的控制，请参考定时器章节的相关部分。

Comparator output and PWM control

LGT8FX8P supports multi-channel PWM output, PWM signal can be used together with comparator modular. Comparator output can be used to shut down PWM signal directly to achieve flexible PWM protection solution.

Related control about PWM output, refer to relevant section about Timer.

Register Definition

CISR—AC1 control and status register

C1SR—AC1 control and status register	
Address:0x2F	Default value：0x80

Bit	7	6	5	4	3	2	1	0
Name	C1D	C1BG	C1O	C1I	C1IE	C1IC	C1IS1	C1IS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Comment
7	C1D	Analog comparator disable When setting C1D to logic one, analog comparator is shut down. When setting C1D to logic zero, analog comparator is open.
6	C1BG	Analog comparator 1 positive input source selection bit. AC1 positive input source is set together by C1PS0 in C1BG and C1XR register, {C1BG, C1PS0} = 00= ACXP is positive input 01= AC1P is positive input 10= output of internal DAC is positive input 11= shut down positive input source of AC1
5	C1O	Output status bit of analog comparator After synchronization, output of analog comparator will connect with C1O bit directly. Output value of analog comparator can be gotten by software reading C1O bit value.
4	C1I	Analog comparator interrupt flag When output event of analog comparator triggers interrupt defined by C1IS, C1I is set. When interrupt enable bit C1IE is one and globe interrupt is set, interrupt occurs. When executing analog comparator interrupt program, C1I will be cleared automatically or by writing one.
3	C1IE	Analog comparator interrupt enable bit When setting C1IE to logic one, and globe interrupt is enabled, AC1 interrupt is enabled. When setting C1IE to logic zero, AC1 interrupt is disabled.
2	C1IC	Analog comparator input capture enable bit C1IC=1, Timer/Counter 1 input capture source is from output of analog comparator. C1IC=0, Timer/Counter 1 input capture source is from external pin ICP1.
1	C1IS1	Analog comparator interrupt mode control high
0	C1IS0	Analog comparator interrupt mode control low. C1IS0 and C1IS1 consist of C1IS[1:0] which is used to control interrupt trigger method of analog comparator.

C0IS[1;0]	Interrupt mode
00	Triggered by rising edge or falling edge of AC1
01	Reserved
10	Triggered by falling edge of AC1
11	Triggered by rising edge of AC1

ADCSRB—ADC control and status register B

ADCSRB—ADC control and status register B								
Address:0x7B				Default value：0x00				
Bit	7	6	5	4	3	2	1	0
Name	CEM01	CEM00	CME11	CME10	ACTS	ADTS2	ADTS1	ADTS0

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	Name	Comment						
7	CEM01	AC0 negative input selection, CME0 = {CME01, CME00}						
6	CME00	00: external port ACXN is used as AC0 negative input 01: ADC multi-channel alternate output is used as AC0 negative input 10: Differential amplifier output is used as AC0 negative input 11: Shut down AC0 negative input source						
5	CME11	AC1 negative input selection, CME1 = {CME11, CME10}						
4	CME10	00: external port ACXN is used as AC1 negative input 01: external port AC1N is used as AC1 negative input 10: internal 1/5 bypass voltage of ADC is used as AC1 negative input 11: output of differential operation is used as AC1 negative input						
3	ACHS	AC trigger source channel selection 0- AC0 output is used as ADC auto-swtich trigger source 1- AC1 output is used as ADC auto-swtich trigger source						
2:0	ADTS	Refer to ADC register section						

C1XR—AC1 Auxiliary control register

C1XR—AC1 Auxiliary control register								
Address:0x3A				Default value：0x00				
Bit	7	6	5	4	3	2	1	0
Name	-	C1OE	C1HYSE	C1PS0	C1WKE	C1FEN	C1FS1	C1FS0
R/W	-	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	Name	Comment						
7	-	reserved						
6	C1OE	AC1 comparator output to external port enable control C1OE=1, AC1 comparator is output to external port PE5 C1OE=0, comparator output to external port is disabled.						
5	C1HYSE	AC1 output delay enable control 1= enable output delay 2= disable output delay						
4	C1PS0	AC1 negative input source selection LOW Negative input source is control together by C1PS0 and C1BG, for detail refer to C1SR register definition.						
3	C1WKE	AC1 is used to enable sleep wake up 1= enable wake-up of comparator output 0= shut down wake-up of comparator output						
2	C1FEN	Comparator digital filter enable control 1= enable digital filter 2= disable digital filter						
1:0	C1FS[1:0]	Comparator digital filter width configuration 00= shut down 01= 32us 10= 64us 11= 96us						

数模转换器(DAC)

- 8 位数模转换输出
- DAC 输出可作为模拟比较器参考输入
- 支持 DAC 输出到外部端口(DAO)
- 可选 VCC/AVREF/IVREF 分压电源

Digital-to-Analog Converter (DAC)

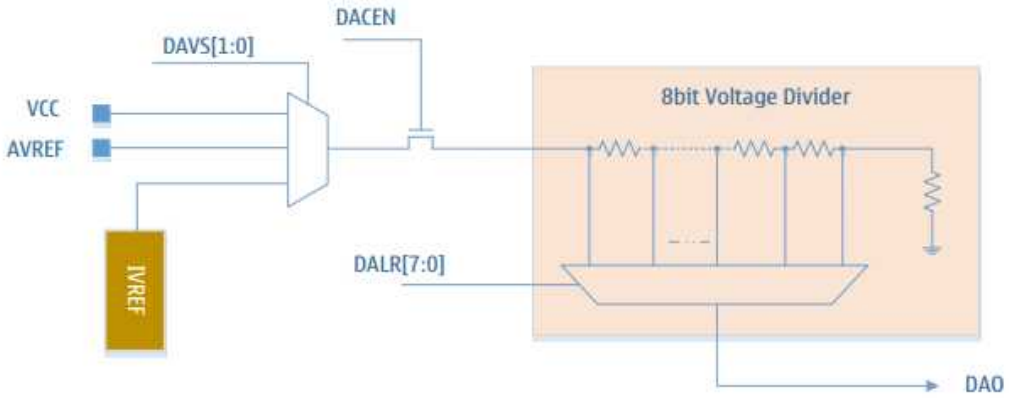
- 8-bit DAC output
- DAC output is used as reference input of analog comparator
- Support DAC output to external port (DAO)
- VCC/AVREF/IVREF voltage division is for option

综述

LGT8FX8P 内部集成一个 8 位可编程数模转换器(DAC)。DAC 的参考电源输入可以选择为来自系统工作电源，内部基准电压源或者来自芯片外部端口 AVREF 输入。 DAC 的输出可选择作为内部比较器 AC0/1 的输入源，也可以直接输出至芯片的外部引脚上作为外部参考使用。当 DAC 输出至外部引脚时，不能直接用于驱动负载，需要通过电压跟随器或其他类似的驱动电路。DAC 内部结构如下图所示：

Overview

LGT8FX8P features an internal 8-bit programmable DAC. Reference power intput of DAC can choose from system working power, internal reference voltage source, or from chip's external port AVREF input. DAC output can be set either as input source of internal comparator AC0/1, or be output to chip's external pin as purpose for external reference. If DAC is output to external pin, it cannot drive loader directly without voltage follower or other similar drive circuit. Internal construction of DAC is shown as below:



Register Definition

DACON—DAC control register

DACON—DAC control register								
Address: 0XA0					0000_0000			
Bit	7	6	5	4	3	2	1	0
	-	-	-	-	DACEN	DAOE	DAVS1	DAVS0
R/W	-	-	-	-	R/W	R/W	R/W	W/R
Bit	Name	Comment						
7:4	-	reserved						
3	DACEN	DAC enable control bit						

		1= enable DAC modular 0= disable DAC modular
2	DAOE	DAC output to external port enable control 1= DAC output to external port PD4 is enabled 0= DAC output to external port is disabled
1	DAVS1	DAC reference voltage source selection bit 1
0	DAVS0	DAC reference voltage source selection bit 0. [DVS1, DVS0] = 00: voltage source is system working power VCC 01: voltage source is external input AVREF 10: voltage source is internal reference voltage 11: shut down DAC reference source and DAC at the same time

DALR—DAC Data Register

DACON—DAC control register								
Address: 0XA1					0000_0000			
Bit	7	6	5	4	3	2	1	0
	DALR[7:0]							
R/W	W/R							
Bit	Name	Comment						
7:0	DALR	DAC data register, setting output voltage in DAC mode Relationship between DAC output voltage and DALR: VDAO = VREF*(DALR + 1)/256 VDAO indicates DAC output analog voltage VREF indicates reference voltagesource, which is set by DAVS of DACON register.						

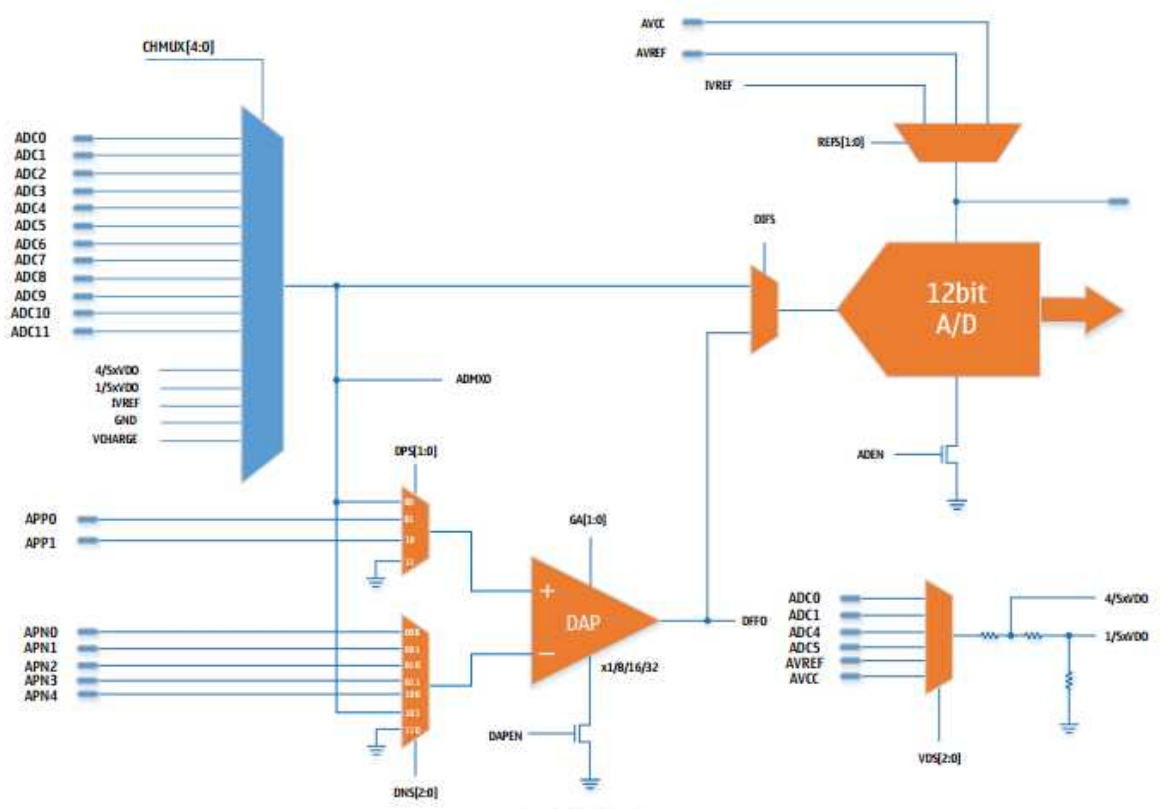
12 位模数转换器(ADC)

