Function Overview

- High Performance Low Power 8-bit LGT8XM Core
- Advanced RISC Architectures
  131 instructions, over 80% single cycle execution
  32x8 general-purpose working registers
  Internal single-cycle multiplier (8x8)
  with up to 32MIPS execution efficiency
  at 32MHz operation
- Non-volatile program and data storage space
  32Kbytes of on-chip, in-circuit programmable FLASH program memory
  2Kbytes internal data SRAM
  Programmable E2PROM emulation
  interface with byte access New
  program encryption algorithm for
  user code security
- Peripheral Controller
  Two 8-bit timers with independent prescaler and support for compare output mode
  Two 16-bit timers with independent prescalers support input capture and compare outputs Internal 32KHz calibratable RC
  oscillator for real-time counter function
  Supports up to 9 PWM outputs with three complementary programmable deadband controls
  12-channel 12-bit high-speed analog-to-digital converter (ADC)
  - Selectable internal, external reference voltage
  - Programmable gain (X1/8/16/32) differential amplification input channels
  - Automatic threshold voltage monitoring mode

  Two analog comparators (AC) to support extended
  internal 1.024V/2.048V/4.096V ±1% calibratable
  reference voltage sources from ADC input
  channels One 8-bit programmable DAC for
  generating reference voltage sources
  Programmable Watchdog Timer (WDT)
  Programmable Synchronous/Asynchronous
  Serial Interface (USART/SPI)
  Synchronous Peripheral Interface (SPI),
  Programmable Master/Slave Operating
  Mode Two-Wire Serial Interface
  (TWI), I2C Master-Slave Mode
  Compatible
  16-bit digital computing acceleration cell (DSC) supporting

direct 16-bit data access access

- Special processor functions
  SWD Dual Wire On-Chip
  Debug/Mass Production Interface
  External Interrupt Source and
  I/O Level Change Interrupt
  Support
  Built-in power-on reset
  circuit (POR) and
  programmable low voltage
  detection circuit (LVD)
  Built-in 1% calibratable
  32MHz RC oscillator
  with frequency doubling
  output support
  Built-in 1% calibratable 32KHz
  RC oscillator
  External support for 32.768KHz
  and 400K~32MHz crystal inputs
  6x high current push-pull driver
  IO to support high speed PWM
  applications

8-bit LGT8XM

RISC Microcontroller with In-System Programmable FLASH Memory

lgt8f88p

lgt8f168p

lgt8f328p

Data book

Version 1.0.5

Applications

Appliances

Motor Drive

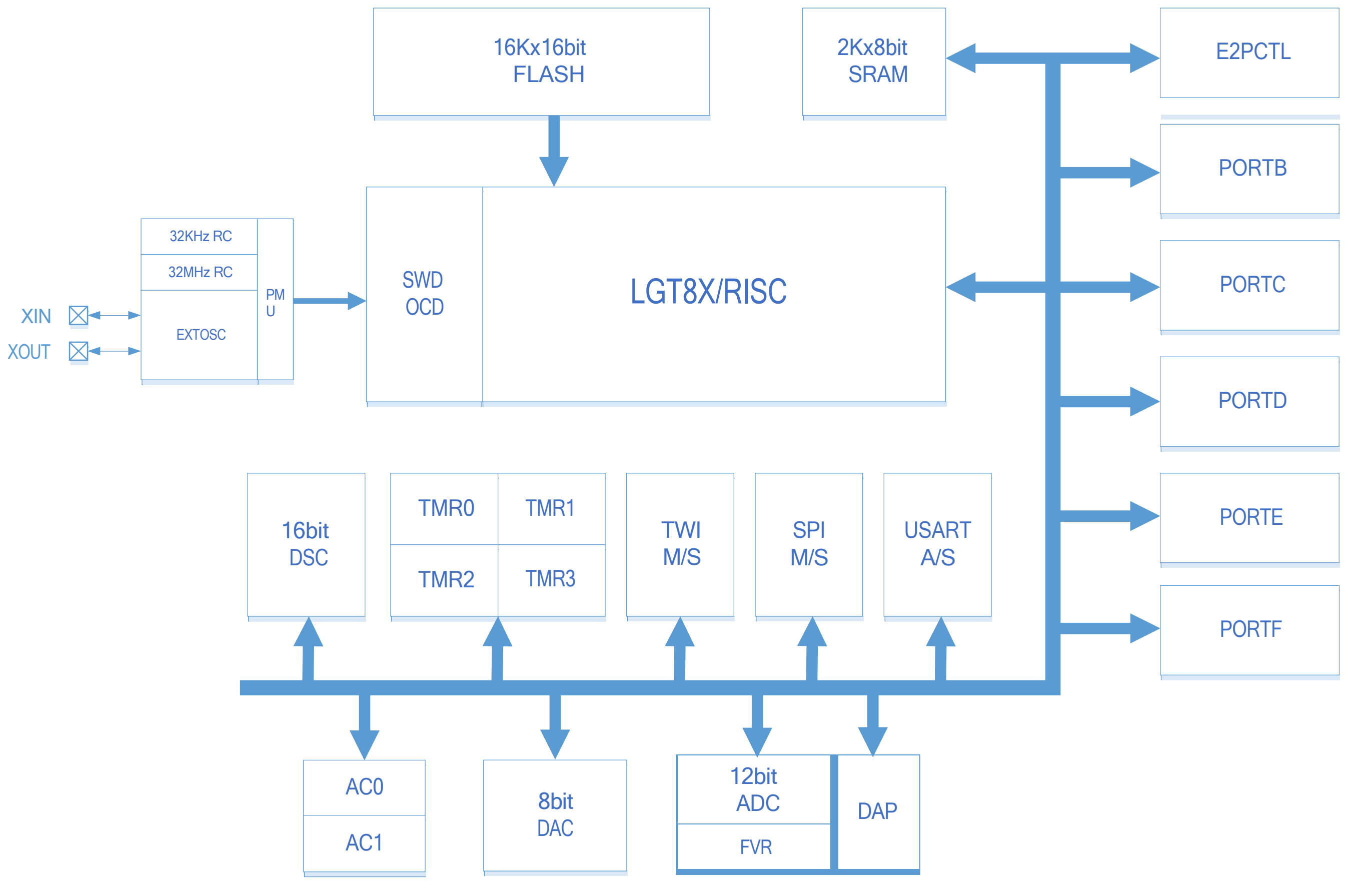
automatic control

- I/O and package: QFP48/32L, SSOP20L
- Lowest power consumption: 1uA@3.3V
- Working environment

  Operating Voltage: 1.8V ~ 5.5V

  Operating frequency: 0 ~ 32MHz Operating temperature: -40C ~ +85C HBM ESD :> 4KV

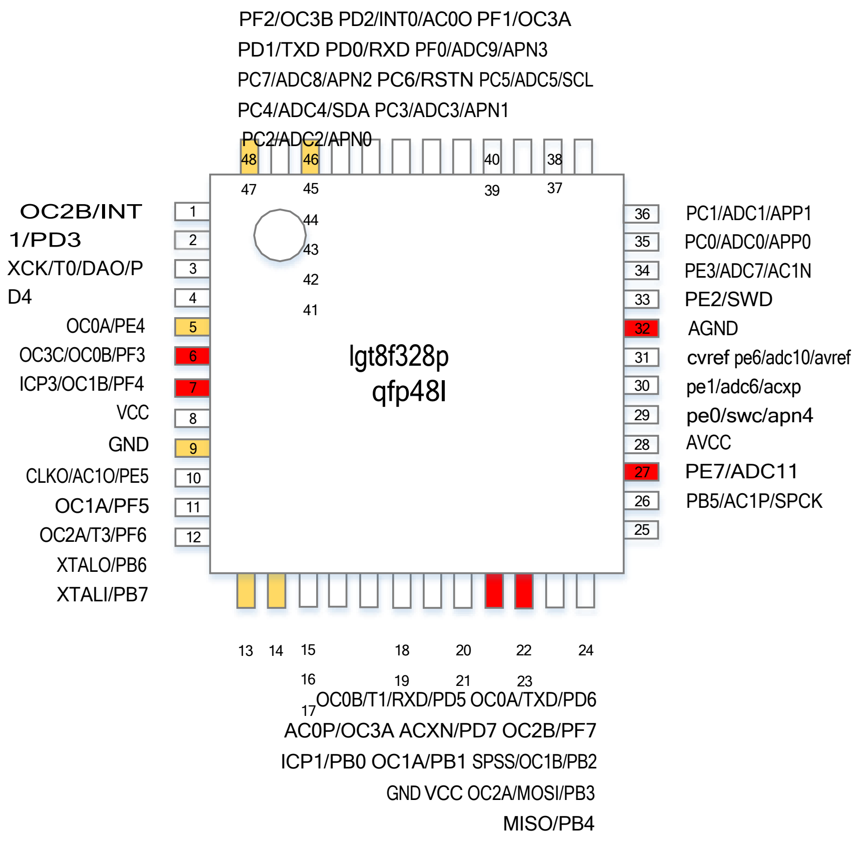
System framework

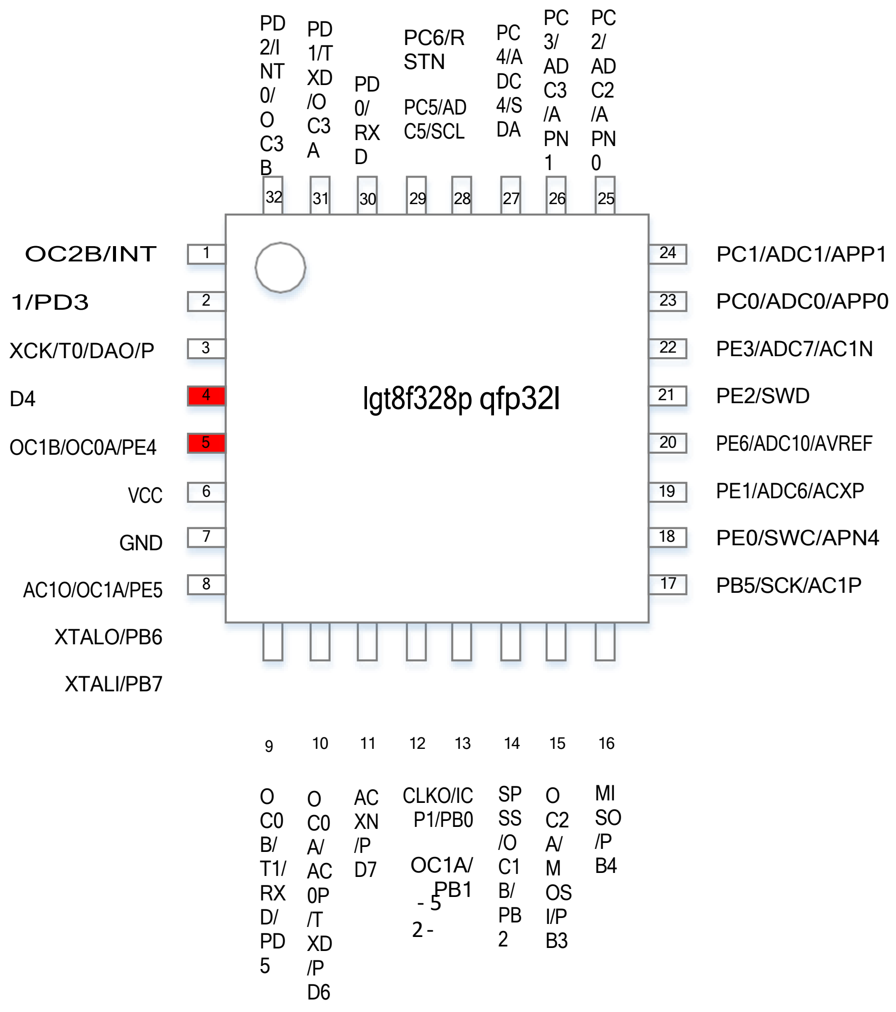


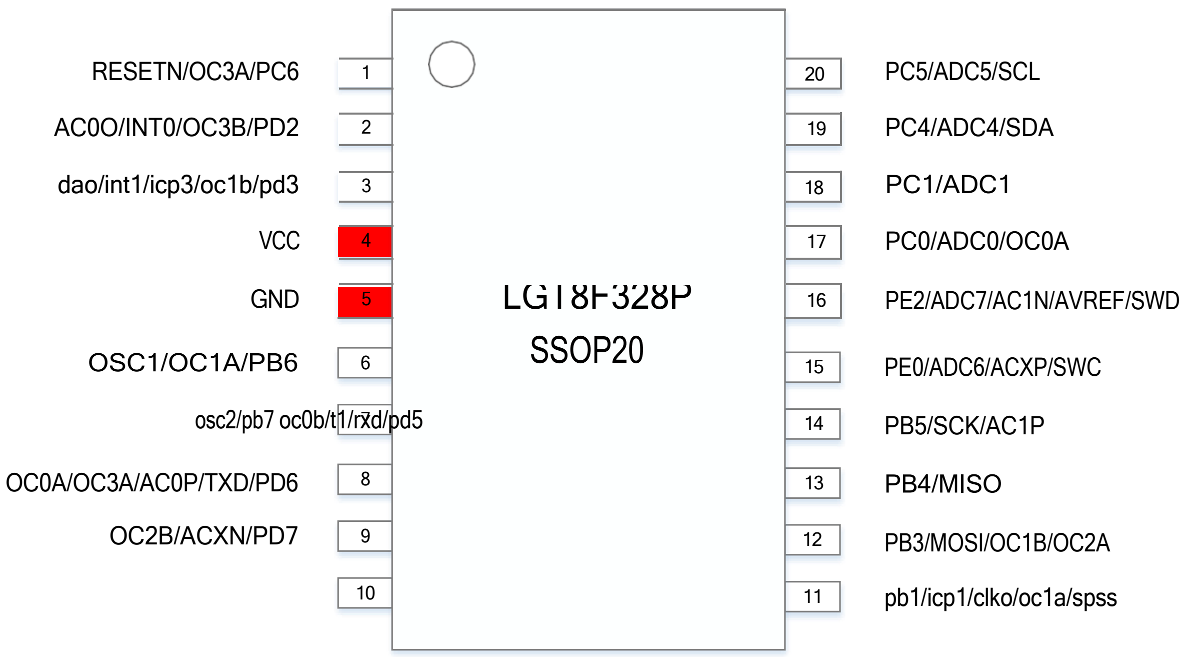
Module Name	Module Function
SWD	Debugging module for both online debugging and ISP functionality
LGT8X	8bit High Performance RISC Cores
E2PCTL	Data FLASH Access Interface Controller
PMU	Power management module, responsible for managing the transition between the operating states of the system
PORTB/C/D/E/F	Universal programmable input and output ports
DSC	16-bit Digital Acceleration Unit
ADC DAP IVREF	8-Channel 12-Bit Analog-to-Digital Converter 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, PWM Controller
WDT	Watchdog reset module
SPI M/S	Master-Slave SPI Controller
TWI M/S	Master-slave dual-wire interface controller, I2C protocol compatible

USART	Synchronous/asynchronous serial transceivers
DAC	8-bit digital-to-analog converter

Encapsulation Definition






LGT8F328P
SSOP20
RESETN/OC3A/PC6
1
AC0O/INT0/OC3B/PD2
2
dao/int1/icp3/oc1b/pd3
3
VCC
4
GND
5
OSC1/OC1A/PB6
6
osc2/pb7 oc0b/t1/rxd/pd5
7
OC0A/OC3A/AC0P/TXD/PD6
8
OC2B/ACXN/PD7
9
10
20
PC5/ADC5/SCL
19
PC4/ADC4/SDA
18
PC1/ADC1
17
PC0/ADC0/OC0A
16
PE2/ADC7/AC1N/AVREF/SWD
15
PE0/ADC6/ACXP/SWC
14
PB5/SCK/AC1P
13
PB4/MISO
12
PB3/MOSI/OC1B/OC2A
11
pb1/icp1/clko/oc1a/spss

Pin Description

The LGT8FX8P series package has the QFP48L package with all pins pinout. All other packages are generated by binding multiple internal I/Os to a single pin on a QFP48 basis. Special care needs to be taken when configuring the pin orientation. The following table lists the pin bindings for the various packages.

QFP48	QFP32	SSOP20	Function description
01	01	03	PD3/INT1/OC2B*
			PD3: Programmable port D3 INT1: External interrupt input 1 OC2B: Timer 2 Compare Match Output B
02	02		PD4/DAO/T0/XCK
			PD4: Programmable port D4 DAO: Internal DAC output T0: Timer0 external clock input XCK: USART synchronous transmission clock
03	03	-	PE4/0C0A*
			PE4: Programmable Port E4 OC0A: Timer 0 Compare Match Output A
04	-	-	PF3/OC3C/OC0B*
			PF3: Programmable Port F3 OC3C: Timer 3 Compare Match Output C OC0B: Timer 0 Compare Match Output B
05	03	03	PF4/OC1B*/ICP3
			PF4: Programmable Port F4 OC1B: Timer 1 Compare Match Output B ICP3: Timer 3 Capture Input
06	04	04	VCC
07	05	05	GND
08	06	-	PE5/AC1O/CLKO*
			PE5: Programmable port E5 C1O: Analog comparator AC1 output CLKO: System clock output
09		06	PF5/OC1A*
			PF5: Programmable Port F5 OC1A: Timer 1 Compare Match Output A
			PF6/T3/OC2A*

10	-	-	PF6: Programmable Port F6 T3: Timer 3 external clock input OC2A: Timer 2 Compare Match Output A
11	07	06	PB6/XTALO
			PB6: Programmable Port B6 XTALO: Crystal IO output port

12	08	07	PB7/XTALI PB7: Programmable port B7` XTALI: Crystal IO input port
13	09	08	pd5/rxd*/t1/oc0b PD5: Programmable Port D5 RXD: USART data reception (optional) T1: Timer 1 external clock input OC0B: Timer 0 Compare Match Output B
14	10	09	PD6/TXD*/OC0A PD6: Programmable Port D6 TXD: USART data send (optional) OC0A: Timer 0 Compare Match Output A
15			AC0P/0C3A AC0P: Analog Comparator 0 positive input OC3A: Timer 3 Compare Match Output A
16	11	10	PD7/ACXN PD7: Programmable Port D7 ACXN: Analog Comparator 0/1 Common Negative Input
17	-		PF7/OC2B PF7: Programmable Port F7 OC2B: Timer 2 Compare Match Output B
18	12	11	PB0/ICP1 PB0: Programmable Port B0 ICP1: Timer 1 captures input
19	13		PB1/OC1A PB1: Programmable Port B1 OC1A: Timer 1 Compare Match Output A
20	14	12	PB2/OC1B/SPSS PB2: Programmable Port B2 OC1B: Timer 1 Compare Match Output B SPSS: SPI Slave Mode Chip Select
21	-	-	GND
22	-	-	VCC
23	15	12	PB3/MOSI/OC2A PB3: Programmable Port B3 MOSI: SPI host output/slave input OC2A: Timer 2 Compare Match Output A
24	16	13	PB4/MISO PB4: Programmable Port B4 MISO: SPI host input/slave output
			PB5/SPCK/AC1P

25	17	14	PB5: Programmable port B5 SPCK: SPI clock signal AC1P: Analog Comparator 1 positive input

26	-	-	PE7/ADC11 PE7: Programmable Port E7 ADC11: ADC Analog Input Channel 11
27	-	-	AVCC: Internal analog circuit power supply
28	18	15	PE0/SWC/APN4 PE0: Programmable Port E0 SWC: SWD Debug Interface Clock APN4: Differential Amplifier Inverted Input Channel 4
29	19		PE1/ADC6/ACXP PE1: Programmable port E1 ADC6: ADC analog input channel 6 ACXP: Analog Comparator 0/1 Male Positive Input
30	20	16	PE6/ADC10/AVREF PE6: Programmable Port E6 ADC10: ADC analog input channel 10 AVREF: ADC external reference input
31	-	-	CVREF: ADC reference voltage output For external 0.1uF filter capacitor only
32	-	-	AGND: Internal analog circuit ground
33	21	16	PE2/SWD PE2: Programmable port E2 SWD: SWD debug interface data line
34	22		PE3/ADC7/AC1N PE3: Programmable Port E3 ADC7: ADC analog input channel 7 AC1N: Analog comparator negative input
35	23	17	PC0/ADC0/APP0 PC0: Programmable Port C0 ADC0: ADC analog input channel 0 APP0: Differential amplifier positive input channel 0
36	24	18	PC1/ADC1/APP1 PC1: Programmable port C1 ADC1: ADC analog input channel 1 APP1: Differential Amplifier Forward Input Channel 1
37	25	-	PC2/ADC2/APN0 PC2: Programmable port C2 ADC2: ADC analog input channel 2 APN0: Differential amplifier reverse input channel 0
38	26	-	PC3/ADC3/APN1 PC3: Programmable Port C3 ADC3: ADC analog input channel 3 APN1: Differential amplifier reverse input channel 1

39	27	19	PC4/ADC4/SDA PC4: Programmable Port C4 ADC4: ADC analog input channel 4 SDA: I2C controller data line
40	28	20	PC5/ADC5/SCL PC5: Programmable Port C5 ADC5: ADC analog input channel 5 SCL: I2C controller clock line
41	29	1	PC6/RESETN PC6: Programmable Port C6 RESETN: External reset input
42	-	-	PC7/ADC8/APN2 PC7: Programmable Port C7 ADC8: ADC analog input channel 8 APN2: Differential Amplifier Inverted Input Channel 2
43	-	-	PF0/ADC9/APN3 PF0: Programmable port F0 ADC9: ADC analog input channel 9 APN3: Differential Amplifier Inverted Input Channel 3
44	30	-	PD0/RXD PD0: Programmable port D0 RXD: USART data receive input
45	31	-	PD1/TXD PD1: Programmable port D1 TXD: USART data transmit output
46		1	PF1/OC3A PF1: Programmable Port F1 OC3A: Timer 3 Compare Match Output A
47	32	2	PD2/INT0/AC0O PD2: Programmable port D2 INT0: External interrupt input 0 AC0O: Analog Compare 0 output
48			PF2/OC3B PF2: Programmable Port F2 OC3B: Timer 3 Compare Match Output B

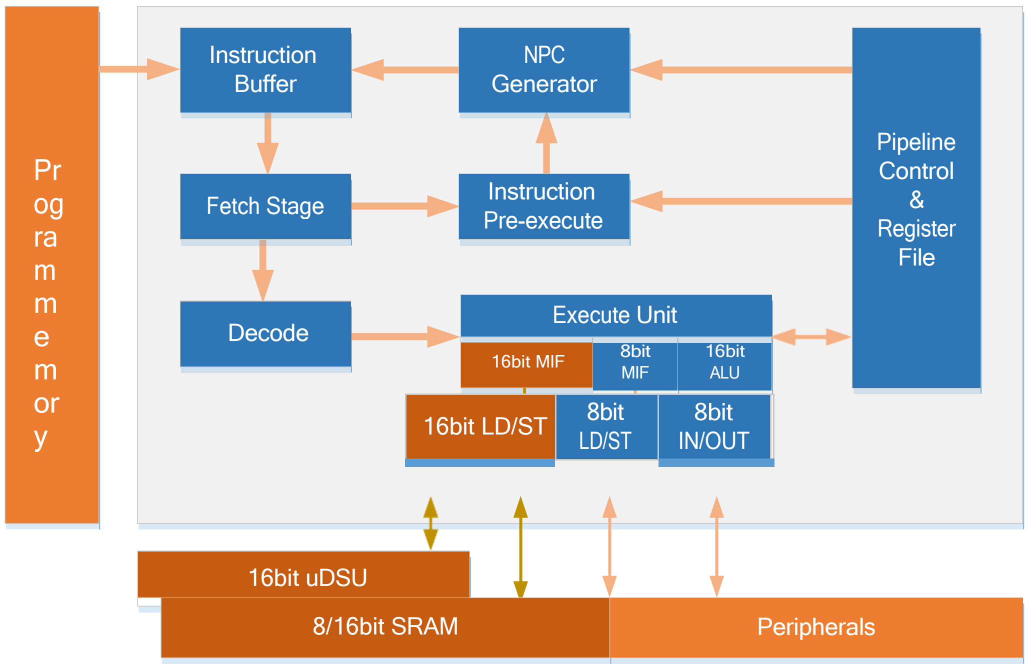
LGT8XM Kernel

- Low power design
- Highly efficient RISC architecture
- 16-bit LD/ST extension (uDSU-specific)
- 130 instructions, of which more than 80% are single-cycle
- Embedded Online Debugging (OCD) support

summarize

This section describes the LGT8XM kernel architecture and functions. The kernel is the brain of the MCU and is responsible for the correct execution of the program. Therefore, the kernel must be able to accurately perform calculations, control peripherals, and handle various interrupts.

Structure of the LGT8XM Kernel



To achieve greater efficiency and parallelism, the LGT8XM kernel uses the Haver architecture - separate data and program buses. Instructions are executed through an optimized two-stage pipeline, which reduces the number of invalid instructions in the pipeline and reduces the amount of accesses to FLASH program memory, thus reducing the power consumption of the kernel. At the same time, the LGT8XM kernel adds an instruction cache (which can cache two instructions at the same time)in the fetch stage, and further reduces the access frequency to FLASH program memory by pre-executing the module in the fetch cycle; after extensive testing, the LGT8XM can reduce the access to FLASH by about 50% compared to other kernels of similar architectures, which greatly reduces the power consumption of the system.

The LGT8XM core has 32 8-bit high-speed access general-purpose working registers (register file), which help to implement single-cycle arithmetic logic operations (ALU).In general, both operands of the ALU operation are derived from the general-purpose working registers, and the result of the ALU operation is written to the register file in one cycle.

The LGT8XM supports single-cycle 16-bit arithmetic operations, which greatly improves the efficiency of indirect addressing. The three special 16-bit registers in the LGT8XM kernel are named X, Y, and Z, and will be described in detail later.

The ALU supports arithmetic logic operations between registers and between constants and registers. individual register operations can also be performed in the ALU. after the ALU operation is completed, the effect of the operation result on the kernel state is updated into the status register (SREG).

Program flow control is implemented through conditional and unconditional jumps/calls that can address so program areas. Most
The LGT8XM instructions are 16-bit. Each program address space corresponds to one 16-bit or 32-bit LGT8XM instruction.

After the kernel responds to an interrupt or subroutine call, the return address (PC) is stored on the stack. The stack is allocated in the system's general data SRAM, so the size of the stack is limited only by the size and usage of the SRAM in the system. All applications that support interrupts or subroutine calls must first initialize the stack pointer register (SP), which can be accessed through IO space. The data SRAM can be accessed through five different addressing modes. the internal storage of the LGT8XM is linearly mapped to a uniform address space. Please refer to the Storage chapter for details.

The LGT8XM kernel contains a flexible interrupt controller, and interrupt functions can be controlled via a global interrupt enable bit in the status register. All interrupts have a separate interrupt vector. The priority of an interrupt corresponds to the interrupt vector address; the smaller the interrupt address, the higher the priority of the interrupt.

The I/O space contains 64 registers that can be directly addressed by the IN/OUT instructions. These registers provide control functions for kernel control and status registers, SPI and other I/O peripherals. This space can be accessed directly via IN/OUT instructions or by mapping them to addresses in data memory space (0x20 – 0x5F). In addition, the LGT8FX8P also contains extended I/O space, and they are mapped to data memory space 0x60 - 0xFF, where they can only be accessed using the ST/STS/STD and LD/LDS/LDD instructions.

To enhance the computing power of the LGT8XM core, a 16-bit LD/ST extension has been added to the instruction pop line. This 16-bit LD/ST extension works in conjunction with the 16-digit acceleration unit (uDSU) to enable efficient 16-bit data operations. The kernel also adds 16-bit access to RAM space. Thus, the 16-bit LD/ST extensions can pass 16-bit data between the uDSU, RAM, and working registers. For details, please refer to the section "Digital Computing Accelerators".

Arithmetic Logic Unit (*ALU*)

The LGT8XM contains an internal 16-bit arithmetic logic unit capable of performing 16 arithmetic operations on data in one cycle. The highly efficient ALU is connected to 32 general-purpose working registers. There are three types of operations in the ALU: arithmetic, logical and bit operations. The ALU section also contains a single-cycle hardware multiplier that can perform direct signed or unsigned operations on two 8-bit registers in a single cycle. Please refer to the instruction set section for details.

Status Register *(SREG)*

The status register mainly holds information about the results resulting from the execution of the most recent ALU operation. This information is used to control the flow of program execution. The status register is updated after the ALU operation is completely finished, which eliminates the need to use a separate compare instruction and can lead to a more compact and efficient code implementation. The status register values are not automatically saved and restored in response to and upon exit from an interrupt, which requires software to implement.

SREG Register Definition

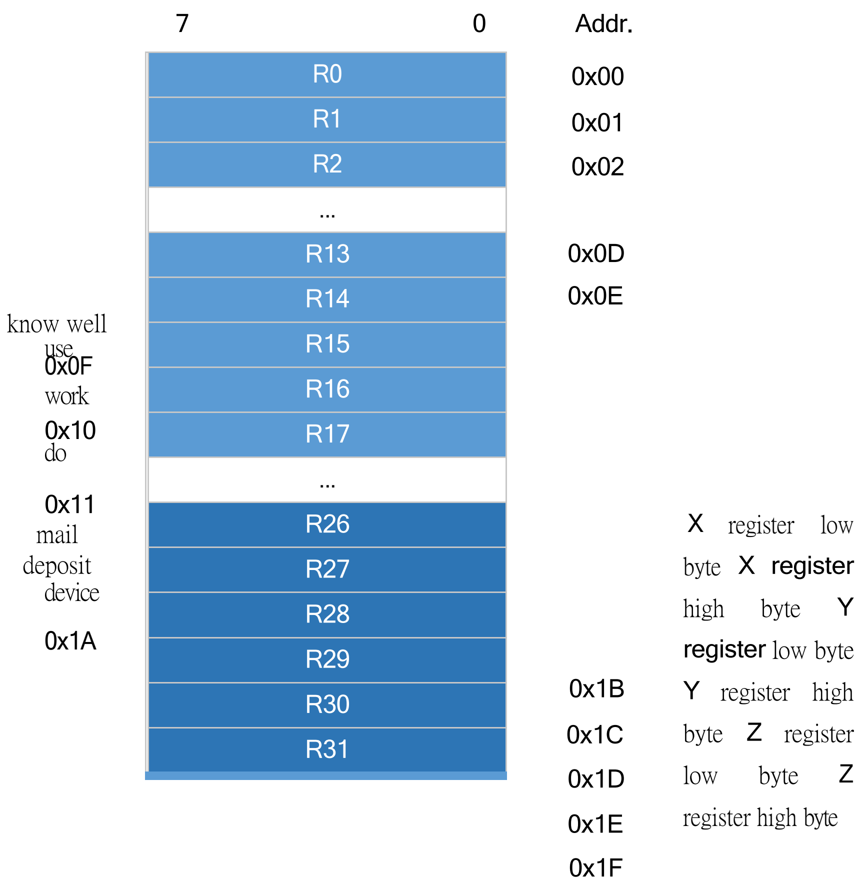
SREG System Status Register								
Address: 0x3F (0x5F)				Default value: 0x00				
Bit	7	6	5	4	3	2	1	0
Name	I	T	H	S	V	N	Z	C
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit Definition								
[0]	C	The rounding flag, indicating that an arithmetic or logical operation resulted in a rounding, see the instruction description for details						
[1]	Z	The zero flag, indicating that the result of an arithmetic or logical operation is zero, see the instruction description section						
[2]	N	Negative flag, indicating that an arithmetic or logical operation has produced a negative number, see the instruction description section						
[3]	V	The overflow flag, indicating that the result of a binary complement operation produces an overflow, see the instruction description section						
[4]	S	Signed bits, equivalent to the result of an iso-or operation between N and V. Refer to the instruction description section for details						
[5]	H	Half-entry flag, useful in BCD operations, indicates a half-entry resulting from a byte operation						
[6]	T	Temporary bits, bit copy (BLD) and bit store (BST) instructions are used in the T bit will be used as a A temporary storage bit is used to temporarily store the value of a bit in the general-purpose register. Refer to the instruction description section for details						
[7]	I	The global interrupt enable bit, which must be set to 1 to enable the kernel to respond to interrupt events. The different interrupt sources are controlled by separate control bits. The global interrupt enable bit is the final barrier controlling the entry of interrupt signals into the kernel. the I bit is automatically cleared by hardware after the kernel responds to an interrupt vector and is automatically set after the execution of the return to interrupt instruction (RETI). The I bit can also be changed using the SEI and CLI instructions, see the instruction description section						

General-purpose working register

The general-purpose working registers are optimized according to the LGT8XM instruction set architecture. In order to achieve the efficiency and flexibility required for kernel execution, the
The LGT8XM's internal general-purpose operating registers support the following access modes.

- One 8-bit read and one 8-bit write operation
- Two 8-bit reads and one 8-bit write operation
- Two 8-bit reads and one 16-bit write operation
- One 16-bit read and one 16-bit write operation

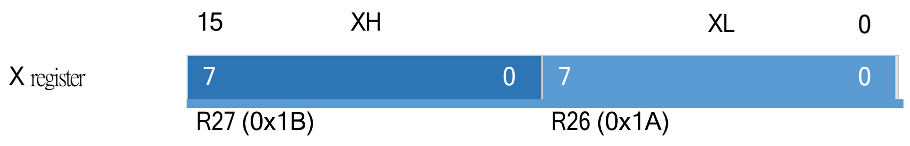
LGT8XM General Purpose Working Register

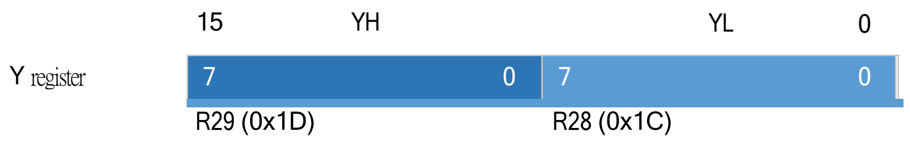


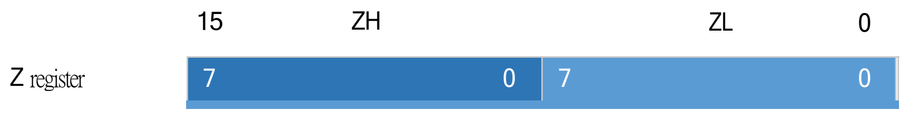
Most of the instructions have direct access to the full range of general-purpose working registers, and most of them are also single-cycle instructions.
As shown above, each register corresponds to an address in the data store, and these general-purpose working registers are mapped to the data store. As soon as possible they do not really exist in SRAM, but this uniformly mapped storage organization gives a lot of flexibility in accessing them. x/y/z registers can be indexed as pointers to any general-purpose register.

X/Y/Z Register

Registers R26...R31 can be combined two by two to form three 16-bit registers. These three 16-bit registers are mainly used as address pointers for indirect addressing accesses, and the X/Y/Z register structure is as follows.







R31 (0x1F) R30 (0x1E)

These registers are used as fixed-offset, auto-increment and auto-decrement address pointers in different addressing modes, as described in the instruction descriptions.

stack pointer

The stack is used to store temporary data, local variables, and the return addresses of interrupts and subroutine calls. It is important to note that the stack is not designed to grow from a high address to a low address. The stack pointer register (SP) always points to the top of the stack. The stack pointer points to the physical space where the data SRAM is located and where the stack space necessary for a subroutine or interrupt call is stored. the PUSH instruction will decrement the stack pointer.

The location of the stack in the SRAM must be set correctly by software prior to subroutine execution or interrupt enable. The stack pointer is normally initialized to point to the highest address of the SRAM. The stack pointer must be set to the high SRAM start address. refer to the System Data Storage section for the address of the SRAM in the System Data Storage map.