- **12 位分辨率，DNL 为±1LSB，INL 为±1.5LSB**
- **最高分辨率时采样率高达 500KSPS**
- **12 路复用的单端输入通道**
- **多路输入可编程增益差分放大器通道**
- **输入电压范围为 0-VCC**
- **内部 1.024V/2.048V/4.096V 参考电压**
- **支持 AVCC 以及外部参考电压输入**
- **内部多输入 1/5、4/5 分压电路**
- **支持正负方向的失调校准**
- **基于中断源的自动开始转换触发模式**
- **支持上/下溢出的自动通道监测**
- **转换结果支持可选对齐模式**
- **转换结束中断请求**

12-bit Analog-to-Digital Converter

- **10-bit Resolution, DNL is ±1LSB, INL is ±1.5LSB**
- **Sampling rate can reach to 5000KSPSat highest resolution**
- **12 Multiplexed Single Ended Input Channels**
- **Multiplexed programmable gain control differential amplifier channel**

- Input voltage range: 0-VCC
- Internal reference voltage 1.024V/2.048V/4.096V
- Support AVCC and external reference voltage input
- Internal 1/5, 4/5 voltage division circuit with multi-input
- Support detuning calibration in positive and negative direction
- Start to switch trigger mode automatically basedon interrupt source
- Support automate channel detection of upper/down overflow
- Optional adjustment mode for switch result
- Swtich end interrupt request



ADC 结构图

概述

模数转换器为一个 12 位的逐次逼近型 ADC。ADC 与一个 17 通道的模拟多路复用器连接，能对来自芯片外部端口 12 路模拟输入以及 5 通道内部电压源进行采样转换。ADC 内部集成一个可编程增益为 x1/x8/x16/x32 的差分运算放大器，放大器输入可来自外部端口或者 ADC 多路复用器的输出。差分运放的结果可作为 ADC 的模拟输入。

ADC 的内部模拟输入源包括来自 ADC 内部的多路输入分压器；内部参考电压源；内部模拟参考地以及来自触摸按键模块的模拟输出。内部多路输入分压器同时输出 4/5，1/5 两路电压；分压器的输入可以选择来自外部端口的电平或者来自系统电源。

ADC 支持失调校准。失调校准的流程由软件控制。失调校准包括正、反两个方向的校准量。失调校准使能后，ADC 控制器将会自动使用正反两个校准值对 ADC 采样结果进行校准。

失调校准的方法请参考本章节相关部分。

Overview

It features a 12-bit successive approximation ADC. The ADC is connected to an 17-channel Analog Multiplexer which samples and converts chip's external port 12-channel analog input and 5-channel interal volage source. ADC is internally integrated a programmable differential operation amplifier with gain control of x1/x8/x16/x32, this amplifer input is from either external port or ADC multiplexer output. Result of differential

operation can be used as ADC analog input. ADC internal analog intput source includes ADC internal multi-channel input voltage divider, internal reference voltage source, internal analog reference location as well as analog output of touch switch modular. Internal multi-channel input voltage divider can output both 4/5 and 1/5 voltage together; Input of voltage divider can choose to be either level of external port or system power.
ADC supports detuning calibration, such calibration procedures are controlled by software, it includes calibration in positive and negative direction. After enabling detuning calibration, ADC controller will adjust sampling result of ADC by automate using positive and negative calibration value.
Detuning calirbration method is referred to relevent part of this chapter.

ADC 的操作

ADC 通过逐次逼近的方法将输入的模拟电压转换成一个 12 位的数字量。最小值代表 GND， 最大值代表基准电压减去 1LSB。基准电压源可以为 ADC 的电源电压 AVCC，外部基准电压 AVREF 或内部 1.024V/2.048V 的参考电压，通过写 ADMUX 寄存器的 REFS 位来选择。

模拟输入通道可以通过写 ADMUX 寄存器的 CHMUX 位来选择。任何 ADC 的输入引脚，外部基 准电压引脚，以及内部参考电压源均可作为 ADC 的单端输入。通过置位 ADTMR 寄存器的 DIFS 可将 ADC 的输入通道切换到内部差分放大器。差分放大器相关输入源以及增益可以通过 DAPCR 寄存器设置。

通过设置 ADCSRA 寄存器的 ADEN 位即可启动 ADC，ADEN 清零时 ADC 并不耗电，因此建议在 进入睡眠模式之前关闭 ADC。

ADC 转换结果为 12 位，存放与 ADC 数据寄存器 ADCH 及 ADCL 中。默认情况下转换结果为右 对齐，但可通过设置 ADMUX 寄存器的 ADLAR 位变为左对齐。

如果设置为转换结果左对齐，且最高只需要 8 位的转换精度，那么只要读取 ADCH 就足够了。 否则要先读取 ADCL，再读取 ADCH，以保证数据寄存器中的内容是同一次转换的结果。一旦 读取 ADCL 后，数据寄存器 ADCL 和 ADCH 被锁存，读取 ADCH 后转换结果即可再更新到数据 寄存器 ADCL 和 ADCH。

ADC 转换结束可以触发中断。即使转换结束发生在读取 ADCL 与 ADCH 之间，中断仍将触发。

ADC Operations

By succesive approximation, ADC tranfer input analog voltage to a 12-bit number. Minnium represents GND while maximum indicates reference voltage minus 1LSB. Reference voltage source, to be set by writing REFS bit in ADMUX register, can be power voltage AVCC of ADC, external reference voltage AVREF or reference voltage of 1.024V/2.048V internally.
Analog input channel can be set by CHMUX in ADMUX register. Single-end input of ADC can be input pin of any ADC, external reference voltage pin as well as internal reference voltage source. By setting DIFS of ADTMR register, input channel of ADC can switch to internal diffential amplifier, whose input source and gain control can be configured by DAPCR register.
ADC will be actived by setting ADEN in ADCSRA register, during clearing ADEN, ADC does not consump power, so it is suggested to shut down ADC before entering to sleep mode.
ADC conversion result is 12-bit, which will be saved in data register ADCH and ADCL. By default, conversion result is right adjustment, however it can change to left adjustment by setting ADLAR in ADMUX register.
If want left adjustment, conversion result only need 8-bit conversioon accurancy the most, in this case reading ADCH is enough, otherwise need to read ADCL first than ADCH in order to ensure that content in data register is the result from the same conversion. Once after reading ADCL, data register ADCL and ADCH will be locked. After reading ADCH, conversion result will update to data register ADCL and ADCH.
ADC conversion end will triggerr interrupt. It implies that conversion end occurs between reading ADCL and ADCH, but interrupt will be trigger still.

启动一次转换

向 ADC 启动转换位 ADSC 位写“1”可以启动单次转换。在转换过程中此位保持为高，直到转换 结束后被硬件清零。如果在转换过程中改变了通道，那么 ADC 会在改变通道前完成这一次 转换。

ADC 转换有不同的触发源。设置 ADCSRA 寄存器的 ADC 自动触发允许位 ADATE 可以使能自动 触发。设置 ADCSRB 寄存器的 ADC 触发选择位 ADTS 可以选择触发源。当所选的触发信号产 生上升沿时，ADC 预分频器复位并开始转换。这提供了一个在固定时间间隔下启动转换的方 法。转换结束后即使触发信号仍然存在，也不会启动一次新的转换。如果在转换过程中触发 信号又产生了一个上升沿，这个上升沿也将被忽略。即使特定的中断被禁止或全局中断使能 位为“0”，其中断标志仍将置位。这样可以在不产生中断的情况下触发一次转换。但是为了 在下次中断事件发生时触发新的转换，必须将中断标志清零。

使用 ADC 中断标志作为触发源，可以在当前进行的转换结束后即开始下一次 ADC 转换。之 后 ADC 便工作于连续转换模式，持续地进行采样并对 ADC 数据寄存器进行更新。第一次转换是通过往 ADCSRA 寄存器的 ADSC 位写“1”来启动。在此模式下，后续的 ADC 转换不依赖于 ADC 中断标志 ADIF 是否置位。

如果使能了自动触发，置位 ADCSRA 寄存器的 ADSC 将启动单次转换。ADSC 标志还可用来检 测转换是否在进行之中。不论转换是如何启动，在转换过程中 ADSC 一直为“1”。

Starting a Conversion

A single conversion is started by writing a logical one to the ADC Start Conversion bit, ADSC. This bit stays high as long as the conversion is in progress and will be cleared by hardware when the conversion is completed. If a different data channel is selected while a conversion is in progress, the ADC will finish the current conversion before performing the channel change.

Alternatively, a conversion can be triggered by various sources. Auto Triggering is enabled by setting the ADC Auto Trigger Enable bit, ADATE in ADCSRA. The trigger source is selected by setting the ADC Trigger Select bits, ADTS in ADCSRB. When a rising edge occurs on the selected trigger signal, the ADC prescaler is reset and a conversion is started.
This provides a method of starting conversions at fixed intervals. If the trigger signal still is set when the conversion completes, a new conversion will not be started. If another positive edge occurs on the trigger signal during conversion, the edge will be ignored. Note that an Interrupt Flag will be set even if the specific interrupt is disabled or the Global Interrupt Enable bit is cleared. A conversion can thus be triggered without causing an interrupt. However, the Interrupt Flag must be cleared in order to trigger a new conversion at the next interrupt event.

Using the ADC Interrupt Flag as a trigger source makes the ADC start a new conversion as soon as the ongoing conversion has finished. The ADC then operates in Free Running mode, constantly sampling and updating the ADC Data Register. The first conversion must be started by writing a logical one to the ADSC bit in ADCSRA. In this mode the ADC will perform successive conversions independently of whether the ADC Interrupt Flag, ADIF is cleared or not.

If Auto Triggering is enabled, single conversions can be started by writing ADSC in ADCSRA . ADSC can also be used to determine if a conversion is in progress. The ADSC bit will be read as one during a conversion, independently of how the conversion was started.

预分频与 ADC 转换时序
在默认条件下，逐次逼近电路需要一个从 300KHz 到 3MHz 的输入时钟以获得最大精度。如 果所需的转换精度低于 12 位，那么输入时钟的频率可以高于 3MHz，以达到更高的采样率。

ADC 模块包括一个预分频器，它可以由系统时钟来产生可接受的 ADC 输入时钟。预分频器通 过 ADCSRA 寄存器的 ADPS 位进行设置。置位 ADCSRA 寄存器的 ADEN 将使能 ADC，预分频器 开始计数。只要 ADEN 位为“1”，预分频器就持续计数，直到 ADEN 被清零。

ADCSRA 寄存器的 ADSC 被置位后，单端转换在下一个 ADC 时钟周期的上升沿开始启动。正 常转换需要 15 个 ADC 时钟周期。ADC 使能（ADCSRA 寄存器的 ADEN 置位）后需要 50 个 ADC 输入时钟周期初始化模拟电路，之后才能有效进行第一次转换。

在 ADC 转换过程中，采样保持在转换启动之后的 1.5 个 ADC 输入时钟开始，而第一次 ADC 转 换的结果输出则发生在启动之后的 14.5 个 ADC 输入时钟。转换结束后，ADC 结果被送入 ADC 数据寄存器，且 ADIF 标志位被置位。ADSC 同时被清零。之后软件可以再次置位 ADSC 标志 或自动触发，从而启动一次新的转换。

Prescaling and Conversion Timing

By default, the successive approximation circuitry requires an input clock frequency between 300kHz and 3MHz to get maximum resolution. If a lower conversion accurancy than 12 bits is needed, the input clock frequency to the ADC can be higher than 3MHz to get a higher sample rate.
The ADC module contains a prescaler, which generates an acceptable ADC input clock from system CPU. The prescaling is set by the ADPS bits in ADCSRA. The prescaler starts counting from the moment the ADC is switched on by setting the ADEN bit in ADCSRA. The prescaler keeps running for as long as the ADEN bit is set to one, and is continuously reset when ADEN is cleared.

When initiating a single ended conversion by setting the ADSC bit in ADCSRA, the conversion starts at the following rising edge of the ADC clock cycle.

A normal conversion takes 15 ADC clock cycles. The first conversion after the ADC is switched on (ADEN in ADCSRA is set) takes 50 ADC input clock cycles in order to initialize the analog circuitry.

In a ADC conversion, the actual sample-and-hold takes place 1.5 ADC clock cycles after the start of a normal conversion and 14.5 ADC clock cycles after the start of an first conversion.

When a conversion is complete, the result is written to the ADC Data Registers, and ADIF is set, meanwhile ADIF is cleared. The software may then set ADSC again or automate trigger, and a new conversion will be initiated.

采样通道与参考电压

ADMUX 寄存器中的 MUX 及 REFS 通过临时寄存器实现了单缓冲。CPU 可对临时寄存器进行随 机访问。在转换启动之前，CPU 可随时对通道及基准源的选择进行配置。为了保证 ADC 有充 足的采样时间，一旦转换开始后，就不允许通道及基准源选择的配置。在转换完成（ADCSRA 寄存器的 ADIF 置位）之后，通道及基准源的选择才会被更新。转换的开始时刻为 ADSC 置位 后的下一个 ADC 输入时钟的上升沿。因此，建议用户在置位 ADSC 之后的一个 ADC 输入时钟 周期内，不要操作 ADMUX 以选择新的通道及基准源。

使用自动触发时，触发事件发生的时间是不确定的。为了控制新设置对转换的影响，在更新 ADMUX 寄存器时要特别小心。若 ADATE 及 ADEN 都置位，则中断时间可以在任意时刻发生， 从而自动触发，启动 ADC 的转换。如果在此期间改变 ADMUX 寄存器的内容，那么用户就无 法辨别下一次转换是基于旧的配置还是新的配置。建议用户在以下安全时刻对 ADMUX 进行 更新：

1） ADATE 或 ADEN 位为“0”；

2） 在转换过程中，但是在触发事件发生后至少一个 ADC 输入时钟周期；

3） 转换结束之后，但是在触发源的中断标志清零之前。如果在上面所提到的任一种情况下更新 ADMUX，那么新配置将在下一次转换前生效。选择 ADC 输入通道时须注意，在启动转换之前先选定通道，在 ADSC 置位后的一个 ADC 时钟 周期之后就可以选择新的模拟输入通道，但最简单的办法是等到转换结束之后再改变通道

ADC 的参考电压源 Vref 反映了 ADC 的转换范围。若单端通道电平超过了 Vref，其转换结果将 接近最大值 0xFFF。Vref 可以是 AVCC，外接 AREF 引脚的电压，内部基准电压源。

使用内部基准(1.024V/2.048V/4.096V)注意事项：

芯片上电后，默认将内部基准校准为 1.024V，用户如果使用 1.024V 的内部基准，可以 直接使用，无需其他操作。但如果需要使用 2.048V 或 4.096V 的内部参考电压，需要自行更 新内部基准的校准值。2.048V/4.096V 的校准值在上电后被加载到寄存器 VCAL2/3(0xCE/0xCC)， 在程序初始化时，将 VCAL2/3 的值读入并写入到 VCAL(0XC8)寄存器即完成校准

Changing Channel or Reference Selection

The MUX and REFS bits in the ADMUX Register are single buffered through a temporary register to which the CPU has random access. Before conversion, CPU can at any time configure the selection of channels and reference voltage.Once the conversion starts, the channel and reference selection is locked to ensure a sufficient sampling time for the ADC.
Selection of channels and reference source only update after conversion completes (ADIF in ADCSRA is set). Note that the conversion starts on the following rising ADC clock edge after ADSC is written. The user is thus advised not to write new channel or reference selection values to ADMUX until one ADC clock cycle after ADSC is written. So it is suggested that users must set ADMUX to select new channels and reference source within a ADC clock after setting ADSC.

If Auto Triggering is used, the exact time of the triggering event can be indeterministic. Special care must be taken when updating the ADMUX Register, in order to control which conversion will be affected by the new settings.
If both ADATE and ADEN are set, an interrupt event can occur at any time which would cause autor triggering , than ADC conversion will start. If the ADMUX Register is changed in this period, the user cannot tell if the next conversion is based on the old or the new settings. ADMUX can be safely updated in the following ways:
a. When ADATE or ADEN is cleared.
b. During conversion, minimum one ADC clock cycle after the trigger event.
c. After a conversion, before the Interrupt Flag used as trigger source is cleared.
When updating ADMUX in one of these conditions, the new settings will become effect before next ADC conversion.

When changing channel selections, must ensure that the correct channel is selected before starting the conversion. New analog input channel can be choosen afte an ADC clock cyle when ADSC is set. Note that the most simple way is to change channels after conversion is complete.

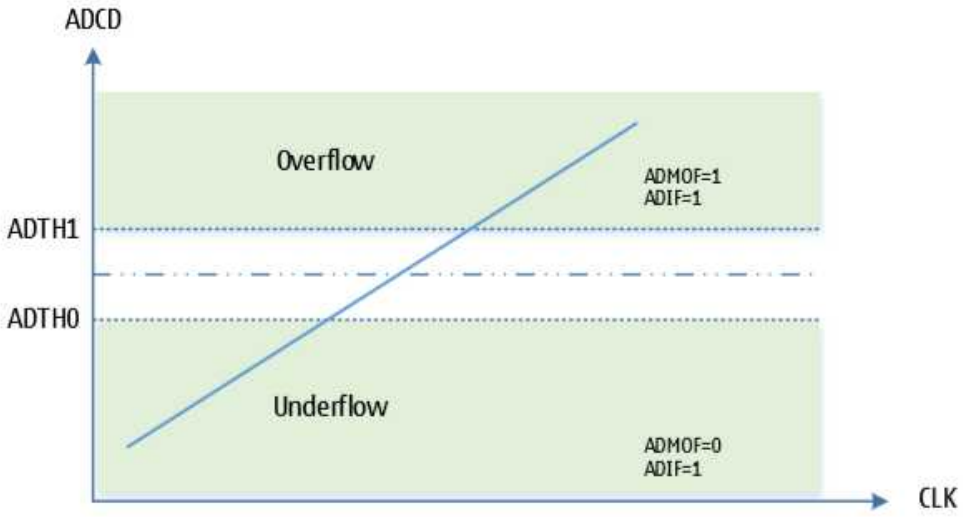
The reference voltage for the ADC (VREF) indicates the conversion range for the ADC. Single ended channels that exceed VREF will result in codes close to 0xFFF. VREF can be selected as either AVCC, internal reference voltage , or external AREF pin.
When using internal reference ((1.024V/2.048V/4.096V), attention should be paid to below:
When chip is on power, internal reference is adjusted to 1.024V by default, it can be used directly without addtional operation if user want to use internal reference of 1.024V. However if 2.048V or 4.096V is wanted for internal reference voltage, users must update internal refernce themself. Calibration value of 2.048V/4.096V will be loaded to register VCAL2/3(0xCE/0xCC) when power on. When programm is initialized, value of VCAL2/3 will be read and written to VCAL(0XC8) register, than a calibration is complete.

自动通道监测

自动通道监测模式用于实时监测选定 ADC 输入通道的电压变化。软件通过置位 ADCSRC 寄存器的 AMEN 位使能自动通道监测功能，ADC 自动转换选定通道的电压，当转换结果在给 定的溢出范围之外，将会置位 ADC 中断标志位(ADIF)，并同时停止自动监测。软件可以通过 中断或查询的方式响应溢出事件。ADMSC 寄存器的 AMOF 位用于指示溢出事件的类型。ADIF 标志位在响应中断复位后自动由硬件清零；在查询模式下， 可由软件写 1 清零。只有当 ADIF 清零，并通过置位 ADCSRC 寄存器的 AMEN 位，才可重新使能自动监测模式。

Auto Channel Monitor

Auto channel monitor mode, which is enabled by setting AMEN bit in ADCSRC register, is used to real-time monitor the voltage change of selected ADC input channel. ADC converts selected channel voltage automate, when conversion result is over a given overflow range, ADC interrupt flag ADIF will be set and Auto Monitor stop as well at the same time, software can response to overflow event by interrupt or enquiry, AMOF bit in ADMSC register can indicate type of overflow event. ADIF is cleared automate after acknowledgment interrupt is reset and can be cleared also by software writing one. Auto monitoring mode can only re-start when ADIF is cleared and AMEN bit in ADCSRC register is set.



为克服单次 ADC 转换结果的不稳定，自动检测支持一个可配置的数字滤波功能。数字滤 波通过对连续转换结果进行检测，只有在限定的连续转换次数内都得到一个一致的结果，才 触发溢出事件。连续转换次数可以通过 ADMSC 寄存器的 AMFC[3:0]位设置。

自动通道监测功能通过 ADCSRC 寄存器的 AMEN 位控制。寄存器 ADT0 用于设置下溢出 的阀值；ADT1 用于设置上溢出的阀值。ADT0/1 为 16 位寄存器。软件置位 AMEN 位后，将会 立刻停止 ADC 当前的转换动作，并复位 ADC 控制状态，之后进入自动转换模式。

在启动自动通道检测模式前，需要设置好检测的通道以及其他相关配置。软件可以随时通过清零 AMEN 寄存器，禁止自动检测模式。

To avoid unstability from a single ADC conversion result, auto-monitoring supports a configurable digital filter, which detects continurous conversion results. Overflow events will be triggered only when a consistent result occurs in a limited continurous conversion times. Times of continurous conversion can be set by AMFC[3:0] in ADMSC register.

Auto-monitoring is set by AMEN in ADCSRC register. Register ADT0 is used to set threhold of underflow, while ADT1 to set threhold of overflow. ADT0/1 is a 16-bit register. Once software setting AMEN bit, it will stop ADC's current conversion immeditely, reset ADC control status and then enter into auto conversion mode. Before start auto channel detection mode, channels and related configuration should be well set. Software can disable auto detection mode by clearing AMEN register at any time.

ADC 失调校准

由于制造工艺的偏差以及电路结构的固有特性， 会造成 ADC 内部比较器电路产生不同 程度失调误差。因此对失调电压进行补偿，对于产生高精度的 ADC 转换结构非常关键。 LGT8FX8P 芯片内部的 ADC 支持失调电压测试相关接口，可以在软件的配合下完成失调的测 量和校准。

失调校准的原理：

失调校准主要是通过改变内部比较器的输入极性，在正、反两个方向测试 ADC 转换结 果。由于正反两个方向失调电压也是表现为两种极性，通过这两次转换结果相减，可以得到 一个中间的失调误差值。正常应用时，将转换结果根据这个失调电压进行相应的调整即可

失调校准流程：

1. 配置 VDS 模块，将 VDS 输入源选择为模拟电源(AVCC)

2. ADC 的参考电压选择为模拟电源(AVCC)

3. ADCSRC[SPN] = 0, ADC 读取 4/5VDO 通道，转换值记录为 PVAL

4. ADCSRC[SPN] = 1, ADC 读取 4/5VDO 通道，转换值记录位 NVAL

5. 将值(NVAL – PVAL) >>1 存储到 OFR0 寄存器

6. ADCSRC[SPN] = 1, ADC 读取 1/5VDO 通道，转换结果记录为 NVAL

7. ADCSRC[SPN] = 0, ADC 读取 1/5VDO 通道，转换结果记录位 PVAL

8. 将值(NVAL – PVAL) >> 1 存储到 OFR1 寄存器

9. 设置 ADCSRC[OFEN]=1 使能失调补偿功能

特别注意：由于失调误差有正负方向， 以上数据以及运算都为有符号操作。

失调校准过程中需要改变 ADC 相关配置，因此建议失调校准在正常使用的配置之前完 成。为了提高校准精度，建议 ADC 读取通道转换时采样多次滤波。

失调校准 OFR0/1 配置完成后，通过 OFEN 位使能自动失调补偿。以后的正常转换后， ADC 控制将根据 ADC 转化结果，自动使用 OFR0/1 进行补偿

ADC Detuning Calibration

Due to production tolerance and their specific features of electronic construction, ADC internal comparator circuit will bring out detuning tolerance in some degree. So it is very critical to compensate detuning voltage in orde to acheive high accurancy of ADC conversion. Internal ADC of LGT8FX8P has ports supporting detuning voltage detection.Detection and calibration of detuning can be complete by the help of software.