Stack pointer-related instructions

command	stack pointer	description
PUSH	Increase 1	Data pressed into the stack
CALL ICALL RCALL	Increase 2	The return address of an interrupt or subroutine call is pressed onto the stack
POP	Reduction 1	Data is removed from the stack
RET RETI	Reduction 2	The return address of an interrupt or subroutine call from the stack

The stack pointer consists of two 8-bit registers allocated in I/O space. The actual length of the stack pointer is system implementation dependent. In some chip implementations of the LGT8XM architecture, the data space is so small that SPL alone is sufficient for addressing, in which case the SPH register will not be present.

SPH/SPL Stack Pointer Register Definition

SPH/SPL Stack Pointer Register		
SPH: 0x3E (0x5E) SPL: 0x3D (0x5D)	Default value: RAMEND	
SP	SP [15:0]	
R/W	R/W	
Bit Definition		
[7:0]	SPL	Stack pointer low 8 bits
[15:8]	SPH	Stack pointer 8 bits higher

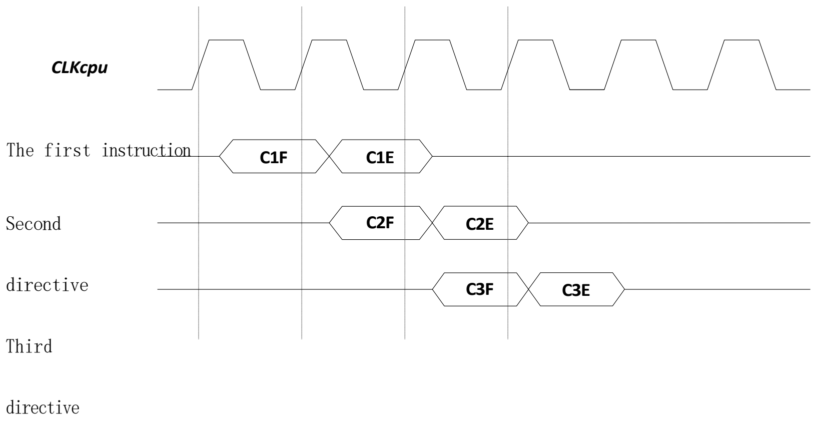
instruction execution timing

This section describes the general timing concepts for instruction execution. LGT8XM kernel is driven by a kernel clock (CLKcpu) that comes directly from the clock source selection circuitry with the system.

The following diagram shows the instruction pipeline execution timing based on the concept of a Haver architecture with fast access register files. This is the timing that makes

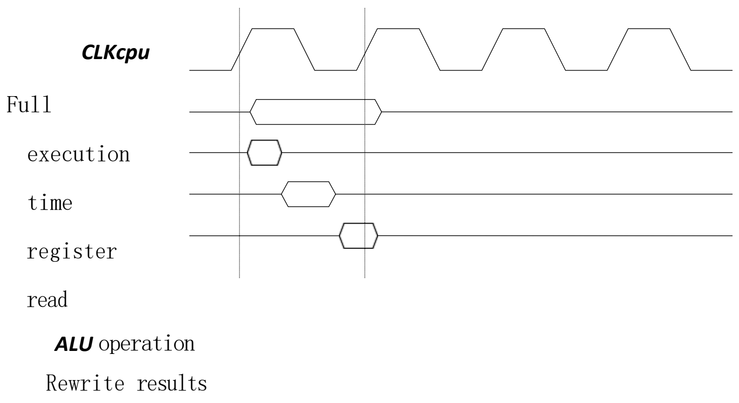
The kernel is physically guaranteed to achieve an execution efficiency of 1MIPS/MHz.

As you can see from the above figure, the second instruction is read at the same time during the execution of the first instruction. When the second instruction enters the execution



line period, while a third instruction is read at the same time. This does not require additional cycles to be spent for reading instructions during the entire execution period, achieving the efficiency of executing one instruction per Monday from a pipeline point of view.

The following diagram shows the access timing of the general-purpose working registers. In one cycle, the ALU operation uses up to two registers as operands and writes the ALU execution result to the target register during this cycle.



Reset and Interrupt Handling

The LGT8XM supports multiple interrupt sources. These interrupts, as well as the reset vector, correspond to a separate program vector entry in program space. In general, all interrupts are controlled by a separate control bit. When this control bit is set and the kernel's global interrupt enable bit is enabled, the kernel can only respond to this interrupt.

The lowest program space is reserved by default for the reset as well as the interrupt vector area.The complete list of interrupts supported by the LGT8FX8P is described in the Interrupts chapter. This list also determines the priority of the different interrupts. The lower the vector address, the higher the corresponding interrupt priority. Reset (RESET) has the highest priority, followed by INT0 - External Interrupt Request 0. The start address of the interrupt vector table (except for the reset vector)can be redefined to the start of any 256-byte alignment, and requires the IVSEL bit in the MCU Control Register (MCUCR) and the IVBASE vector base address register This is achieved through the IVSEL bit in the MCU Control Register (MCUCR) and the IVBASE Vector Base Address Register.

When the kernel responds to an interrupt, the global interrupt enable flag of I is automatically cleared by hardware. The user can implement interrupt nesting by enabling the I bit. The I bit is automatically set after the execution of the return from interrupt instruction (RETI), so that subsequent interrupts can be responded to normally.

There are basic types of interrupts. The first type is triggered by an event, and the interrupt flag bit is set after the interrupt event occurs. For this type of interrupt, after the kernel responds to the interrupt request, the current PC value is directly replaced with the actual interrupt vector address and the corresponding interrupt service subroutine is executed, while the hardware automatically clears the interrupt flag bit. The interrupt flag bit can also be cleared by writing 1 to the location of the interrupt flag bit. If the interrupt enable bit is cleared when an interrupt occurs, the interrupt flag bit will still be set to record the interrupt event. By the time the interrupt is enabled, this recorded interrupt event will be responded to immediately. Similarly, if the global interrupt enable bit (SERG. I) is cleared when an interrupt occurs, the corresponding interrupt flag bit is set to record the interrupt event, and so on

By the time the global interrupt enable bit is set, these recorded interrupts will be executed in order of priority.

The second interrupt type is an interrupt that keeps responding when the interrupt condition is always present. This type of interrupt does not require an interrupt flag bit. If the interrupt condition disappears before the interrupt is enabled, this interrupt will not be responded to.

When the LGT8XM kernel exits from the interrupt service subroutine, the execution flow returns to the main program. One or more instructions are executed in the main program before responding to other waiting interrupt requests.

Note that the System Status Register (SREG) is not automatically saved upon entering interrupt service, nor is it automatically restored upon returning from interrupt service. It must be handled by software.

When interrupts are disabled with the CLI instruction, they will be disabled immediately. All interrupts that occur after the CLI instruction will not be responded to. Even interrupts that occur at the same time as the CLI instruction is executed will not be responded to. The following example shows how to use the CLI to avoid interrupts disrupting the EEPROM write timing.

Interrupt response time

The LGT8XM kernel is optimized for interrupt response so that any interrupt must be responded to within 4 system clock cycles. after 4 system clock cycles, the interrupt service subroutine enters the execution cycle. During these four clocks, the PC value prior to the interrupt is pressed onto the stack and the system execution flow jumps to the interrupt vector corresponding to the interrupt service program. If the interrupt occurs during a multi-cycle instruction execution, the kernel will ensure that the current instruction is properly executed to the end. If the interrupt occurs while the system is in the sleep state (SLEEP), the interrupt response requires an additional 4 clock cycles.This added clock period is used for the synchronization period of the wakeup operation from the selected hibernation mode. For a detailed description of the hibernation mode, refer to the section on power management.

It takes 2 clock cycles to return from the interrupt service subroutine.During these 2 clock cycles, the PC recovers from the stack, the stack pointer is added by 2, and the global interrupt control bit is automatically enabled.

Storage unit

summarize

This section describes the different internal memory cells of the LGT8FX8P family.ȠLGT8XM architecture supports two main types of internal storage, data storage and program storage. the LGT8FX8P also contains data FLASH internally, and the EEPROM interface data storage function can be implemented through the internal controller.ȠIn addition, the LGT8FX8P system contains special memory cells for system configuration information and the chip's global device number (GUID).

The LGT8FX8P series chips include four different models of LGT8F88P/168P/328P;the peripherals and packages of the four models are fully compatible, the difference is the FLASH program memory and the internal data SRAM, the following table clearly describes the different memory configurations of the LGT8FX8P series chips.

DEVICE	FLASH	SRAM	E2PROM	interrupting vector
LGT8F88P	8KB	1KB	2KB	1 command word
LGT8F168P	16KB	1KB	4KB	2 command words
LGT8F328P	32KB	2KB	Configurable to 0K/1K/2K/4K/8K (shared with FLASH)	2 command words

The LGT8F328P does not have a separate FLASH space inside for emulating the E2PROM interface; the memory space for emulating the E2PROM is shared with the program FLASH, and the user can choose the appropriate configuration according to the application requirements.

Due to the unique implementation of the emulated E2PROM interface, the system requires twice the program FLASH space to emulate the E2PROM storage space. For example, for the LGT8F328P, if the user configures 1KB of E2PROM space, 2KB bytes of program space will be reserved, leaving 30KB of FLASH space for storing the program.

LGT8F328P Program FLASH and E2PROM Shared Configuration Table.

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

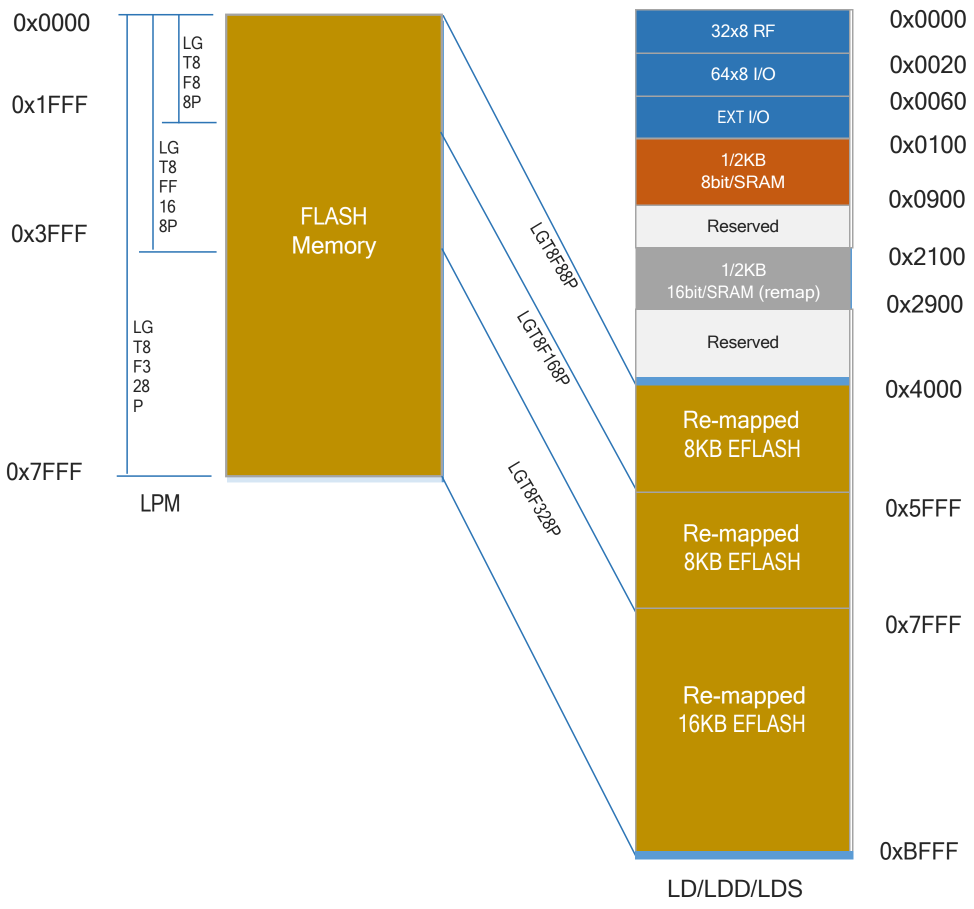
System programmable *FLASH* program memory unit

The LGT8FX8P series microcontrollers include 8K/16K/32K bytes of on-chip, in-circuit programmable FLASH program memory.

The LGT8FX8P has an internal FLASH interface controller that enables in-system programming (ISP) and self-updating of the program. For implementation details, please refer to the FLASH Interface Controller section in this chapter.

The program space can also be accessed (read) directly via the LPM instruction, a feature that enables application-related constant lookups

Table. The FLASH program space is also mapped into the system data store, and the user can also access the FLASH space using LD/LDD/LDS. The program space is mapped to the address range starting from 0x4000 in the data store.
As shown in the figure below.



SRAM data storage unit

The LGT8FX8P family of microcontrollers is a relatively complex microcontroller that supports many different types of peripherals whose controllers are allocated in 64 I/O register spaces. They can be accessed directly through IN/OUT instructions. Other peripherals have their control registers allocated in the 0x60 ~ 0xFF area, which can only be accessed via ST/STS/STD and LD/LDS/LDD instructions as this space is mapped into the data storage space.

The LGT8FX8P's system data storage space starts at address 0 and maps the general-purpose working register file, I/O space, extended I/O space, and internal data SRAM space, respectively. The first 32 byte addresses correspond to the 32 general-purpose working registers of the LGT8XM core. The next 64 addresses are the standard I/O space that can be accessed directly by the IN/OUT instructions. The next 160 addresses are the extended I/O space, followed by up to 2K bytes of data SRAM, starting at 0x4000 and ending at 0xBFFF, which maps the FLASH program memory cells.

The 1K/2K bytes of SRAM in the system are mapped into two separate spaces. The space starting at 0x0100 and ending at 0x0900 is read and written by the kernel in 8-bit width. The area starting at 0x2100 and ending at 0x2900 is a 16-bit wide access space. The system RAM is mapped to the higher addresses starting at 0x2100 and is mainly used to work with the uDSU module for efficient 16-bit data storage. During programming, the normal 8-bit addressable variable address is added with an offset of 0x2000 to switch to 16-bit access mode.

The system supports five different addressing modes that can cover the entire data space: direct access, indirect access with offset, indirect access, indirect access with decremented address before access, and indirect access with incremented address after access. The general-purpose working registers R26 to R31 are used as address pointers for indirect accesses. Indirect accesses can address the entire data storage space. Indirect accesses with offset addresses can address up to 63 address spaces in the vicinity of the Y/Z register as the base address.

When using the register indirect access mode that supports automatic address increment/decrement, address registers X/Y/Z are automatically decremented/incremented by hardware before/after the access occurs. Refer to the instruction set description section for details.

The 16-bit registers X/Y/Z and their associated auto-addressing modes (increment, decrement) also have a very important role to play in the 16-bit extended mode. 16-bit extended mode allows auto-increment and decrement addressing with variables using the increment/decrement modes of LD/ST. This mode is very effective when performing arithmetic operations on arrays. For details, please refer to the chapter on "Digital Computing Accelerators (uDSU)".

General Purpose *I/O* Registers

The I/O space of the LGT8FX8P has three general-purpose I/O registers, GPIOR2/1/0, which can be accessed using the IN/OUT instructions and are used to store user-defined data.

Peripheral register space

For detailed definitions of the I/O spaces, refer to the "Register Overview" section of the LGT8FX8P datasheet.

LGT8FX8P So all peripherals are assigned to I/O space. All I/O space addresses can be accessed by LD/LDS/LDDD and ST/STS/STD instructions. The accessed data are passed through 32 general-purpose working registers. The I/O registers between 0x00 ~ 0x1F can be accessed by the bit addressing instructions SBI and CBI. The value of a bit in one of these registers can be detected using the SBIS and SBIC instructions to control the flow of program execution. Refer to the instruction set description section for details.

When accessing the I/O registers using the IN/OUT instructions, they must be addressed between 0x00 ~ 0x3F. When accessing I/O space using the LD or ST instructions, the I/O space must be accessed via the mapped address of the I/O space in the system data memory unified mapping space (plus an offset of 0x20). Some other peripheral registers allocated in the extended I/O space (0x60 ~ 0xFF) can only be accessed using the ST/STS/STD and LD/LDS/LDD instructions.

For compatibility with future devices, reserved bits must be written 0 during write operations. write operations cannot be performed on reserved I/O space.

Some registers include status flags that need to be written 1 to clear them. Note that the CBI and SBI instructions only support specific bits, so the CBI/SBI can only work on registers that contain these status flags. In addition, the CBI/SBI instructions can only work on registers in the address range of 0x00 to 0x1F.

FLASH Controller *(E2PCTL)*

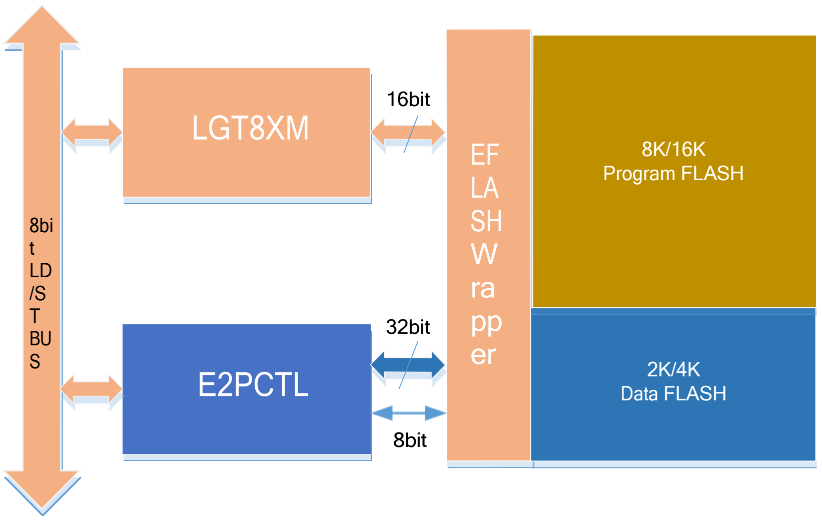
The LGT8FX8P internal implementation integrates a flexible and reliable EFLASH read/write controller, which can use the existing data FLASH storage space in the system to realize byte read/write access to the storage space to realize E2PROM-like storage applications;the E2PROM interface emulation adopts the

erase balance algorithm, which can increase the usage cycle of data FLASH by about 1 times and can
The E2PROM interface emulation uses an erase balancing algorithm to increase the data FLASH cycle by a factor of 1, enabling more than 100,000 erase cycles.

The E2PCTL controller also implements online erasure of the FLASH program space, which allows online self-writing via software.

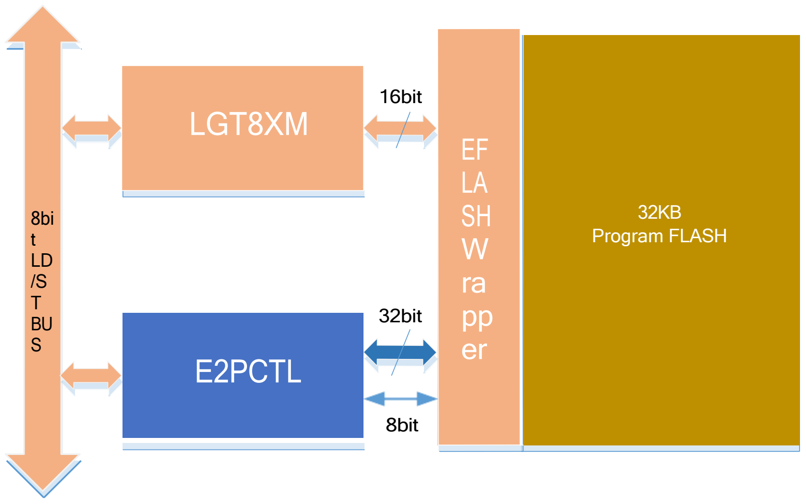
The firmware can be upgraded automaticallyAccess to the program FLASH program space via the FLASH controller supports only page erase (1024 bytes) and 32-bit wide read and write access.

LGT8F88D/168D E2PCTL Controller Structure Diagram



E2PCTL supports 8-bit and 32-bit read/write widths when accessing data FLASH space by emulating the E2PROM function. When accessing the program FLASH space, page erase and 32-bit data read/write are supported. The LGT8FX8P internal FLASH has a 32-bit minimum memory cell, so 32-bit access is recommended, especially for write operations. 32-bit access for read and write operations is not only efficient, but also helps to protect the erase life of the FLASH memory cell.

LGT8F328P E2PCTL Controller Structure Diagram



The LGT8F328P has no extra data FLASH inside. Therefore, the LGT8XM kernel shares 32K bytes of internal FLASH memory with the E2PCTL. The user can divide the 32K bytes of FLASH space into program space and data space as needed. The size of the emulated E2PROM space can be set by configuring the E2PCTL controller, which implements the emulated E2PROM logic using a page-swap mode with an algorithm in pages (1K bytes). So to simulate 1K bytes of E2PROM space, 2K bytes of FLASH space is required, and so on, to implement 4K bytes of E2PROM, 8K bytes of FLASH space is required. For details, please refer to the description of the E2PCTL algorithm implementation.

E2PCTL Data Register

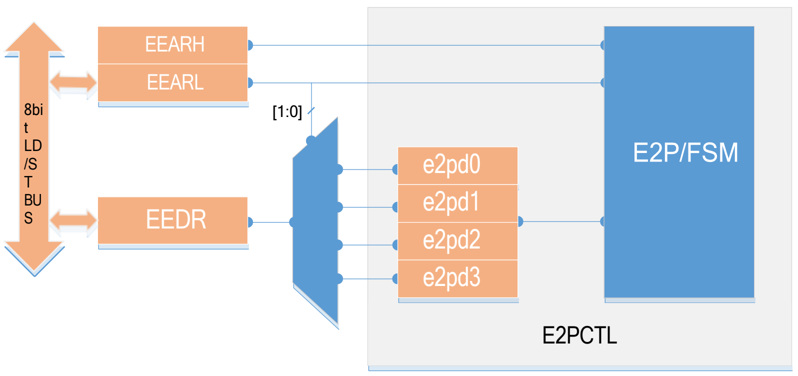
The E2PCTL controller has an internal 4-byte data cache (E2PD0~3), and this 4-byte cache makes up the final access

32-bit data interface in FLASH space.

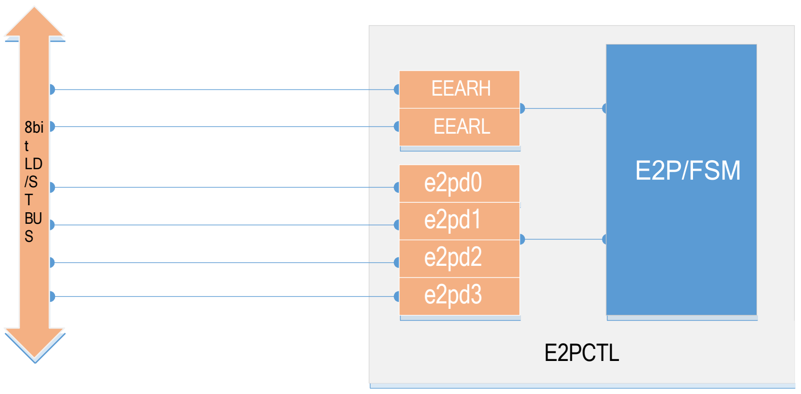
When the E2PCTL controller is operating in byte read/write mode, the EEDR serves as the interface for reading and writing byte data, and the E2PCTL adds the address information of EEARL[1:0] to load data into the correct data cache, and completes the other three bytes of data based on the data at the current FLASH destination address, eventually updating the combined complete 32-bit data into the FLASH.

When the E2PCTL is operating in 32-bit read/write mode, it is still possible to read and write a complete 32-bit data using the EEDR register as a common data interface and the EEARL[1:0] as the address addressing internal data cache. In addition, direct access to the registers in IO space (E0~3) is also possible using the data cache mapping.

Schematic of data access when E2PCTL is operating in 8-bit byte read/write mode.



Schematic of data access when E2PCTL is operating in 32-bit word read/write mode.



Byte mode is used for backward compatibility with the byte read/write mode of the LGT8FX8D. The LGT8FX8P has a 32-bit interface width for the built-in FLASH, so using 32-bit read/write mode will bring great benefits to read/write efficiency and FLASH erase life, so it is recommended to use 32-bit read/write mode.

E2PCTL Analog E2PROM Interface Algorithm

As we know, FLASH memory must be erased before it can be written,and the erase operation is done on a page by page basis.Therefore, in order to update one byte of data in the page, you need to first erase the whole page data, then update the target address data, and recover the other bytes of data in the page at the same time, the whole operation is not only time consuming, but also brings the risk of data loss due to power accident.

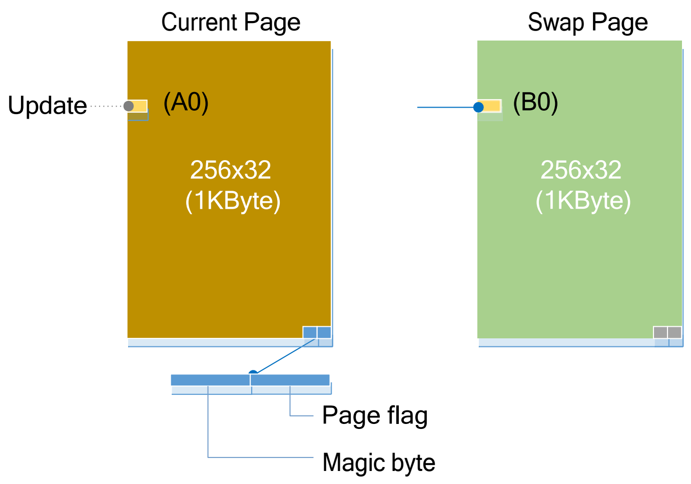
The E2PCTL uses an internal page swapping algorithm to emulate the E2PROM, and the page swapping algorithm mode ensures that when a page erase operation is performed, the original data is not lost due to unexpected circumstances such as power loss. The page swap algorithm also uses two page spaces

The alternate use of the interchange also increases the lifetime of the analog E2PROM space.

In terms of efficiency, the E2PCTL controller implements a continuous data update mode that reduces the repetitive erasing process caused by repeated data updates.

In terms of implementation, E2PCTL manages each page separately and occupies the last 2 bytes of a page as page status information. Therefore, when using E2PROM emulation space larger than 1K, users need to pay attention to the special handling of addresses across the 1K space. Because the last 2 bytes of each 1K space are reserved for E2PCTL, the user cannot read or write to these 2 bytes of space normally.

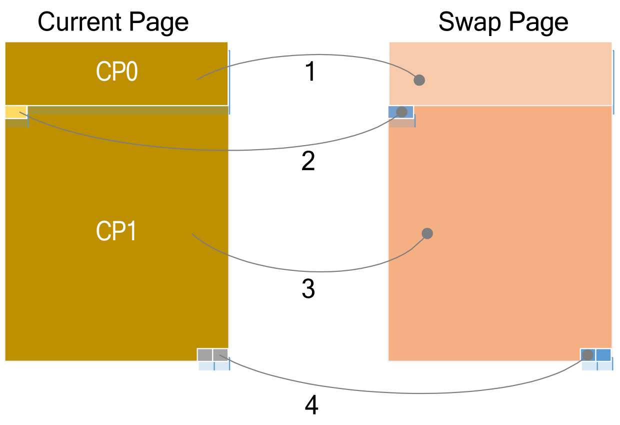
The following diagram illustrates the logic of the E2PCTL based page switching algorithm.



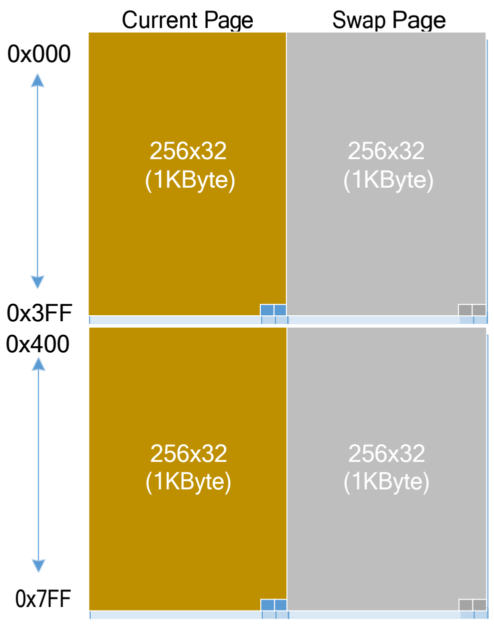
As shown in the figure, E2PCTL uses two pages internally to simulate a page-sized E2PROM space. E2PCTL uses the last 2 bytes of the page to store page information. When we need to update a byte in the page, such as byte A0 in the figure above. First, we do not erase the current page, but the swap page. Then the current page is divided into 3 parts. First is the data before A0, we make this part of space CP0, then the data after A0, this part of space is CP1. E2PCTL will copy the data corresponding to CP0 to the corresponding address of the swap page according to the user configuration, then write the data that needs to be updated to the corresponding address of the swap page (B0), and finally the data of CP1 is copied to the swap page.

After completing the above operation, the data has been exchanged, but the page state has not been updated. Therefore if a power down or other abnormality occurs before that, this update operation, because it is not completed, the previous data is not destroyed, ensuring the integrity of the data. If everything goes well, E2PCTL will write the updated page state to the page information of the previous exchange page at the end of the CP1 exchange of data, enabling the current page replacement. Thereafter, the exchanged page becomes the current page.

The E2PCTL page exchange process is shown in the following diagram (1->2->3->4).



When the system is configured with an E2PROM emulation space larger than 1K, E2PCTL still implements the emulation algorithm for E2PROM space in the smallest unit of pages. For example, if the user configures a 2K E2PROM area, E2PCTL will actually take up 4 pages (4K) of space. A group of 2 pages is used to implement the simulation of a page size E2PROM space.



Note that the user-configured 2K bytes of E2PROM space is not contiguous, as the last 2 bytes of each page will be used to store page status information.

E2PCTL Continuous Programming Mode

Since updating via E2PCTL results in a page swap, the page swap process will erase the swap page, and the page erase is not only time consuming, but also increases the loss of FLASH life. Therefore, E2PCTL adds a continuous write mode. In sequential write mode, the user can continuously update the E2PROM area and only at the end of the sequential address will the page swap be performed, making sequential mode more efficient for applications that need to continuously update a block of data.

Continuous programming mode E2PCTL control register ECCR SWM bit is enabled. When the continuous mode is enabled, subsequent write operations will write data directly to the address corresponding to the swap page, and in SWM mode, the write operation will not perform a CP0/1 area data copy operation. Before writing the last byte, software disables the continuous mode via SWM and then executes the write, after which the E2PCTL performs a complete CP0/1 copy operation and updates the page status information.

E2PCTL Reads and writes FLASH program space

The E2PCTL controller enables read and write access to the program FLASH space. Unlike the analog E2PROM, access to the program FLASH space via E2PCTL requires complete software control. The steps are as follows.

1. To erase the target page, the target page needs to be erased first before updating the data, the page address is given through the EEAR register. Erase command control for FLASH pages, refer to the definition of the EECR register.
2. Write program FLASH space must be in 32-bit minimum units. Setting data via E2PD0~3.
3. the destination address is given by t h e EEAR register and the address EEAR[1:0] will be ignored.

The E2PCTL read/write program FLASH space enables the In-line Program Update (IAP) function, which is useful in some applications where application data needs to be updated in the field and where product custom updates need to be provided.

E2PCTL Interface Operating Procedure

The E2PCTL controller works mainly through four registers: E2PCTL control status r e g i s t e r s EECR and ECCR; data registers EEDR (E2PD0~E2PD3) and address registers EEAR (EEARL/EEARH).

T h e ECCR register is used to set the operating status of the E2PCTL, most of the status needs to be set before the E2PCTL operates, this process is usually implemented during the system initialization. the SWM bit in t h e ECCR register is used to enable the continuous write mode, this control bit needs to be set during the implementation of continuous write operation.

The EECR register is used to control the select operation type for selecting operation commands, such as set read and erase commands.

EEDR registers for 8-bit byte mode interface and E2PD0 to 3 for 32-bit mode read and write operations.

T h e EEAR register is used to set the target address for reads and writes, and also to set the page address for page erase operations. The page address is aligned in page bits, and the size of a page is 1K bytes.

Access to *FLASH* program space via the *E2PCTL* interface.

The E2PCTL interface enables reading, writing and erasing of the FLASH program space. Read and write to FLASH space is only supported for 32-bit access width. Erase operations are in page bits, 1K bytes per page size (256x32).

The E2PCTL does not support sequential mode for writing FLASH program space, the user needs to complete the write operation in sequence. The following is the flow of erasing FLASH program space.

1. Program *FLASH* Page Erase Operation
   - Set EEAR[14:0] as the address of the target page to be erased, with a program FLASH page size of 1K bytes.
     Therefore EEAR[14:10] will be used as the page address and EEAR[9:0] is set to 0
   - Set EEPM[3:0] = 1X01, where EEPM[2] can be set to 0 or 1
   - Set EEMPE = 1, while EEPE = 0
   - Set EEPE = 1 in four cycles to start t h e program FLASH erase process

2. Program *FLASH* Programming Operations
   - Write E2PD0~3 to prepare 32-bit programming data
   - Set the EEAR as the destination address, where the address is 4-byte aligned
   - Set EEPM[3:0] = 1X10, where EEPM[2] can be set to 0 or 1

- Set EEMPE = 1, while EEPE = 0
- Set EEPE = 1 in four cycles to start the FLASH programming process

Access to the *E2PROM* analog space via the *E2PCTL* interface.

The E2PCTL controller provides logical access to the data FLASH space through the analog E2PROM interface. The analog E2PROM supports read and write access to 8-bit,16-bit, and 32-bit data widths. 8-bit byte mode provides better compatibility with the E2PROM interface. The 32-bit mode is the recommended read/write mode because it improves storage efficiency and FLASH lifetime. the E2PROM analog interface supports sequential read/write mode, which is recommended for data applications that require multiple sequential address updates at once.

For the LGT8F88P/168P, the data FLASH is a separate memory space. There is no need to configure and enable the FLASH data space through the ECCR register. LGT8F328P does not have a separate data FLASH space, and the data FLASH shares 32K bytes of FLASH space with the program FLASH. It is necessary to enable the data FLASH partition function through the ECCR register and configure the size of the data FLASH through the ECS[1:0] bits of the ECCR registerAfter the configuration takes effect,other usage is the same as LGT8F88P/168P. When the FLASH controller implements the E2PROM interface, it has internally implemented the automatic erasure of the data FLASH when necessary. logic, so the EPROM erase command is optional and this command is only used when the user needs to perform an erase alone.

The EECR registers control the erase/write timing of the FLASH, including the program FLASH and E2PROM, and the specific operation type needs to be set by EEPME and EEPM[3:0] of the EECR registers. The E2PROM read operation is relatively simple, after setting the target address and mode, the write EERE bit will read the 32-bit data corresponding to the target address into the FLASH controller, and the user can read the byte of interest through the EEDR register. The FLASH controller does not implement a read operation to the program FLASH space, the user can easily read it using LPM or through the program FLASH using the LD/LDD/LDS instruction at the address of the data unified mapping space.

1. 8-Bit Mode, Programmed *E2PROM*
   - Set the destination address to the EEARH/L register
   - Set new data to EEDR register
   - Set EEPM[3:1] = 000, EEPM[0] can be set to 0 or 1
   - Set EEMPE = 1, while EEPE = 0
   - Set EEPE = 1 in four cycles

   When setup is complete, the FLASH controller will initiate a programming operation, during which the CPU will remain at the current instruction address until the operation is completeDuring the programming process, if the data FLASH needs to be erased, the FLASH controller will automatically start the erase process.