Detuning calirbration principle:

Detuning carlibaration mainly changes input polarity of internal comparator, detects ADC conversion result in positive and negative directions. Because detuning voltage in positive and negative direction have also two polarity, an intermediate detuning tolerance can be calculated by minus between these two conversion results. In normal application, based on detuning voltage conversion result is adjusted accordingly.

Detuing calibration flow:

1. Setting VDS modular, VDS input source is set as analog powr (AVCC)
2. Referance voltage of ADC is set as analog power (AVCC)
3. ADCSRC[SPN] = 0, ADC read 4/5VDO channel, conversion result is record as PVAL
4. ADCSRC[SPN] = 1, ADC read 4/5VDO channel, conversion result is record as NVAL
5. Save (NVAL – PVAL) >>1 to OFRO register
6. ADCSRC[SPN] = 1, ADC read 1/5VDO channel, conversion result is record as NVAL
7. ADCSRC[SPN] =0, ADC read 1/5VDO channel, conversion result is record as PVAL
8. Save (NVAL – PVAL) >>1 to OFR1 register
9. Setting ADCSRC[OFEN]=1 to enable detuning compensation

Special care must be taken to : Due to positve and negative dierection of detuning tolerance, all datas and operations mentioned above should be with symbols.

During detuning calibration, relevant configuration of ADC must be changed, so it is suggested to complete detuning calibration before normally used configuration. To increase calibration accurancy, sampling is suggested with serveral filter when ADC reads channel conversion.

After detuning calibration OFR0/1 configuration, auto detuning compensation will be enabled by OFEN bit. Than after normal conversion, ADC control will use OFR0/1 for compensation based on ADC conversion results.

上面介绍的失调校准方法，基于在一个测试环境和测试输入下的失调。当系统环境改变 后，ADC 的失调也会随之变化。因此如果能够实现实时的校准补偿，对于克服器件随工作环 境变化而导致的性能差异，提高 ADC 测量精度， 非常重要。

这里提供一种建议使用的算法，基于失调校准算法的原理，可以实现动态补偿工作环境 带来的失调误差，获得一致准确的测试结果。 这种方法无需计算失调电压，也不用使能失调补偿(OFEN)。算法只需要通过 SPN 控制 ADC 转换的极性，在不同 SPN 下采样两个测量结果，两个结果中由于失调引入的误差表现为 正负两种方向，因此我们可以简单的通过相加求平均的方法抵消失调产生的误差。

我们假设当在 ADC 转换时，失调引入的测试误差为 VOFS，因此控制 SPN 进行连续两次 ADC 转换，所得到的 ADC 转换结果可以表示为：

SPN = 1 时， VADC1 = VREL + VOFS1

SPN = 0 时， VADC0 = VREL – VOFS0

我们将两次测量结果相加，即可消除掉 VOFS 对实际采样输入 VREL 产生的影响。由于电路 的匹配特性， VOFS1 和 VOFS0 可能不会完全相同，但总体上仍然可以实现补偿失调误差的效果。

动态失调补偿算法流程：

根据应用需要初始化 ADC 转换参数

设置 SPN=1，启动 ADC 采样，记录 ADC 采样结果为 VADC1

设置 SPN=0，启动 ADC 采样，记录 ADC 采样结果为 VADC2 4. (VADC1 + VADC2) >> 1 即为本次 ADC 的转换结果

实际应用中，可以将这种算法与取样平均算法结合，可以得到更加理想的效果。

ADN Dynamic Calibration

The above mentioned carliabration method is for detuning based on a single test environment and test input. However ADC detuning will change accordingly when system environment is changing. So real-time calibaration compensation is very important with regard to avoid performance tolerance of devices when it is in a changing environment and to increase ADC measurement accurancy.

In this case a algorithm, based on principle of detuning calibration algorithm, is recommended. This algorithm is to control ADC conversion polarity by SPN, in different SPN to sample two measurement results whose tolerance has both positive and negative direction because of detuning, so detuning tolerance can be offset easily by average value from plus operation.

We assume that during ADC conversion measurement tolerance from detuning is indicated by VOFS. Than ADC conversion result can be indicated as below by setting SPC to make two continurous ADC conversion:

When SPN = 1， VADC1 = VREL + VOFS1

WhenSPN = 0， VADC0 = VREL -VOFS0

By plusing these two measurement results, V_{OFS}'s affection to actual sampling input V_{REL} will be eliminated. Due to match feature of circuit, V_{OFS1} and V_{REL0} will not be the same but in general detuning tolerance compensation is acheived.

Dynamic detuning compensation algorithm procedures:

1. As per application demand to initialize ADC conversion parameter

2. Setting SPN=1, start ADC sampling, record ADC sampling result as VADC1

3. Setting SPN=0, start ADC sampling, record ADC sampling result as VADC2

4. If (VADC1 + VADC2) >> 1, then it is this ADC conversion result

In actual application, to acheive much better result by combination of above algorithm and sampling average method.

Register Definition

ADC Register List

Register	Address	Default value	Comments
ADCL	0x78	0x00	ADC data low byte register
ADCH	0x79	0x00	ADC data high byte register
ADCSRA	0x7A	0x00	ADC control and status register A
ADCSRB	0x7B	0x00	ADC control and status register B
ADMUX	0x7C	0x00	ADC multi-channel selection control register
ADCSRC	0x7D	0x01	ADC control and status register C
DIDR0	0x7E	0x00	Digital input disable control register 0
DIDR1	0x7F	0x00	Digital input disable control register 1
DAPCR	0xDC	0x00	Differential amplifier control register
OFR0	0xA3	0x00	Detuning compensation register 0
OFR1	0xA4	0x00	Detuning compensation register 1
ADTOL	0xA5	0x00	In auto monitoring underflow threhold low 8-bit
ADTOH	0xA6	0x00	In auto monitoring underflow threhold high 8-bit
ADT1L	0xAA	0x00	In auto monitoring overflow threhold low 8-bit
ADT1H	0xAB	0x00	In auto monitoring overflow threhold high 8-bit
ADMSC	0xAC	0x01	Auto monitoring status and control register
ADCSRD	0xAD	0x00	ADC control and status register D

ADCL--ADC Data Low Byte Register

ADCL--ADC Data Low Byte Register								
Address: 0x78				Default value:0x00				
Bit	7	6	5	4	3	2	1	0
Name0	ADC7	ADC6	ADC5	ADC4	ADC3	ADC2	ADC1	ADC0
Name1	ADC3	ADC2	ADC1	ADC0	-	-	-	-
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial	0	0	0	0	0	0	0	0
Bit	Name	Comment						
7:0	ADC[7:0]/ ADC[3:0]	ADCL--ADC Data Low Byte Register When ADLAR is zero, ADC output data is saved in registe by adjustment of low byte, i.e. as indicated by Name0, ADCL is ADC[7:0]; When ADLAR is one, ADC output data is saved in registe by adjustment of high byte, i.e. ADCL high 4-bit is ADC[3:0], low 4-bit makes no sense as indicated by Name1.						

ADCL--ADC Data High Byte Register

ADCL--ADC Data High Byte Register								
Address: 0x79				Default value:0x00				
Bit	7	6	5	4	3	2	1	0
Name0	-	-	-	-	ADC11	ADC10	ADC9	ADC8
Name1	ADC11	ADC10	ADC9	ADC8	ADC7	ADC6	ADC5	ADC4
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial	0	0	0	0	0	0	0	0
Bit	Name	Comment						
7:0	ADC[11:8]/	ADC Data Low Byte Register						

	ADC[11:4]	When ADLAR is zero, ADC output data is saved in registe by adjustment of low byte, i.e. as indicated by Name0, ADCH low 4-bit is ADC[11:8] and high 4-bit is meaningless; When ADLAR is one, ADC output data is saved in registe by adjustment of high byte, i.e. ADCH is ADC[11:4] as indicated as Name1.

ADCSRA--ADC control and status register A

ADCL--ADC Data High Byte Register								
Address: 0x7A				Default value:0x05				
Bit	7	6	5	4	3	2	1	0
Name0	ADEN	ADSC	ADATE	ADIF	ADIE	ADPS2	ADPS1	ADPS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial	0	0	0	0	0	0	0	0

Bit	Name	Comment
7	ADEN	ADC enable control bit When setting ADEN to one, ADC is enabled When setting ADEN to zero, ADC is disabled
6	ADSC	ADC start conversion In single conversion mode, ADSC enable will start a conversion. In continurous conversion mode ADSC enable will start the first conversion.
5	ADATE	ADC auto triggering enable control bit When setting ADATE to one, auto triggering is enabled. Rising edge of selected trigger signal starts one conversion. Trigger source is control by ADTS in ADCSRB register. When setting ADATE to zero, auto triggering is disabled.
4	ADIF	ADC interrupt flag bit ADIF is set when ADC completes one conversion and updates data register. When ADC interrupt enable bit ADIE is one and globe interrupt is set, ADC interrupt is generated. ADIF is cleared by executing interrupt or by writing one to this bit
3	ADIE	ADC interrupt control bit When setting ADIE to one and globe interrupt is set, ADC interrupt is enabled. When setting ADIE to zero , ADC interrupt is disabled.
2:0	ADPS[2:0]	ADC prescaler selection control bit ADPS selects prescale factor generated by system clock

ADPS[2:0]	Prescale factor
0	2
1	2
2	4
3	8
4	16
5	32(default)
6	64
7	128

ADCSRB--ADC control and status register B

ADCSRB--ADC control and status register B								
Address: 0x7B				Default value:0x00				
Bit	7	6	5	4	3	2	1	0
Name0	ACME01	ACME00	ACME11	ACME10	ACTS	ADTS2	ADTS1	ADTS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial	0	0	0	0	0	0	0	0

Bit	Name	Comment
7	ACME01	Comparator 0 negative input selection 00: negative select external input ACIN0 01: negative select ADC multiplexer output 1x: nagative select operation amplifer 0 output
6	ACME00	
5	ACME11	Comparator 1 negative input selection 00: negative select external input ACIN2 01: negative select ADC multiplexer output 1x: nagative select operation amplifer 1 output
4	ACME10	
3	ACTS	AC trigger source channel selection 0- AC0 output used as ADC auto conversion trigger source 1- AC1 output used as ADC auto conversion trigger source
2:0	ADTS[2:0]	ADC auto trigger source selection control bit When setting ADATE is one, auto triggering is enabled, trigger source is control by ADTS. When setting ADATE is zero, auto ADTS setting is useless. Rising edge of selected trigger singal interrupt flag will start one conversion. When trigger source is switched from interrupt clear to interupt enable, trigger singal will generate a rising edge, if ADEN is set in this case, ADC will also start a conversion. When in continurous conversion mode (ADTS=0), Auto-triggering is disabled.

ADTS[2:0]	Trigger source
0	Continurous conversion mode
1	Comparator 0/1
2	External interrupt 0
3	Timer/Counter 0 compare match
4	Timer/Counter 0 overflow
5	Timer/Counter 1 compare match B
6	Timer/Counter 1 overflow
7	Timer/Counter 1 input capture event

ADMUX--ADC mutl-channel selection control register

ADMUX--ADC mutl-channel selection control register								
Address: 0x7C				Default value:0x00				
Bit	7	6	5	4	3	2	1	0
Name0	REFS1	REFS0	ADLAR	CHMUX4	CHMUX3	CHMUX2	CHMUX1	CHMUX0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial	0	0	0	0	0	0	0	0

Bit	Name	Comment		
7:6	REFS[1:0]	In combination with REFS2 in ADCSRD register to select ADC reference voltage source. By setting REFS control bit to select reference voltage, if changing REFS setting during conversion, such change only effect after conversion is complete.		
		REFS2, REFS[1:0]	Reference voltage selection	
		0_00	AREF	
		0_01	AVCC	
		0_10	On-chip 2.048V reference voltage source	
		0_11	On-chip 1.024V reference voltage source	
		1_00	On-chip 4.096V reference voltage source	
5	ADLAR	Conversion result left adjustment enable control bit When setting ADLAR to one, conversion result in ADC data register is left adjustment When setting ADLAR to zero, conversion result in ADC data register is right adjustment		
4:0	CHMUX[4:0]	ADC input source selection control bit		
		CHMUX[4:0]	Singe-end input source	Comment
		0_0000	PC0	external input port
		0_0001	PC1	
		0_0010	PC2	
		0_0011	PC3	
		0_0100	PC4	
		0_0101	PC5	
		0_0110	PE1	
		0_0111	PE3	
		0_1001	PC7	
		0_1010	PF0	
		0_1011	PE6	
		0_1100	PE7	
		0_1110	4/5VDO	Internal voltage division circuit
		0_1000	1/5VDO	
		0_1101	IVREF	Internal reference
		0_1111	AGND	Analog ground
		0_xxxx	DACO	Internal DAC output

ADCSRC--ADC control status register C

ADCSRC--ADC control status register C								
Address: 0x7D				Default value:0x00				
Bit	7	6	5	4	3	2	1	0
Name0	OFEN	-	SPN	AMEN	-	SPD	DIFS	ADTM
R/W	R/W	-	R/W	R/W	-	R/W	R/W	R/W
Bit	Name	Comment						
7	OFEN	1=Enable detuning compensate 0= Disable detuning compensate						

6	-	Unimplemented
5	SPN	ADC conversion input polarity control, only used in detuning calibration. In normal mode it must be cleared.
4	AMEN	Channel auto monitoring enable: 1: enable channel auto monitoring 0: disable channel auto monitoring
3	-	Unimplemented
2	SPD	0= ADC low speed conversion mode 1= ADC high speed conversion mode, only used for analog input with low resistance
1	DIFS	0= ADC conversion from ADC multiplexer 1= ADC conversion from internal differential amplifier
0	ADTM	Testing mode, internal reference voltage is output from AVREF port

DIDR0--Digital input disable control register 0

DIDR0--Digital input disable control register 0								
Address: 0x7E				Default value:0x00				
Bit	7	6	5	4	3	2	1	0
Name0	PE3D	PE1D	PC5D	PC4D	PC3D	PC2D	PC1D	PC0D
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	Name	Comment						
7	PE3D	1= shut down PE3 digital input function						
6	PE1D	1= shut down PE1 digital input function						
5	PC5D	1= shut down PC5 digital input function						
4	PC4D	1= shut down PC4 digital input function						
3	PC3D	1= shut down PC3 digital input function						
2	PC2D	1= shut down PC2 digital input function						
1	PC1D	1= shut down PC1 digital input function						
0	PC0D	1= shut down PC0 digital input function						

DIDR1--Digital input disable control register 1

DIDR0--Digital input disable control register 1								
Address: 0x7F				Default value:0x00				
Bit	7	6	5	4	3	2	1	0
Name0	PE7D	PE6D	PE0D	C0PD	PF0D	PC7D	PD7D	PD6D
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	Name	Comment						
0	PD6D	1= shut down PD6 digital input function						
1	PD7D	1= shut down PD7 digital input function						
2	PC7D	1= shut down PC7 digital input function						
3	PF0D	1= shut down PF0 digital input function						
4	C0PD	1= shut down AC0P digital input function (LQFP48)						
5	PE0D	1= shut down PE0 digital input function						
6	PE6D	1= shut down PE6 digital input function						
7	PE7D	1= shut down PE7 digital input function						

ADCSRD--ADC control register D

ADCSRD--ADC control register D								
Address: 0xAD				Default value:0x00				
Bit	7	6	5	4	3	2	1	0
Name0	BGEN	REFS2	IVSEL1	IVSEL0	-	VDS2	VDS1	VSD0
R/W	R/W	R/W	R/W	R/W	-	R/W	R/W	R/W
Bit	Name	Comment						
6	BGEN	Internal reference globe control enable, 1=enable						
5	REFS2	In combination with REFS in ADMUX register, used to select reference voltage of ADC conversion, referring to definition of REFS in ADMUX.						
5:4	IVSEL	When ADC reference voltage is set as VCC or AVREF, IVSEL is used to control internal reference of output voltage: 00 = 1.024V 01 = 2.048V 1x = 4.096V						
3	-	reserved						
2:0	VDS[2:0]	Voltage division circuit input source selection 000/111 =shut down voltage division circuit modular 001 = ADC0 010 = ADC1 011 = ADC4 100 = ADC5 101 = external reference input (AVREF) 110 = system power						

DAPCR--Differential operation amplifier control register

DAPCR--Differential operation amplifier control register								
Address: 0xDC				Default value:0x00				
Bit	7	6	5	4	3	2	1	0
Name0	DAPEN	GA1	GA0	DNS2	DNS1	DNS0	DPS1	DPS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	Name	Comment						
7	DAPEN	1= enable differential amplifier 1= disable differential amplifier						
6:5	GA[1:0]	Differential amplifier gain control 00 = x1 01 = x8 10 = x16 11 = x32						
4:2	DNS[2:0]	Differential amplifier reverse input source selection bit 000 = ADC2/APN0 001 = ADC3/APN1 010 = ADC8/APN2						

		011 = ADC9/APN3 100 = PE0/APN4 101 = ADC multiplexer 110 = AGND 111 = disable differential amplifier reverse input
1:0	DPS[1:0]	Differential amplifier forward input source selection bit 00 = ADC multiplexer 01 = ADC0/APP0 10 = ADC1/APP1 11 = AGND

OFR0--Detuning compensate register 0

OFR0--Detuning compensate register 0								
Address: 0xA3				Default value:0x00				
Bit	7	6	5	4	3	2	1	0
Name0	OFR0[7:0]							
R/W	W/R							
Bit	Name	Comment						
7:0	OFR0	Detuning compensate register 0 OFR0 is value with symbol, saved in format of two's-compliment						

OFR0--Detuning compensate register 1

OFR0--Detuning compensate register 1								
Address: 0xA4				Default value:0x00				
Bit	7	6	5	4	3	2	1	0
Name0	OFR1[7:0]							
R/W	W/R							
Bit	Name	Comment						
7:0	OFR1	Detuning compensate register 1 OFR1 is value with symbol, saved in format of two's-compliment						

ADMSC--ADC channel monitoring status control register

ADMSC--ADC channel monitoring status control register								
Address: 0xAC				Default value:0x01				
Bit	7	6	5	4	3	2	1	0
Name	AMOF	-	-	-	AMFC3	AMFC2	AMFC1	AMFC0
R/W	-	-	-	-	R/W	R/W	R/W	R/W
Bit	Name	Comment						
7	AMOF	Auto-monitoring overflow event type flag bit 1=overflow 0=underflow						
6:4	-	Unimplemented						
3:0	AMFC	Auto-monitoring digital filter control bit 0000 = diable configuration 0001 = one conversion, no filter 0010 = two in consistent						

		0011 = three in consistent ..... 1110 = 14 in consistent 1111 = 15 in consistent

ADT0L--Auto-monitoring underflow threhold low 8-bit

ADT0L--Auto-monitoring underflow threhold low 8-bit								
Address: 0xA5				Default value:0x00				
Bit	7	6	5	4	3	2	1	0
Name	ADT0L[7:0]							
R/W	W/R							
Bit	Name	Comment						
7:0	ADT0L	Auto-monitoring underflow threhold register low 8-bit						

ADT0h--Auto-monitoring underflow threhold high 8-bit

ADT0L--Auto-monitoring underflow threhold high 8-bit								
Address: 0xA6				Default value:0x00				
Bit	7	6	5	4	3	2	1	0
Name	ADT0h[7:0]							
R/W	W/R							
Bit	Name	Comment						
7:0	ADT0H	Auto-monitoring underflow threhold register high 8-bit						

ADT1L--Auto-monitoring overflow threhold low 8-bit

Address: 0xAA				Default value:0x00				
Bit	7	6	5	4	3	2	1	0
Name	ADT1L[7:0]							
R/W	W/R							
Bit	Name	Comment						
7:0	ADT1L	Auto-monitoring overflow threhold register low 8-bit						

ADT1H--Auto-monitoring overflow threhold high 8-bit

ADT1H--Auto-monitoring overflow threhold high 8-bit								
Address: 0xAB				Default value:0x00				
Bit	7	6	5	4	3	2	1	0
Name	ADT1H[7:0]							
R/W	W/R							
Bit	Name	Comment						
7:0	ADT1H	Auto-monitoring overflow threhold register high 8-bit						

VCAL-- Internal reference calibration register

VCAL-- Internal reference calibration register								
Address: 0xC8				Default value:0x00				
Bit	7	6	5	4	3	2	1	0

Name	VCAL[7:0]	
R/W	W/R	
Bit	Name	Comment
7:0	VCAL	Internal reference calibration register. After power on, load calibration value of 1.024V by default. Writing calibration value of other reference voltage to this register can acheive calibration of relevant reference. I.e. reference is set as 2.048V, writing VCAL2 to this register, calibration of internal reference to 2.048V is complete.

VCAL1-- 1.024V reference calibration register

VCAL1-- 1.024V reference calibration register								
Address: 0xCD				Default value:0x00				
Bit	7	6	5	4	3	2	1	0
Name	VCAL1[7:0]							
R/W	R/0							
Bit	Name	Comment						
7:0	VCAL1	Calibration coefficient of 1.024V internal reference						

VCAL2-- 2.048V reference calibration register

VCAL2-- 2.048V reference calibration register								
Address: 0xCE				Default value:0x00				
Bit	7	6	5	4	3	2	1	0
Name	VCAL2[7:0]							
R/W	R/0							
Bit	Name	Comment						
7:0	VCAL2	Calibration coefficient of 2.048V internal reference						

VCAL3-- 4.096V reference calibration register

VCAL3-- 4.096V reference calibration register								
Address: 0xCC				Default value:0x00				
Bit	7	6	5	4	3	2	1	0
Name	VCAL3[7:0]							
R/W	R/0							
Bit	Name	Comment						
7:0	VCAL3	Calibration coefficient of 4.096V internal reference						

Page 247---page 258, refer to data sheet of Chinese version.