2. 32-Bit Mode, Programmed *E2PROM*
   - Prepare 32-bit data via E2PD0~3
   - Set the destination address to the EEARH/L register. Note that this is a byte-aligned address, and the FLASH controller uses EEAR[15:2] as the address to access the FLASH.
   - Set EEPM[3:1] = 010, EEPM[0] can be set to 0 or 1
   - Set EEMPE = 1, while EEPE = 0
   - Set EEPE = 1 in four cycles

3. 8-bit Mode, Read *E2PROM*
   - Set the destination address to the EEARH/L register
   - Set EEPM[3:1] = 000
   - Set EERE = 1 to initiate E2PROM read operation

- Wait 2 cycles (perform two NOP operations)
- The data corresponding to the destination address is updated to the EEDR register

4. *32-Bit Mode, Read E2PROM*
   - Set EEARH/L as the destination address, address is 4-byte aligned
   - Set EEPM[3:1] = 010 to enable 32-bit interface mode
   - Set EERE = 1 to start E2PROM read operation
   - Wait 2 system clock cycles (execute two NOP instructions)

E2PCTL access emulates E2PROM space and supports continuous programming mode. Continuous access mode is very efficient for applications that require one block of data to be updated at a time, and also contributes to the longevity of the FLASH. The sequential programming mode only supports 32-bit wide data programming operations.

Continuous access mode is enabled through the SWM bit in the ECCR register. after SWM is enabled, the next operations to write analog E2PROM space through the E2PCTL are in continuous programming mode. In continuous programming mode, the E2PCTL controller automatically handles the page change based on the data in the target address. However, if a page swap occurs during continuous programming mode, the controller does not automatically swap the data in the CP0/1 area during continuous programming, nor does it update the page information.

When continuous programming reaches the last operation, turn off the continuous programming mode by clearing the SWM bit, and then start the last programming operation in the non-SWM mode. After programming, the E2PCTL will automatically copy the data in the CP0/1 area to the swap page and update the information of the swap page to make it the current valid page, thus completing the whole continuous programming operation.

5. *Continuous programming mode operation procedure.*
   1. Configure the size of the data FLASH via ECCR and enable the SWM bit
   2. Emulating E2PROM regions using 32-bit mode programming
   3. If this is not the last operation, go back to step 2 to continue programming the next data
   4. If the last programming is reached, first disable the continuous programming mode via SWM and then complete the last programming using the procedure in step 2

E2PCTL Efficient FLASH Data Management

In addition to the continuous programming mode, the E2PCTL controller also provides independent control of the data exchange copy during page exchange through the CP0/1 bits of the ECCR registers, which are used to control the exchange of data in the CP0/1 area of the current page during page exchange, respectively. By clearing the CP0/1 bit, the data in the corresponding area of the current page will not be exchanged during the page exchange. This section provides an efficient management method that will take advantage of this feature.

The most time-consuming operation in the FLASH data update process occurs in the swap page erase process. Therefore, we can address a data management method that minimizes the number of page erasures, both to improve programming efficiency and to reduce lifetime loss.

Here we provide a reference algorithm for data block-based data management applications.

1. Assume that the user data is only a complete block of data with an integer multiple of the block size of 4 bytes.
2. Each data update will update a complete block of data
3. In addition to the user data, the data block information also needs to hold a block management information

These three conditions allow us to take full advantage of E2PCTL's continuous programming mode and automatic page swapping mechanism to achieve an efficient approach to FLASH data management.

Since the data is updated in one block of the same size each time and the address information pointing to the next block of data is stored in each data structure, we can program the FLASH in address order each time we update the data without having to do a CP0/1 data copy. Also, since the data is updated to an erased area each time, no page erasure occurs.

When the last piece of data is written, the next data area pointed to by its structure information returns to the start address of the page. When another data write operation occurs thereafter, the E2PCTL will initiate a page erase process and update the currently active page.

Protective measures for FLASH operation

If the VCC voltage is low, the FLASH erase operation may cause an error because the voltage is too low.

Erase operation errors of FLASH/data at low voltage can be caused by two reasons. First, normal FLASH erase operations require a minimum operating voltage below which the operation will fail and result in data errors. The second reason is that the kernel runs at a certain frequency, which also requires a minimum voltage, and when it falls below this voltage, it will cause an error in the execution of the instruction, thus making the FLASH operation incorrect.

Similar problems can be avoided by the following simple methods.

Put the system into a reset state when the supply voltage is low. This can be achieved by configuring the internal low voltage detection circuit (VDT). If the VDT detects that the current operating voltage is below the set threshold, the VDT will output a reset signal. If the threshold value of the VDT does not meet the needs of the application, consider adding an external reset circuit.

Register Description

FLASH Address Register - EEARH/EEARL

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 lower 8 bits.
[14:8]	EEARH	EFLASH/E2PROM access address high 7 bits
[15]	-	keep sth. unused

EEAR[14:2] is used to access the entire program space aligned in 4 bytes when using the E2PCTL controller to access the program FLASH area.EEAR[1:0] is used only when accessing the data register EEDR. Refer to the description of the EEDR data registers below for details.The E2PCTL controller supports 8/16/32-bit modes; in either mode, the EEAR here is addressed in byte alignment.

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	E2PCTL Data Register
	E2PD0	For accessing the lowest byte in 16/32-bit mode

FLASH Data Register - E2PD1

E2PD1 - E2PCTL Data Register 1

E2PD1: 0x5A	Default value: 0x00
bits	E2PD1[7:0]

R/W	R/W	
Bit Definition		
[7:0]	E2PD1	High 8 bits for storing 16-bit data in 16-bit mode High 8 bits for storing low 16-bit data in 32-bit mode

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	Lower 8 bits for storing high 16-bit data in 32-bit mode

FLASH Data Register - E2PD3

E2PD3 - FLASH Data Register 3		
E2PD3: 0x5C		Default value: 0x00
Bits	E2PD3[7:0]	
R/W	R/W	
Bit Definition		
[7:0]	E2PD3	High 8 bits for storing high 16-bit data in 32-bit mode

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	ECCR Write Enable Control Before modifying the ECCR, you must first write 1 to WEN and then update the contents of the ECCR register within 6 system cycles
[6]	EEN	E2PROM enable, valid only for LGT8F328P 1: Enabling E2PROM emulation will reserve some space from 32KFLASH 0: E2PROM emulation disabled, 32K FLASH all for program space
[5]	ERN	Write 1 will reset the E2PCTL controller
[4]	SWM	Continuous write mode for analog E2PROM controller operation
[3]	CP1	Page Switching CP1 Area Enable Control
[2]	CP0	Page Switching CP0 Area Enable Control

[1:0]	ECS[1:0]	E2PROM Space Configuration 00: 1KB E2PROM, 30KB program FLASH 01: 2KB E2PROM, 28KB program FLASH

		10: 4KB E2PROM, 24KB program FLASH 11: 8KB E2PROM, 16KB program FLASH

FLASH Access Control Register - EECR

EECR - FLASH/E2PROM control register								
EECR: 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]	EFLASH/EPROM access mode control bit				
		[3]	[2]	[1]	[0]	Model description
		0	0	0	control	8-bit mode read/write E2PROM (default)
		0	0	1	control	16-bit mode read/write E2PROM
		0	1	0	control	32-bit mode read/write E2PROM
		1	control	0	0	E2PROM erase (optional operation)
		1	control	0	1	Program FLASH Erase (Page Erase)
		1	control	1	0	Program FLASH Programming
		1	control	1	1	Resetting FLASH/E2PROM Controller
[3]	EERIE	FLASH/E2PROM Ready interrupt enable control. Write 1 enable, write 0 disable. When After the EEPE is automatically cleared by hardware, the E2PROM ready interrupt is active. In the EPROM This interrupt will not be generated during the operation				
[2]	EEMPE	FLASH/E2PROM programming operation enable control bit EEMPE is used to control whether EEPE is valid or not.When EEMPE is set to 1 and EEPE is set to 0 at the same time, setting EEPE to 1 will initiate the programming operation in the following four cycles. Otherwise, the programming operation is invalid. After four cycles, EEMPE is automatically cleared to zero				
[1]	EEPE	FLASH/E2PROM programming operation enable bit				
[0]	EERE	E2PROM read enable bit, data will be valid after two system cycles				

General Purpose I/O Register - GPIOR2

GPIOR2 - General Purpose 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-purpose I/O register 2 for storing user-defined data

General Purpose I/O Register - GPIOR1

GPIOR1 - General Purpose I/O Register 1		
GPIOR1: 0x2A (0x4A)		Default value: 0x00
Bits	GPIOR1[7:0]	
R/W	R/W	
initial value	0x00	
Bit Definition		
[7:0]	GPIOR1	General-purpose I/O register 1 for storing user-defined data

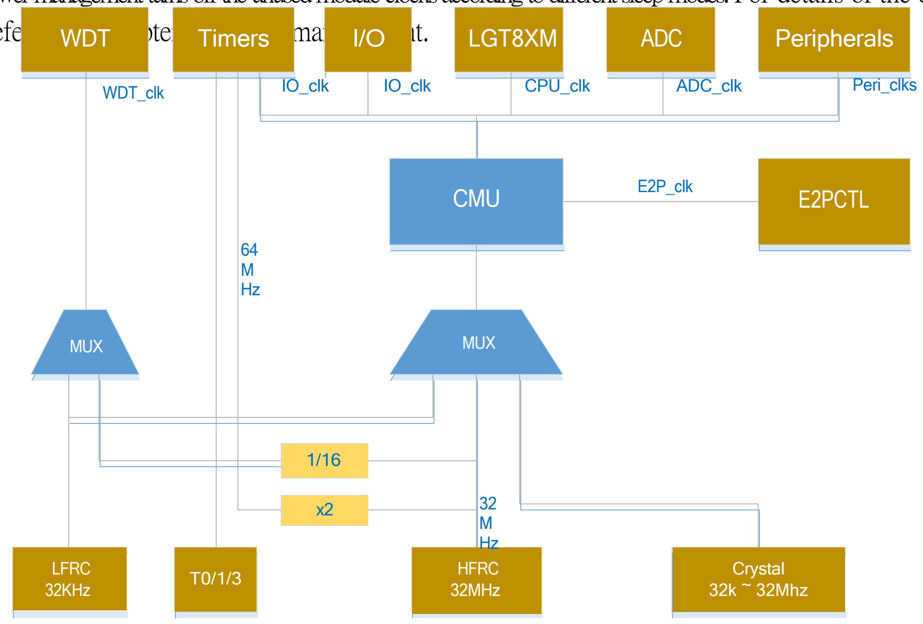
General Purpose I/O Register - GPIOR0

GPIOR0 - General Purpose I/O Register 0		
GPIOR0: 0x1E (0x3E)		Default value: 0x00
Bits	GPIOR0[7:0]	
R/W	R/W	
initial value	0x00	
Bit Definition		
[7:0]	GPIOR0	General-purpose I/O register 0 for storing user-defined data

System Clocking and Configuration

System Clock Distribution

The LGT8FX8P supports multiple clock inputs. The system can operate on three main clock sources, namely the internal 32KHz
Calibratable RC oscillator, internal 32MHz calibratable RC oscillator and external 400KHz ~ 32MHz crystal input.
The following figure shows the distribution of the LGT8FX8P clocking system. The CMU is the center of the entire clock management, responsible for the system clock division, generating independent clocks for different modules and controlling the clocks, etc. In general applications, not all clocks are working at the same time. In general applications, not all the clocks work at the same time. To reduce system power consumption, the system power management turns off the unused module clocks according to different sleep modes. For details of the operation, please refe te ma t.



CPU_clk

Used to drive the LGT8XM kernel and the operation of the SRAM. For example, driving general-purpose work registers, status registers, etc. After the CPU clock is stopped, the kernel will not continue to execute instructions and perform calculations. After the system executes the SLEEP instruction and goes into sleep mode, the kernel clock will be turned off.

Peri_clk

The IO clock is also used to drive external interrupt modules. When a peripheral clock is stopped due to hibernation, some peripheral parts of the system can be used to wake up the system to operate in independent clock or asynchronous mode. For example, the address recognition function of TWI can wake up most of the hibernation modes, when the address recognition part is working in asynchronous mode.

E2P_clk

The E2P_clk clock is used to generate the FLASH interface access timing.E2P_clk generates the timing for accessing the E2PCTL access FLASH interface.E2P_clk is fixed at 32 divisions (1MHz) from the internal 32MHz HFRC oscillator. If the user needs to use the E2PCTL module to read or write to the internal program FLASH or data FLASH space, the internal 32MHz oscillator needs to be enabled in advance.

Asy_clk

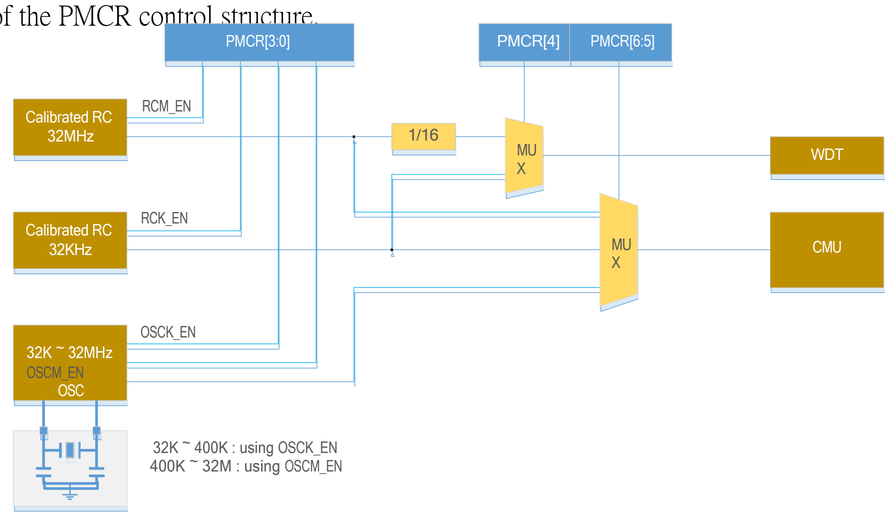
Asynchronous timer clock. The timer/counter can be driven directly using an external clock or a crystal (32.768K). This independent clock mode allows the timer to remain running while the system is processing sleep mode.

WDT_clk

Internal watchdog timer clock source can be configured to select either the internal 32KHz LFRC oscillator or 16 divisions (2MHz) from the internal 32MHz HFRC. When the system is powered up, the watchdog default clock source is the 32KHz LFRC oscillator.

Clock Source Selection

The LGT8FX8P supports 4 types of clock source inputs, and the user can use the PMCR register to enable control of the clock source and to complete the switching of the master clock. The following is a diagram of the PMCR control structure.



The LGT8FX8P internal OSC oscillator can operate in both high frequency and low frequency modes, and the user needs to control the internal OSC oscillator to operate in the correct mode according to the actual size of the external crystal. The same internal RC oscillator is also divided into high frequency and low frequency. the lowest 4 bits of the PMCR register are used to control these four clock sources. The control relationship is as follows.

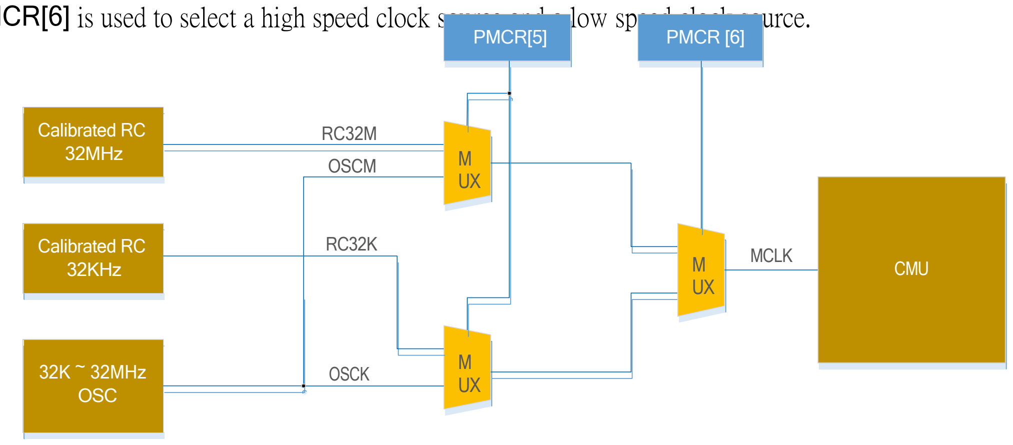
PMCR	Corresponding clock source
PMCR[0]	32MHz RC Enable Control, 1 Enable, 0 Off
PMCR [1]	32KHz RC Enable Control, 1 Enable, 0 Off
PMCR [2]	400K ~ 32MHz OSC Mode Enable, 1 Enable, 0 Disable
PMCR [3]	32K ~ 400K OSC Mode Enable, 1 Enable, 0 Disable

When the LGT8FX8P system is powered up, it uses 32MHz RC as the system clock source by default, and the core operates at 8 divisions of the clock source (4MHz). The user can change the default configuration by setting the PMCR register as well as the system prescaler register (CLKPR).

If the user needs to change the master clock source configuration, it is necessary to ensure that the switched clock source is in a stable operating state before switching the clock. Therefore, you need to enable the required clock source via PMCR[3:0] before switching the master clock source, and wait until the clock is stable before switching.

When the user switches the master clock to the external crystal, although the user enables the external crystal, it is not excluded that the crystal cannot be oscillated due to configuration error or crystal failure. If you switch to the external crystal at this time, the system will stop working after the switch. Therefore, from the system reliability consideration, it is recommended to open the watchdog timer to avoid such problems from the software design point of view.

After the clock source is enabled and waiting for stability, master clock can be switched via PMCR[6:5]. Where PMCR[5] is used to select whether it is an internal RC oscillator and an external crystal, and PMCR[6] is used to select a high speed clock source and a low speed clock source.



Master clock source selection.

PMCR [6]	PMCR [5]	master clock source
0	0	Internal 32MHz RC oscillator (system default)
0	1	External 400K ~ 32MHz High Speed Crystal
1	0	Internal 32KHz RC oscillator
1	1	External 32K ~ 400KHz Low Speed Crystal

Clock Source Control Timing

To protect the PMCR register from accidental modifications, modifications to the PMCR register need to be performed by strictly installing the specified timing sequence. The highest bit (PMCR[7]) of the PMCR register is used to implement timing control. The user must first set PMCR[7] to 1 before modifying the other bits of PMCR, and change the value of the other PMCR registers within 6 cycles of the set 1 operation. after 6 cycles, direct modification of PMCR will fail.

The following is an example of the recommended procedure for switching to an external high-speed crystal.

(1) Enabling the clock source
- Set PMCR[7] = 1
- Set PMCR[2] = 1 for six cycles to enable the external high-speed mode external crystal
- Wait for the external crystal to stabilize (waiting time varies depending on the crystal, generally us level waiting is sufficient)

(2) Switching the master clock source
- Set PMCR[7] = 1
- Set PMCR[6:5] = 01 for six cycles and the system will automatically switch the operating clock to the external crystal
- Perform several NOP operations to improve stability (optional operation)

[Note]: In the above operation of switching the master clock, make sure that the current system clock is working properly, and turn off the previous internal *RC* oscillator only after switching to the external crystal.

System clock prescaling control

The LGT8FX8P has an internal system clock prescaler that can be controlled through the Clock Prescaler Register (CLKPR). This feature can be used to reduce system power consumption when the system does not require very high processing power. The prescaler setting is valid for all clock sources supported by the system. Clock prescaling can affect the kernel execution clock as well as so synchronize peripherals.

When switching between different clock prescaler settings, the system clock prescaler ensures that no burrs are generated during the switching process and will already ensure that there are no intermediate states with excessive frequencies. The crossover switch is performed immediately, and the system clock switches to the new crossover clock after at most 2 to 3 current system clock cycles when the register change takes effect.

In order to avoid misuse of the clock division registers, modifications to CLKPR must also follow a special timing flow: the

- Set the clock prescaler change enable bit (CLKPCE) to 1, CLKPR other bits to 0
- Write the desired value to CLKPS in four cycles, while CLKPCE writes 0

Before changing the clock prescaler register, the interrupt function needs to be disabled to ensure that the write timing is complete. Refer to the Register Description section of this chapter for the specific definition of the master clock prescaler register CLKPR.

Internal *RC* oscillator calibration

The LGT8FX8P contains two internal calibratable RC oscillators, both of which are calibrated to within ±1% accuracy. One 32MHz RC is used by default for the system operating clock.

Before the LGT8FX8P is shipped, the internal 32MHz HFRC and 32KHz LFRC are calibrated and the calibration values are written to the system configuration information area. During system power saving, these calibration values will be read into the internal registers, and recalibration of the RC frequencies will be achieved through the registers.

The calibration registers are located in the IO address space and can be read and written by the user program. For applications with special requirements for frequency, the frequency output of the internal oscillator can be adjusted by modifying the calibration register. Modifying the calibration registers does not change the factory configuration information, and the calibration registers will be restored to the factory settings upon system re-powering or a user initiated configuration bit reload operation.

Register Definition

32MHz HFRC Oscillator Calibration Register - RCMCAL

RCMCAL - 32MHz HFRC Calibration Register		
RCMCAL: 0x66	Default: Factory Configured	
Bits	RCCAL[7:0]	
R/W	R/W	
Bit Definition		
[7:0]	RCCAL	When the system is powered up, the register values will be replaced by the RC calibration values in the system configuration information.

32KHz RC Oscillator Calibration Register - RCKCAL

RCKCAL - 32MHz RC Calibration Register		
RCKCAL: 0x67	Default: Factory setting	
Bits	RCKCAL [7:0]	
R/W	R/W	
Bit Definition		
[7:0]	RCKCAL	Write the calibration value to the RCKCAL register to complete the calibration of the 32KHz RC oscillator

Clock Source Management Register - PMCR

PMCR - Clock Source Management Register								
PMCR: 0xF2				Default value: 0x03				
Bits	PMCE	CLKFS/CLKSS	WCLKS	OSCKEN	OSCMEN	RCKEN	RCMEN	
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, 0 disable
[1]	RCKEN	Internal 32KHz RC oscillator enable control, 1 enable, 0 disable
[2]	OSCMEN	External HF crystal enable control, 1 enable, 0 disable
[3]	OSCKEN	External low frequency crystal enable control, 1 enable, 0 disable
[4]	WCLKS	WDT clock source selection. 0 - Selects 16 divisions of the internal 32MHz HFRC oscillator 1 - Internal 32KHz LFRC oscillator
[5]	CLKSS	Master clock source selection control, select the clock source type, refer to the clock source selection section
[6]	CLKFS	Master clock source frequency control, select the clock frequency type, refer to the clock source selection section
[7]	PMCE	PMCR register change enable control bit. This bit must be set first before changing the other positions of the PMCR, and then the values of the other bits must be set within four cycles.

Master Clock Prescaler Register - CLKPR

CLKPR - Master Clock Prescaler 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 Bit				
		PS3	PS2	PS1	PS0	Crossover parameters
		0	0	0	0	1

		0	0	0	1	2
		0	0	1	0	4
		0	*0*	*1*	*1*	*8 (default configuration)*

		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				retain
[4]	-	keep sth. unused				
[5]	CKOEN0	Sets whether the system clock is output on the PB0 pin				
[6]	CKOEN1	Sets whether the system clock is output on the PE5 pin				
[7]	WCE	Clock Prescaler Change Clock Control Before changing other bits in the CLKPR register, CKWEN must first be set to 1 individually, and then the other bits must be set for the next four system cycles. CKWEN is automatically cleared to zero at the end of four cycles.				

Power Management

summarize

The LGT8FX8P offers a wide variety of hibernation modes and module controllers, allowing the user to achieve the optimal low-power configuration depending on the application.

The LGT8FX8P does not automatically turn off analog function modules such as ADC, DAC, comparator (AC), low voltage reset module (LVD), etc. when it enters hibernation mode. Software needs to turn off unwanted analog functions before entering hibernation and restore the correct state after the system wakes up, according to the application requirements.

The LGT8FX8P supports a variety of sleep modes, including a dedicated ADC noise cancellation mode to eliminate interference from the digital portion of the ADC power supply during ADC conversion. Otherwise, all others are power control modes, divided into five.

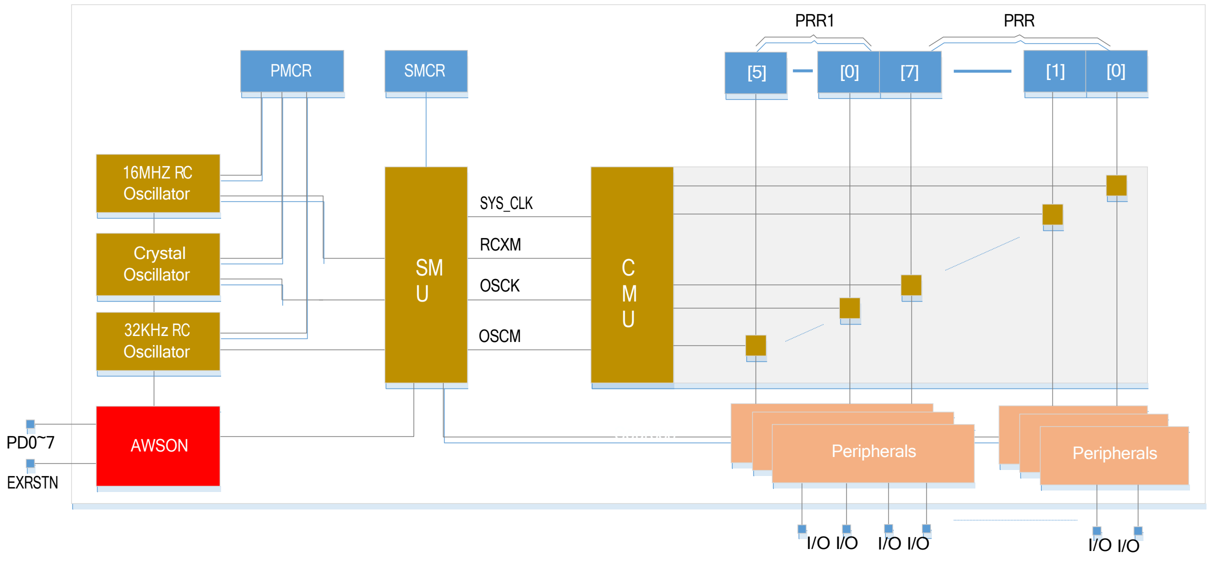
dormant mode	Function description
Idle mode (IDLE)	Only the kernel clock is turned off, other peripheral modules work normally, and all valid interrupt sources can wake up the kernel
Power saving mode (Save)	Same as DPS0 mode, Save mode is compatible with LGT8FX8D
Power down mode (DPS0)	As with Save mode, supported wake-up sources include. ● All pin level changes ● Watchdog timed wake-up ● TMR2 wake-up in asynchronous mode
Power down mode (DPS1)	Turn off all internal and external oscillators，supported wake-up sources include. ● External level change on all pins ● External interrupt 0/1 ● Watchdog timer operating at 32K LFRC
Power down mode (DPS2)	Power off the kernel，lowest power mode, supported wakeup sources include. ● External reset ● PORTD pin level change ● LPRC Timed Wakeup (128ms/256ms/512ms/1s) Note that the wake-up process from DPS2 is the same as a power-on reset

The LGT8FX8P supports Deep Sleep DPS2, in which the system internal LDO is powered down,the core registers, all peripheral controllers and SRAM are powered down, and the data in them will not be maintained, the FLASH memory cells are also powered down, so the DPS2 mode can achieve the minimum power consumption of the system. The power-down mode can be woken up by a change in the level of the port D (PORTD) pin, or by selecting a 5-step timed wake-up.The timer of DPS2 for wake-up has an accuracy of about 15% because it does not support calibration, and is only suitable for low accuracy timed wake-up applications.

When the system wakes up from DPS2 mode, it will first turn on the LDO, which is the same process as the power-up process. The chip will perform the full power-on reset boot process, load the configuration information, and then run the program from the address pointed to by the reset vector.

Modes other than DPS2 do not turn off internal power, during hibernation, all register information as well as RAM None of the data is lost. After waking up, the kernel continues execution from the last instruction before hibernation.

System power management schematic.



As shown above, the LGT8FX8P controls the power consumption of the whole system mainly through the Sleep Mode Controller (SMU) and the Clock Management Unit (CMU). In terms of power saving level, we can classify the power consumption into 4 levels:

The first level is to control the module operating clock through the PRR register, saving the dynamic power consumption of the system operation by turning off the clock of the unused module. In general, the power savings that can be achieved at this level are not significant.

The second stage is done by switching the main clock source to a low frequency clock and turning off the unused clock source modules as well as other analog modules, this mode basically gives very substantial system operating power and sleep power.

The third level is by putting the system into power-down mode (DPS1). The DPS1 mode allows the LGT8FX8P to obtain extreme standby power consumption, and after waking up from power-down mode, the software can read the state before reset through the MCUSR register.

The fourth level is the power down mode (DPS2), this mode will turn off the kernel power and can achieve the lowest system power consumption. Because the kernel power is turned off, all data information will be lost in this mode. A power-on reset process is executed immediately after wake-up and the system starts running again from the reset vector.

AWSON Power Management

Power-down mode DPS2 is a new power consumption mode compared to the LGT8FX8D.DPS2 mode is used for applications with higher requirements for sleep power consumption. When entering DPS2 mode, the system maintains only one static module (AWSON) in operation and all other circuits are in a complete power-down state.

The AWSON module is dedicated to hibernation and wake-up control in DPS2 mode and consists of IO wake-up control logic and a low-power LPRC. Software control of AWSON is achieved through the IOCWK register and the DPS2R register.

The IOCWK register is used to control the wake-up function for PD0~7 level changes.The DPS2R register is used to control the DPS2 mode as well as the LPRC function mode. Refer to the Register Definition section at the end of this section for specific information.

Before using DPS2 mode, the software sets IOCWK to enable the desired wake-up IO, or enables LPRC and configures the timed wake-up period through the DPS2R register, and then enables DPS2 mode through the DPS2EN bit of the DPS2R register. After the setup is completed, the software needs to set the DPS2 sleep mode through the SMCR register, and then execute the SLEEP instruction to enter sleep.

Sleep mode and wake-up source

The LGT8FX8P supports five hibernation modes, and the user can select the appropriate hibernation mode according to the application requirements.The SMCR register contains the hibernation mode control settings, and the core enters hibernation mode after the SLEEP instruction is executed. To obtain a more optimal hibernation power consumption, it is recommended to turn off all unused clocks and analog modules before the core enters hibernation mode. However, it should be noted that some wake-up sources require an operating clock for their generation, so if you need to use such wake-up sources, keep the relevant clock sources operating.

Hibernation mode and wake-up mode.

dormant mode	Effective clock				wake-up call source							
	kernel clock	Peripheral Clock	clocks	asynchronous clock	Pin level change	External Interrupts	address matching	disruptions	End of conversion	Watchdog Overflow	Peripheral Interrupts	Level change
Idle mode (IDLE)		X	X	X	X	X	X	X	X	X	X	X
ADC Noise Suppression			X	X	X	X	X	X	X	X		X
Power saving mode (SAVE)				X	X	X	X	X		X		X
Power down mode (DPS0) (With RC32K)				X	X	X		X		X		X
Power down mode (DPS1) (Without RC32K)				X	X	X		X				X
Power down mode (DPS2) (Without LDO)												X

If you need to enter the above 5 sleep modes, the SE bit in SMCR must be set to 1 to enable the sleep mode control. SM0/1/2 in the SMCR is used to select a different hibernation mode. Please refer to the following description for specific information.

While the MCU is in sleep mode, if the wake-up source is valid, the MCU will be woken up after 4 cycles to continue executing instructions. If the interrupt remains valid, the interrupt will also respond immediately and enter the interrupt service subroutine. If a system reset occurs in SLEEP mode, the MCU will also be woken up and execution will begin from the reset vector.

When the MCU is in Power/Off mode, the system can be woken up via external interrupt INT0/1. After waking up, the MCU will be woken up from

The execution continues in the position before sleep.

Idle mode *(IDLE)*

When SM2...0 is set to 000 and the SLEEP instruction is executed, the MCU enters IDLE mode, which will turn off the core operating clock and all other peripherals will work normally.

The IDLE mode can be woken up by external interrupts as well as internal interrupts, etc. It is recommended to turn off the comparator and ADC if they are not needed as wake-up sources.

IDLE mode does not get a significant power reduction because it only turns off the clock for the kernel to runIDLE mode also stops the kernel from executing and fetching instructions, so it can reduce the power consumption of the internal program FLASH operation.

However, IDLE mode has a more flexible wake-up mode, allowing users to obtain more optimal operating power by reducing the system master clock and turning off unneeded modules.

ADC Noise Suppression Mode

When SM2...0 is set to 001 and the SLEEP instruction is executed, the MCU enters ADC noise suppression mode. In this mode, the core and most of the peripherals will stop working, and the ADC, external interrupts, TWI address matching, WDT and Timer/Counter 2 working in asynchronous clock mode will work normally.

ADC noise all the time mode is mainly used to provide a good working environment for ADC conversion.Reduce the high frequency interference of the digital module to the analog conversion. After entering this mode, the ADC will automatically start the sample conversion, and after the converted data is saved to the ADC data register, the ADC end-of-conversion interrupt wakes up the MCU from the ADC noise mode.

Power saving mode *(Save)*

When SM2...0 is set to 010 and the SLEEP instruction is executed, the MCU enters the Save mode. In this mode, the system will turn off the operating clocks of all modules. External interrupts, TWI address matching, and WDTs operating in independent clock source mode can generate wake-up signals in this mode because the operating clocks of all modules are turned off.

This mode turns off all modules except the main clock source. To achieve more optimal power consumption, it is recommended to switch the system master clock to internal 32K RC or external 32KHz low frequency crystal before entering this mode, and then turn off all unused clock sources and analog modules.

Power down mode *DPS0*

When SM[2:0] is set to 110 and the SLEEP instruction is executed, the MCU will enter into DPS0 mode. After entering DPS0, all clock sources are turned off except for the internal 32KHz RC. This mode can be woken up by external interrupt INT0/1; if the interrupt function of WDT is enabled, the timed wake-up can also be realized by WDT.

Power down mode *DPS1*

When SM[2:0] is set to 011 and the SLEEP instruction is executed, the MCU will enter into DPS1 mode. After entering DPS1, all clock sources of the system are turned off. This mode can use the level change of IO, watchdog wake-up.

Power down mode *DPS2*

Set SM[2:0] to 111 and enable the AWSON module via DPS2EN in the DPSR2 register, which will enter the DPS2 mode after executing the SLEEP instruction. After entering DPS2 mode, the system turns off the kernel power. So the registers and RAM data will be lost. The wake-up process from DSP2 is the same as the power-on reset process.

In DPS2 mode, since the kernel voltage is turned off and the register information is lost, the control state of the port will all revert to the input state, and all IO output drivers and pull-up controls will be turned off.

FLASH power control and fast wake-up

When the system is in SLEEP mode, the core will not continue executing instructions and can optionally power down the FLASH for lower standby power consumption. This feature can be controlled by the FPDEN bit of the MCUCR register.

In power-down mode, the system can be woken up using an external interrupt or WDT. To filter out possible interference from external signals, the internal wake-up circuit contains a configurable filter circuit that allows the user to select the appropriate filter width as required. The configuration of the filter circuit can be implemented through the FWKPEN of MCUCR register.

MCUCR [FWKPEN] Filter width control.

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

Register Description

Sleep Mode Control Register - SMCR

SMCR - Sleep Mode Control Register							
SMCR: 0x33(0x53)			Default value: 0x00				
Bits			SM2	SM1	SM0	SE	
R/W	-		R/W	R/W	R/W	R/W	
Bit Definition							
[0]	SE	The SE bit protects the system from accidental entry into hibernation mode. It is recommended to clear the SE bit immediately after waking up.					
[3:1]	SM	Hibernation Mode Selection					
		SM2	SM1	SM0	Model description		
		0	0	0	IDLE Mode		
		0	0	1	ADC Noise Suppression Mode		
		0	1	0	Save Mode		
		0	1	1	DPS1 mode		
		1	1	0	DPS0 mode		
		1	1	1	DPS2 mode		
		Others			keep sth. unused		
[7:4]	-	keep sth. unused					

Power Save Control Register - PRR

PRR - Power Save Control Register								
PRR: 0x64					Default value: 0x00			
PRR	PRTWI	PRTIM2	PRTIM0	-	PRTIM1	PRSPI	PRUART0	PRADC
R/W	R/W	R/W	R/W	-	R/W	R/W	R/W	R/W
Bit Definition								
[0]	PRADC	Set to 1 to turn off the ADC controller clock						
[1]	PRUART0	Set to 1 to turn off the clock for the USART0 module						
[2]	PRSPI	Set to 1 to turn off the SPI module's clock						
[3]	PRTIM1	Set to 1 to turn off the clock for Timer/Counter 1						
-	-	keep sth. unused						
[5]	PRTIM0	Set to 1 to turn off the clock for timer/counter 0						
[6]	PRTIM2	Set to 1 to turn off the clock for Timer/Counter 2						
[7]	PRTWI	Set to 1 to turn off the TWI module's clock						

Power Save Control Register - PRR1

PRR1 - Power Save Control Register 1								
PRR1: 0x65				Default value: 0x00				
PRR1			PRWDT	-	PRTIM3	PREFL	PRPCI	-
R/W			R/W	-	R/W	R/W	R/W	-
Bit Definition								
[0]	-	keep sth. unused						
[1]	PRPCI	Set to 1 to turn off external pin changes and the external interrupt module clock						
[2]	PREFL	Set to 1 to turn off the FLASH controller interface clock						
[3]	PRTIM3	Set to 1 to turn off the TMR3 controller's clock						
[4]	-	keep sth. unused						
[5]	PRWDT	Set to 1 to turn off the WDT counter clock						
[7:6]	-	keep sth. unused						

MCU Control Register - MCUCR

MCUCR - MCU Control Register								
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 bitwhich needs to be set first before updating the MCUCR and then completing the update of the MCUCR register within 6 cycles						
[1]	IVSEL	Interrupt vector select bit, after this position 1, the interrupt vector address will be set according to IVBASE The value of the register is mapped to the new address						
[2]	IFAIL	the system configuration bit load failure flag bit. 0 = Configuration information checks out 1 = Configuration information failed to load						
[3]	IRLD	Write 1 will reload the system configuration information						
[4]	PUD	Global pull-up ban bit 0 = Enables global pull-up control 1 = Turn off pull-up resistors for all IOs						
[5]	EXRFD	External reset filter disable bit 0 = (190us) digital filter with external reset enabled 1 = Digital filter circuit with external reset disabled						
[6]	FPDEN	Flash Power/down Enable Control 0: FLASH remains powered on after system SLEEP 1: FLASH power failure after system SLEEP						

[7]	FWKEN	Fast Wake-Up Mode Enable Control, valid for Power/Off mode only 0: 260us filter delay 1: 32us filter delay

PD Group Level Change Wake-Up Control Register - IOCWK

IOCWK - PD group level change wake-up control 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	Set the corresponding bit to 1 to enable the pin level change wake-up function of the PD group IO						

DPS2 Mode Control Register - DPS2R

DPS2R - DPS2 Mode Control Register								
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 timed wake-up setting. 00 = 128ms 01 = 256ms 10 = 512ms 11 = 1s						
[2]	LPRCE	LPRC Enable Control 0 = Disable LPRC timer 1 = Enable LPRC timer						
[3]	DPS2E	DPS2 mode enable control bit 0 = Disable DPS2 mode 1 = Enables DPS2 mode						
[7:4]	-	retain						

System Control and Reset

summarize

After a system reset, all I/O registers are set to their initial values and program execution begins at the reset vector. an RJMP - relative jump instruction must be used to jump to the reset handler at the interrupt vector address of the LGT8FX8P. If the program does not use the interrupt, the interrupt vector is not enabled, and the interrupt vector area can be used to store the user's program code.

Immediately after the reset is active, all I/O ports enter their initial states. Most I/Os are initialized to the input and turn off the internal pull-up resistor. I/Os that have analog input functions are also initialized to digital I/O functions.

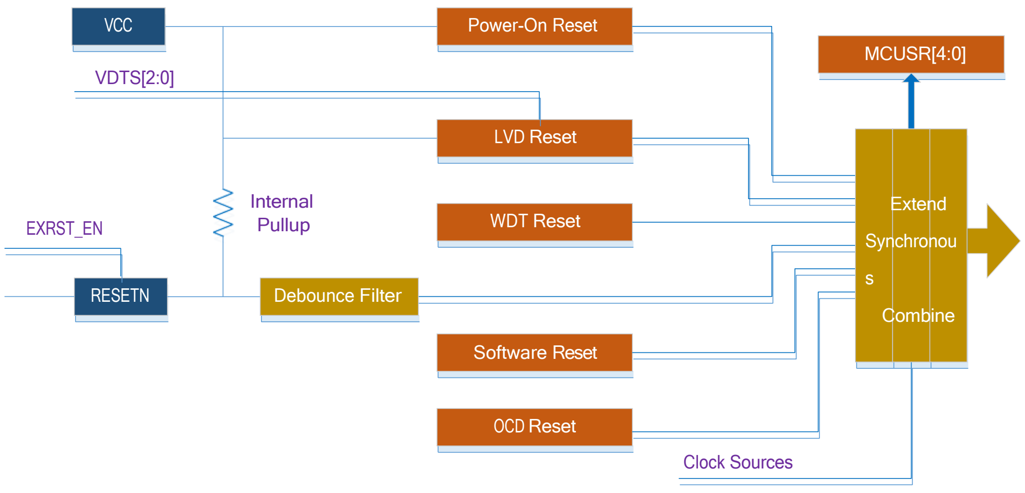
When the reset becomes invalid, the internal timing counter of the LGT8FX8P starts and is used to spread the reset. The width of the spread reset signal is used to ensure that the power supply and the clock and other modules in the system are brought to a stable state.

reset source

The LGT8FX8P supports a total of six reset sources.

- Power-on Reset: Power-on reset is active when the system is operating at a low voltage to the reset threshold of the internal POR module.
- External Reset: A low pulse of a certain width on the external reset pin of the chip, external reset is valid.
- Watchdog Reset: After enabling the watchdog module, the system will reset if the watchdog timer times out.
- Low voltage reset: The LGT8FX8P has an internal low voltage detection module (LVD), when the system operating power is below LVD
  The MCU will also be reset when the reset threshold is set.
- Software Reset: The LGT8FX8P has a dedicated internal software-triggered reset register, which allows the user to reset the MCU at any time.
- OCD Reset: The OCD reset is issued by the debugger module and is used to reset

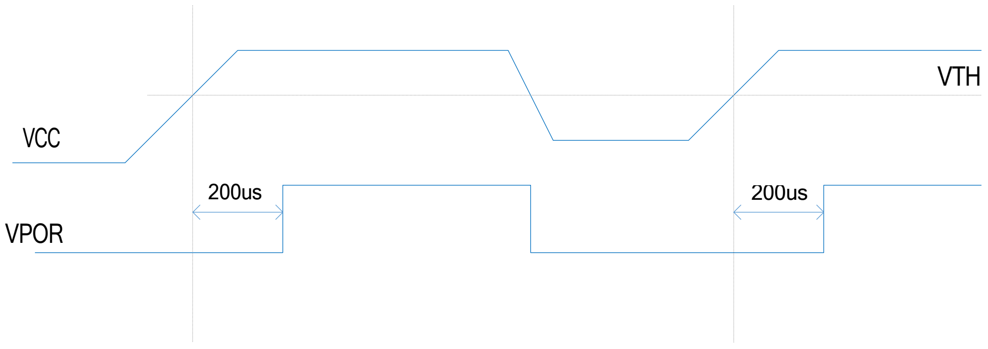
the MCU core directly. Reset system architecture diagram.



Power on reset

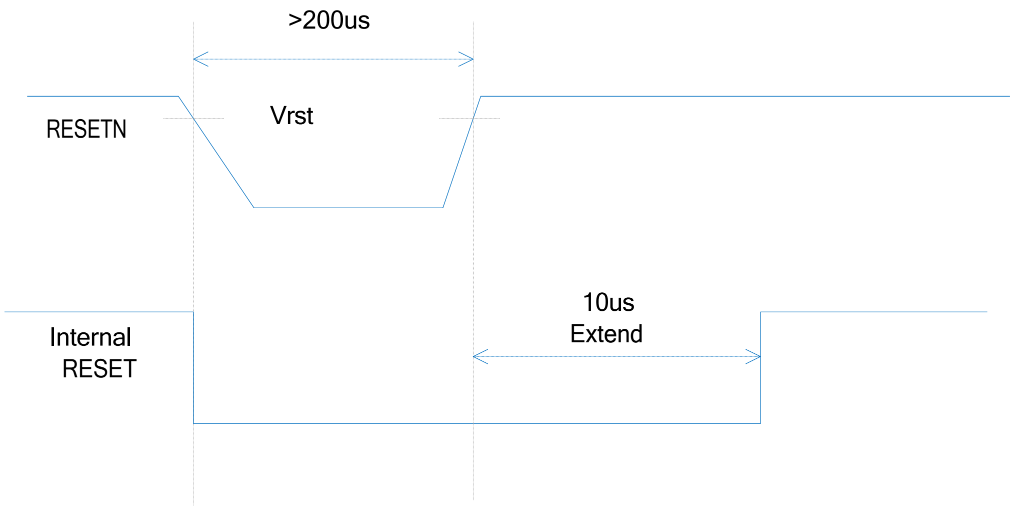
The power-on reset signal is generated by the internal voltage detection circuit. The power-on reset signal is valid when the system power supply (VCC) is below the detection threshold. For the detection threshold of power-on reset, refer to the Electrical Parameters section.

The power-on reset circuit ensures that the chip is in a reset state during power-up and that the chip is able to start operating from a known stable state after power-up. The power-on reset signal is also spread by the chip's internal counter to ensure that the various internal analog modules, such as the RC oscillator, can enter a stable operating state after power-up.



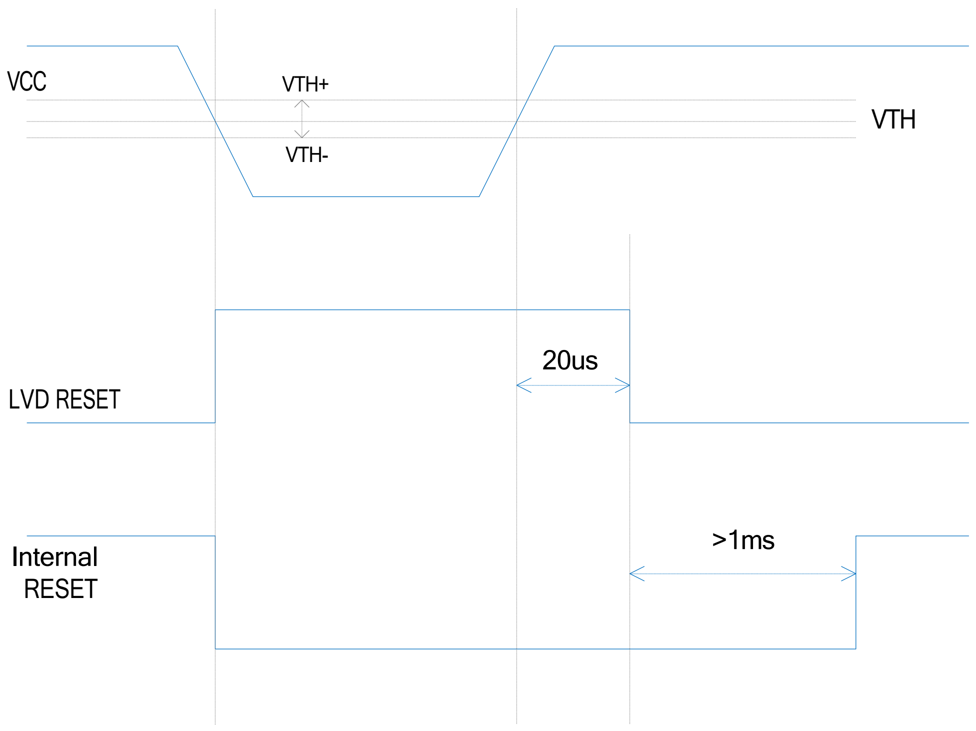
External reset

An external reset is immediately valid when a low level is applied to the external reset pin (RSTN). The width of the low level is greater than one minimum reset pulse width required. The external reset is an asynchronous reset, so it can reset the chip even if the chip is not clocked, and the LGT8FX8P external reset pin can also be used as a general purpose I/O. After the chip is powered on, it is used as the external reset function by default. The user can disable the external reset function of this pin through register configuration, so that it can be used as a normal I/O. For details, please refer to the description of IOCR register.



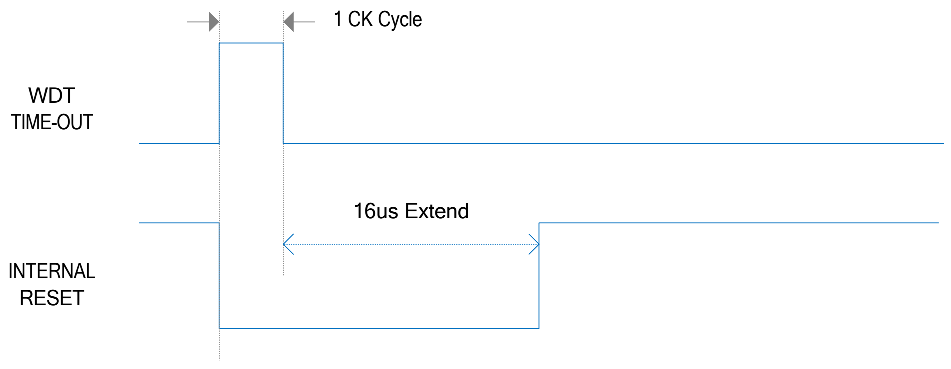
Low voltage detection (LVD) reset

The LGT8FX8P contains an internal programmable low voltage detection (LVD) circuit.LVD also detects voltage changes in VCC, but unlike power-on reset, LVD can select the threshold value of the detected voltage. The user can select between different voltage thresholds by directly manipulating the VDTCR register.The voltage detection circuit of LVD has a hysteresis characteristic of ±10mV to ±50mV for filtering out the jitter of VCC voltage. When LVD is enabled, if the voltage of VCC drops to the set reset threshold, LVD reset will be effective immediately. When VCC increases above the reset threshold, the internal reset unfolding circuit is activated to continue the reset spread for at least 1 millisecond.



Watchdog reset

When the watchdog timer overflows, a one-cycle system reset signal will be generated immediately if the watchdog system reset function is enabled. The watchdog reset signal will also be universally spread by the internal delay counter. For detailed operation of the watchdog controller, refer to the detailed description section below.



Software Reset, OCD Reset

Software reset is triggered by the user by manipulating the sixth bit of the VDTCR register, and the timing of the software reset is exactly similar to that of a watchdog reset. Internally, the reset signal is spread out by 16us.

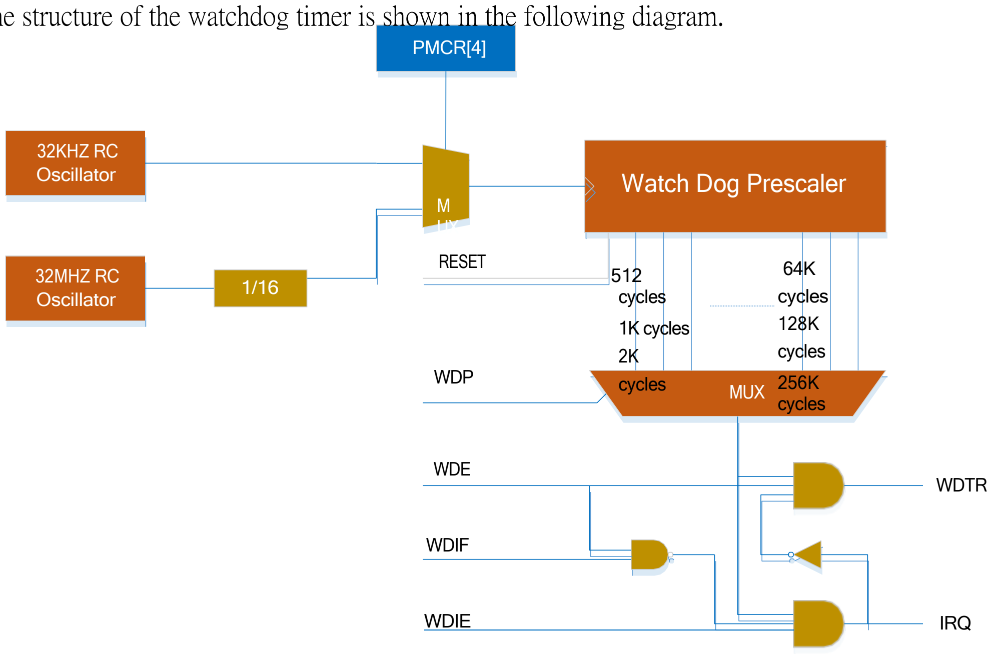
OCD reset is generated by the debugger unit inside the chip. OCD reset is generally controlled by the debugger and cannot be triggered by user software.

Watchdog Timer

- Clock selectable internal 32KHz RC or internal 32MHz RC with 16 divisions (2MHz)
- Supports interrupt mode, reset mode, and reset interrupt mode
- Timer timeout up to 8 seconds

The LGT8FX8P contains an enhanced Watchdog Timer (WDT) module internally.The WDT timer operates either on an internal 32KHz RC oscillator or on a 16 division of an internal 32MHz RC oscillator.The WDT counter can output an interrupt or a system reset signal after an overflow. In normal use, a software WDR - watchdog timer reset instruction is required to restart the counter before it overflows. If the system does not execute the WDR instruction, the WDT will generate an interrupt or system reset.

The structure of the watchdog timer is shown in the following diagram.



In interrupt mode, an interrupt request signal is generated after a WDT overflow. This interrupt can be used as a wakeup signal for sleep mode, or as a general system timer. For example, this interrupt can be used to limit the execution time of an operation and to terminate a current task in an overflow. In system reset mode, the WDT generates a system reset signal immediately after the counter overflows. The most typical use is for preventing the system from dying or running away. The third mode, the reset interrupt mode, combines both interrupt and reset functions. First the system will respond to the WDT interrupt function and switch to the reset mode immediately after exiting the WDT interrupt reset program. This function can support saving some more critical parameter information before resetting.

To prevent the WDT from being accidentally disabled, the operation to turn off the WDT must follow a tightly defined timing sequence. The following code describes how to turn off the watchdog timer. The following example assumes that interrupts are already disabled so that the entire operation flow is not interrupted.

Example code for watchdog enable and disable operations.

assembly code

```
WDT_OFF:
    ; Turn off global interrupt
    CLI
    ; Reset watchdog timer
    WDR
    ; Clear WDRF in MCUSR
    IN r16, MCUSR
    ANDI r16, ~(1 <<
    WDRF) OUT MCUSR,
    r16
    Write logical one to WDCE and WDE
    Keep old Prescaler setting to prevent unintentional time-out
    LDS r16, WDTCSR
    ORI r16, (1 << WDCE) | (1 << WDE)
    STS WDTCSR, r16
    ; Turn off WDT
    LDI r16, (0 << WDE)
    STS WDTCSR, r16
    ; Turn on global interrupt
    SEI
    RET
```

C code

```
void WDT_OFF(void)
{
    __disable_interrupt();
    __watchdog_reset();
    /* Clear WDRF in MCUSR */
    MCUSR &= ~(1 << WDRF);
    /* Write logical one to WDCE and WDE */
    /* Keep old Prescaler setting to prevent unintentional time-out */
    WDTCSR |= (1 << WDCE) | (1 << WDE);
    /* Turn off WDT */
    WDTCSR = 0x00;
    __enable_interrupt();
}
```

[Usage tips]

If the WDT is accidentally enabled, such as a program running away, the chip will be reset, but the WDT will still be in the enabled state. If the WDT is not handled in the user code, this will result in a cyclic reset. To avoid this, it is recommended that the user software clear the Watchdog Reset Flag bit (WDRF) and the WDE control bit in the initialization routine.

The following code describes how to change the timeout value of the watchdog timer.

assembly code

```
WDT_TOV_Change:
    ; Turn off global interrupt
    CLI
    ; Reset watchdog timer
    WDR
    Start timed sequence
    LDS r16, WDTCSR
    ORI r16, (1 << WDCE) | (1 << WDE)
    STS WDTCSR, r16
    -- Got for cycles to set the new value from here --
    Set new time-out value = 64k cycles
    LDI r16, (1 << WDE) | (1 << WDP2) | (1 << WDP0)
    STS WDTCSR, r16
    Finished setting new value, used 2 cycles --
    ; Turn on global interrupt
    SEI
    RET
```

C code

```
void WDT_TOV_Change(void)
{
    __disable_interrupt();
    __watchdog_reset();
    /* Start timed sequence */
    WDTCSR |= (1 << WDCE) | (1 << WDE);
    /* Set new time-out value = 64K cycles */
    WDTCSR |= (1 << WDE) | (1 << WDP2) | (1 << WDP0);
    __enable_interrupt();
}
```

[Instructions for use]

Before changing the WDP configuration bits, it is recommended that the watchdog timer be reset. This is because changing the WDP bit to a relatively small timeout period will likely cause the watchdog to time out and reset.

Register Definition

Low Voltage Detect (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	Low voltage detection module enable control, 1 enable, 0 disable
[1]	VDREN	Low voltage reset function enable control, 1 enable, 0 disable
[4:2]	VDTS	Low pressure detection threshold configuration position 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]	-	keep sth. unused
[6]	SWR	Soft reset enable bit, clearing this bit will generate a software reset
[7]	WCE	VDTCR value change enable bit Before the user can change the value of the VDTCR register, he must first write 1 to this bit and change the value of the other VDTCR bits for the next 6 clock cycles. After four cycles the WCE is automatically cleared and the update operation to the VDTCR register is invalid.

IO Function Multiplexing Register - PMX2

PMX2 - IO Function Multiplex Register								
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	PC6 pin is reset by default, setting this bit to 1 will disable the external reset function, after the reset function is disabled, PC6 can be used as a normal I/O
1	E6EN	PE6 pin is analog input function by default, setting this bit to 1 will turn off the analog input function, this pin can be used as GPIO
2	XIEN	External clock input enable control
4:3	-	keep sth. unused

5	STSC0	Low-speed crystal start-up control
6	STSC1	High-speed crystal start-up control
7	WCE	IOCR value change enable bit Before the user can change the value of the IOCR register, he must first write 1 to this bit in

		The value of the other IOCR bits is changed for the next 6 clock cycles. After four cycles the WCE is automatically cleared to zero and the update operation to the IOCR register is invalid.

MCU Status Register - MCUSR

MCUSR - IO Special Function Control Register								
MCUSR: 0x34(0x54)				Default value: 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 0 to clear zero
[1]	EXTRF	External reset flag, automatically cleared by power-on reset, or write 0 to clear zero
[2]	BORF	Low voltage detection reset, power-on reset auto-zero, or write 0 to clear zero
[3]	WDRF	Watchdog reset flag, automatically cleared by power-on reset, or write 0 to clear zero
[4]	OCDRF	OCD Debugger reset flag, automatically cleared by power-on reset, or write 0 to clear zero
[5]	PDRF	Wake-up flag from Power/off mode as described in the Power Management chapter.
[6]	-	keep sth. unused
[7]	SWDD	SWD interface disable bit. Writing 1 will turn off the SWD interface. When the SWD interface is closed, debugging and ISP operations will not be possible. If the SWD interface is turned off in the user program, the internal program can be disabled by pulling RESET low during power-up, and then debug and ISP operations can be performed. To avoid misuse of the SWDD, the user needs to update the SWDD bit for the first time. The SWDD is written again in the following four cycles to take effect.

[Usage tips].

In order to use the reset flag information more accurately and efficiently, it is recommended that the user try to read the reset flag before the initialization of the program and then clear it to zero.

Watchdog Control Status Register - WDTCSR

WDTCSR - WDT Control and Status Register								
Address: 0x60					Default value: 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 bit. The WDIF bitis set when the WDT is operating in interrupt mode and an overflow occurs. The WDT interrupt is generated when the WDT interrupt enable bit WDIE is "1" and the global interrupt is set. Execute The WDIF bit is cleared when the WDT interrupts, and can be cleared by writing a "1" to the WDIF bit.
[6]	WDIE	WDT interrupt enable control bit. When the WDIE bit is set to "1" and the global interrupt is set, the WDT interrupt is enabled.

		When the WDIE bit is set to "0", the WDT interrupt is disabled. The WDIE bit and the WDE bit together determine the watchdog operating mode, as shown in the table below.

WDE	WDIE	mode	Post spill action
0	0	stop	not
0	1	interrupt mode	disruptions
1	0	reset mode	reset (a dislocated joint)
1	1	Interrupt reset mode	Reset after interruption

[5]	WDP3	WDT Prescaler Selection Control Bit 3. WDP[3] and WDP[2:0] make up the WDT prescaler selection bits WDP[3:0], which are used to set the overflow period of the WDT.
[4]	WDTOE	WDT off enable control bit. The WDTOE bit must be set when the WDE bit is to be cleared, otherwise the WDT will not be turned off. When the WDTOE bit is set, hardware will clear the WDTOE bit after 4 clock cycles.
[3]	WDE	WDT enable control bit. When the WDE bit is set to "1", WDT is enabled. When the WDE bit is set to "0", the WDT is enabled. WDT is banned. The WDE can only be cleared with the WDTOE position bit. To turn off the already enabled WDT, must operate in the following timing sequence. 1. Set both the WDTOE and WDE bits, even if the WDE is already set, and a "1" must be written to the WDE bit before the shutdown operation can begin. 2. Write "0" to the WDE bit for the next 4 clock cycles. This will turn off the WDT. The WDT reset system flag WDRF (located in the MCUSR register) set when the WDE bit is "1" and the WDT overflow resets the system. The WDE bit is set when the WDRF bit is in the set state. Therefore, to clear the WDE bit, the WDRF bit must be cleared first.
[2:0]	WDP	WDT prescaling factor selection control. Used to set the overflow period of the WDT. It is recommended to change the value of the WDP while the WDT is not counting, changing the value of the WDP during the counting process will produce an unanticipated WDT overflow.

Watchdog prescaler selection list.

WDP3	WDP2	WDP1	WDP0	Watchdog Timer Number of overflow cycles	32KHz clocks	2MHz clocks

0	0	0	0	2K cycles	64ms	1ms
0	0	0	1	4K cycles	128ms	2ms
0	0	1	0	8K cycles	256ms	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	keep sth. unused
1	0	1	1	
1	1	0	0	
1	1	0	1	
1	1	1	0	
1	1	1	1	

Interrupts and interrupt vectors

- 28 interrupt sources
- Programmable vector start address

The interrupt resources of LGT8F88P/168P/328P are basically the same, the main difference is that the interrupt vector of LGT8F88P is 1 instruction word (16 bits), while the interrupt vector of LGT8F168P/328P is 2 instruction words.

LGT8F88P Interrupt Vector List

LGT8F88P Interrupt vector list.

number	vector address	Interrupt source signal	Interrupt source description
1	0x0000	RESET	external reset, power-on reset, watchdog reset. SWD Debug Reset, Low Voltage Reset
2	0x0001	INT0	External interrupt request 0
3	0x0002	INT1	External interrupt request 1
4	0x0003	PCI0	Pin level interrupt 0
5	0x0004	PCI1	Pin level interrupt 1
6	0x0005	PCI2	Pin level interrupt 2
7	0x0006	WDT	Watchdog overflow interrupt
8	0x0007	TC2 COMPA	Timer 2 Compare Match A Interrupt
9	0x0008	TC2 COMPB	Timer 2 Compare Match B Interrupt
10	0x0009	TC2 OVF	Timer 2 overflow interrupt
11	0x000A	TC1 CAPT	Timer 1 Input Capture Interrupt
12	0x000B	TC1 COMPA	Timer 1 Compare Match A Interrupt
13	0x000C	TC1 COMPB	Timer 1 Compare Match B Interrupt
14	0x000D	TC1 OVF	Timer 1 overflow interrupt
15	0x000E	TC0 COMPA	Timer 0 Compare Match A Interrupt
16	0x000F	TC0 COMPB	Timer 0 Compare Match B Interrupt
17	0x0010	TC0 OVF	Timer 0 Overflow interrupt
18	0x0011	SPI STC	SPI end-of-serial-transfer interrupt
19	0x0012	USART RXC	USART Receive end interrupt
20	0x0013	USART UDRE	USART Data Register Air Break
21	0x0014	USART TXC	USART end-of-send interrupt
22	0x0015	ADC	ADC end-of-conversion interrupt
23	0x0016	EE_RDY	EEPROM Ready Interrupt
24	0x0017	ANA_COMP	Analog Comparator 0 Interrupt
25	0x0018	TWI	Two-wire serial interface interrupt
26	0x0019	ANA_COMP1	Analog Comparator 1 Interrupt
27	0x001A	-	retain
28	0x001B	PCI3	Pin level interrupt 3

29	0x001C	PCI4	Pin level interrupt 4
30	0x001D	TC3_INT	Timer 3 Interrupt

LGT8F168P/328P Interrupt Vector List

LGT8F168P/328P Interrupt vector list.

nu mb er	vector address	Interrupt source signal	Interrupt source descriptio n
1	0x0000	RESET	external reset, power-on reset, watchdog reset. SWD Debug Reset, Low Voltage Reset
2	0x0002	INT0	External interrupt request 0
3	0x0004	INT1	External interrupt request 1
4	0x0006	PCI0	Pin level interrupt 0
5	0x0008	PCI1	Pin level interrupt 1
6	0x000A	PCI2	Pin level interrupt 2
7	0x000C	WDT	Watchdog overflow interrupt
8	0x000E	TC2 COMPA	Timer 2 Compare Match A Interrupt
9	0x0010	TC2 COMPB	Timer 2 Compare Match B Interrupt
10	0x0012	TC2 OVF	Timer 2 overflow interrupt
11	0x0014	TC1 CAPT	Timer 1 Input Capture Interrupt
12	0x0016	TC1 COMPA	Timer 1 Compare Match A Interrupt
13	0x0018	TC1 COMPB	Timer 1 Compare Match B Interrupt
14	0x001A	TC1 OVF	Timer 1 overflow interrupt
15	0x001C	TC0 COMPA	Timer 0 Compare Match A Interrupt
16	0x001E	TC0 COMPB	Timer 0 Compare Match B Interrupt
17	0x0020	TC0 OVF	Timer 0 Overflow interrupt
18	0x0022	SPI STC	SPI end-of-serial-transfer interrupt
19	0x0024	USART RXC	USART Receive end interrupt
20	0x0026	USART UDRE	USART Data Register Air Break
21	0x0028	USART TXC	USART end-of-send interrupt
22	0x002A	ADC	ADC end-of-conversion interrupt
23	0x002C	EE_RDY	EEPROM Ready Interrupt
24	0x002E	ANA_COMP	Analog comparator interrupt
25	0x0030	TWI	Two-wire serial interface interrupt
26	0x0032	ANA_COMP1	Analog Comparator 1 Interrupt
27	0x0034	-	retain
28	0x0036	PCI3	Pin level interrupt 3
29	0x0038	PCI4	Pin level interrupt 4
30	0x003A	TC3_INT	Timer 3 Interrupt

The LGT8FX8P's reset vector is executed from address 0x0000. All vector addresses except the reset vector can be redirected to a 512-byte aligned starting address using the IVSEL in the MCUCR register and the IVBASE register.

Interrupt vector processing

The following code is used to illustrate reset and interrupt vector programming using the LGT8F88P as an example only.

Assembly Code Example - LGT8F88P		
address	code	instructions
0x000 0x001 0x002 0x003 0x004 0x005 0x006 0x007 0x008 0x009 0x00A 0x00B 0x00C 0x00D 0x00E 0x00F 0x010 0x011 0x012 0x013 0x014 0x015 0x016 0x017 0x018 0x019 0x01A 0x01B ; 0x01C (RESET :) 0x01D 0x01E 0x01F 0x020 0x021	RJMP RESET RJMP EXT_INT0 RJMP EXT_INT1 RJMP PCINT0 RJMP PCINT1 RJMP PCINT2 RJMP WDT RJMP TIM2_COMPA RJMP TIM2_COMPB RJMP TIM2_OVF RJMP TIM1_CAPT RJMP TIM1_COMPA RJMP TIM1_COMPB RJMP TIM1_OVFR RJMP TIM0_COMPA RJMP TIM0_COMPB RJMP TIM0_OVF RJMP SPI_STC RJMP USART_RXC RJMP usart_udre rjmp USART_TXC RJMP ADC RJMP EE_RDY RJMP ANA_COMP RJMP TWI NOP NOP RJMP PCI3	Reset vector external interrupt 0 External interrupt 1 Pin level change interrupt 0 Pin level change interrupt 1 Pin level change interrupt 2 Watchdog timer interrupt Timer 2 Compare Match Group A Interrupt Timer 2 Compare Match Group B Interrupt Timer 2 Overflow Interrupt Timer 1 Capture interrupt Timer 1 Compare Match Group A Interrupt Timer 1 Compare Match Group B Interrupt Timer 1 Overflow Interrupt Timer 0 Compare Match Group A Interrupt Timer 0 Compare Match Group B Interrupt Timer 0 Overflow Interrupt SPI transfer completion interrupt USART receive completion interrupt USART data register air break USART send completion interrupt ADC conversion completion interrupt EEPROM controller ready for interrupt comparator interrupt TWI Controller Interrupt Reserved Address reserved address Pin level change interrupt 3 Start of main program Set the stack pointer to the top RAM address

	LDI r16, high(RAMEND) OUTSPH, r16 LDI r16, low(RAMEND) OUTSPL, r16 SEI ……	Enabling global interrupts

Register Definition

MCU Control Register - MCUCR

MCUCR - MCU Control Register								
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 bitwhich needs to be set first before updating the MCUCR and then completing the update of the MCUCR register within 6 cycles
[1]	IVSEL	Interrupt vector select bit, after this position 1, the interrupt vector address will be set according to IVBASE The value of the register is mapped to the new address
[2]	IFAIL	the system configuration bit load failure flag bit. 0 = Configuration information checks out 1 = Configuration information failed to load
[3]	IRLD	Write 1 will reload the system configuration information
[4]	PUD	Global pull-up ban bit 0 = Enables global pull-up control 1 = Turn off pull-up resistors for all IOs
[5]	EXRFD	External reset filter disable bit 0 = (190us) digital filter with external reset enabled 1 = Digital filter circuit with external reset disabled
[6]	FPDEN	Flash Power/down Enable Control 0: FLASH remains powered on after system SLEEP 1: FLASH power failure after system SLEEP
[7]	FWKEN	Fast Wake-Up Mode Enable Control, valid for Power/Off mode only 0: 260us filter delay 1: 32us filter delay

Interrupt vector base address register - IVBASE

IVBASE - Interrupt vector base address register	
IVBASE: 0x75	Default value: 0x00
IVBASE	IVBASE[7:0]
R/W	R/W

Bit Definition		
[7:0]	IVBASE	If IVSEL is 1, the interrupt vector (except the reset vector) will be remapped on a 512-byte page with IVBASE as the base address. The base address of the mapped interrupt vector is: (IVBASE << 8) + the corresponding vector address

External Interrupts

- 2 external interrupt sources
- Configurable level or edge-triggered interrupts
- Can be used as a wake-up source in sleep mode

summarize

External interrupts are triggered by the INT0 and INT1 pins. As long as the external interrupt is enabled, the interrupt can be triggered even if these 2 pins are configured as outputs. This can be used to generate software interrupts. The external interrupt can be triggered by a rising edge, falling edge or low level and is configured by the External Interrupt Control Register EICRA. When the external interrupt is enabled and configured to be level triggered (INT0 and INT1 pins only), the interrupt will always be generated as long as the pin level is low. rising or falling edge interrupt triggering on INT0 and INT1 pins requires the IO clock to be working properly, while low level triggered interrupts on both INT0 and INT1 pins are asynchronously detected. Except for the idle mode, the IO clock is stopped in all other sleep modes. Therefore, both external interrupts can be used as wakeup sources in sleep modes other than idle mode.