寄存器速查表

Addr	Name	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Extended IO Register									
$F6	GUID3	GUID Byte 3							
$F5	GUID2	GUID Byte 2							
$F4	GUID1	GUID Byte 1							
$F3	GUID0	GUID Byte 0							
$F2	PMCR	PMCE	CLKFS	CLKSS	WCLKS	OSCKEN	OSCMEN	RCKEN	RCMEN
$F0	PMX2	WCE	STOSC1	STOSC0	-	-	XIEN	E6EN	C6EN
$EE	PMX0	PMXCE	C1BF4	C1AF5	C0BF3	C0AC0	SSB1	TXD6	RXD5
$ED	PMX1	-	-	-	-	-	C3AC	C2BF7	C2AF6
$EC	TCKSR	-	F2XEN	TC2XF1	TC2XF0	-	AFCKS	TC2XS1	TC2XS0
$E2	PSSR	PSS1	PSS3	-	-	-	-	PSR3	PSR1
$E1	OCPUE	PUE7	PUE6	PUE5	PUE4	PUE3	PUE2	PUE1	PUE0
$E0	HDR	-	-	HDR5	HDR4	HDR3	HDR2	HDR1	HDR0
$DE	DAPTE	DAPTE	-	-	-	-	-	-	-
$DD	DAPTR	DAPTP	DAP Trimming						
$DC	DAPCR	DAPEN	GA1	GA0	DNS2	DNS1	DNS0	DPS1	DPS0
$D8									
$D7									
$D6									
$D5									
$D4									
$D2									
$D1									
$D0									
$CF	LDOCR	WCE				PDEN	VSEL2	VSEL1	VSEL0
$CE	VCAL2	Calibration value for 2.048V internal reference							
$CD	VCAL1	Calibration value for 1.024V internal reference							
$CC	VCAL3	Calibration value for 4.096V internal reference							
$C8	VCAL	Internal Voltage Reference calibration register							
$C6	UDR	USART Data Register							
$C5	UBRRH	-	-	-	-	USART Baud Rate Register High			
$C4	UBRRL	USART Baud Rate Register Low							
$C2	UCSRC	UMSEL1	UMSEL0	UPM1	UPM0	USBS0	UCSZ01	UCSZ00	UCPOL0
$C1	UCSRB	RXCIE0	TXCIE0	UDRIE0	RXEN0	TXEN0	UCSZ02	RXB80	TXB80
$C0	UCSRA	RXC0	TXC0	UDRE0	FE0	DOR0	UPE0	U2X0	MPCM0
$BD	TWAMR	TWI Address Mask							-
$BC	TWCR	TWINT	TWEA	TWSTA	TWSTO	TWWC	TWEN	-	TWIE
$BB	TWDR	TWI Data							
$BA	TWAR	TWI Address							TWGCE

Addr	Name	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
$B9	TWSR	TWI Status bits					-	TWPS	
$B8	TWBR	TWI Bit Rate register							
$B6	ASSR	INTCK	-	AS2	TCN2UB	OCR2AUB	OCR2BUB	TCR2AUB	TCR2BUB
$B4	OCR2B	Timer 2 Output Compare Register B							
$B3	OCR2A	Timer 2 Output Compare Register A							
$B2	TCNT2	Timer 2 Counter Register							
$B1	TCCR2B	FOC2A	FOC2B	-	-	WGM22	CS2		
$B0	TCCR2A	COM2A1	COM2A0	COM2B1	COM2B0	-	-	WGM21	WGM20
$AF	DPS2R	-	-	-	-	DPS2E	LPRCE	TOS1	TOS0
$AE	IOCWK	IOCD7	IOCD6	IOCD5	IOCD4	IOCD3	IOCD2	IOCD1	IOCD0
$AD	ADCSRD	BGEN	REFS2	IVSEL1	IVSEL0	-	VDS2	VDS1	VDS0
$AC	ADMSC	AMOF	-	-	-	AMFC3	AMFC2	AMFC1	AMFC0
$AB	ADT1H	ADC Auto-monitor Overflow threshold high byte							
$AA	ADT1L	ADC Auto-monitor Overflow threshold low byte							
$A9	PORTE	Port Output E (for compatible with LGT8FX8D)							
$A8	DDRE	Data Direction E (for compatible with LGT8FX8D)							
$A7	PINE	Port Input E (for compatible with LGT8FX8D)							
$A6	ADT0H	ADC Auto-monitor Underflow threshold high byte							
$A5	ADT0L	ADC Auto-monitor Underflow threshold low byte							
$A4	OFR1	ADC positive offset trimming							
$A3	OFR0	ADC negative offset trimming							
$A1	DALR	DAC data register							
$A0	DACON	-	-	-	-	DACEN	DAOE	DAVS1	DAVS0
$9F	OCR3CH	Compare output register high byte of Timer3 C channel							
$9E	OCR3CL	Compare output register low byte of Timer3 C channel							
$9D	DTR3H	Dead-band register high byte of Timer3							
$9C	DTR3L	Dead-band register low byte of Timer3							
$9B	OCR3BH	Compare output register high byte of Timer3 B channel							
$9A	OCR3BL	Compare output register low byte of Timer3 B channel							
$99	OCR3AH	Compare output register high byte of Timer3 A channel							
$98	OCR3AL	Compare output register low byte of Timer3 A channel							
$97	ICR3H	Input capture register high byte of Timer3							
$96	ICR3L	Input capture register low byte of Timer3							
$95	TCNT3H	Counter register high byte of Timer3							
$94	TCNT3L	Counter register low byte of Timer3							
$93	TCCR3D	Control register D of Timer3							
$92	TCCR3C	Control register C of Timer3							
$91	TCCR3B	Control register B of Timer3							
$90	TCCR3A	Control register A of Timer3							
$8D	DTR1H	Dead-band register high byte of Timer1							
$8C	DTR1L	Dead-band register low byte of Timer1							
$8B	OCR1BH	Timer 1 Output Compare B High							

Addr	Name	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
$8A	OCR1BL	Timer 1 Output Compare B Low							
$89	OCR1AH	Timer 1 Output Compare A High							
$88	OCR1AL	Timer 1 Output Compare A Low							
$87	ICR1H	Timer 1 Input Capture High							
$86	ICR1L	Timer 1 Input Capture Low							
$85	TCNT1H	Timer 1 Counter High							
$84	TCNT1L	Timer 1 Counter Low							
$83	TCCR1D	DSX17	DSX16	DSX15	DAX14	-	-	DSX11	DSX10
$82	TCCR1C	FOC1A	FOC1B	DOC1B	DOC1A	DTEN1	-	-	-
$81	TCCR1B	ICNC1	ICES1	-	WGM13	WGM12	CS1		
$80	TCCR1A	COM1A1	COM1A0	COM1B1	COM1B0	-	-	WGM11	WGM10
$7F	DIDR1	PE7D	PE6D	PE0D	C0PD	PF0D	PC7D	PD7D	PD6D
$7E	DIDR0	PE3D	PE1D	PC5D	PC4D	PC3D	PC2D	PC1D	PC0D
$7D	ADCSRC	OFEN	-	SPN	AMEN	-	SPD	DIFS	ADTM
$7C	ADMUX	REFS1	REFS0	ADLAR	CHMUX				
$7B	ADCSRB	CME01	CME00	CME11	CME10	-	ADTS		
$7A	ADCSRA	ADEN	ADSC	ADATE	ADIF	ADIE	ADPS		
$79	ADCH	ADC Data High							
$78	ADCL	ADC Data Low							
$76	DIDR2	-	PB5D	-	-	-	-	-	-
$75	IVBASE	Interrupt Vector Base Address							
$74	PCMSK4								
$73	PCMSK3	PCINT[39:32]							
$71	TIMSK3			ICIE3	-	OCIE3C	OCIE3B	OCIE3A	TOIE3
$70	TIMSK2	-	-	-	-	-	OCIE2B	OCIE2A	TOIE2
$6F	TIMSK1	-	-	ICIE1	-	-	OCIE1B	OCIE1A	TOIE1
$6E	TIMSK0	-	-	-	-	-	OCIE0B	OCIE0A	TOIE0
$6D	PCMSK2	PCINT[23:16]							
$6C	PCMSK1	PCINT[15:8]							
$6B	PCMSK0	PCINT[7:0]							
$69	EICRA	-	-	-	-	ISC11	ISC10	ISC01	ISC00
$68	PCICR	-	-	-	PCIE4	PCIE3	PCIE2	PCIE1	PCIE0
$67	RCKCAL	RC32K Calibration							
$66	RCMCAL	RC32M Calibration							
$65	PRR1	-	-	PRWDT	-	PRTIM3	PREFL	PRPCI	-
$64	PRR0	PRTWI	PRTIM2	PRTIM0	-	PRTIM1	PRSPI	PRUART0	PRADC
$62	VDTCR	WCE	SWR	-	VDTS			VDREN	VDTEN
$61	CLKPR	WCE	CKOE1	CKOE0	-	CLKPS			
$60	WDTCSR	WDIF	WDIE	WDP3	WDCE	WDE	WDP2	WDP1	WDP0
DirectIO Register									
$5F	SREG	I	T	H	S	V	N	Z	C
$5E	SPH	Stack Point High							

Addr	Name	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
$5D	SPL	Stack Point Low							
$5C	E2PD3	E2PCTL Data register byte 3							
$5B	C1TR	AC1 trimming data							
$5A	E2PD1	E2PCTL Data register byte1							
$59	DSAH	DSA[31:16] access port of uDSC							
$58	DSAL	DSA[15:0] access port of uDSC							
$57	E2PD2	E2PCTL Data register byte 2							
$56	ECCR	WEN	EEN	ERN	SWM	CP1	CP0	ECS1	ECS0
$55	MCUCR	FWKEN	FPDEN	SWR	PUD	IRLD	IFAIL	IVSEL	WCE
$54	MCUSR	SWDD	-	-	OCDRF	WDRF	BORF	EXTRF	PORF
$53	SMCR	-	-	-	-	SM			SE
$52	C0TR	AC0 Trimming register							
$51	C0XR	-	C0OE	C0HYSE	C0PS0	C0WKE	C0FEN	C0FS1	C0FS0
$50	C0SR	C0D	C0BG	C0O	C0I	C0IE	C0IC	C0IS	
$4F	DTR0	TC0 Dead-band timing control register							
$4E	SPDR	SPI Data register							
$4D	SPSR	SPIF	WCOL	-	-	-	DUAL	-	SPI2X
$4C	SPCR	SPIE	SPE	DORD	MSTR	CPOL	CPHA	SPR	
$4B	GPIOR2	General Purpose Register 2							
$4A	GPIOR1	General Purpose Register 1							
$49	TCCR0C	DSX07	DSX06	DSX05	DSX04	-	-	DSX01	DSX00
$48	OCR0B	Timer 0 Output Compare Register B							
$47	OCR0A	Timer 0 Output Compare Register A							
$46	TCNT0	Timer 0 Counter							
$45	TCCR0B	FOC0A	FOC0B	OC0AS	DTEN0	WGM02	CS02	CS01	CS00
$44	TCCR0A	COM0A1	COM0A0	COM0B1	COM0B0	DOC0B	DOC0A	WGM01	WGM00
$43	GTCCR	TSM	-	-	-	-	-	PSRASY	PSRSYNC
$42	EEARH	E2PCTL Address High							
$41	EEARL	E2PCTL Address Low							
$40	E2PD0	E2PCTL Data byte 0							
$3F	EECR	EEPM2	EEPM2	EEPM1	EEPM0	EERIE	EEMWE	EEWE	EERE
$3E	GPIOR0	General Purpose Register 0							
$3D	EIMSK	-	-	-	-	-	-	INT1	INT0
$3C	EIFR	-	-	-	-	-	-	INTF1	INTF0
$3B	PCIFR	-	-	-	-	PCIF3	PCIF2	PCIF1	PCIF0
$3A	C1XR	-	C1OE	C1HYSE	C1PS0	C1WKE	C1FEN	C1FS1	C1FS0
$39	SPFR	RDFULL	RDEMPT	RDPTR1	RDPTR0	WRFULL	WREMPT	WRPTR1	WRPTR0
$38	TIFR3	-	-	ICF3	-	-	OCF3B	OCF3A	TOV3
$37	TIFR2	-	-	-	-	-	OCF2B	OCF2A	TOV2
$36	TIFR1	-	-	ICF1	-	-	OCF1B	OCF1A	TOV1
$35	TIFR0	-	-	-	-	-	OCF0B	OCF0A	TOV0
$34	PORTF	Port Output of Group F							