If the level-triggered interrupt is used as a wake-up source in power save mode, the changed level must be held for a certain amount of time to wake up the MCU to reduce the MCU's sensitivity to noise. The requested level must be held long enough for the MCU to end the wake-up process and then trigger the level interrupt.

Register Definition

Register List

processor register	address	default value	description
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

EICRA – External Interrupt Control Register A								
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	description						
7:4	-	Reserved.						
3	ISC11	INT1 pin interrupt trigger mode control bit high.						

2	ISC10	INT1 pin interrupt trigger mode control bit low. External interrupt 1 is triggered by the INT1 pin when the global interrupt is set and the corresponding interrupt mask control bit of the GICR register is set. The interrupt is triggered as described in the table. The MCU first samples the level on the INT1 pin before edge detection. If the edge trigger method or level change trigger method is selected, then pulses with a duration greater than one system clock cycle will trigger the interrupt; pulses that are too short are not guaranteed to trigger the interrupt. If low power is selected

		flat trigger method, then the low level must be held until the current instruction execution is complete before the interrupt is triggered.
1	ISC01	INT0 pin interrupt trigger mode control bit high.
0	ISC00	INT0 pin interrupt trigger mode control bit low. External interrupt 0 is triggered by the INT0 pin when the global interrupt is set and the corresponding interrupt mask control bit of the GICR register is set. The interrupt is triggered as described in the table. The MCU first samples the level on the INT0 pin before edge detection. If the edge trigger method or level change trigger method is selected, then pulses with a duration greater than one system clock cycle will trigger the interrupt; pulses that are too short are not guaranteed to trigger the interrupt. If the low trigger method is selected, then the low level must be held until the current instruction is completed before the interrupt is triggered.

The external interrupt 1 trigger method is shown in the following table.

External interrupt 1 Trigger mode control

ISC1[1:0]	descri ption
0	External pin INT1 Low trigger
1	External pin INT1 Rising or falling edge trigger
2	External pin INT1 falling edge trigger
3	External pin INT1 Rising edge trigger

The external interrupt 0 trigger method is shown in the following table.

External interrupt 0 Trigger mode control

ISC0[1:0]	descri ption
0	External pin INT0 Low trigger
1	External pin INT0 Rising or falling edge trigger
2	External pin INT0 falling edge trigger
3	External pin INT0 Rising edge trigger

External Interrupt Mask Register - EIMSK

EIMSK - External Interrupt Mask Register								
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	description						
7:2	-	retain						

1	INT1	External pin 1 Interrupt enable control bit. When the INT1 bit is set to "1" and the global interrupt is set, the external pin 1 interrupt is enabled and the wake-up function is enabled. Even if the INT1 pin is configured as an output, the interrupt will be generated as soon as the pin level changes accordingly. When the INT1 bit is set to "0", external pin 1 interrupt is disabled and the wake-up function is Prohibition.

0	INT0	External pin 0 Interrupt enable control bit. When the INT0 bit is set to "1" and the global interrupt is set, the external pin 0 interrupt is enabled and the wake-up function is enabled. Even if the INT0 pin is configured as an output, the interrupt will be generated as soon as the pin level changes accordingly. When the INT0 bit is set to "0", external pin 0 interrupt is disabled and the wake-up function is Prohibition.

External Interrupt Flag Register - EIFR

EIFR - External Interrupt Flag Register								
Address: 0x3C					Default value: 0x00			
Bit	7	6	5	4	3	2	1	0
Name	-	-	-	-	-	-	INTF1	INTF0
R/W	-	-	-	-	-	-	R/W	R/W

Bit	Name	description
7:2	-	Reserved.
1	INTF1	External pin 1 Interrupt flag bit. INTF1 is set when an edge-triggered external pin 1 interrupt occurs. When an external pin 1 interrupt is triggered low, the INTF1 bit is not set. If the external pin 1 interrupt enable INT1EN bit is "1" at this time and the global interrupt flag is set, an external pin 1 interrupt will be generated. INTF1 will be automatically cleared when this interrupt service routine is executed, or by writing a "1" to the INTF1 bit.
0	INTF0	External pin 0 Interrupt flag bit. INTF0 is set when an edge-triggered external pin 0 interrupt occurs. The INTF0 bit is not set when an external pin 0 interrupt is triggered low. If the external pin 0 interrupt enable INT0EN bit is "1" at this time and the global interrupt flag is set, an external pin 0 interrupt will be generated. INTF0 will be automatically cleared when this interrupt service routine is executed, or by writing a "1" to the INTF0 bit.

Operational accelerator *(uDSC)*

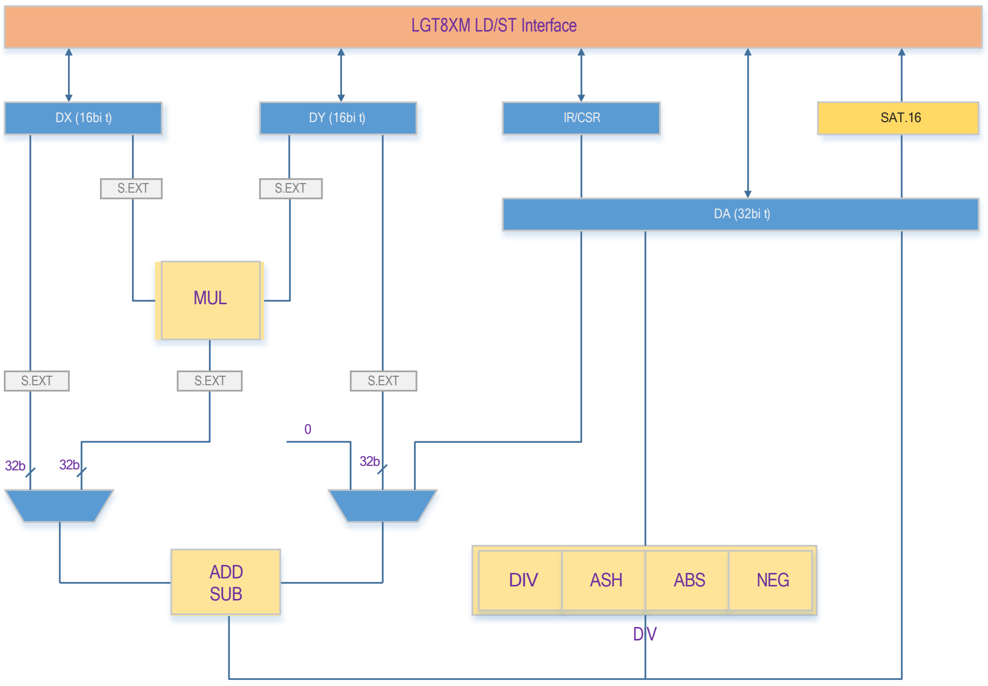
- 16-bit storage mode (LD/ST)
- 32-bit accumulator (DX)
- Single-cycle 16-bit multiplier (MUL)
- 32-bit Arithmetic Logic Unit (ALU)
- 16-bit saturation operation (SD)
- 8 Cycle 32/16 Divider
- Single cycle multiply-add/subtract operations (MAC/MSC)

summarize

The Digital Computing Accelerator (uDSC), as an arithmetic co-processing module of the LGT8XM kernel, implements a 16-bit digital signal processing unit in conjunction with the 16-bit LD/ST mode of the LGT8XM kernel. It can satisfy the processing of most control-type digital signals.

uDSC Functions Internal as well as functional.

1. 16-bit operand register DX/DY
2. 32-bit accumulator register DA
3. Single-cycle 17-bit multiplier (can implement 16-bit signed/unsigned multiplication operations)
4. 32-bit ALU (allows 16/32-bit addition, subtraction and shift operations)
5. 16-bit saturation operation (for storing the result into RAM space)
6. 32/16 Divider, 8 cycles to complete the operation



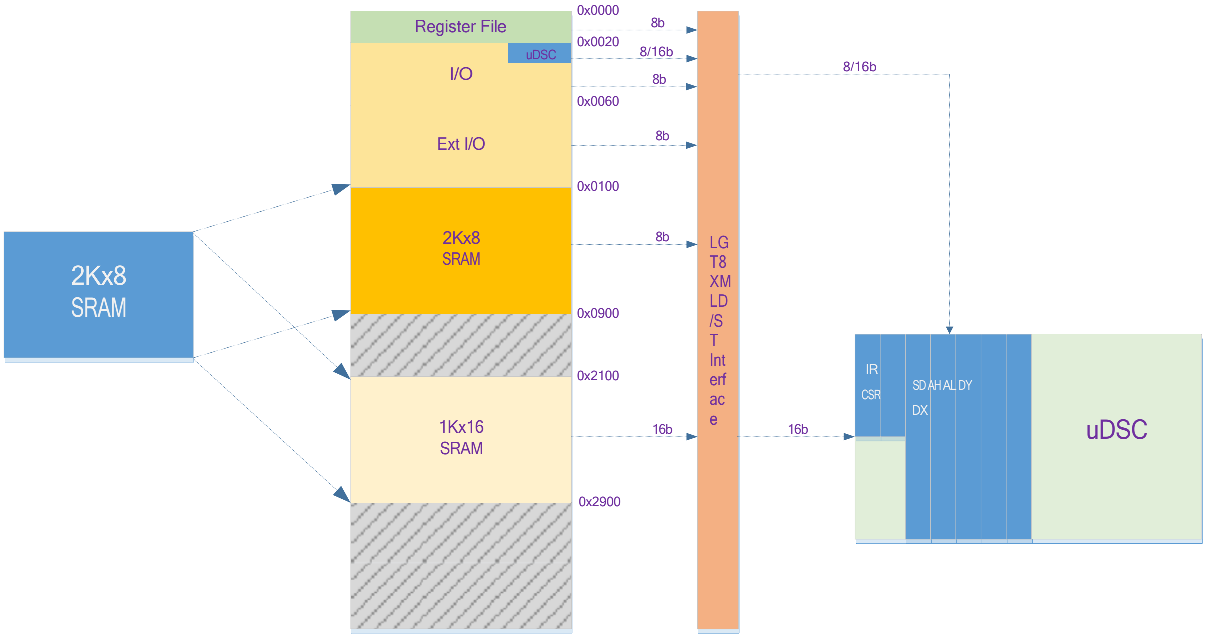
uDSC Structure Diagram

16-bit *LD/ST* operating mode

To increase the efficiency of the uDSC in handling large amounts of data, the LGT8XM kernel implements a dedicated 16-bit LD/ST memory channel that allows efficient 16-bit data exchange between the uDSC and SRAM and general-purpose register files using the LDD/STD instructions.

In order not to disrupt the normal LD/ST instruction system, LGT8XM kernel remaps the SRAM space to 0x2100~0x28FF. When accessing the SRAM from 0x2100~0x28FF using the LD/ST instruction, the kernel automatically enables the 16-bit LD/ST function to open the direct access channel between the SRAM and uDSC.

The following figure shows the data space address distribution of the LGT8XM kernel.



As shown above, the LGT8XM kernel can access 16-bit data storage directly between the DX/DY/DA registers of the uDSC and the SRAM using the LD/ST instructions. The internal registers of the uDSC are also mapped to I/O space, and access to the uDSC registers is divided into two modes: 8/16.

In addition to the DX/DY/DA registers for arithmetic, the uDSC contains two other 8-bit registers: the uDSC control status register CSR and the arithmetic instruction register IR. CSR/IR can only be accessed in byte units through I/O space; DX/DY/AL/AH are accessed in 16-bit mode. They can be accessed using IN/OUT and LD/ST/LDD/STD/LDS/STD instructions.

The uDSC-related control state and data registers are mapped to IO space and can be addressed directly using IN/OU instructions, allowing 8/16-bit data access in one instruction cycle.

The CSR is used to control the operating mode of the uDSC and to record the status flag bits of the current operation performed by the uDSC.IR controls the specific operation implemented by the uDSC. uDSC supports most of the operations that will complete in one cycle, and the division operation requires seven wait cycles, or the flag bits in the CSR register can be used to determine whether the current division operation is complete.

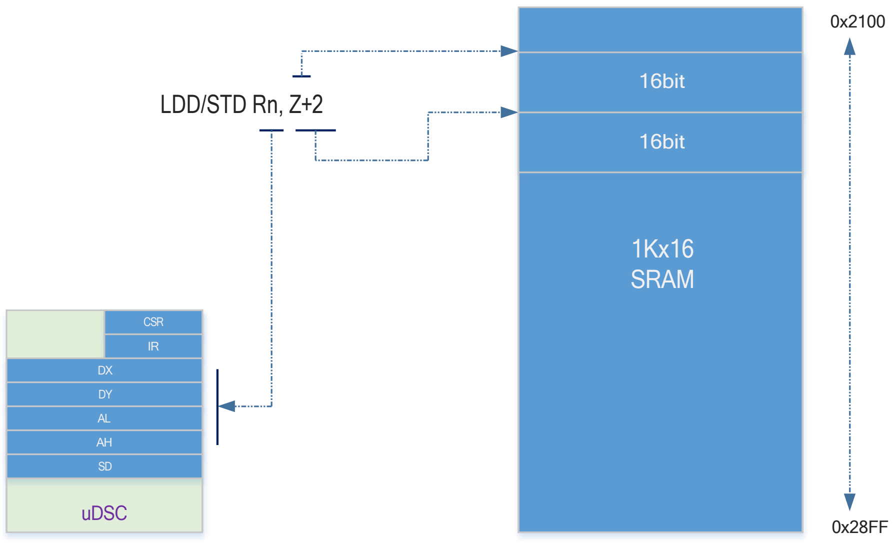
The standard LD/ST instruction uses the LGT8XM internal general-purpose working register as the LD/ST data and X/Y/Z as the target address. When the target address falls in the 16-bit SRAM mapping space, the meaning of the LD/ST instruction operands changes, where X/Y/Z remains as the target address and the meaning of the general-purpose working register addressing is handled in two ways depending on the uDSC mapping mode. uDSC's mapping mode only works for accessing 0x2100~0x28FF addresses. The mapping mode is set via bit 6 (MM) of the CSR register.

In 16-bit LD/ST mode, the instruction "LDD Rn, Z+q" means to load the 16-bit data at address [Z] into the data register of uDSC, and then add an offset "q" to the value of Z. The meaning of Rn in this context is related to the mapping mode CSR[MM] as follows.

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

LD/ST, LDS/STS in the LGT8XM instruction set can access the 0x2100~0x28FF area, but the Y/Z+q addressing method of LDD/STD is more efficient. By using the Y/Z+q addressing method of the LDD/STD instruction, the instruction can be executed and the data can be accessed in one cycle and the address pointer can be automatically moved to the next destination address.

The Y/Z+q offset addressing mode of the LGT8XM kernel's standard LDD/STD instruction uses [Y/Z+q] as the address of the 8-bit data when the instruction is executed, and the value of Y/Z does not increase after execution is completed. When LDD/STD is used to address addresses in the 0x2100 to 0x28FF range, the behavior of the LDD/STD instruction changes:the instruction is executed using [Y/Z] as the 16-bit data address, and after execution, the value of Y/Z is increased by the offset specified by "q". This feature allows us to increase the efficiency of sequential addressing, which can be achieved by setting "q=2" for sequential 16-bit data addressing.



Mapping between variable addresses and 16-bit mode addresses

The LGT8XM is an 8-bit processor with data access in bytes.2K bytes of data space are built into the LGT8F328P. This space is mapped to addresses 0x0100~0x08FF. C/C++ compilation automatically assigns variables between 0x0100~0x08FF. If we have a 16-bit array defined in C/C++ that needs to use uDSC for operations, we need to first map the variable's address to the 16-bit LD/ST access address area (0x2100~0x28FF). The method is simple, just add an offset of 0x2000 to the variable's address.

uDSC Operation instruction definition

The software specifies the operations to be implemented through the IR registers of uDSC. uDSC's operations are all performed in the DX/DY/DA

The user can use 16-bit LD/ST channels to quickly exchange data directly between DX/DY/DA and SRAM. Users can exchange data directly and quickly in DX/DY/DA and SRAM using 16-bit LD/ST channels.

classify	IR[7:0]								Function Description
ADD/SUB	0	0	S1	0	0	1	0	1	DA = DX + DY
	0	0	S1	0	0	0	0	1	DA = DX - DY
	0	0	0	1	1	1	0	1	DA = DY
	0	0	S1	1	1	0	0	1	DA = -DY
	0	0	S1	1	0	1	1	1	DA = DA + DY
	0	0	S1	1	0	0	1	1	DA = DA - DY
MAC/MSC	0	1	S12	S02	0	1	0	0	DA = DX * DY
	0	1	S12	S02	0	0	0	0	DA = -DX * DY
	0	1	S12	S02	1	1	0	0	DA = (DX * DY) >> 1
	0	1	S12	S02	1	0	0	0	DA = (-DX * DY) >> 1
	0	1	S12	S02	0	1	1	S	DA = DA + DX * DY
	0	1	S12	S02	1	1	1	S	DA = (DA + DX * DY) >> 1
	0	1	S12	S02	0	0	1	S	DA = DA - DX * DY
	0	1	S12	S02	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

Description.

1. S indicates whether the ex-officio operation is a signed or unsigned operation
2. S1 indicates whether DX is a signed number, S2 indicates whether DY is a signed number
3. N3...0 is the number of four-bit shift bits, allowing for up to 15-bit shift operations
4. - Indicates that the value of this bit is not meaningless and can be set to 0 or 1. It is recommended to set it to 0

Register Definition

name (of a thing)	IO Address	Function Description
DCSR	0x20 (0x00)	uDSC Control Status Register
DSIR	0x21 (0x01)	Operational instruction register
DSSD	0x22 (0x02)	Result of 16-bit saturation operation of the accumulator DSA
DSDX	0x10 (0x30)	Operand DSDX, 16-bit read/write access
DSDY	0x11 (0x31)	Operand DSDY, 16-bit read/write access
DSAL	0x38 (0x58)	32-bit a c c u m u l a t o r DSA[15:0], 16-bit read/write access
DSAH	0x39 (0x59)	32-bit a c c u m u l a t o r DSA[31:16], 16-bit read/write access

DSCR - Control Status Register

DSCR - 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; refer to the 16-bit operating mode description for detailed definitions. 0 = fast access mode, 1 = IO mapping mode
5	D1	Division completion flag, 1 = operation complete
4	D0	Division operation divided by 0 flag bit
3	-	Unimplemented
2	N	Operation results in negative flag bits

1	Z	The operation results in a zero flag bit
0	C	32 Adder in/out flags

DSIR - Operational Instruction Register

DSIR - uDSC Operational 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	The uDSC operation instruction. See the description in the section "Definition of arithmetic instructions".						

DSDX - Operand register DSDX

DSDX - uDSC operand 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 operand register DSDX														

DSDY - Operand register DSDY

DSDY - uDSC operand register DY																
Address: 0x31 (0x11)										Default value: 0000_0000						
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	DSDY [15:0]															
R/W	R/W															
Bit	Name	description														
15:0	DSDY	16-bit operand register DSDY														

DSAL - Lower 16 bits of the 32-bit accumulator DA

DSAL - lower 16 bits of uDSC operand register DSA																
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	Lower 16 bits of the 32-bit accumulator DSA														

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

DSAH - the high 16 bits of the uDSC operand register DSA																
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 bits of the 32-bit a c c u m u l a t o r DSA														

DSSD - DA saturation register

DSSD- 16-bit DA saturation 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 saturation result for 32-bit a c c u m u l a t o r DSA														

uDSC Application Examples

Example 1. Basic configuration and operations

The following is a simple subroutine (AVRGCC) that implements a 16-bit multiplication operation and returns a 32-bit result. : unsigned long dsu_xmuluu (unsigned short dy, unsigned short dx);

The following is the assembly code for this C function:

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

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} =
                in        AL
                ret       r24, DSAH          ; {r25, r24} =
                          AH
```

General Purpose Programmable Port *(GPIO)*

summarize

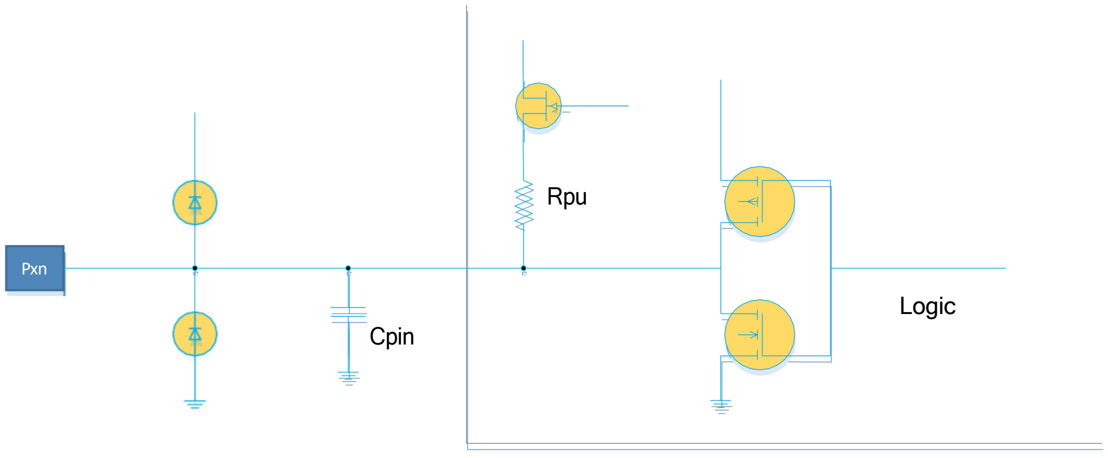
All MCUs implemented on the LGT8XM core family have I/O port read-modify-write functionality. This means that the state of one port can be changed individually using the SBI and CBI instructions without affecting any other I/O. Similarly, changing the direction of a port or controlling its pull-up resistor can also be done.

Most of the I/Os of the LGT8FX8P have symmetrical drive characteristics, capable of driving and absorbing large currents.The I/Os have a two-stage drive capability, allowing the user to control the drive capability of each group of I/Os.The drive capability of the I/Os can directly drive some LEDs.

Most of the I/Os of the LGT8FX8P can drive up to 30mA and can be used directly to drive segment LEDs.

All I/Os have separate ESD protection diodes directly at VCC and GND and are designed to withstand at least up to 5000V ESD pulse.

I/O Equivalent Circuit Diagram.



All registers in this chapter are described in a uniform manner, with lowercase "x" indicating the alphabetic name of the port and lowercase "n" indicating the bit number in the port. However, when using a port register in a program, the exact register name must be used. For example, PORTB3, which represents the third bit of PORTB, is uniformly represented by PORTxn. For detailed definition of I/O related registers, please refer to the register description section.

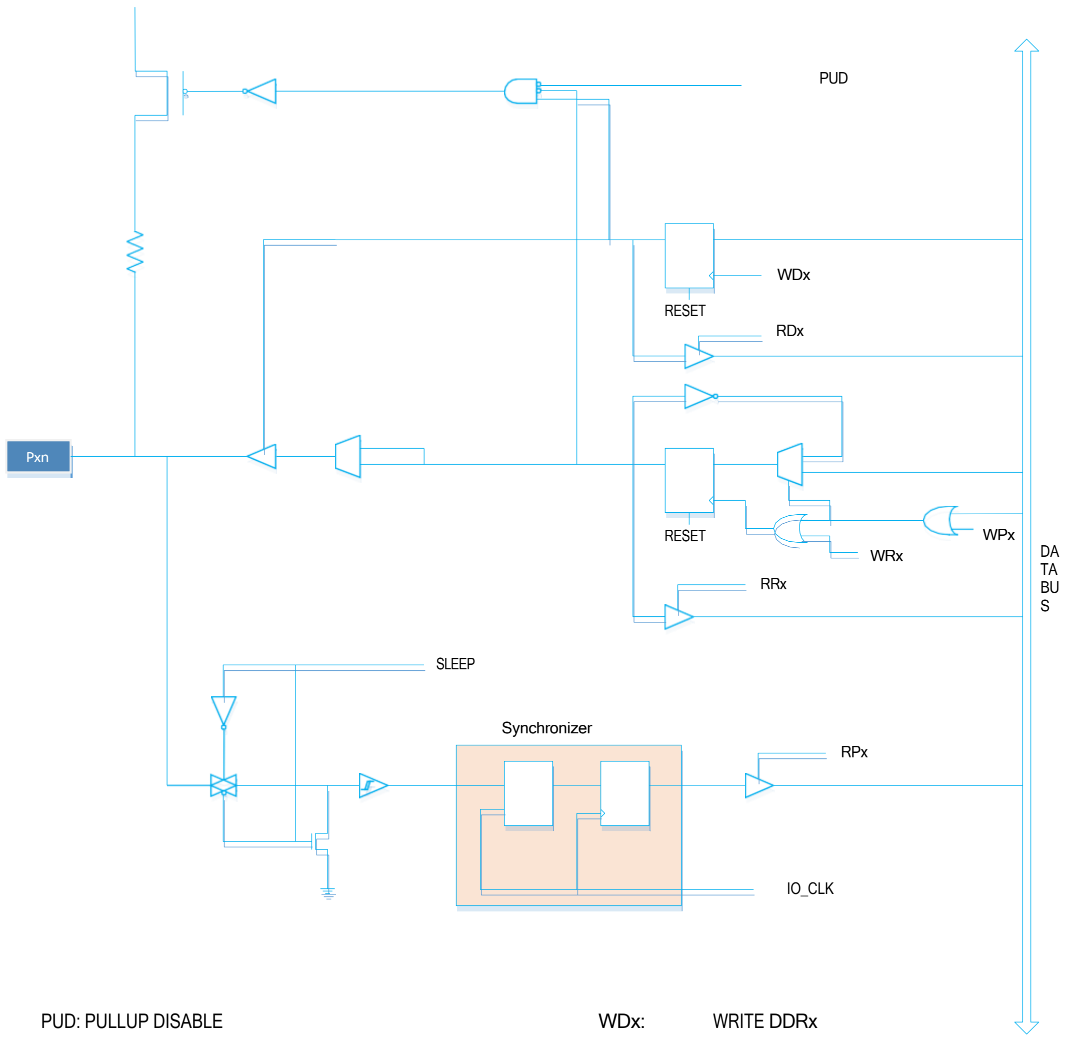
Each port is allocated three I/O register spaces, they are: port data output register (PORTx), port direction register (DDRx), and port data input register (PINx). The port data input register is a read-only register. The PUD bit in the MCUCR register is used to control the pull-up resistors of all I/Os, and when the PUD bit is 1, the pull-up resistors of so I/Os will be disabled.

Most I/Os are multiplexed for other peripheral functions in addition to their general-purpose input/output functions. Refer to the section on Port Function Multiplexing for specific multiplexing functions.

Note that enabling multiplexing of some ports does not affect the use of these ports as digital I/O. Also, some multiplexing functions may require the I/O registers to control the input/output direction of the port. The specific settings will be described in the documentation for each multiplexing module.

Universal input/output ports

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：

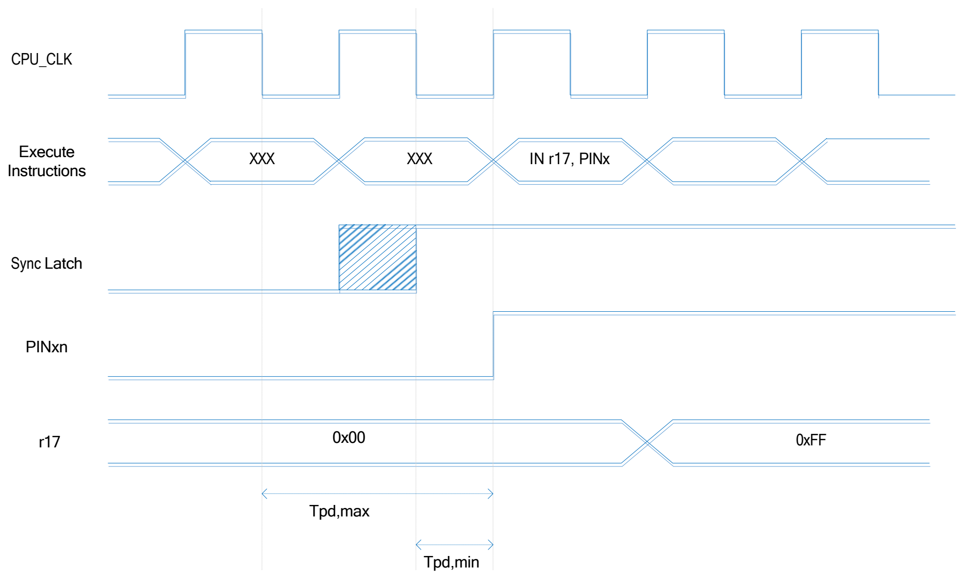


PUD: PULLUP DISABLE		WDx:	WRITE DDRx
SLEEP: SLEEP CONTROL		RDx:	READ DDRx
IO_CLK:	I/O CLOCK	WRx:	WRITE PORTx
		RRx:	READ PORTx REGISTER
		RPx:	READ PORTx PIN
		WPx:	WRITE PINx REGISTER

Port Usage Configuration

Each port is controlled by three register bits: DDxn, PORTxn, and PINxnwhere DDxn is used to enable the use of the DDRx register access, PORTxn can be accessed through the PORTx register, and PINxn can be accessed through the PINx register.

The DDRxn register bit is used to set the input/output direction of the port. If DDxn is set to 1, the Pxn port is configured as an output port. If DDxn is set to 0, Pxn is configured as an input port.



If the PORTxn bit is written 1 and this port is configured as an input port, the pull-up resistor for this port is active. If you want to disable the pull-up resistor for the port, PORTxn must be written to 0 or this port must be configured as an output port.

The reset initialization state of the port is the input state and the pull-up resistor is invalid.

PORTxn is set to 1 while this port is configured as an output port and the external port will be driven high. If PORTxn is set to 0, the port will be driven low.

Input/output switching

When the I/O state toggles between tri-state ([DDxn, PORTxn]) = 0b00) and output high ([DDxn, PORTxn] = 0b11), there will be an intermediate state where the port pulls up or the output is low. Usually, pull-up resistors are acceptable because in a high-resistance environment, the difference between driving to high and pulling up is not important. If this is not the case, the pull-up of the so port can be turned off via the PUD bit in the MCUCR register.

Likewise, the same problem occurs when switching between pull-up enabled input and output low. The user must use a tri-state ([DDxn, PORTxn] = 0b00) or an output high ([DDxn, PORTxn] = 0b11) as an intermediate state.

Port driver configuration table.

DDxn	PORTxn	PUD	Port Status	pull up	Function description
0	0	X	importation	prohibit	Tri-state (High-Z)
0	1	0	importation	enable	Input + internal pull-up mode
0	1	1	importation	prohibit	Tri-state (High-Z)
1	0	X	exports	prohibit	Output low (fan-in)
1	1	X	exports	prohibit	Output high (fan-out)

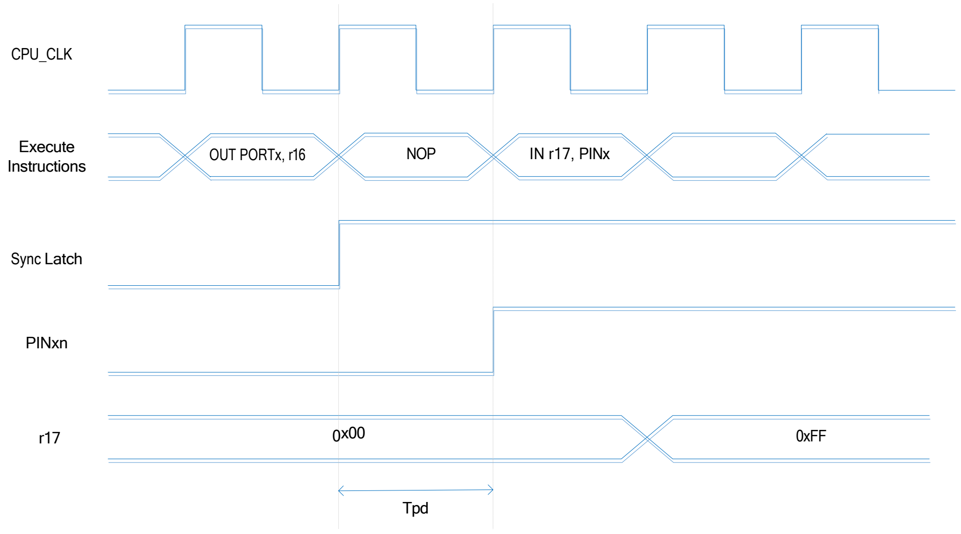
read port value

The current state of the port can be read from the PINxn register bits regardless of how the port direction bit DDxn is set. To avoid the sub-stability created by reading the port directly, the PINxn register bit is the result of the port going through a synchronizer. The synchronizer is a latch and a register together, so there is a small delay between the value of PINxn and the current port. This delay is the result of the presence of the synchronizer and is at most 1½ system cycles.

We assume that the system cycle starts at the first falling edge of the system clock, the latch latches the data when the clock is low, and the data passes straight through the latch when the clock is high, as shown in the shaded portion of the figure above. With the clock low, the port data is latched

is stored and is registered to the PINxn register on the rising edge of the next clock. Tpd,max in the above figure and Tpd,min
is the maximum and minimum delay of the port data, divided into 1.5 cycles and 0.5 cycles.

To read the port value set by the software, insert a null instruction into the write and read byte support of the I/O (NOP). The timing is shown in the following diagram.



The following code illustrates how to set Port B pins 0/1 to high and 2/3 to low, define pins 4 to 7 as inputs and enable the pull-up resistors on pins 6 and 7. The pin values are then read back into the general purpose work register, and a NOP instruction is inserted directly into the pin outputs and inputs as described previously.

assembly code
; 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 code
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;

Input Enable and Sleep Control

From the I/O equivalent circuit diagram we can see that the digital inputs can be clamped to ground level under the control of the SLEEP signal.The SLEEP signal is controlled by the MCU's hibernation controller and various hibernation modes. This ensures that after going into hibernation, the system does not suffer from leakage due to floating port inputs.

The SLEEP control role of the port is replaced by an external interrupt function. If the external interrupt request is invalid, SLEEP control can still function. the SLEEP control function is also replaced by some other second functions, see the description of the port's second functions below.

Quickly flip port status

The port state is set to the IO of the output, and the port state can be changed through the PORTn register. If you need to flip the current port output state, you usually need to read the current port state PINx first, and then write back to PORTn register to complete the flip. LGT8FX8P provides another more efficient way to flip the port state, by writing 1 to PINx register directly, you can flip the specified port state. For example, if we write PINB[3] to 1, we can flip the port state of PB3. This is very useful for applications that need to generate output clocks.

Digital/Analog Multiplexed Ports

Some of the LGT8FX8P ports are mixed analog/digital function multiplexed ports. The mixed port's are used as analog inputs, except for the output PD4 of the internal DAC. When the port is used as an analog function, the software needs to set the port to input mode and turn off the internal pull-ups as needed to avoid any impact on the analog revenue.The DIDR0~2 registers are used to turn off the digital input channel of the mixed function port to avoid excess power loss to the digital circuit from the analog input.DIDRx does not turn off the digital output function of the port.

High Current Push-Pull Driver Port

The LGT8FX8P supports up to 6 high-current push-pull drive ports, supporting up to 80mA push-pull drive. Considering the maximum over-current capability of the chip VCC, it is not recommended to turn on 6 high-current drives at the same time. Especially for QFP32 package with only one power port, it is recommended not to turn on and drive more than 4 high-current loads at the same time.

The normal port is driven at 12mA and the software needs to enable the high current drive capability of the port through the HDR register. The ports with high current drive capability are as follows.

HDR Port	QFP48	QFP32	HDR	Function description
PD5	PD5	PD5	HDR[0]	N/A
PD6	PD6	PD6	HDR [1]	N/A
PF1	PF1	PD1 PF1	HDR [2]	PD1 in QFP32 package is internally equivalent to PD1 in QFP48 Connected in parallel with PF1
PF2	PF2	PD2 PF2	HDR [3]	PD2 in QFP32 package is internally equivalent to PD2 in QFP48 Connected in parallel with PF2
PF4	PF4	PE4 PF4	HDR[4]	The PE4 of the QFP32 package is internally equivalent to the PF4 of the QFP48 Connected in parallel with PE4

PF5	PF5	PE5 PF5	HDR [5]	The PE5 of the QFP32 package is internally equivalent to the PF5 of the QFP48 Connected in parallel with PE5

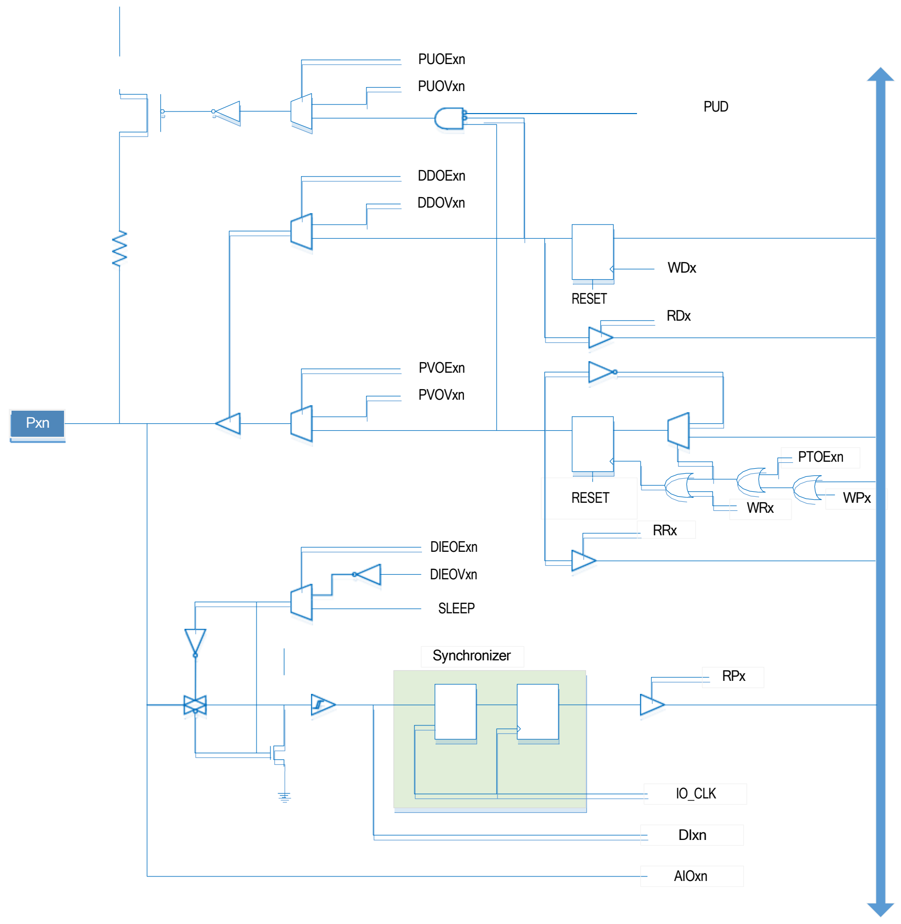
Handling of idle ports

If some ports are not being used, it is recommended to drive them to a fixed level. In any case, floating pins will introduce more power consumption and will cause the system to become unstable under strong disturbances.

The easiest way to give the port a fixed level is to open the pull-up resistor on the port. It is important to note that pull-up resistors are disabled during the power-on reset. The pull-up resistor approach can also introduce excess leakage. Therefore it is recommended to use an external pull-up or pull-down resistor connection. Connecting the port directly to power or ground is not recommended, because if these pins are configured as outputs, there is a risk of very high currents passing by the port, causing damaging effects on the chip.

Port Multiplexing Function

Most ports have multiplexing functions, and the equivalent circuit below illustrates the control of the port by the port multiplexing function. These multiplexing functions are not necessarily present with so port pins.



PUOExn:	Pxn PULL-UP OVERRIDE ENABLE	PUD:	PULLUP DISABLE
PUOVxn:	Pxn PULL-UP OVERRIDE VALUE	WDx:	WRITE DDRx
DDOExn:	Pxn DATA DIRECTION OVERRIDE 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

General description of the multiplexed function control signal.

signal	full name	Function Description
PUOE	Pull-up multiplexing enable	If this bit is 1, pull-up enable is controlled by PVOV; if this bit is 0, pull-up enable is controlled by DDxn, PORTxn and PUD together
PUOV	pull-up multiplexed value	If PUOE is 1, a 1 in this bit will enable the pin pull-up resistor, otherwise it will disable the pin pull-up resistor
DDOE	Port Direction Multiplexing Enable	Sub-bit is 1, pin output enable is controlled by DDOE, otherwise by DDxn control
DDOV	Port Direction Multiplexing Value	If DDOE is 1, a sub-bit of 1 will enable the pin's output function, otherwise the pin's output is turned off
PVOE	Port Data Multiplexing Enable	If the sub-bit is 1 and the pin output is enabled, the output value of the pin will be controlled by PVOV, otherwise it is controlled by PORTxn
PVOV	Port Data Multiplexing Value	Reference PVOE Function Description
PTOE	Port Flip Multiplexing Enable	The secondary bit is 1 and the PORTxn bit will be flipped
DIEOE	Digital input enable multiplex enable	If the secondary bit is 1, port digital input enable will be controlled by DIEOV; otherwise, the MCU's operating state will control
DIEOV	Digital input enable multiplexed values	If DIEOE is 1, the digital input function of the port will be controlled by the sub-bit, independent of the MCU operating state
DI	Digital Inputs	This is the digital input signal that is fed to the alternative function module. As you can see from the circuit diagram under I/O etc., this value is after the Schmitt trigger but before the I/O input synchronizer. This signal is connected to the Peripheral Module, which will be used as needed to To synchronize the process
AIO	Analog input	Analog input/output signal, this signal is directly connected to the PAD of the I/O and can be used as an analog bi-directional signal. This signal is directly connected to the ports of the internal ADC, comparator, and other analog modules

The following sub-section will briefly describe the multiplexing function of each

pin and the associated control signals. Port B Multiplexing Functions

pins	Description of the multiplexing function
PB7	XTALI/TOSC2 (external master crystal pin XI) PCINT7 (pin level change interrupt 7)
PB6	XTALO/TOSC1 (external master crystal pin XO) PCINT6 (pin level change

PB3	MOSI (SPI bus master output/slave input) OC2A (Timer/Counter 2 Compare Match Output A) PCINT3 (Pin Level Change Interrupt 3)
PB2	SSN (SPI bus slave device selection input) OC1B (Timer/Counter 1 Compare Match Output B) PCINT2 (Pin Level Change Interrupt 2)
PB1	OC1A (Timer/Counter 1 Compare Match Output A) PCINT1 (Pin Level Change Interrupt 1)
PB0	ICP1 (Timer/Counter 1 capture input) CLKO (System clock output) PCINT0 (pin level change interrupt 0)

XTALI/TOSC2/PCINT7 - Port *B* Pin *7*

XTALI: External crystal pin XI. This pin will not be used as I/O when used as the clock signal for the crystal. TOSC2: Timer external crystal pin 2. When the internal RC is configured as the main operating clock of the chip and the asynchronous timer function is enabled (ASSR register configuration), this pin will be used as the external crystal pin of the timer. When AS2 of the ASSR register is set to 1 and EXCLK is set to 0, the asynchronous clock function of Timer/Counter 2 using external crystal is enabled and PB7 will be disconnected from the internal I/O port and become the inverted output pin of the internal oscillation amplifier. In this mode, the external crystal is connected to the pin.

PCINT7: Pin level change interrupt 7. PB7 is an external interrupt source.

If PB7 is used for crystal pins, the values of DDB7,PORTB7 and PINB7 will have no meaning.

XTALO/TOSC1/PCINT6- Port *B* Pin *6*

XTALO: External crystal pin XO.

TOSC1: Timer external crystal pin 1. When the internal RC is configured as the main operating clock of the chip and the asynchronous timer function is enabled (ASSR register configuration), this pin will be used as the external crystal pin of the timer. When AS2 of ASSR register is set to 1 and EXCLK is set to 0, the asynchronous clock function of Timer/Counter 2 using external crystal is enabled and PB6 will be the input pin of internal oscillation amplifier with internal I/O port port.In this mode, the external crystal is connected to the pin.

PCINT6: Pin level change interrupt 6. PB6 is an external interrupt source.

If PB6 is used for crystal pins, the values of DDB6,PORTB6 and PINB6 will have no meaning.

SCK/PCINT5- Port *B* Pin *5*

SCK: SPI controller master device clock output,slave device clock input. When the SPI controller is configured as a slave device, this pin will be configured as an input pin and not be controlled by DDB5.When the SPI controller is configured as a master device, the direction of this pin is controlled by DDB5. When this pin is forced as an input by the SPI,the pull-up resistor can still be controlled via the PORTB5 bit.

PCINT5: Pin level change interrupt. pb5 is the external interrupt source.

MISO/PCINT4- Port *B* Pin *4*

MISO: SPI controls master device data input and slave device data output. When the SPI is configured as a master device, this pin will be forced as an input and is not controlled by DDB4. When the SPI is

100 -

used as a slave device, the data side of this pin

The direction is controlled by DDB4. When this pin is forced as an input by the SPI controller, its pull-up resistor can still be used by
PROTB4 control.
PCINT4: Pin level change interrupt. pb4 is the external interrupt source.

MOSI/OC2A/PCINT3- Port *B* Pin *3*

MOSI: SPI controller master device data output, slave device data input. When the SPI is configured as a slave device, this pin will be forced as an input and is not controlled by DDB3. When the SPI controller is configured as a master device, the method of this pin is controlled by DDB3. When this pin is forced as an input by SPI control, its pull-up resistor can still be controlled via PORTB3.
OC2A: Group A compare match output for Timer/Counter 2. PB3 can be used as an external output for Timer/Counter 2 compare match. In this case, the pin must be set to output via DDB3. Also, OC2A is the PWM mode output pin of Timer 2.
PCINT3: Pin level change interrupt. pb3 is the external interrupt source.

SSN/OC1B/PCINT2- Port *B* Pin *2*

SSN: SPI slave device chip select input. When the SPI controller is configured as a slave device, this pin will be forced as an input and is not controlled by DDB2. As a slave device, the SPI controller is active when SSN is driven low. When the SPI controller is configured as a master device, the direction of this pin is controlled by DDB2. When this pin is forced as an input by the SPI controller, the pull-up resistor can still be controlled via PORTB2.
OC1B: Group B compare match output for Timer/Counter 1. PB2 can be used as an external output for Timer/Counter 1 compare match. In this case, the pin must be set to output via DDB2. Also, OC1B is the PWM mode output pin of Timer 1.
PCINT2: Pin level change interrupt. pb2 is the external interrupt source.

OC1A/PCINT1- Port *B* Pin *1*

OC1A: Group A compare match output for Timer/Counter 1. PB1 can be used as an external output for Timer/Counter 1 compare match. In this case, the pin must be set to output via DDB1. Also, OC1A is the PWM mode output pin of Timer 1.
PCINT1: Pin level change interrupt. pB1 is the external interrupt source.

ICP1/CLKO/PCINT0- Port *B* Pin *0*

ICP1: Captured input pin for Timer/Counter 1
CLKO: System operating clock output, when CLKOE bit in CLKPR register is 1, this pin will be forced to output, not controlled by DDB0. The output frequency is the current operating clock frequency of the system.
PCINT0: Pin level change interrupt. pb0 is the external interrupt source.

Port C Multiplexing Function

pins	Description of the multiplexing function
PC7	ADC8 (ADC input channel 8) APN2 (DAP Reverse Input 2) PCINT15 (pin level change input 15)
PC6	RESETN (external reset input) PCINT14 (pin level change input 14)
PC5	ADC5 (ADC input channel 5) SCL (TWI clock line) PCINT13 (pin level change input 13)
PC4	ADC4 (ADC input channel 4) SDA (TWI data line) PCINT12 (pin level change input 12)
PC3	ADC3 (ADC input channel 3) PCINT11 (pin level change input 11)
PC2	ADC2 (ADC input channel 2) PCINT10 (pin level change input 10)
PC1	ADC1 (ADC input channel 1) PCINT9 (pin level change input 9)
PC0	ADC0 (ADC input channel 0) PCINT8 (Pin level change input 8)

ADC8/APN2/PCINT15- Port C Pin 6

ADC8: ADC external input channel 8
APN2: Inverted input port of differential amplifier 2
PCINT15: Pin level change interrupt. With the external reset input function of this pin turned off, PC7 can be used as an external interrupt source.

RESETN/PCINT14- Port C Pin 6

RESETN: External reset input pinAfter power-on resetthis pin defaults to the external reset function. It can be turned off via the IOCR register. After turning off the external reset function, this pin can be used as a general-purpose I/O. However, it should be noted that during power-on and other reset processes, this pin defaults to reset input, so if the user needs to use the general-purpose I/O function of this pin, the external circuit must not affect the power-on and reset processes of the chip, and it is recommended to configure this pin as I/O for output function and add an appropriate external pull-up resistor.
PCINT14: Pin level change interrupt. With the external reset input function of this pin turned off, PC6 can be used as an external interrupt source.

SCL/ADC5/PCINT13- Port C Pin 5

SCL: TWI interface clock signal. after TWEN position 1 in the TWCR register, the TWI interface is enabled and PC5 will be
TWI control, which becomes the clock signal for the TWI interface.
ADC5: ADC input channel 5. The DIDR register is used to turn off the digital function of the digital-analog

multiplexed I/O to avoid the digital section from being used for the digital input.

The effect of division on analog circuits. Refer to the section on ADCs for details.
PCINT13: Pin level change interrupt 13

SDA/ADC4/PCINT12- Port C Pin *4*

SDA: TWI interface data signal. after TWEN position 1 in the TWCR register, the TWI interface is enabled and PC4 will be
TWI control, which becomes the data signal for the TWI interface.
ADC4: ADC input channel 4. The DIDR register is used to disable the digital function of the digital-analog multiplexed I/O to avoid the effect of the digital part on the analog circuitry. Refer to the ADC related chapter for details.
PCINT12: Pin level change interrupt 12

ADC3/APN1/PCINT11- Port *C* Pin *3*

ADC3: ADC input channel 3. The DIDR register is used to disable the digital function of the digital-analog multiplexed I/O to avoid the effect of the digital part on the analog circuitry. Refer to the ADC related chapter for details.
APN1: Differential amplifier reverse input 1
PCINT11: Pin level change interrupt 11

ADC2/APN0/PCINT10- Port *C* Pin *2*

ADC2: ADC input channel 2. The DIDR register is used to disable the digital function of the digital-analog multiplexed I/O to avoid the effect of the digital part on the analog circuitry. Refer to the ADC related chapter for details.
APN0: Differential amplifier reverse input 0
PCINT10: Pin level change interrupt 10

ADC1/APP1/PCINT9- Port *C* Pin *1*

ADC1: ADC input channel 1. The DIDR register is used to disable the digital function of the digital-analog multiplexed I/O to avoid the effect of the digital part on the analog circuitry. Refer to the ADC related chapter for details.
APP1: Differential amplifier forward input 1
PCINT9: Pin level change interrupt 9

ADC0/APP0/PCINT8- Port *C* Pin *0*

ADC0: ADC input channel 0. The DIDR register is used to turn off the digital function of the digital-analog multiplexed I/O to avoid the effect of the digital part on the analog circuitry. Please refer to the ADC related chapter for details.
APP0: Differential amplifier positive input 0
PCINT8: Pin level change interrupt 8

Port D Multiplexing Function

pins	Description of the multiplexing function
PD7	ACXN (Analog Comparator 0/1 Common Negative Input) PCINT23 (Pin Level Change Interrupt 23)
PD6	AC0P (QFP32: Analog Comparator 0 positive input) OC0A (Timer/Counter 0 compare match output A) OC3A (QFP32: Timer/Counter 3 Compare Match Output A) PCINT22 (pin level change interrupt 22)
PD5	T1 (Timer/Counter 1 external count clock input) OC0B (Timer/Counter 0 compare match output B) PCINT21 (Pin level change interrupt 21)
PD4	XCK (USART external clock in/out) DAO (internal 8bit DAC analog output) T0 (Timer/Counter 0 external count clock input) PCINT20 (pin level change interrupt 20)
PD3	INT1 (External interrupt input 1) OC2B (Timer/Counter 2 Compare Match Output B) PCINT19 (Pin Level Change Interrupt 19)
PD2	INT0 (external interrupt input 0) AC0O (comparator 0 output) OC3B (QFP32: Timer/Counter 3 Compare Match Output B) PCINT18 (pin level change interrupt 18)
PD1	TXD (USART data output) OC3A (QFP32: Timer/Counter 3 Compare Match Output A) PCINT17 (Pin Level Change Interrupt 17)
PD0	RXD (USART data input) PCINT16 (pin level change interrupt 16)

ACXN/OC2B/PCINT23- Port *D* Pin *7*

ACXN: Analog Comparator 0/1 Common Negative Input
OC2B: Group B compare match output for Timer/Counter 2. PD7 can be used as an external output for Timer/Counter 2 compare match. In this case, the pin must be set to output via DDD7. Also, OC2B is the PWM mode output pin of Timer 2;
PCINT23: Pin level change interrupt 23

AC0P/OC0A/PCINT22- Port *D* Pin *6*

AC0P: Analog comparator 0 positive input.
OC0A: Group A compare match output for Timer/Counter 0. PD6 can be used as an external output for Timer/Counter 0 compare match. In this case, the pin must be set to output via DDD6. Also, OC0A is the PWM mode output pin of Timer 0

PCINT22: Pin level change interrupt 22

T1/OC0B/PCINT21- Port *D* Pin *5*

T1: External count clock input for Timer/Counter 1

OC0B: Group B compare match output for Timer/Counter 0. PD5 can be used as an external output for Timer/Counter 0 compare match. In this case, the pin must be set to output via DDD5. Also, OC0B is the PWM mode output pin of Timer 0

PCINT21: Pin level change interrupt 21

XCK/T0/DAO/PCINT20- Port *D* Pin *4*

XCK: External clock signal for synchronous
mode USART T0: External count clock
input for timer/counter 0 DAO: Internal 8-bit
DAC analog output

PCINT20: Pin level change interrupt 20

INT1/OC2B/PCINT19- Port *D* Pin *3*

INT1: External interrupt input 1

OC2B: Group B compare match output for Timer/Counter 2. PD3 can be used as an external output for Timer/Counter 2 compare match. In this case, the pin must be set to output via DDD3. Also, OC2B is the PWM mode output pin of Timer 2

PCINT19: Pin level change interrupt 19

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

INT0: External interrupt input 0

OC3B: Timing Counter 3 Compare Match Output B. In the QFP32 package only, PD2 and QFP48/PF2 are combined into one IO, so the OC3B function on PF2 will also be output from PD2

AC0O: Analog Comparator 0 The comparison result is output directly. Controlled by AC0FR register

PCINT18: Pin level change interrupt 18

TXD/OC3A/PCINT17- Port *D* Pin *1*

TXD: Transmit data (USART data output). when the USART transmitter is enabled, PD1 will be forced to be an output, not subject to
Control of DDD1

OC3A: Timing counter 3 compare match output A. In QFP32 package only, PD1 and QFP48/PF1 are combined into one IO, so the OC3A function on PF1 will also be output from PD1

PCINT17: Pin level change interrupt 17

RXD/PCINT16- Port *D* Pin *0*

RXD: Transmit data (USART data input.) When enabled by the USART receiver, PD0 will be forced as an input and is not controlled by DDD0. When the pin is forced as input by USART, the pull-up resistor can still control PCINT16: pin level change interrupt 16 via PORTD0 bit

Port E Multiplexing Function

pins	Description of the multiplexing function
PE7	ADC11 (ADC input channel 11) PCINT31 (pin level change interrupt 31)
PE6	AVREF (QFP32: ADC external reference voltage) ADC10 (ADC input channel 10) PCINT30 (pin level change interrupt 30)
PE5	CLKO (System Clock Output) AC1O (Analog Comparator 1 Output) PCINT29 (pin level change interrupt 29)
PE4	OC0A (Timer/Counter 0 Compare Configuration Output A) PCINT28 (Pin Level Change Interrupt 28)
PE3	ADC7 (ADC input channel 7) AC1N (analog comparator 1 negative input) PCINT27 (pin level change interrupt 27)
PE2	SWD (SWD debugger data line) PCINT26 (pin level change interrupt 26)
PE1	ADC6 (ADC input channel 6) ACXP (Analog to Machine 0/1 Common Positive Input) PCINT25 (Pin Level Change Interrupt 25)
PE0	SWC (SWD debugger clock input) APN4 (differential amplifier reverse input 4) PCINT24 (pin level change interrupt 24)

ADC11/PCINT31- Port *E* Pin *7*

ADC11: ADC external input channel 11
PCINT31: Pin level change interrupt 30

AVREF/ADC10/PCINT30- Port *E* Pin *6*

AVREF: ADC external reference power input, when used for analog functions, the corresponding digital I/O needs to be set as input and the pull-up resistor needs to be closed to avoid interference from digital circuits to analog circuits
ADC10: ADC analog input channel 10
PCINT30: Pin level change interrupt 30

CLKO/AC1O/PCINT29- Port *E* Pin *5*

CLKO: This function is the same as the CLKO function of PB0. Can be used as a backup pin to PB0/CLKO

AC1O: Analog Comparator 1 Output

PCINT29: Pin level change interrupt 29

OC0A/PCINT28- Port E Pin 4

OC0A: Group A compare match output for Timer/Counter 0. PE4 can be used as an external output for Timer/Counter 0 compare match. In this case, the pin must be set to output via DDE4. Also, OC0A is the PWM mode output pin of Timer 0.
PCINT28: Pin level change interrupt 28

ADC7/ AC1N/PCINT27- Port E Pin 3

ADC7: ADC input channel 7. The DIDR register is used to turn off the digital function of the digital-analog multiplexed I/O to avoid the effect of the digital portion on the analog circuitry. Refer to the ADC related chapter for details.
AC1N: Analog Comparator 1 negative input
PCINT27: Pin level change interrupt 27

SWD/PCINT26- Port E Pin 2

SWD: SWD debugger data line. the PE2 defaults to the SWD function. The user can set the MCUSR register SWDD by setting the
Position 1 Disables the SWD debugger function. the debugging function will not be available when SWD is disabled.
PCINT26: Pin level change interrupt 26

ADC6/ACXP/PCINT25- Port E Pin 1

ADC6: ADC input channel 6. The DIDR register is used to disable the digital function of the digital-analog multiplexed I/O to avoid the effect of the digital portion on the analog circuitry. Refer to the ADC related chapter for details.
ACXP: Analog Comparator 0/1 Male Positive Input
PCINT25: Pin level change interrupt 25

SWC/APN4/PCINT24- Port E Pin 0

SWC: SWD debugger clock line.PE0 defaults to the SWC function. The user can set the MCUSR register SWDD by setting the
Position 1 disables the SWD debugger function. when SWD is disabled, the debugging function will not be available
APN4: Differential amplifier reverse input 4
PCINT24: Pin level change interrupt 24

Port F Multiplexing Function

pins	Description of the multiplexing function
PF7	OC2B (Timer/Counter 2 Compare Match Output B) PCINT39 (Pin Level Change Interrupt 39)
PF6	T3 (Timer/Counter 3 external clock input) OC2A (Timer/Counter 2 compare match output A) PCINT38 (Pin level change interrupt 38)
PF5	OC1A (Timer/Counter 1 Compare Match Output A) PCINT37 (Pin Level Change Interrupt 37)
PF4	OC1B (Timer/Counter 1 compare configuration output B) ICP3 (Timer/Counter 3 external capture input) PCINT36 (Pin level change interrupt 36)
PF3	OC0B (Timer/Counter 0 Compare Configuration Output B) PCINT35 (Pin Level Change Interrupt 35)
PF2	OC3B (Timer/Counter 3 Compare Match Output B) PCINT34 (Pin Level Change Interrupt 34)
PF1	OC3A (Timer/Counter 3 Compare Match Output A) PCINT33 (Pin Level Change Interrupt 33)
PF0	ADC9 (ADC external input channel 9) APN3 (differential amplifier reverse input 3) PCINT32 (pin level change interrupt 32)

OC2B/PCINT39 - Port *F* Pin *7*

OC2B: Timer/Counter 2 Compare Match Output B. Output selection is controlled by the PMX1 register
PCINT39: Pin level change interrupt 39

OC2A/T3/PCINT38 - Port *F* Pin *6*

OC2A: Timer/Counter 2 Compare Match Output A. Output selection is controlled by PMX1 register
T3: Timer/Counter 3 External clock input
PCINT38: Pin level change interrupt 38

0C1A/PCINT37 - Port *F* Pin *5*

OC1A: Timer/Counter 1 Compare Match Output A. Output selection is controlled by the PMX0 register
PCINT37: Pin level change interrupt 37

ICP3/OC1B/PCINT36 - Port *F* Pin *4*

OC1B: Group B compare match output for Timer/Counter 1. Output selection is controlled by the PMX0 register

ICP3: Timer/Counter 3 External capture input

PCINT36: Pin level change interrupt 36

OC3C/OC0B/PCINT35- Port F Pin 3

OC0B: Group B compare match output for Timer/Counter 0. Output selection is controlled by the PMX0 register
OC3C: Timer/Counter 3's Group C Compare Match Output
PCINT35: Pin level change interrupt 35

OC3B/PCINT34- Port F Pin 2

OC3B: Timer/Counter 3's Group B Compare Match output
PCINT34: Pin level change interrupt 34

OC3A/PCINT33- Port F Pin 1

OC3A: Group B compare match output of Timer/Counter 3. Output selection is controlled by the PMX1 register
PCINT33: Pin level change interrupt 33

ADC9/APN3/PCINT32- Port F Pin 0

ADC9: ADC external mode input channel 9
APN3: Differential amplifier reverse
input 3
PCINT32: Pin level change interrupt 32

Register Definition

Port B Output Data Register - PORTB

PORTB - Port B Output Data Register								
PORTB: 0x05(0x25)					Default value: 0x00			
Bits	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	Group B Port Output Register						

Port B Direction Register - DDRB

DDRB - Port B Direction Register								
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 group direction control bits; 1 = output, 0 = input						

Port B Input Data Register - PINB

PINB - Port B input data storage								
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	Group B port status register. Reading PINB directly obtains the current status of the port; writing PINBn bit 1 will flip the output status of PORTBn						

Port C Output Data Register - PORTC

PORTC - Port C Output Data Register								
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	Group C Port Output Register						

Port C Direction Register - DDRC

DDRC - Port C Direction Register								
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	Group C port direction control bits; 1 = output, 0 = input						

Port C Input Data Register - PINC

PINC - Port C input data storage								
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	Group C port status register; read PINC to get current port status write PINC will flip current port output						

Port D Output Data Register - PORTD

PORTD - Port D Output Data Register

PORTD: 0x0B(0x2B)				Default value: 0x00				
Bits	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	Group D Port Output Register

Port D Direction Register - DDRD

DDRD - Port D Direction Register								
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	Group D Port Output Direction Control Register						

Port D Input Data Register - PIND

PIND - Port D Input Data Storage								
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	D Group Port Status Register Read PIND to get current port level status Write PINDn to 1, flip the state of the corresponding bit of PORTDn						

Port E Output Data Register - PORTE

PORTE – Port E Output Data Register								
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]	PORTE	Group E Port Output Register						

Port E Direction Register - DDRE

DDRE – Port E Direction Register								
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	Group E Port Direction Control Register						

Port E Input Data Register - PINE

PINE - Port E Input Data Storage								
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	E Group Port Status Register Read PINE to get current port level status Write PINEn to 1, flip the state of the PORTEn bit						

Port F Output Register - PORTF

PINF - Port F Input Data Storage								
PORTF: 0x14(0x34)				Default value: 0x00				
Bits	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]	PORTF	F Group Port Status Register The input mode port, corresponding to a bit written 1 will turn on the internal pull-up Output mode port, corresponding bit written 1 will drive output high						

Port F Directional Control Register - DDRF

DDRF - Port F Directional Control Register								
DDRF: 0x13(0x33)				Default value: 0x00				
Bits	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]	DDRF	Group F Port Direction Control Register						

Port F Status Register - PINF