Addr	Name	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
$33	DDRF	Data Direction of Group F							
$32	PINF	Port Input of Group F							
$31	DSDY	DSDY access port of uDSC							
$30	DSDX	DSDX access port of uDSC							
$2F	C1SR	C1D	C1BG	C1O	C1I	C1IE	C1IC	C1IS	
$2E	PORTE	Port Output of Group E							
$2D	DDRE	Data Direction of Group E							
$2C	PINE	Port Input of Group E							
$2B	PORTD	Port Output of Group D							
$2A	DDRD	Data Direction of Group D							
$29	PIND	Port Input of Group D							
$28	PORTC	Port Output of Group C							
$27	DDRC	Data Direction of Group C							
$26	PINC	Port Input of Group C							
$25	PORTB	Port Output of Group B							
$24	DDRB	Data Direction of Group B							
$23	PINB	Port Input of Group B							
$22	DSSD	DSSD access port of uDSC							
$21	DSIR	Instruction regiter of uDSC							
$20	DSCR	DSUEN	MM	D1	D0	-	DSN	DSZ	DSC

指令集速查表

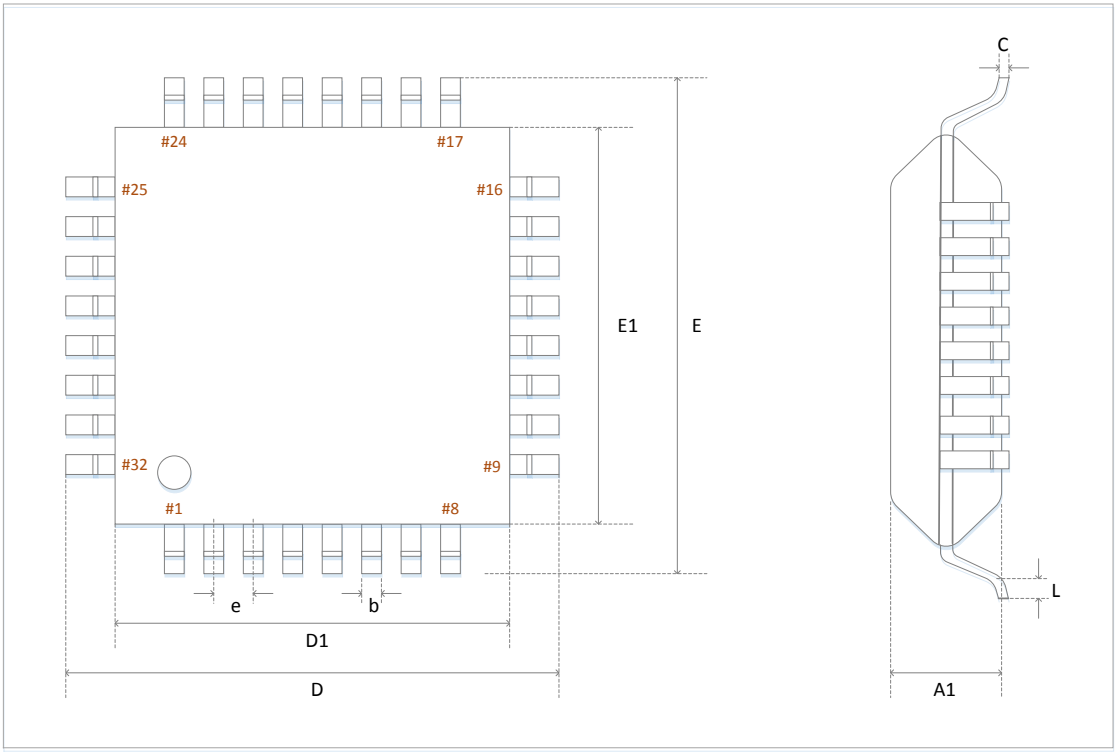
指令	操作数	描述	操作	标记位	周期
算术逻辑运算指令					
ADD	R_d, R_r	寄存器相加	$R_d \leftarrow R_d + R_r$	Z,C,N,V,H	1
ADC	R_d, R_r	带进位的寄存器相加	$R_d \leftarrow R_d + R_r + C$	Z,C,N,V,H	1
ADIW	R_{dl}, K	立即数与字相加	$R_{dh}:R_{dl} \leftarrow R_{dh}:R_{dl} + K$	Z,C,N,V,S	1
SUB	R_d, R_r	寄存器相加减	$R_d \leftarrow R_d - R_r$	Z,C,N,V,H	1
SUBI	R_d, K	寄存器减常数	$R_d \leftarrow R_d - K$	Z,C,N,V,H	1
SBC	R_d, R_r	带借位的寄存器相加减	$R_d \leftarrow R_d - R_r - C$	Z,C,N,V,H	1
SBCI	R_d, K	带借位的寄存器减常数	$R_d \leftarrow R_d – K - C$	Z,C,N,V,H	1
SBIW	R_{dl}, K	立即数与字相减	$R_{dh}:R_{dl} \leftarrow R_{dh}:R_{dl} - K$	Z,C,N,V,S	1
AND	R_d, R_r	逻辑与	$R_d \leftarrow R_d \& R_r$	Z,N,V	1
ANDI	R_d, K	寄存器逻辑与常数	$R_d \leftarrow R_d \& K$	Z,N,V	1
OR	R_d, R_r	逻辑或	$R_d \leftarrow R_d \mid R_r$	Z,N,V	1
ORI	R_d, K	寄存器逻辑或常数	$R_d \leftarrow R_d \mid K$	Z,N,V	1
EOR	R_d, R_r	寄存器异或	$R_d \leftarrow R_d \oplus R_r$	Z,N,V	1
COM	R_d	反码	$R_d \leftarrow \$FF - R_d$	Z,C,N,V	1
NEG	R_d	2 禁制补码	$R_d \leftarrow \$00 - R_d$	Z,C,N,V,H	1
SBR	R_d, K	设置寄存器中的位	$R_d \leftarrow R_d \vee K$	Z,N,V	1
CBR	R_d, K	清寄存器中的位	$R_d \leftarrow R_d \vee (\$FF – K)$	Z,N,V	1
INC	R_d	递增	$R_d \leftarrow R_d + 1$	Z,N,V	1
DEC	R_d	递减	$R_d \leftarrow R_d - 1$	Z,N,V	1
TST	R_d	测试为 0 或负数	$R_d \leftarrow R_d \& R_d$	Z,N,V	1
CLR	R_d	清寄存器	$R_d \leftarrow R_d \oplus R_d$	Z,N,V	1
SER	R_d	寄存器全设置为 1	$R_d \leftarrow \$FF$	None	1
MUL	R_d, R_r	无符号乘法	R_1: $R_0 \leftarrow R_d \times R_r$	Z,C	1
MULS	R_d, R_r	有符号乘法	R_1: $R_0 \leftarrow R_d \times R_r$	Z,C	1
MULSU	R_d, R_r	有符号数乘无符号数	R_1: $R_0 \leftarrow R_d \times R_r$	Z,C	1
FMUL	R_d, R_r	无符号乘法，移位	R_1: $R_0 \leftarrow (R_d \times R_r) << 1$	Z,C	1
FMULS	R_d, R_r	有符号乘法，移位	R_1: $R_0 \leftarrow (R_d \times R_r) << 1$	Z,C	1
FMULSU	R_d, R_r	有符号数乘无符号数，移位	R_1: $R_0 \leftarrow (R_d \times R_r) << 1$	Z,C	1
跳转指令					
RJMP	K	相对跳转	PC ← PC + K + 1	None	1
IJMP		间接跳转（到 Z 指向地址）	PC ← Z	None	2
JMP	K	直接跳转	PC ← K	None	2
RCALL	K	相对地址子程序调用	PC ← PC + K + 1	None	1
ICALL		间接子程序调用（Z 指向地址）	PC ← Z	None	2
CALL	K	直接子程序调用	PC ← K	None	2
RET		子程序返回	PC ← Stack	None	2
RETI		中断返回	PC ← Stack	I	2

指令	操作数	描述	操作	标记位	周期
跳转指令（续）					
CPSE	R_d, R_r	相等即跳转	If(R_d=R_r) PC ← PC + 2 or 3	None	1/2
CP	R_d, R_r	比较	R_d - R_r	Z,N,V,C,H	1
CPC	R_d, R_r	带进位比较	R_d - R_r – C	Z,N,V,C,H	1
CPI	R_d, K	与立即数比较	R_d - K	Z,N,V,C,H	1
SBRC	R_r, b	位为 0 即跳过下一条指令	If(R_r(b)=0) PC ← PC + 2 or 3	None	1/2
SBRS	R_r, b	位为 1 即跳过下一条指令	If(R_r(b)=1) PC ← PC + 2 or 3	None	1/2
SBIC	P, b	I/O 位为 0 即跳过下一条指令	If(P(b)=0) PC ← PC + 2 or 3	None	1/2
SBIS	P, b	I/O 位为 1 即跳过下一条指令	If(P(b)=1) PC ← PC + 2 or 3	None	1/2
BRBS	s, k	状态标记为 1 即跳转	If(SREG(S)=1) PC ← PC + K + 1	None	1/2
BRBC	s, k	状态标记为 0 即跳转	If(SREG(S)=0) PC ← PC + K + 1	None	1/2
BREQ	k	相等即跳转	if (Z = 1) then PC ← PC + k + 1	None	1/2
BRNE	k	不等即跳转	if (Z = 0) then PC ← PC + k + 1	None	1/2
BRCS	k	进位则跳转	if (C = 1) then PC ← PC + k + 1	None	1/2
BRCC	k	无进位则跳转	if (C = 0) then PC ← PC + k + 1	None	1/2
BRSH	k	不小于则跳转	if (C = 0) then PC ← PC + k + 1	None	1/2
BRLO	k	小于则跳转	if (C = 1) then PC ← PC + k + 1	None	1/2
BRMI	k	为负则跳转	if (N = 1) then PC ← PC + k + 1	None	1/2
BRPL	k	为正则跳转	if (N = 0) then PC ← PC + k + 1	None	1/2
BRGE	k	有符号的不小于即跳转	if (N⊕V= 0) then PC ← PC + k + 1	None	1/2
BRLT	k	有符号的小于 0 即跳转	if (N⊕V= 1) then PC ← PC + k + 1	None	1/2
BRHS	k	半进位为 1 则跳转	if (H = 1) then PC ← PC + k + 1	None	1/2
BRHC	k	半进位为 0 则跳转	if (H = 0) then PC ← PC + k + 1	None	1/2
BRTS	k	T 置位则跳转	if (T = 1) then PC ← PC + k + 1	None	1/2
BRTC	k	T 清零则跳转	if (T = 0) then PC ← PC + k + 1	None	1/2
BRVS	k	溢出则跳转	f (V = 1) then PC ← PC + k + 1	None	1/2
BRVC	k	不溢出则跳转	f (V = 0) then PC ← PC + k + 1	None	1/2
BRIE	k	全局中断使能则跳转	f (I = 1) then PC ← PC + k + 1	None	1/2
BRID	k	全局中断禁止则跳转	f (I = 0) then PC ← PC + k + 1	None	1/2
数据传输指令					
MOV	Rd, Rr	寄存器之间移动数据	Rd ← Rr	None	1
MOVW	Rd, Rr	移动一个字的数据	Rd+1:Rd ← Rr+1:Rr	None	1
LDI	Rd, K	加载立即数	Rd ← K	None	1
LD	Rd, X	间接加载	Rd ← (X)	None	1/2
LD	Rd, X+	间接加载，地址递增	Rd ← (X), X ← X + 1	None	1/2
LD	Rd, -X	地址递减，间接加载	X ← X - 1, Rd ← (X)	None	1/2
LD	Rd, Y	间接加载	Rd ← (Y)	None	1/2
LD	Rd, Y+	间接加载，地址递增	Rd ← (Y), Y ← Y + 1	None	1/2
LD	Rd, -Y	地址递减，间接加载	Y ← Y - 1, Rd ← (Y)	None	1/2
LDD	Rd, Y+q	带偏移量的间接加载	Rd ← (Y + q)	None	1/2
LD	Rd, Z	间接加载	Rd ← (Z)	None	1/2

LD	Rd, Z+	间接加载，地址递增	Rd ← (Z), Z ← Z+1	None	1/2
LD	Rd, -Z	地址递减，间接加载	Z ← Z - 1, Rd ← (Z)	None	1/2
LDD	Rd, Z+q	带偏移量的间接加载	Rd ← (Z + q)	None	1/2
LDS	Rd, k	直接从 SRAM 中加载	Rd ← (k)	None	2
ST	X, Rr	间接存储	(X) ← Rr	None	1
ST	X+, Rr	间接存储，地址递增	(X) ← Rr, X ← X + 1	None	1
ST	-X, Rr	地址递减，间接存储	X ← X - 1, (X) ← Rr	None	1
ST	Y, Rr	间接存储	(Y) ← Rr	None	1
ST	Y+, Rr	间接存储，地址递增	(Y) ← Rr, Y ← Y + 1	None	1
ST	-Y, Rr	地址递减，间接存储	Y ← Y - 1, (Y) ← Rr	None	1
STD	Y+q, Rr	带偏移量的间接存储	(Y + q) ← Rr	None	1
ST	Z, Rr	间接存储	(Z) ← Rr	None	1
ST	Z+, Rr	间接存储，地址递增	(Z) ← Rr, Z ← Z + 1	None	1
ST	-Z, Rr	地址递减，间接存储	Z ← Z - 1, (Z) ← Rr	None	1
STD	Z+q, Rr	带偏移量的间接存储	(Z + q) ← Rr	None	1
STS	k, Rr	直接存储到 SRAM 中	(k) ← Rr	None	2
LPM		加载程序空间数据	R0 ← (Z)	None	2
LPM	Rd, Z	加载程序空间数据	Rd ← (Z)	None	2
LPM	Rd, Z+	加载程序数据，地址递增	Rd ← (Z), Z ← Z+1	None	2
LD	Rd, Z+	间接加载，地址递增	Rd ← (Z), Z ← Z+1	None	1
LD	Rd, -Z	地址递减，间接加载	Z ← Z - 1, Rd ← (Z)	None	1
LDD	Rd, Z+q	带偏移量的间接加载	Rd ← (Z + q)	None	1
LDS	Rd, k	直接从 SRAM 中加载	Rd ← (k)	None	2
IN	Rd, P	读端口	Rd ← P	None	1
OUT	P, Rr	写端口	P ← Rr	None	1
PUSH	Rr	压栈	STACK ← Rr	None	1
POP	Rd	出栈	Rd ← STACK	None	1/2
SBI	P, b	设置 IO 寄存器	I/O(P, b) ← 1	None	1
CBI	P, b	清零 IO 寄存器	I/O(P, b) ← 0	None	1
LSL	Rd	逻辑左移	Rd(n+1) ← Rd(n), Rd(0) ← 0	Z,C,N,V	1
LSR	Rd	逻辑右移	Rd(n) ← Rd(n+1), Rd(7) ← 0	Z	1
ROL	Rd	包含进位的循环左移	Rd(0)←C, Rd(n+1) ← Rd(n), C←Rd(7)	Z	1
ROR	Rd	包含进位的循环右移	Rd(7)←C, Rd(n) ← Rd(n+1), C←Rd(0)	Z	1
ASR	Rd	算术右移	Rd(n) ← Rd(n+1), n=0:6	Z	1
SWAP	Rd	位交换	Rd(3:0) ← Rd(7:4), Rd(7:4) ← Rd(3:0)	None	1
BSET	s	设置状态位	SREG(s) ← 1	SREG(s)	1
BCLR	s	清零状态位	SREG(s) ← 0	SREG(s)	1
BST	Rr, b	存储到 T 位	T ← Rr(b)	T	1
BLD	Rd, b	读出 T 位到寄存器	Rd(b) ← T	None	1
SEC		设置进位标志	C ← 1	C	1
CLC		清楚进位标志	C ← 0	C	1

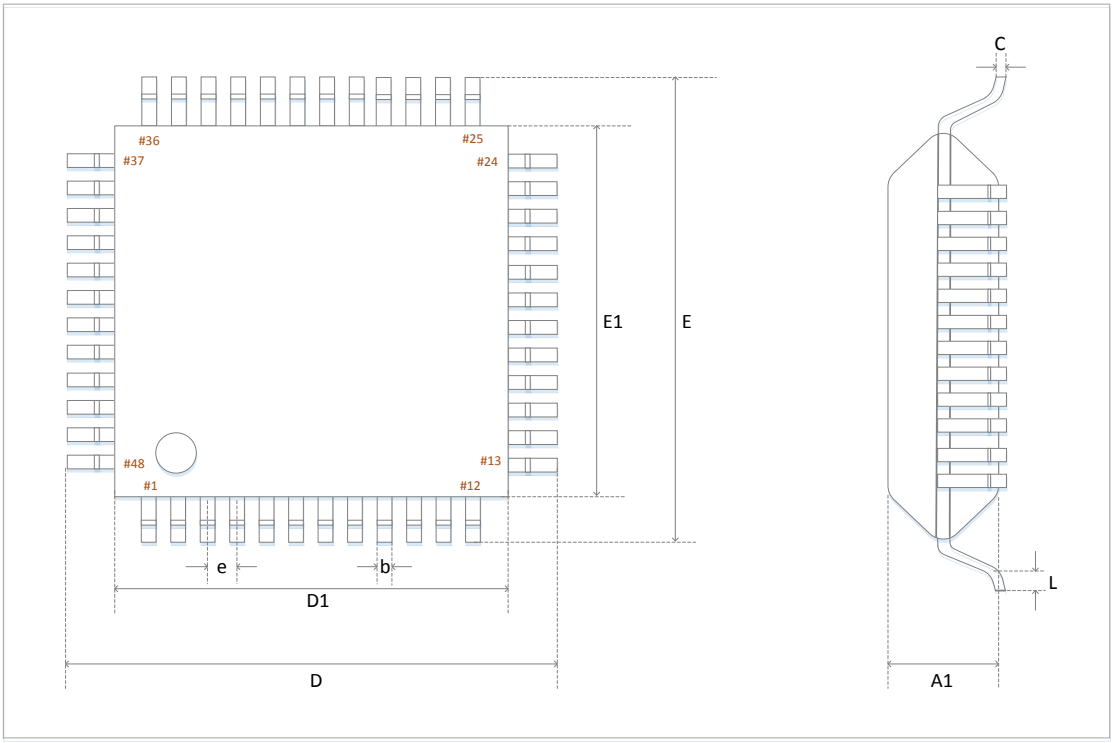
SEN		设置负数标志	N ← 1	N	1
CLN		清除负数标志	N ← 0	N	1
SEZ		设置零标志	Z ← 1	Z	1
CLZ		清除零标志	Z ← 0	Z	1
SEI		使能全局中断	I ← 1	I	1
CLI		禁制全局中断	I ← 0	I	1
SES		设置符号测试标志	S ← 1	S	1
CLS		清除符号测试标志	S ← 0	S	1
SEV		设置二进制补码溢出标志	V ← 1	V	1
CLV		清除二进制补码溢出标志	V ← 0	V	1
SET		设置 T 位（SREG）	T ← 1	T	1
CLT		清除 T 位（SREG）	T ← 0	T	1
MCU 控制指令					
NOP		空指令		None	1
SLEEP		进入休眠模式		None	1
WDR		看门狗复位		None	1
BREAK		软断点	仅用于调试目的	None	N/A
NOP		空指令		None	1
SLEEP		进入休眠模式		None	1

封装参数



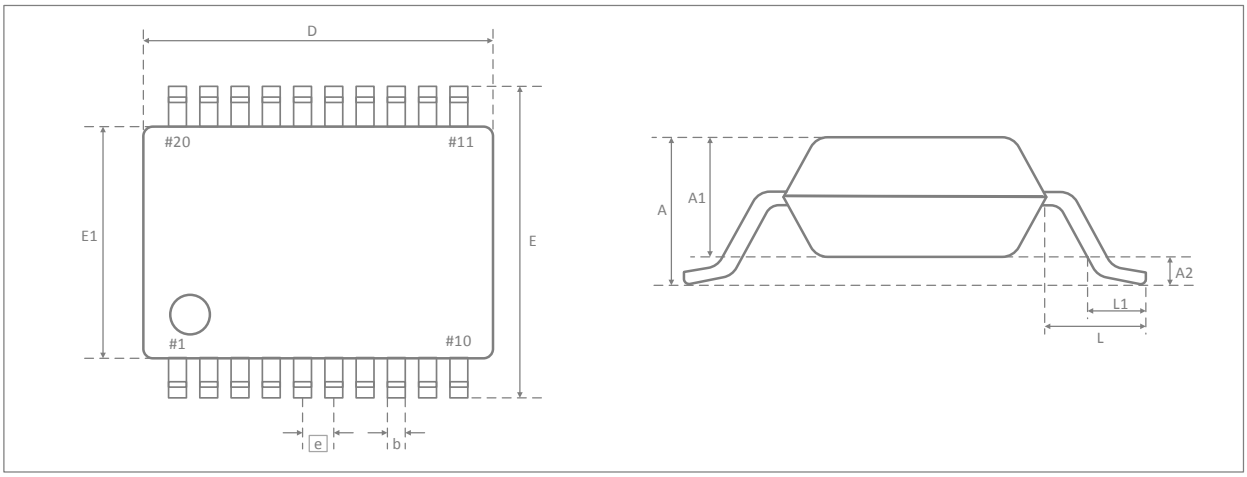
LQFP32 通用尺寸定义

字符代号	最小值	典型值	最大值	单位
D	8.90	9.00	9.10	mm
D1	6.90	7.00	7.10	mm
b	0.2	0.30	0.4	mm
e	0.75	0.80	0.85	mm
E	8.90	9.00	9.10	mm
E1	6.90	7.00	7.10	mm
C	-	0.10	-	mm
L	0.55	0.60	0.65	mm
A1	-	1.40	-	mm



LQFP48 通用尺寸定义

字符代号	最小值	典型值	最大值	单位
D	8.80	9.00	9.20	mm
D1	6.80	7.00	7.20	mm
b	0.17	0.22	0.27	mm
e	-	0.50BSC	-	mm
E	8.80	9.00	9.20	mm
E1	6.80	7.00	7.20	mm
C	0.09	-	0.2	mm
L	0.45	0.60	0.75	mm
A1	1.35	1.40	1.45	mm



SSOP20L 通用尺寸定义

字符代号	最小值	典型值	最大值	单位
D	6.90	7.20	7.50	mm
A2	0.03	0.05	0.07	mm
b	0.22	0.30	0.38	mm
e	-	0.65	-	mm
E	7.40	7.80	8.20	mm
E1	5.00	5.30	5.60	mm
L1	0.55	-	0.95	mm
L	-	-	-	mm
A1	-	2.0	-	mm

版本历史

V1.0.4 2017/11/15	更正 SSOP20 PIN8/11 的定义
V1.0.3 2017/6/23	增加 SSOP20 封装定义 更新 TMR3 中断标记位的操作说明
V1.0.2 2017/5/15	更新 TMR0/TRM1/TMR3 中关于自动 PWM 关闭和重启的说明 更新 SPI 章节中对 SPI 中断处理的说明以及更新 SPFR 寄存器的说明
V1.0.1 2017/2/13	删除 I2C1 部分，此功能不可用 完善了部分寄存器的定义
V1.0.0 2016/12/29	初始版本