PINF - Port F Status Register								
PINF: 0x12(0x32)				Default value: 0x00				
Bits	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]	PINF	F Group Port Status Register Read PINF to get the current level status of port F PINFn Write 1, flip the state of the corresponding bit of PORTFn						

Port Driver Control Register - HDR

HDR0 - Port Driver Control Register								
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								

[7:6]	-	keep sth. unused
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	PD6 output drive control; 1 = 80mA drive, 0 = 12mA drive
0	HDR0	PD5 output drive control; 1 = 80mA drive, 0 = 12mA drive

Port Multiplexing Control Register 0- PMX0

PMX0 - Port Multiplexing Control Register 0								
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 control; the WCE bit needs to be written to 1 before updating the PMX0/1 register, and the update to PMX0/1 is completed within the next 6 system cycles.
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	OC0B Auxiliary Output Control 1 = OC0B output to PF3 0 = OC0B output to PD5
3	C0AC0	OC0A Auxiliary Output Control The 0C0A output is controlled by the C0AC0 bit in conjunction with the C0AS bit of the TCCR0B register. { C0AC0, C0AS} = 00 = OC0A output to PD6 01 = 0C0A Output to PE4 10 = 0C0A Output to PC0 11 = OC0A 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 TXD Auxiliary Output Control 1 = TXD output to PD6, 0 = TXD output t o PD1
0	RXD5	Serial RXD Auxiliary Input Control 1 = RXD input from PD5, 0 = RXD input from PD0

Port Multiplexing Control Register 1- PMX1

PMX1 - Port Multiplexing Control Register 1								
PMX1: 0xED					Default value: 0x00			
Bit	-	-	-	-	-	C3AC	C2BF7	C2AF6
R/W	-	-	-	-	-	R/W	R/W	R/W
Bit Definition								
[7:3]	-	keep sth. unused						
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						
Instructions for use								
PMX0/1 Shared Register Update Protection Control Bit PMX0[7], When updating PMX1, refer to the PMX0 register for a description of PMX0[7] control.								

Port Multiplexing Control Register 2 – PMX2

PMX2 - Port Multiplexing Control Register 2								
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]	WCE	PMX2 update enable control; before updating the PMX2 register, you need to write The WCE bit is 1 and the update to PMX2 is completed within the next 6 system cycles.						
[6]	STSC1	High-speed crystal IO start-up circuit control STSC1 is automatically enabled after enabling the high speed crystal via PMCR. STSC1 is automatically cleared when the system clock is switched to the external high speed crystal. The software can also manually clear STSC1 after the crystal is stabilized, which has turned off the crystal start circuit to save power.						
[5]	STSC0	Low-speed crystal IO start circuit control STSC0 is automatically enabled when the low-speed crystal is enabled via PMCR. STSC0 is automatically cleared when switching the system clock to the external low-speed crystal. The software can also manually clear STSC0 after the crystal has stabilized and has turned off the						

		crystal start circuit to save power.
[4:3]	-	keep sth. unused
[2]	XIEN	Enabling the external clock input requires also enabling the external crystal
[1]	E6EN	Enables PE6 generic IO function; default PE6 is AVREF function
[0]	C6EN	Enables general purpose IO function of PC6; default PC6 is external reset input

Pin level change interrupt

- 40 pin level change interrupt sources
- 5 interrupt entries

a general narrative

Pin level change interrupts are triggered by the PBn, PCn, PDn, PEn and PFn pins.As long as the pin level change interrupt is enabled, the interrupt can be triggered even if these pins are configured as outputs. This can be used to generate software interrupts.

Any enabled PBn pin flip will trigger the pin level interrupt PCI0, an enabled PCn pin flip will trigger PCI1, an enabled PDn pin flip will trigger PCI2, and an enabled PEn pin flip will trigger PCI3. The enable of each pin change interrupt is controlled by the PCMSK0 to 4 registers, respectively. All pin level change interrupts are asynchronously detected and can be used as a wake-up source in some sleep modes.

Register Definition

Pin Change Interrupt Register List

processor register	address	default value	description
PCICR	0x68	0x00	Pin change interrupt control register
PCIFR	0x3B	0x00	Pin change interrupt flag register
PCMSK0	0x6B	0x00	Pin change interrupt mask register 0
PCMSK1	0x6C	0x00	Pin change interrupt mask register 1
PCMSK2	0x6D	0x00	Pin change interrupt mask register 2
PCMSK3	0x73	0x00	Pin change interrupt mask register 3
PCMSK4	0x74	0x00	Pin change interrupt 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. Pin change interrupt 4 is enabled when the PCIE4 bit is set to "1" and the global interrupt is enabled. A change in the level of any of the enabled PFn pins will generate a PCI4 interrupt. the enable of the PFn pin interrupt can be controlled separately by the PCMSK4 register. When the PCIE3 bit is set to "0", pin change interrupt 3 is disabled.
3	PCIE3	Pin change interrupt enable control bit 3. When the PCIE3 bit is set to "1" and the global interrupt is enabled, pin change interrupt 3 is enabled.

		A change in the level of any of the enabled PEn pins generates a PCI3 interrupt.The enable of the PEn pin interrupt can be controlled separately by the PCMSK3 register. When the PCIE3 bit is set to "0", pin change interrupt 3 is disabled.
2	PCIE2	Pin change interrupt enable control bit 2. When the PCIE2 bit is set to "1" and the global interrupt is enabled, pin change interrupt 2 is enabled. A change in the level of any of the enabled PDn pins will generate a PCI2 interrupt. the enable of the PDn pin interrupt can be controlled separately by the PCMSK2 register. When the PCIE2 bit is set to "0", pin change interrupt 2 is disabled.
1	PCIE1	Pin change interrupt enable control bit 1. Pin change interrupt 1 is enabled when the PCIE1 bit is set to "1" and the global interrupt is enabled. A change in the level of any of the enabled PCn pins will generate a PCI1 interrupt. the enable of the PCn pin interrupt can be controlled separately by the PCMSK1 register. When the PCIE1 bit is set to "0", pin change interrupt 1 is disabled.
0	PCIE0	Pin change interrupt enable control bit 0. When the PCIE0 bit is set to "1" and the global interrupt is enabled, pin change interrupt 0 is enabled. A change in the level of any of the enabled PBn pins will generate a PCI0 interrupt. the enable of the PBn pin interrupt can be controlled separately by the PCMSK0 register. When the PCIE0 bit is set to "0", pin change interrupt 0 is disabled.

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. A change in the level of any of the enabled PFn pins will set PCIF4. when both PCIE4 and the global interrupt are set, the MCU will jump to the PCI4 interrupt entry address. the enable of the PFn pin interrupt can be controlled by the PCMSK4 register respectively. Executing an interrupt service routine or writing a "1" to the PCIF4 bit will clear the PCIF4 bit.
3	PCIF3	Pin change interrupt flag bit 3. A change in the level of any of the enabled PEn pins will set PCIF3. when both PCIE3 and the global interrupt are set, the MCU will jump to the PCI3 interrupt entry address. the enable of the PEn pin interrupt can be controlled by the PCMSK3 register respectively. Executing an interrupt service routine or writing a "1" to the PCIF3 bit will clear the PCIF3 bit.

2	PCIF2	Pin change interrupt flag bit 2. A change in the level of any of the enabled PDn pins will set PCIF2. When both PCIE2 and the global interrupt are set, the MCU will jump to the PCI2 interrupt entry address. enable of the PDn pin interrupt can be controlled separately by the PCMSK2 register. Executing an interrupt service routine or writing a "1" to the PCIF2 bit will clear the PCIF2 bit.
1	PCIF1	Pin change interrupt flag bit 1.

		A change in the level of any of the enabled PCn pins will set PCIF1. when both PCIE1 and the global interrupt are set, the MCU will jump to the PCI1 interrupt entry address. the enable of the PCn pin interrupt can be controlled by the PCMSK1 register respectively. Executing an interrupt service routine or writing a "1" to the PCIF1 bit will clear the PCIF1 bit.
0	PCIF0	Pin change interrupt flag bit 0. A change in the level of any of the enabled PBn pins will set PCIF0. When both PCIE0 and the global interrupt are set, the MCU will jump to the PCI0 interrupt entry address. the enable of the PBn pin interrupt can be controlled by the PCMSK0 register respectively. Executing an interrupt service routine or writing a "1" to the PCIF0 bit will clear the PCIF0 bit.

PCMSK0 - Pin Change Interrupt Mask Register 0

PCMSK0 – Pin Change 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. When the PCINT7 bit is set to "1", the PB7 pin level change interrupt is enabled. a level change on the PB7 pin will set PCIF0, and if the PCIE0 bit and the global interrupt are set, the PCI0 interrupt will be generated. When the PCINT7 bit is set to "0", the PB7 pin level change interrupt is disabled.
6	PCINT6	Pin change enable mask bit 6. When the PCINT6 bit is set to "1", the PB6 pin level change interrupt is enabled. a level change on the PB6 pin will set PCIF0, and if the PCIE0 bit and the global interrupt are set, the PCI0 interrupt will be generated. When the PCINT6 bit is set to "0", the PB6 pin level change interrupt is disabled.
5	PCINT5	Pin change enable mask bit 5. When the PCINT5 bit is set to "1", the PB5 pin level change interrupt is enabled. a level change on the PB5 pin will set PCIF0, and if the PCIE0 bit and the global interrupt are set, the PCI0 interrupt will be generated. When the PCINT5 bit is set to "0", the PB5 pin level change interrupt is disabled.
4	PCINT4	Pin change enable mask bit 4. When the PCINT4 bit is set to "1", the PB4 pin level change interrupt is enabled. a level change on the PB4 pin will set PCIF0, and if the PCIE0 bit and the global interrupt are set, the PCI0 interrupt will be generated. When the PCINT4 bit is set to "0", the PB4 pin level change interrupt is disabled.

3	PCINT3	Pin change enable mask bit 3. When the PCINT3 bit is set to "1", the PB3 pin level change interrupt is enabled. a level change on the PB3 pin will set PCIF0, and if the PCIE0 bit and the global interrupt are set, the PCI0 interrupt will be generated. When the PCINT3 bit is set to "0", the PB3 pin level change interrupt is disabled.
2	PCINT2	Pin change enable mask bit 2. When the PCINT2 bit is set to "1", the PB2 pin level change interrupt is enabled. a level change on the PB2 pin will set PCIF0, and if the PCIE0 bit and the global interrupt are set, the PCI0 interrupt will be generated. When the PCINT2 bit is set to "0", the PB2 pin level change interrupt is disabled.
1	PCINT1	Pin change enable mask bit 1. When the PCINT1 bit is set to "1", the PB1 pin level change interrupt is enabled.PB1 pin

		A level change on the PCIF0 will set PCIF0, and if the PCIE0 bit and the global interrupt are set, it will generate PCI0 interrupt.When the PCINT1 bit is set to "0", the PB1 pin level change interrupt is disabled.
0	PCINT0	Pin change enable mask bit 0. When the PCINT0 bit is set to "1", the PB0 pin level change interrupt is enabled. a level change on the PB0 pin will set PCIF0, and if the PCIE0 bit and the global interrupt are set, it will generate PCI0 interrupt.When the PCINT0 bit is set to "0", the PB0 pin level change interrupt is disabled.

PCMSK1 - Pin change interrupt mask register 1

PCMSK1 – Pin Change 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. When the PCINT15 bit is set to "1", the PC7 pin level change interrupt is enabled. a level change on the PC7 pin will set PCIF1, and if the PCIE1 bit and the global interrupt are set, the PCI1 interrupt will be generated. When the PCINT15 bit is set to "0", the PC7 pin level change interrupt is disabled.
6	PCINT14	Pin change enable mask bit 14. When the PCINT14 bit is set to "1", the PC6 pin level change interrupt is enabled. a level change on the PC6 pin will set PCIF1, and if the PCIE1 bit and the global interrupt are set, the PCI1 interrupt will be generated. When the PCINT14 bit is set to "0", the PC6 pin level change interrupt is disabled. Stop.
5	PCINT13	Pin change enable mask bit 13. When the PCINT13 bit is set to "1", the PC5 pin level change interrupt is enabled. a level change on the PC5 pin will set PCIF1, and if the PCIE1 bit and the global interrupt are set, the PCI1 interrupt will be generated. When the PCINT13 bit is set to "0", the PC5 pin level change interrupt is disabled.
4	PCINT12	Pin change enable mask bit 12. When the PCINT12 bit is set to "1", the PC4 pin level change interrupt is enabled. a level change on the PC4 pin will set PCIF1, and if the PCIE1 bit and the global interrupt are set, the PCI1 interrupt will be generated. When the PCINT12 bit is set to "0", the PC4 pin level change interrupt is disabled.
3	PCINT11	Pin change enable mask bit 11. When the PCINT11 bit is set to "1", the PC3 pin level change interrupt is enabled. a level change on the PC3 pin will set PCIF1, and if the PCIE1 bit and the global interrupt are set, the PCI1 interrupt will be generated. When the PCINT11 bit is set to "0", the PC3 pin level change interrupt is disabled.

2	PCINT10	Pin change enable mask bit 2. When the PCINT10 bit is set to "1", the PC2 pin level change interrupt is enabled. a level change on the PC2 pin will set PCIF1, and if the PCIE1 bit and the global interrupt are set, the PCI1 interrupt will be generated. When the PCINT10 bit is set to "0", the PC2 pin level change interrupt is disabled.

		Stop.
1	PCINT9	Pin change enable mask bit 1. The PC1 pin level change interrupt is enabled when the PCINT9 bit is set to "1." A level change on the PC1 pin will set PCIF1, and if the PCIE1 bit and the global interrupt are set, it will generate PCI1 interruptWhen the PCINT9 bit is set to "0", the PC1 pin level change interrupt is disabled.
0	PCINT8	Pin change enable mask bit 0. When the PCINT8 bit is set to "1", the PC0 pin level change interrupt is enabled. a level change on the PC0 pin will set PCIF1, and if the PCIE1 bit and the global interrupt are set, the PCI1 interrupt will be generated.When the PCINT8 bit is set to "0", the PC0 pin level change interrupt is disabled.

PCMSK2 - Pin change interrupt mask register 2

PCMSK2 – Pin Change Mask Register 2								
Address: 0x6D					Default value: 0x00			
Bits	7	6	5	4	3	2	1	0
	PCINT23	PCINT22	PCINT21	PCINT20	PCINT19	PCINT18	PCINT17	PCINT16
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. When the PCINT23 bit is set to "1", the PD7 pin level change interrupt is enabled. a level change on the PD7 pin will set PCIF2, and if the PCIE2 bit and the global interrupt are set, a PCI2 interrupt will be generated. When the PCINT23 bit is set to "0", the PD7 pin level change interrupt is disabled.
6	PCINT22	Pin change enable mask bit 6. When the PCINT22 bit is set to "1", the PD6 pin level change interrupt is enabled. a level change on the PD6 pin will set PCIF2, and if the PCIE2 bit and the global interrupt are set, a PCI2 interrupt will be generated. When the PCINT22 bit is set to "0", the PD6 pin level change interrupt is disabled.
5	PCINT21	Pin change enable mask bit 21. When the PCINT21 bit is set to "1", the PD5 pin level change interrupt is enabled. a level change on the PD5 pin will set PCIF2, and if the PCIE2 bit and the global interrupt are set, a PCI2 interrupt will be generated. When the PCINT21 bit is set to "0", the PD5 pin level change interrupt is disabled.
4	PCINT20	Pin change enable mask bit 20. When the PCINT20 bit is set to "1", the PD4 pin level change interrupt is enabled. a level change on the PD4 pin will set PCIF2, and if the PCIE2 bit and the global interrupt are set, a PCI2 interrupt will be generated. When the PCINT20 bit is set to "0", the PD4 pin level change interrupt is disabled.

3	PCINT19	Pin change enable mask bit 19. When the PCINT19 bit is set to "1", the PD3 pin level change interrupt is enabled. a level change on the PD3 pin will set PCIF2, and if the PCIE2 bit and the global interrupt are set, a PCI2 interrupt will be generated. When the PCINT19 bit is set to "0", the PD3 pin level change interrupt is disabled.
2	PCINT18	Pin change enable mask bit 18.

		When the PCINT18 bit is set to "1", the PD2 pin level change interrupt is enabled. a level change on the PD2 pin will set PCIF2, and if the PCIE2 bit and the global interrupt are set, a PCI2 interrupt will be generated. When the PCINT18 bit is set to "0", the PD2 pin level change interrupt is disabled.
1	PCINT17	Pin change enable mask bit 17. When the PCINT17 bit is set to "1", the PD1 pin level change interrupt is enabled. a level change on the PD1 pin will set PCIF2, and if the PCIE2 bit and the global interrupt are set, a PCI2 interrupt will be generated. When the PCINT17 bit is set to "0", the PD1 pin level change interrupt is disabled.
0	PCINT16	Pin change enable mask bit 16. When the PCINT16 bit is set to "1", the PD0 pin level change interrupt is enabled. a level change on the PD0 pin will set PCIF2, and if the PCIE2 bit and the global interrupt are set, a PCI2 interrupt will be generated. When the PCINT16 bit is set to "0", the PD0 pin level change interrupt is disabled.

PCMSK3 - Pin Change Interrupt Mask Register 3

PCMSK3 – Pin Change Mask Register 3								
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. When the PCINT31 bit is set to "1", the PE7 pin level change interrupt is enabled. a level change on the PE7 pin will set PCIF3, and if the PCIE3 bit and the global interrupt are set, a PCI3 interrupt will be generated. When the PCINT31 bit is set to "0", the PE7 pin level change interrupt is disabled.
6	PCINT30	Pin change enable mask bit 30. When the PCINT30 bit is set to "1", the PE6 pin level change interrupt is enabled. a level change on the PE6 pin will set PCIF3, and if the PCIE3 bit and the global interrupt are set, a PCI3 interrupt will be generated. When the PCINT30 bit is set to "0", the PE6 pin level change interrupt is disabled.
5	PCINT29	Pin change enable mask bit 39. When the PCINT29 bit is set to "1", the PE5 pin level change interrupt is enabled. a level change on the PE5 pin will set PCIF3, and if the PCIE3 bit and the global interrupt are set, a PCI3 interrupt will be generated. When the PCINT29 bit is set to "0", the PE5 pin level change interrupt is disabled.

4	PCINT28	Pin change enable mask bit 28. When the PCINT28 bit is set to "1", the PE4 pin level change interrupt is enabled. a level change on the PE4 pin will set PCIF3, and if the PCIE3 bit and the global interrupt are set, a PCI3 interrupt will be generated. When the PCINT28 bit is set to "0", the PE4 pin level change interrupt is disabled.
3	PCINT27	Pin change enable mask bit 27.

		When the PCINT27 bit is set to "1", the PE3 pin level change interrupt is enabled. a level change on the PE3 pin will set PCIF3, and if the PCIE3 bit and the global interrupt are set, a PCI3 interrupt will be generated. When the PCINT27 bit is set to "0", the PE3 pin level change interrupt is disabled.
2	PCINT26	Pin change enable mask bit 26. When the PCINT26 bit is set to "1", the PE2 pin level change interrupt is enabled. a level change on the PE2 pin will set PCIF3, and if the PCIE3 bit and the global interrupt are set, a PCI3 interrupt will be generated. When the PCINT26 bit is set to "0", the PE2 pin level change interrupt is disabled.
1	PCINT25	Pin change enable mask bit 25. When the PCINT25 bit is set to "1", the PE1 pin level change interrupt is enabled. a level change on the PE1 pin will set PCIF3, and if the PCIE3 bit and the global interrupt are set, a PCI3 interrupt will be generated. When the PCINT25 bit is set to "0", the PE1 pin level change interrupt is disabled.
0	PCINT24	Pin change enable mask bit 24. When the PCINT24 bit is set to "1", the PE0 pin level change interrupt is enabled. a level change on the PE0 pin will set PCIF3, and if the PCIE3 bit and the global interrupt are set, a PCI3 interrupt will be generated. When the PCINT24 bit is set to "0", the PE0 pin level change interrupt is disabled.

PCMSK4 - Pin Change Interrupt Mask Register 4

PCMSK4 – Pin Change 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. When the PCINT39 bit is set to "1", the PF7 pin level change interrupt is enabled. a level change on the PF7 pin will set PCIF4, and if the PCIE4 bit and the global interrupt are set, a PCI4 interrupt will be generated. When the PCINT39 bit is set to "0", the PF7 pin level change interrupt is disabled.
6	PCINT38	Pin change enable mask bit 38. When the PCINT38 bit is set to "1", the PF6 pin level change interrupt is enabled. a level change on the PF6 pin will set PCIF4, and if the PCIE4 bit and the global interrupt are set, a PCI4 interrupt will be generated. When the PCINT38 bit is set to "0", the PF6 pin level change interrupt is disabled.

5	PCINT37	Pin change enable mask bit 37. When the PCINT37 bit is set to "1", the PF5 pin level change interrupt is enabled. a level change on the PF5 pin will set PCIF4, and if the PCIE4 bit and the global interrupt are set, a PCI4 interrupt will be generated. When the PCINT37 bit is set to "0", the PF5 pin level change interrupt is disabled.
4	PCINT36	Pin change enable mask bit 36.

		When the PCINT36 bit is set to "1", the PF4 pin level change interrupt is enabled. a level change on the PF4 pin will set PCIF4, and if the PCIE4 bit and the global interrupt are set, a PCI4 interrupt will be generated. When the PCINT36 bit is set to "0", the PF4 pin level change interrupt is disabled.
3	PCINT35	Pin change enable mask bit 35. When the PCINT35 bit is set to "1", the PF3 pin level change interrupt is enabled. a level change on the PF3 pin will set PCIF4, and if the PCIE4 bit and the global interrupt are set, a PCI4 interrupt will be generated. When the PCINT35 bit is set to "0", the PF3 pin level change interrupt is disabled.
2	PCINT34	Pin change enable mask bit 34. When the PCINT34 bit is set to "1", the PF2 pin level change interrupt is enabled. a level change on the PF2 pin will set PCIF4, and if the PCIE4 bit and the global interrupt are set, a PCI4 interrupt will be generated. When the PCINT34 bit is set to "0", the PF2 pin level change interrupt is disabled.
1	PCINT33	Pin change enable mask bit 33. When the PCINT33 bit is set to "1", the PF1 pin level change interrupt is enabled. a level change on the PF1 pin will set PCIF4, and if the PCIE4 bit and the global interrupt are set, a PCI4 interrupt will be generated. When the PCINT33 bit is set to "0", the PF1 pin level change interrupt is disabled.
0	PCINT32	Pin change enable mask bit 32. When the PCINT31 bit is set to "1", the PF0 pin level change interrupt is enabled. a level change on the PF0 pin will set PCIF4, and if the PCIE4 bit and the global interrupt are set, a PCI4 interrupt will be generated. When the PCINT32 bit is set to "0", the PF0 pin level change interrupt is disabled.

Timer/Counter *0 (TMR0)*

- 8-bit counter
- Two separate comparison units
- Automatically clears the counter and automatically loads it when a comparison match occurs
- Phase-corrected PWM output without interference pulses
- Frequency generator
- External event counter
- 10-bit clock prescaler
- Overflow and compare match interrupts
- With dead time control
- 6 selectable trigger sources automatically turn off the PWM output
- High-speed clock mode generates high-speed, high-resolution (500KHz@7Bit) PWM

summarize

TC0 is a general-purpose 8-bit timer counter module that supports PWM output for accurate waveform generation.TC0 contains a count clock generation unit, an 8-bit counter, waveform generation mode control unit and two output comparison units. Meanwhile, TC0 can share a 10-bit prescaler with TC1, or use a 10-bit prescaler independently. The prescaler is a 2x multiplier of the system clock clkio or the high-speed clock rcm2x (internal 32M RC oscillator output clock rc32m)
The waveform generation mode control unit controls the counter's operating mode and the generation of the comparison output waveform. Clkt0 can be generated by internal or external clock source. When the counter count value TCNT0 reaches the maximum value (equal to the maximum value 0xFF or the output comparison register OCR0A, defined as TOP, defines the maximum value as MAX to indicate the difference), the counter is cleared or decremented by one. When the counter count value TCNT0 reaches the minimum value (equal to 0x00, defined as BOTTOM), the counter will perform a plus one operation. When the counter's count value TCNT0 reaches OCR0A/OCR0B, also known as when a comparison match occurs, it will clear or set the output comparison signal OC0A/OC0B to generate the PWM waveform. When the insertion dead time is enabled, the set dead time (the number of count clocks corresponding to the DTR0 register) will be inserted into the generated PWM waveform. The software can turn off the waveform output of OC0A/OC0B by clearing the COM0A/COM0B bits to zero, or set the corresponding trigger source, and the hardware will automatically clear the COM0A/COM0B bits to turn off the waveform output of OC0A/OC0B when the trigger event occurs.

The counting clock can b e generated from an internal or external clock source, and the selection of the clock source and the division of the frequency is made by the TCCR0B register located in the
CS0 bit to control, see TC0 and TC1 prescaler chapters for detailed description.

The counter is 8 bits long and supports bi-directional counting. The waveform generation mode, i.e. the counter's operating mode, is controlled by the WGM0 bits located in the TCCR0A and TCCR0B registers. Depending on the operating mode, the counter implements a clear, plus one or minus one operation for each count clock Clkt0. The count overflow flag TOV0 bit in the TIFR0 register is set when a count overflow occurs. A TC0 count overflow interrupt can be generated when the interrupt is enabled.

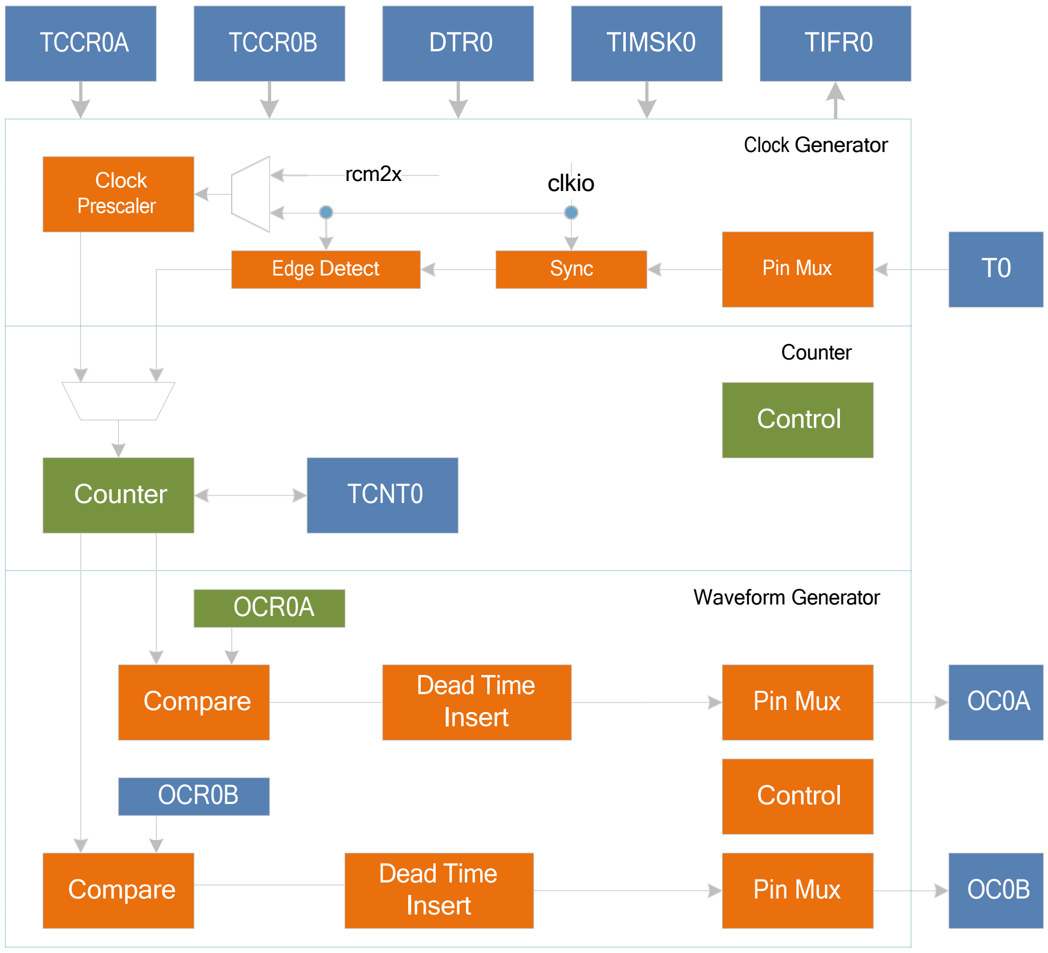
The output compare unit compares the count value TCNT0 with the values of the output compare registers OCR0A and OCR0B. When TCNT0 equals OCR0A or OCR0B, a compare match is called to occur and the output compare flag OCF0A or OCF0B bit located in the TIFR0 register is set. A TC0 output compare match interrupt can be generated when the interrupt is enabled.

Note that in PWM operation mode, the OCR0A and OCR0B registers are double-buffered registers. In the normal mode and

In CTC mode, the double buffer function is disabled. When the count reaches the maximum or minimum value, the value in the buffer register is updated to the comparison registers OCR0A and OCR0B synchronously. See the description in the Operating Modes section for details.

The waveform generator generates the output compare waveform signals OC0A and OC0B using compare match and count overflow, etc. according to the waveform generation mode control and compare output mode control. the specific generation mode is described in the Operating Mode and Registers section. To output the output comparison waveform signals OC0A and OC0B to the corresponding pins, the data direction register of the pin must also be set to output.

The following diagram shows the internal structure of TC0, which contains one count clock generation unit, one 8-bit counter, two output comparison units and two waveform generation control units.



TC0 Structure Diagram

workin g mode

Timing Counter 0 has four different operating modes, including Normal mode, Clear on Compare Match (CTC) mode, Fast Pulse Width Modulation (FPWM) mode and Phase Correction Pulse Width Modulation (PCPWM) mode, selected by the Waveform Generation Mode control bits WGM0[2:0]. These four modes are described in detail below. Since there are two independent output comparison units, denoted by "A" and "B" respectively, the two output comparison unit channels are denoted by lowercase "x".

normal mode

Normal mode is the simplest mode of operation of the timer counter, in which the waveform generation mode control bits WGM0[2:0]=0 and the maximum value of the count is MAX (0xFF) In this mode, the count is incremented by one for each count clock, and when the counter reaches TOP overflow, it returns to BOTTOM and starts accumulating again. The timer overflow flag TOV0 is set in the same count clock where the count value TCNT0 goes to zero; in this mode the TOV0 flag is like the 9th count bit, except that it is only set and not cleared. The overflow interrupt service routine automatically clears the TOV0 flag, which can be used by software to increase the resolution of the timer. There are no special circumstances to consider in normal mode, and a new count value can be written at any time.
The waveform of the output comparison signal OC0x can be obtained only when the data direction register of the OC0x pin is set to output.When COM0x=1
When a comparison match occurs, the OC0x signal is flipped and the frequency of the waveform in this case can be calculated using the following equation.

$f_{oc0xnormal} = fsys/(2*N*256)$

where N denotes the prescaling factor (1, 8, 64, 256 or 1024)
The output comparison unit can be used to generate interrupts, but interrupts are not recommended in normal mode, as they can take up too much CPU
of time.

CTC model

When WGM0[2:0]=2 is set, Timer 0 enters CTC mode, and the maximum value of count TOP is OCR0A. In this mode, the count is incremented by one for each count clock, and the counter is cleared when the counter value TCNT0 equals TOP. OCR0A defines the maximum value of count, which is also the resolution of the counter. This mode allows the user to easily control the frequency of the compare match output and also simplifies the operation of the external event count.
When the counter reaches the maximum value of the count, the output compare match flag OCF0 is set and an interrupt will be generated when the corresponding interrupt enable is set. The OCR0A register, the maximum value of the count, can be updated in the interrupt service program. In this mode OCR0A does not use double buffering, so be careful when updating the maximum value to near minimum with the counter operating with no prescaler or very low prescaler. If the value written to OCR0A is less than the TCNT0 value at the time, the counter will lose a compare match. The counter has to count to TOP and then to the OCR0A value starting from BOTTOM before the next compare match occurs. As in normal mode, the count value returns to the BOTTOM with the TOV0 flag set in the count clock. The waveform of the output comparison signal OC0x can be obtained only when the data direction register of the OC0x pin is set to output. When COM0x=1, the comparison match will flip the OC0x signal, in which case the frequency of the waveform can be calculated by the following formula.

$f_{oc0xctc} = fsys/(2*N*(1+OCR0x))$

where N denotes the prescaling factor (1, 8, 64, 256 or 1024)
From the equation, it can be seen that when setting OCR0A to 0x0 and no prescaler, an output waveform with a maximum frequency of fsys/2 can be obtained.

Fast PWM Mode

When WGM0[2:0]=3 or 7 is set, Timer 0 enters the Fast PWM mode, which can be used to generate high frequency PWM waveforms with a maximum count value of TOP of MAX(0xFF) or OCR0x, respectively. The counters accumulate from the minimum value of 0x00 to TOP and then return to BOTTOM to recount. When the count value TCNT0 reaches OCR0x or BOTTOM, the output compare

signal OC0x is set or cleared, depending on the compare output mode COM0x setting, as detailed in the register description. Due to the unidirectional operation, the fast PWM mode operates at twice the frequency of the phase correction PWM mode with bidirectional operation. The high frequency feature makes the fast PWM mode suitable for power regulation, rectification, and DAC applications. The high-frequency signal reduces the size of external components (inductors, capacitors, etc.), thus reducing system cost.
When the count value reaches its maximum value, the timer overflow flag TOV0 will be set and the value of the comparison buffer will be updated

to the compare value. If the interrupt is enabled, the compare buffer OCR0x register can be updated in the interrupt service program.

The waveform of the output comparison signal OC0x can be obtained only when the data direction register of the OC0x pin is set to output. The frequency of the waveform can be calculated by the following equation.

$f_{oc0xfpwm}$ = fsys/(N*(1+TOP))

where N denotes the prescaling factor (1, 8, 64, 256 or 1024)

When TCNT0 and OCR0x are matched by comparison, the waveform generator sets (clears) the OC0x signal, and when TCNT0 is cleared, the waveform generator clears (sets) the OC0x signal to generate a PWM waveform. The resulting polar value of OCR0x will generate a special PWM waveform. When OCR0x is set to 0x00, the output PWM is a narrow spike pulse for every (1+TOP) count clock. When OCR0x is set to the maximum value, the output waveform is a continuous high or low level.

Phase Correction PWM Mode

When WGM0[2:0]=1 or 5 is set, Timer 0 enters phase correction PWM mode, and the maximum value of count TOP is MAX(0xFF) or OCR0A, respectively. The counter operates in both directions, incrementing from BOTTOM to TOP, then decrementing to BOTTOM, and then repeating this operation. The count changes direction when it reaches both TOP and BOTTOM, and the count value stays on TOP or BOTTOM for only one count clock. When the count value TCNT0 matches OCR0x during incrementing or decrementing, the output comparison signal OC0x will be cleared or set, depending on the setting of the comparison output mode COM0x. Compared to unidirectional operation, the maximum frequency available for bidirectional operation is smaller, but its excellent symmetry is more suitable for motor control.

The phase correction PWM mode sets the TOV0 flag when the count reaches BOTTOM and updates the comparison buffer value to the comparison value when the count reaches TOP. If the interrupt is enabled, the comparison buffer OCR0x register can be updated in the interrupt service program.

The waveform of the output comparison signal OC0x can be obtained only when the data direction register of the OC0x pin is set to output. The frequency of the waveform can be calculated by the following equation.

$f_{oc0xpcpwm}$ = fsys/(N*TOP*2)

where N denotes the prescaling factor (1, 8, 64, 256 or 1024)

During incremental counting, the waveform generator clears (sets) the OC0x signal when TCNT0 matches OCR0x. During decrement counting, the waveform generator sets (clears) the OC0x signal when TCNT0 matches OCR0x. The resulting extreme value of OCR0x generates a special PWM waveform. When OCR0x is set to the maximum or minimum value, the OC0x signal output will remain low or high.

To ensure symmetry of the output PWM wave on both sides of the minimum value, there are two cases where the OC0x signal is also flipped when no comparison matching occurs. The first case is when the value of OCR0x changes from the maximum value 0xFF to other data. When OCR0x is the maximum value and the count value reaches its maximum, the output of OC0x is the same as the result of the comparison match during the previous descending count, i.e., OC0x remains unchanged. At this point the comparison value is updated to the new OCR0x value (not 0xFF), and the OC0x value is held until it is flipped when the comparison match occurs during ascending counting. At this point, the OC0x signal is not centered symmetrically on the minimum value, so it is necessary to flip the OC0x signal when TCNT0 reaches its maximum value, which is the first case of flipping the OC0x signal when a comparison match does not occur. The second case is when TCNT0 starts counting from a value higher than OCR0x, thus losing a comparison match and causing an asymmetric situation. Again, the OC0x signal needs to be flipped to achieve symmetry on both sides of the minimum.

Automatic shutdown and restart of PWM output

When the DOC0x bit of TCCR0A register is set high, the auto-off function of PWM output will be enabled. When the trigger condition is met, the hardware will clear the corresponding COM0x bit, disconnect the PWM output signal OC0x from its output pin, and switch to the general-purpose IO output to realize the auto-off of PWM output. At this time, the state of the output pins can be controlled by the output of the general-purpose IO port.

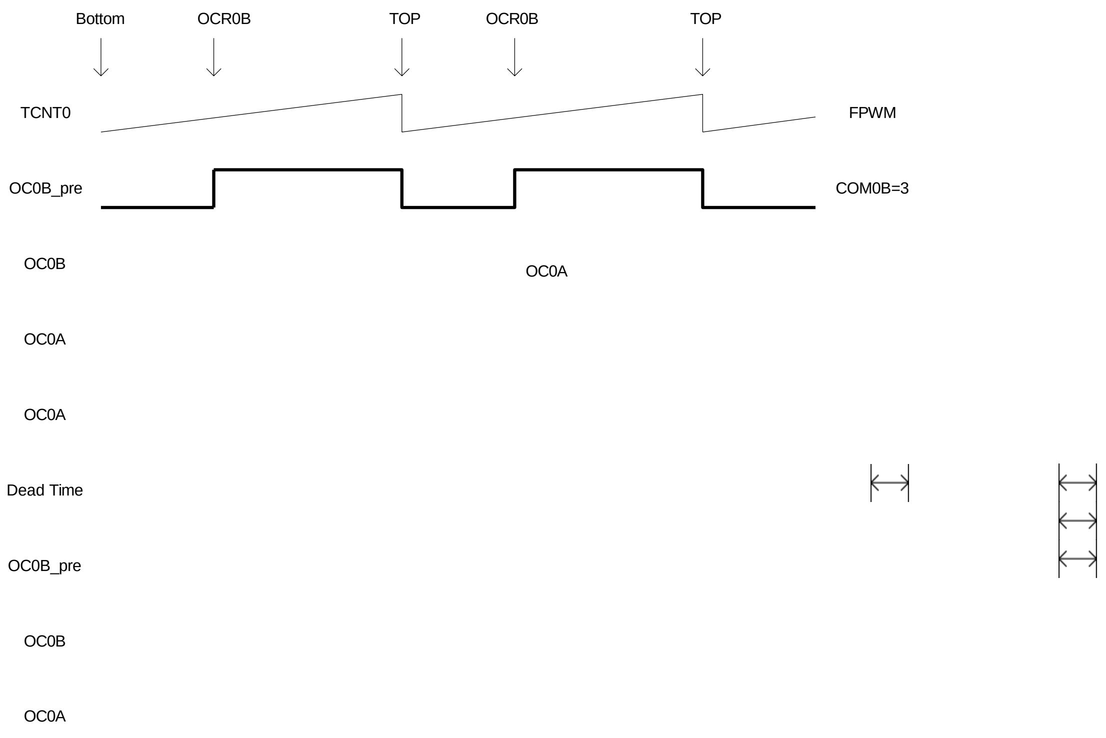
After the PWM output auto-off is enabled, the trigger condition needs to be set, and the DSX0n bit of the TCCR0C register is used to select the trigger source. The trigger sources are analog comparator interrupt, external interrupt, pin level change interrupt, and timer overflow interrupt, as described in t h e TCCR0C register. When one or more trigger sources are selected as trigger conditions, the hardware will clear the COM0x bit to turn off the PWM output while these interrupt flag bits are set.

When a trigger event occurs to turn off the PWM output, the timer module does not have the corresponding interrupt flag bit, and the software needs to read the interrupt flag bit of the trigger source to know the trigger condition and the trigger event.

When the PWM output is automatically turned off and the output needs to be restarted again, the software simply resets the COM0x bit to switch the OC0x signal output to the appropriate pin. Note that the timer does not stop working after an automatic shutdown occurs, and the state of the OC0x signal is always updated. The software can set the COM0x bit again to output the OC0x signal after a timer overflow or comparison match occurs, so that a clear PWM output state can be obtained.

Dead time control

When DTEN0 is set to "1", the function of inserting dead time is enabled, and the output waveforms of OC0A and OC0B will insert the set dead time based on the waveform generated by the comparison output of channel B. The length of the time is the time value corresponding to the count clock number of DTR0 register. As shown in the figure below, the dead time insertion of both OC0A and OC0B is based on the comparison output waveform of channel B. When COM0A and COM0B are both "2" or "3", the waveform polarity of OC0A is the same as that of OC0B; when COM0A and COM0B are "2" or "3" respectively, the waveform polarity of OC0A is the same as that of OC0B. When COM0A and COM0B are "2" or "3" respectively, the waveform polarity of OC0A is opposite to that of OC0B.



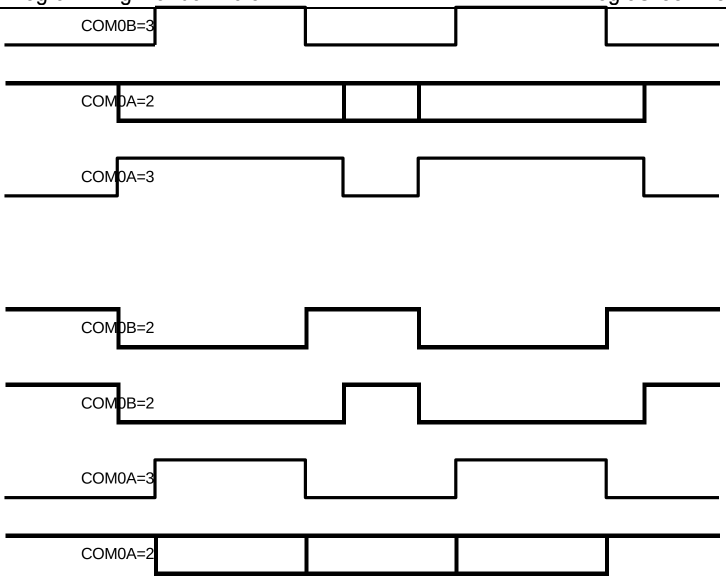


Figure 1 TC0 Dead Time Control in FPWM Mode

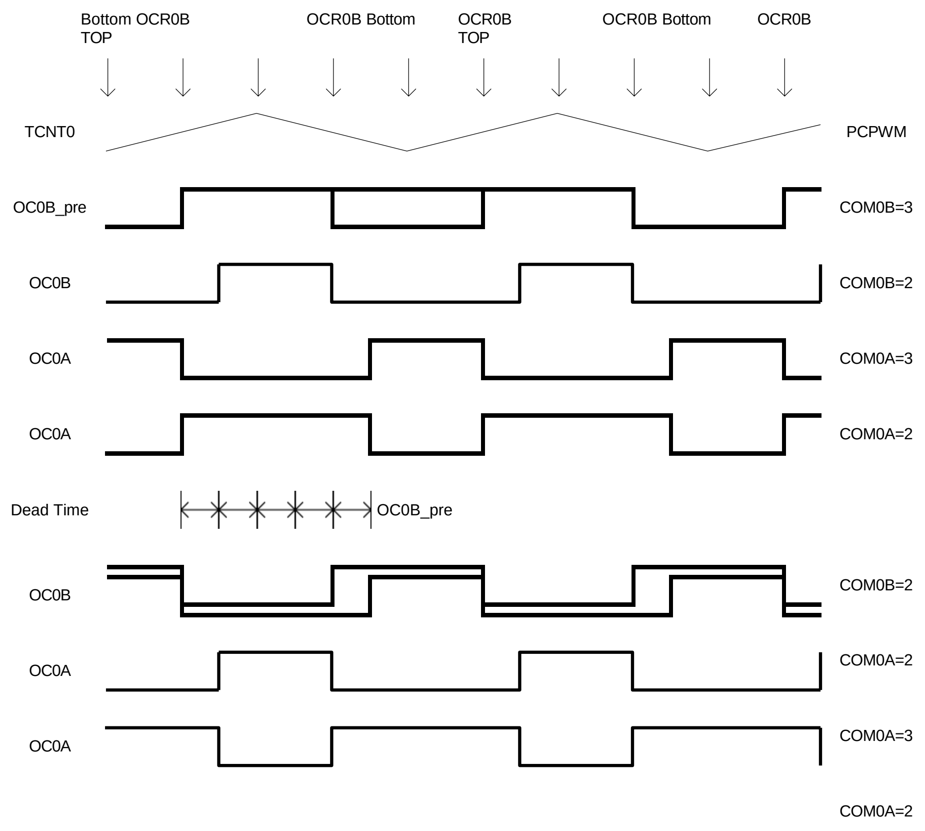


Figure 2 TC0 Dead Time Control in PCPWM Mode

When DTEN0 is set to "0", the function of inserting dead time is disabled, and the output waveforms of OC0A and OC0B are the waveforms generated by their respective comparison outputs.

High-speed clock mode

The high speed clock mode uses a higher frequency clock as the clock source for counting, which is used to generate higher speed and higher resolution PWM waveforms. This high-frequency clock is generated by doubling the output clock of the internal 32M RC oscillator, rc32m. Therefore, before entering the high frequency mode, the internal 32M RC oscillator needs to be enabled for frequency doubling, i.e., set the F2XEN bit of the TCKCSR register and wait for a certain time until the output of the doubled clock signal is stable. Then, the TC2XS0 bit of TCKCSR can be set to put the timer into high speed clock mode.

In this mode, the system clock is asynchronous to the high-speed clock, and some of the registers (see TC0 register list) are operating in the high-speed clock domain, so the configuration and reading of such registers is also asynchronous, and care needs to be taken when operating them.

There are no special requirements for non-sequential read and write operations to registers under the high-speed clock domain, while for sequential read and write operations, you need to wait for a system clock, which can be done as follows.

1) Write register A.
2) Waiting for a system clock (NOP or register under the OS clock)
3) Read or write register A or B.
4) Wait for a system clock (NOP or register under OS clock)

When reading the registers under the high-speed clock domain, all registers except TCNT0 can be read directly. When the counter is still counting, the value of TCNT0 will change with the high-speed clock, and you can pause the counter (set CS0 to zero) and then read the value of TCNT0.

Register Definition

TC0 Register List

processor register	address	default value	description
TCCR0A*	0x44	0x00	TC0 control register A
TCCR0B*	0x45	0x00	TC0 Control Register B
TCNT0*	0x46	0x00	TC0 Count Value Register
OCR0A*	0x47	0x00	TC0 Output comparison register A
OCR0B*	0x48	0x00	TC0 Output comparison register B
DSX0*	0x49	0x00	TC0 Trigger Source Control Register
DTR0*	0x4F	0x00	TC0 Dead Time Register
TIMSK0	0x6E	0x00	Timing counter 0 Interrupt mask register
TIFR0	0x35	0x00	Timing counter 0 Interrupt flag register
TCKCSR	0xEC	0x00	TC Clock Control and Status Register

[Note]
Registers with "*" work under the system clock and high-speed clock domain, registers without "*" work under the system clock domain only.

TC0 Control Register A- TCCR0A

TCCR0A -TC0 Control Register A								
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	description
7	COM0A1	TC0 Compare Match A Output Mode Control High. COM0A1 and COM0A0 together form the compare output mode control COM0A[1:0], which is used to control the output waveform of OC0A. If either bit 1 or bit 2 of COM0A is set, the output compare waveform occupies the OC0A pin, but the data direction register of this pin must be set high to output this waveform. The control of the output compare waveform by COM0A is different in different operating modes, as described in the comparison output mode control table.

6	COM0A0	TC0 Compare Match A Output Mode Control Low. COM0A0 and COM0A1 together form the compare output mode control COM0A[1:0], which is used to control the output waveform of OC0A. If either bit 1 or bit 2 of COM0A is set, the output comparison waveform occupies the OC0A pin, but the data direction register of this pin must be set high to output this waveform. In different operating modes, COM0A has the same effect on the output comparison waveform. The control of the comparison waveform is also different, as described in the comparison output mode control table.
5	COM0B1	TC0 Compare Match B Output Mode Control High. COM0B1 and COM0B0 together form the compare output mode control COM0B[1:0], which is used to control the output waveform of OC0B. If either bit 1 or bit 2 of COM0B is set, the

		The output compare waveform occupies the OC0B pin, although the data direction register of this pin must be set high to output this waveform. COM0B controls the output compare waveform differently in different operating modes, as described in the compare output mode control table.
4	COM0B0	TC0 Compare Match B Output Mode Control Low. COM0B0 and COM0B1 together form the compare output mode control COM0B[1:0], which is used to control the output waveform of OC0B. If either bit 1 or bit 2 of COM0B is set, the output compare waveform occupies the OC0B pin, but the data direction register of this pin must be set high to output this waveform. The control of COM0B on the output compare waveform differs in different operating modes, as described in the Compare Output Mode Control Table.
3	DOC0B	TC0 Turn off the output compare enable control high. When the DOC0B bit is "1", the trigger source off output comparison signal OC0B is enabled. When a trigger event occurs, the hardware automatically clears the COM0B bit to turn off the waveform output of OC0B. The software can turn on the PWM output again by setting COMB. When the DOC0B bit is "0", the trigger source off output comparison signal OC0B is disabled.
2	DOC0A	TC0 turns off the output compare enable control low. When the DOC0A bit is set to "1", the trigger source off output comparison signal OC0A is enabled. When a trigger event occurs, the hardware automatically turns off the waveform output of OC0A. When the DOC0A bit is set to "0", the trigger source is turned off and the output comparison signal OC0A is disabled. Stop. When a trigger event occurs, the waveform output of OC0A will not be turned off.
1	WGM01	TC0 Waveform Generation Mode Control Neutral. WGM01 and WGM00, WGM02 together form the waveform generation mode control WGM0[2:0], which controls how the counter counts and how the waveform is generated,as specified in the waveform Generate pattern table descriptions.
0	WGM00	TC0 Waveform Generation Mode Control Low. WGM00 together with WGM01 and WGM02 form the waveform generation mode control WGM0[2:0], which controls the counter counting mode and waveform generation mode, as described in the waveform generation mode table.

TC0 Control Register B- TCCR0B

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

Bit	Name	description
7	FOC0A	TC0 forces the output to compare the A control bit. When operating in non- PWM mode, a compare match can be generated by writing a "1" to the forced output compare bit FOC0A. The forced compare match will not set the OCF0A flag, nor will it reload or clear the timer, but output pin OC0A will be updated accordingly to the COM0A setting, just as if a compare match had actually occurred. The return value of reading FOC0A is always zero.
6	FOC0B	TC0 forces the output to compare the B control bit.

		When operating in non- PWM mode, a compare match can be generated by writing a "1" to the forced output compare bit FOC0B. The forced compare match will not set the OCF0B flag, nor will it reload or clear the timer, but output pin OC0B will be updated accordingly to the COM0B setting, just as if a compare match had actually occurred. The return value of reading FOC0B is always zero.
5	OC0AS	OC0A output port selection control bit. When OC0AS bit is set to "0", the waveform of OC0A is output from pin PD6; when OC0AS bit is set to "1", the waveform of OC0A is output from pin PE4 (valid under QFP32 package)
4	DTEN0	TC0 Dead time enable control bit. When DTEN0 is set to "1", the dead time insertion is enabled.OC0A and OC0B both insert the dead time on top of the waveform generated by the B channel comparison output, and the interval of the inserted dead time is determined by the count time corresponding to the DTR0 register. The polarity of the OC0A output waveform is determined by the corresponding relationship between COM0 and COM0B, as shown in the table of waveform polarity after OC0A inserts the dead time. When the DTEN0 bit is set to "0",dead time insertion is disabled and the OC0A and OC0B The waveform is the waveform generated by the respective comparison output.
3	WGM02	TC0 Waveform Generation Mode Control High. WGM02 together with WGM00 and WGM01 form the waveform generation mode control WGM0[2:0], which controls the counting mode of the counter and the waveform generation mode,as described in the waveform generation mode table.
2	CS02	TC0 Clock Select Control High. Used to select the clock source for timer counter 0.
1	CS01	TC0 Clock Select Control Neutral. Used to select the clock source for timer counter 0.
0	CS00	TC0 Clock Select Control Low. Used to select the clock source for timer counter 0.

CS0[2:0]	description
0	No clock source, stop counting
1	clksys
2	clksys/8, from prescaler
3	clksys/64, from prescaler
4	clksys/256, from prescaler
5	clksys/1024 from prescaler
6	External clock T0 pin, falling edge triggered
7	External clock T0 pin, rising edge triggered

The following table shows the control of the comparison output mode on the output comparison waveform in non- PWM modes (i.e. normal mode and CTC mode).

COM0x[1:0]	description
0	OC0x disconnected, general purpose IO port operation
1	Flip OC0x signal when comparing matches
2	Clear OC0x signal when comparing matches
3	Set OC0x signal when comparing matches

The following table shows the control of the comparison output mode on the output comparison waveform in fast PWM mode.

COM0x[1:0]	description
0	OC0x disconnected, general purpose IO port operation
1	retain
2	Clear OC0x signal for comparison match, set OC0x signal for maximum match
3	Set OC0x signal for comparison match, clear OC0x signal for maximum match

The following table shows the control of the output comparison waveform by the comparison output mode in phase correction mode.

COM0x[1:0]	description
0	OC0x disconnected, general purpose IO port operation
1	retain
2	Clear OC0x signal when comparing matches in ascending count, set OC0x signal when comparing matches in descending count
3	Set OC0x signal when comparing matches in ascending count, clear OC0x signal when comparing matches in descending count

The following table shows the waveform generation mode control.

WGM0[2:0]	working mode	TOP Value	Update OCR0X time	Position TOV0 moment
0	Normal	0xFF	immediately	MAX
1	PCPWM	0xFF	TOP	BOTTOM
2	CTC	OCR0A	immediately	MAX
3	FPWM	0xFF	TOP	MAX
4	retain	-	-	-
5	PCPWM	OCR0A	TOP	BOTTOM
6	retain	-	-	-
7	FPWM	OCR0A	TOP	TOP

The following table shows the polarity control of the OC0A signal output waveform when the dead time is enabled.

Polarity Control of OC0A Signal Output Waveform in Dead Time Enable Mode

DTEN0	COM0A[1:0]	COM0B[1:0]	description
0	-	-	OC0A signal polarity is controlled by the OC0A compare output mode
1	0	-	OC0A disconnected, general purpose IO port operation
1	1	-	retain
1	2	2	OC0A signal has the same polarity as OC0B signal
		3	The OC0A signal is opposite in polarity to the OC0B signal

1	3	2	The OC0A signal is opposite in polarity to the OC0B signal
		3	OC0A signal has the same polarity as OC0B signal

[Attention].

The polarity of the OC0B signal output waveform is controlled by the OC0B compare output mode, the same as the unenabled dead time mode.

TC0 Control Register C - TCCR0C

TCCR0C - *TC0* Control Register *C*								
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	description
7	DSX07	TC0 Trigger Source Select Control Enable Bit 7. When the DSX07 bit is set to "1", TC1 overflow is enabled as the trigger source for turning off the output comparison signal waveform OC0A/OC0B. When the DOC0A/DOC0B bit is "1", the rising edge of the interrupt flag register bit of the selected trigger source will automatically turn off the waveform output of OC0A/OC0B. When the DSX07 bit is set to "0", TC1 overflow is used to close the output comparison signal wave. The trigger source of shape OC0A/OC0B is disabled.
6	DSX06	TC0 Trigger Source Select Control Enable Bit 6. When the DSX06 bit is set to "1", TC2 overflow is enabled as the trigger source to turn off the output comparison signal waveform OC0A/OC0B. When the DOC0A/DOC0B bit is "1", the rising edge of the interrupt flag register bit of the selected trigger source will automatically turn off the waveform output of OC0A/OC0B. When the DSX06 bit is set to "0", TC2 overflow is used to close the output comparison signal wave. The trigger source of shape OC0A/OC0B is disabled.
5	DSX05	TC0 Trigger Source Select Control Enable Bit 5. When the DSX05 bit is set to "1", the pin level change 0 is enabled as the trigger source for turning off the output comparison signal waveform OC0A/OC0B. When the DOC0A/DOC0B bit is "1", the rising edge of the interrupt flag register bit of the selected trigger source will automatically turn off the waveform output of OC0A/OC0B. When the DSX05 bit is set to "0", the pin level changes by 0 as an off output comparison The trigger source of the signal waveform OC0A/OC0B is disabled.
4	DSX04	TC0 Trigger Source Select Control Enable Bit 4. When the DSX04 bit is set to "1", external interrupt 0 is enabled as the trigger source for turning off the output comparison signal waveform OC0A/OC0B. When the DOC0A/DOC0B bit is "1", the rising edge of the interrupt flag register bit of the selected trigger source will automatically turn off the waveform output of OC0A/OC0B. When the DSX04 bit is set to "0", external interrupt 0 is used to turn off the

		output comparison signal The trigger source of waveform OC0A/OC0B is disabled.
3:2	-	keep sth. unused
1	DSX01	TC0 Trigger Source Select Control Enable Bit 1. When the DSX01 bit is set to "1", Analog Comparator 1 is enabled as the trigger source for turning off the output comparison signal waveform OC0A/OC0B. When the DOC0A/DOC0B bit is "1", the rising edge of the interrupt flag register bit of the selected trigger source will automatically turn off the waveform output of OC0A/OC0B.

		When the DSX01 bit is set to "0", Analog Comparator 1 is disabled as the trigger source for turning off the output comparison signal waveform OC0A/OC0B.
0	DSX00	TC0 Trigger Source Select Control Enable Bit 0. When the DSX00 bit is set to "1", analog comparator 0 is enabled as the trigger source for turning off the output comparison signal waveform OC0A/OC0B. When the DOC0A/DOC0B bit is "1", the rising edge of the interrupt flag register bit of the selected trigger source will automatically turn off the waveform output of OC0A/OC0B. When the DSX00 bit is set to "0", analog comparator 0 is used to turn off the output comparison signal. The trigger source of waveform OC0A/OC0B is disabled.

The following table shows the selection control of the trigger source for the waveform output.

Turn off trigger source selection control for OC0A/OC0B waveform output

DOC0x	DSX0n=1	trigge r sourc e	des crip tion
0	-	-	DOC0x bit is "0", the trigger source off waveform output function is disabled
1	0	Analog Comparator 0	The rising edge of ACIF0 will turn off the OC0x waveform output
1	1	Analog comparator 1	The rising edge of ACIF1 will turn off the OC0x waveform output
1	4	External interrupt 0	The rising edge of INTF0 will turn off the OC0x waveform output
1	5	Pin level change 0	The rising edge of PCIF0 will turn off the OC0x waveform output
1	6	TC2 Overflow	The rising edge of TOV2 will turn off the OC0x waveform output
1	7	TC1 Overflow	The rising edge of TOV1 will turn off the OC0x waveform output

Caution.

1) DSX0n=1 means that when bit n of the DSX0 register is 1, each register bit can be set at the same time.

TC0 Count Value Register - TCNT0

TCNT0 -*TC0* Counter Value Register								
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	R/W	R/W
Bit	Name	description						

7:0	TCNT0	TC0 Count value register. The TCNT0 register allows direct read and write access to the counter's 8-count value.CPU writes to the TCNT0 register prevent a compare match from occurring on the next timer clock cycle, even if the timer has been stopped. This allows the TCNT0 register to be initialized to the same value as OCR0 without triggering an interrupt. If the value written to TCNT0 is equal to or bypasses the OCR0 value, the comparison match is lost, resulting in incorrect waveform generation results. The timer stops counting when no clock source is selected, but the CPU can still access TCNT0.CPU Write counters have higher priority than clear or add/drop operations.

TC0 Output Compare Register A- OCR0A

OCR0A - *TC0* Output Compare Register *A*								
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	R/W	R/W

Bit	Name	descri ption
7:0	OCR0A	TC0 Output comparison register. OCR0A contains an 8-bit data that is compared to the counter value TCNT0 without interruption. The compare match can be used to generate an output compare interrupt or to generate a waveform on the OC0A pin. When using PWM mode, the OCR0A register uses double-buffered registers. In contrast, the double-buffering function is disabled in normal operating mode and match clear mode. Double buffering synchronizes updating the OCR0A register with the count maximum or minimum moment, thus preventing the generation of asymmetrical PWM pulses and eliminating interference pulses. When using the double buffer function, the CPU accesses the OCR0A buffer register and disables the double buffer function. The CPU accesses the OCR0A itself when it can.

TC0 Output Compare Register B- OCR0B

OCR0B - TC0 Output Compare Register B								
Address: 0x48				Default value: 0x00				
Bit	7	6	5	4	3	2	1	0
Name	OCR0B7	OCR0B6	OCR0B5	OCR0B4	OCR0B3	OCR0B2	OCR0B1	OCR0B0
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	descri ption
7:0	OCR0B	TC0 output compares the B register. OCR0B contains an 8-bit data that is compared to the counter value TCNT0 without interruption. The compare match can be used to generate an output compare interrupt or to generate a waveform on the OC0B pin. When using PWM mode, the OCR0B register uses double-buffered registers. In contrast, the double buffering is disabled in normal operating mode and match clear mode. Double buffering synchronizes updating the OCR0B register with the count maximum or minimum moment, thus preventing the generation of asymmetrical PWM pulses and eliminating interference pulses. When using the double buffer function, the CPU accesses the OCR0B buffer

		register and disables double buffering The CPU accesses the OCR0B itself at the time of the function.

TC0 Interrupt Mask Register - TIMSK0

TIMSK0 - TC0 Interrupt Mask Register								
Address: 0x6E					Default value: 0x00			
Bit	7	6	5	4	3	2	1	0
	-	-	-	-	-	OCIE0B	OCIE0A	TOIE0
R/W	-	-	-	-	-	R/W	R/W	R/W

Bit	Name	description
7:3		Reserved.
2	OCIE0B	TC0 Output Compare B Match interrupt enable bit. When the OCIE0B bit is "1" and the global interrupt is set, TC0 outputs a compare B match interrupt enable. The interrupt is generated when a compare match occurs, i.e., when the OCF0B bit in TIFR0 is set. When the OCIE0B bit is "0", the TC0 output compare B match interrupt is disabled.
1	OCIE0A	TC0 Output Compare A Match interrupt enable bit. When the OCIE0A bit is "1" and the global interrupt is set, TC0 outputs a compare A match interrupt enable. The interrupt is generated when the compare match occurs, i.e., when the OCF0A bit in TIFR0 is set. When the OCIE0A bit is "0", the TC0 output compare A match interrupt is disabled.
0	TOIE0	TC0 Overflow interrupt enable bit. When the TOIE0 bit is "1" and the global interrupt is set, TC0 overflow interrupt is enabled. When the TC0 The interrupt is generated when an overflow occurs, i.e., when the TOV0 bit in the TIFR is set. When the TOIE0 bit is "0", the TC0 overflow interrupt is disabled.

TC0 Interrupt Flag Register - TIFR0

TIFR0 - TC0 Interrupt Flag Register								
Address: 0x35					Default value: 0x00			
Bit	7	6	5	4	3	2	1	0
	OC0A	OC0B	-	-	-	OCF0B	OCF0A	TOV0
R/W	R/O	R/O	-	-	-	R/W	R/W	R/W

Bit	Name	description

7	OC0A	Output the comparison waveform signal OC0A. The output comparison waveform signal, OC0A, can be read but not written by software.The software can read the value of OC0A bit to get the polarity of the comparison waveform signal to be output before enabling the OC0A signal to be output to its corresponding IO pin, and can change its polarity by configuring the COM0A bit and setting the FOC0A bit, so as to avoid the problem of outputting the OC0A signal before enabling the OC0A signal. to its corresponding IO pins and then generates excess interference pulses.
6	OC0B	Output the comparison waveform signal OC0B. The output comparison waveform signal OC0B is readable but not writable by software.The software can read the OC0B bit before enabling the OC0B signal to be output to its corresponding IO pin.

		value to obtain the polarity of the comparison waveform signal to be output, and can change its polarity by configuring the COM0B bit and setting the FOC0B bit to avoid excess interference pulses after enabling the OC0B signal to be output to its corresponding IO pin.
5:3		retain
2	OCF0B	TC0 output compares the B match flag bits. When TCNT0 equals OCR0B, the compare unit gives a match signal and sets the compare flag OCF0B. If the output compare B interrupt enable OCIE0B is "1" and the global interrupt flag is set, the output compare B interrupt will be generated. OCF0B will be cleared automatically when this interrupt service routine is executed, or by writing a "1" to the OCF0B bit.
1	OCF0A	TC0 output compares the A match flag bits. When TCNT0 is equal to OCR0A, the compare unit gives a match signal and sets the compare flag OCF0A. If the output compare A interrupt enable OCIE0A is "1" and the global interrupt flag is set, the output compare A interrupt is generated. OCF0A will be cleared automatically when this interrupt service routine is executed, or by writing a "1" to the OCF0A bit.
0	TOV0	TC0 Overflow flag bit. If the overflow interrupt enable TOIE0 is "1" and the global interrupt flag is set, an overflow interrupt will be generated. TOV0 will be cleared automatically when this interrupt service routine is executed, or by writing a "1" to the TOV0 bit.

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	description
[7:4]	DTR0H	TC0 Dead time register high. When the DTEN0 bit of the TCCR0B register is "1", OC0A and OC0B form a complementary output and the dead time control is enabled. The dead time inserted on the OC0B channel is determined by DTR0H and the length of time is the time corresponding to the DTR0H count clock.

[3:0]	DTR0L	TC0 Dead time register low. When the DTEN0 bit of the TCCR0B register is "1", OC0A and OC0B form a complementary output and the insertion dead time control is enabled, the dead time inserted on the OC0A channel is determined by DTR0L and the length of time is the time corresponding to the DTR0H count clock.

TCKCSR - TC Clock Control and Status Register

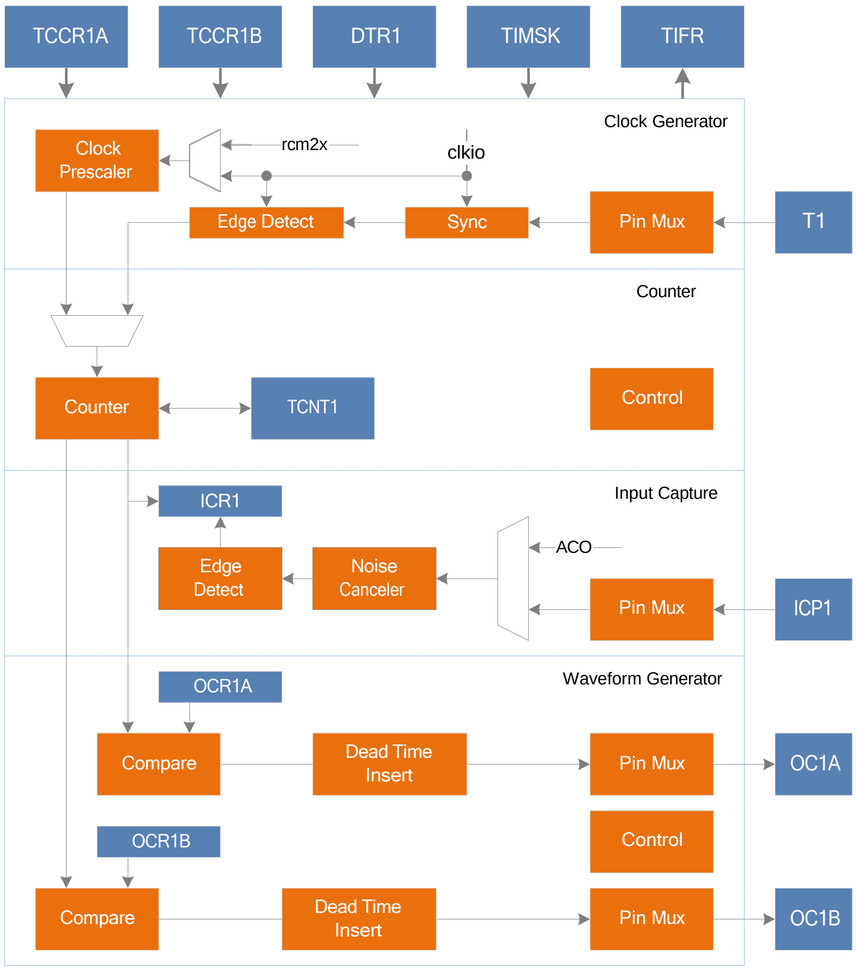
TCKSCR - TC Clock Control and Status Register								
Address: 0xEC					Default value: 0x00			
Bit	7	6	5	4	3	2	1	0
Name	-	F2XEN	TC2XF1	TC2XF0	-	AFCKS	TC2XS1	TC2XS0
R/W	-	R/W	R	R	-	R/W	R/W	R/W

Bit	Name	descri ption
7	-	retain
6	F2XEN	RC 32M multiplier output enable control bit. When the F2XEN bit is set to "1", the multiplier output of the 32M RC oscillator is enabled to output a 64M high-speed clock. When the F2XEN bit is set to "1", the multiplier output of the 32M RC oscillator is disabled. A 64M high-speed clock cannot be output.
5	TC2XF1	TC High-speed clock mode flag bit 1. See Timer Counter 1 register description.
4	TC2XF0	TC High-speed clock mode flag bit 0. When the TC2XF0 bit is read as "1", it indicates that Timer 0 is operating in high-speed clock mode, and when it is "0", it indicates that Timer 0 is operating in system clock mode.
3:2	-	Reserved.
1	TC2XS1	TC High-speed clock mode selection control bit 1. See Timer Counter 1 register description.
0	TC2XS0	TC High Speed Clock Mode Select Control Bit 0. When the TC2XS0 bit is set to "1", Timer 0 is selected to operate in high-speed clock mode. When the TC2XS0 bit is set to "0", Timer 0 is selected to operate on the system clock Model.

Timer/Counter 1 (TMR1)

- True 16-bit design, allowing 16-bit PWM
- 2 independent output comparison units
- Double-buffered output comparison register
- 1 input capture unit
- Input Capture Noise Suppressor
- Automatically clears the counter and automatically loads it when comparing matches
- PWM with phase correction without interference pulses
- Variable PWM Cycle
- Frequency generator
- External event counter
- 4 independent interrupt sources
- PWM with dead time control
- 6 selectable trigger sources automatically turn off the PWM output
- High-speed clock mode generates high-speed high-resolution (500KHZ@7BIT) PWM

summarize



TC1 Structure Diagram

TC1 is a general-purpose 16-bit timer counter module that supports PWM output for accurate waveform generation.TC1 contains a 16-bit counter, waveform generation mode control unit, two independent output comparison units and an input capture unit. TC1 can also share a 10-bit prescaler with TC0 or use a 10-bit prescaler independently. The prescaler divides the system clock clkio or the high-speed clock rcm2x (2 times the output clock rc32m of the internal 32M RC oscillator) to generate the counter clock Clkt1. The waveform generation mode control unit controls the counter's operation mode and the generation of the comparison output waveform. Depending on the operating mode, the counter implements zero, plus one or minus one operation for each count clock Clkt1.Clkt1 can be generated by internal or external clock source. When the counter count value TCNT1 reaches the maximum value (equal to the maximum value 0xFFFF or a fixed value or the output compare register OCR1A or the input capture register ICR1, defined as TOP, and the maximum value MAX to show the difference),the counter will perform a zero or minus one operation. When the counter count value TCNT1 reaches the minimum value (equal to 0x0000, defined as BOTTOM), the counter will perform a plus one operation. When the counter's count value TCNT1 reaches OCR1A or OCR1B, also known as when a comparison match occurs, it will clear or set the output comparison signal OC1A or OC1B to generate the PWM waveform. When the insertion dead time is enabled, the set dead time (the number of count clocks corresponding to the DTR1 register) will be inserted into the generated PWM waveform. When the input capture function is enabled, the counter is triggered to start or stop counting, and the ICR1 register will record the count value during the trigger cycle of the capture signal. The software can turn off the waveform output of OC1A/OC1B by clearing the COM1A/COM1B bit to zero, or set the corresponding trigger source, and the hardware will automatically clear the COM1A/COM1B bit to turn off the waveform output of OC1A/OC1B when the trigger event occurs.

The count clock can be generated from either an internal or external clock source. The selection of the clock source and the divider selection is controlled by the CS1 bit located in the TCCR1B register, as described in detail in the TC0 and TC1 prescaler chapters.

The counter is 16 bits long and supports bi-directional counting.The waveform generation mode, i.e., the counter's operating mode, is controlled by the WGM1 bits located in the TCCR1A and TCCR1B registers. Depending on the operating mode, the counter implements a clear, plus one or minus one operation for each count clock Clkt1. The count overflow flag TOV1 bit located in the TIFR1 register is set when a count overflow occurs. A TC1 count overflow interrupt can be generated when the interrupt is enabled.

The Output Compare Unit compares the count value TCNT1 with the values of the Output Compare Registers OCR1A and OCR1B. When TCNT1 equals OCR1A or OCR1B, a compare match is said to occur and the Output Compare Flag OCF1A or OCF1B bit located in the TIFR1 register is set. A TC1 output compare match interrupt can be generated when the interrupt is enabled.
Note that in PWM operation mode, the OCR1A and OCR1B registers are double-buffered registers. In the normal mode and CTC mode, the double buffer function is disabled. When the count reaches the maximum or minimum value, the value in the buffer register is updated to the comparison registers OCR1A and OCR1B synchronously. See the Operating Modes section description for details.

The waveform generator generates the output compare waveform signals OC1A and OC1B using compare match and count overflow, etc. according to the waveform generation mode control and compare output mode control. the details of the generation mode are described in the Operating Mode and Registers chapter. To output the output comparison waveform signals OC1A and OC1B to the corresponding pins, the data direction register of the pin must also be set to output.

working mode

Timing Counter 1 has six different operating modes, including Normal mode (Normal)Clear on Compare Match (CTC) mode, Fast Pulse Width Modulation (FPWM) mode, Phase Correction Pulse Width Modulation (PCPWM) mode, Phase Frequency Correction Pulse Width Modulation (PFCPWM) mode, and Input Capture (ICP) mode. The selection is made by the waveform generation mode control bits WGM1[3:0]. These six modes are described in detail below. Since there are two independent output comparison units, denoted by "A" and "B" respectively, a lowercase "x" is used to denote these two output comparison unit channels.

normal mode

Normal mode is the simplest mode of operation of the timer counter, in which the waveform generation mode control bits WGM1[3:0]=0 and the maximum value of the count is MAX (0xFFFF) In this mode, the count is incremented by one for each count clock, and when the counter reaches TOP overflow, it returns to BOTTOM and starts accumulating again. The timer overflow flag TOV1 is set in the same count clock where the count value TCNT1 goes to zero; in this mode the TOV1 flag is like the 17th count bit, except that it is only set and not cleared. The overflow interrupt service routine automatically clears the TOV1 flag, which can be used by software to increase the resolution of the timer. There are no special circumstances to consider in normal mode, and a new count value can be written at any time.

The waveform of the output comparison signal OC1x can be obtained only when the data direction register of the OC1x pin is set to output.When COM1x=1
When a comparison match occurs, the OC1x signal is flipped and the frequency of the waveform in this case can be calculated using the following equation.

$f_{oc1xnormal}$ = fsys/(2*N*65536)

where N denotes the prescaling factor (1, 8, 64, 256 or 1024)

The output comparison unit can be used to generate interrupts, but interrupts are not recommended in normal mode, as they can take up too much CPU
of time.

CTC model

When WGM1[3:0]=4 or 12 is set, Timer 1 enters CTC mode. When WGM1[3]=0, the maximum count value TOP is OCR1A, when WGM1[3]=1, the maximum count value TOP is ICR1. The following is an example of CTC mode with WGM1[3:0]=4. In this mode, the count mode is incremented by one for each count clock, and the counter is cleared when the counter value TCNT1 equals TOP. This mode allows the user to easily control the frequency of the compare match output, and also simplifies the operation of external event counting.

When the counter reaches TOP, the output compare match flag OCF1 is set and an interrupt will be generated when the corresponding interrupt enable is set. The OCR1A register can be updated in the interrupt service program. In this mode OCR1A does not use double buffering, so be careful when updating the maximum value to a value close to the minimum with the counter operating with no prescaler or a very low prescaler. If the value written to OCR1A is less than the then current TCNT1 value, the counter will lose a compare match. The counter has to count to MAX before the next compare match occurs, and then count from BOTTOM to OCR1A. as in normal mode, the count value returns to the count clock at 0x0 to set the TOV1 flag.

The waveform of the output comparison signal OC1x can be obtained only when the data direction register of the OC1x pin is set to output. The frequency of the waveform can be calculated using the following equation.

$f_{oc1xctc}$ = fsys/(2*N*(1+OCR1A))

where N denotes the prescaling factor (1, 8, 64, 256 or 1024)
From the equation, it can be seen that when setting OCR1A to 0x0 and no prescaler, an output waveform with a maximum frequency of fsys/2 can be obtained.

When WGM1[3:0]=12 is similar to WGM1[3:0]=4, just replace the one associated with OCR1A with ICR1.

Fast PWM Mode

When WGM1[3:0]=5, 6, 7, 14 or 15 is set, Timer 1 enters Fast PWM mode with count maximum TOP of 0xFF, 0x1FF, 0x3FF, ICR1 or OCR1A respectively, which can be used to generate high frequency PWM waveform. The fast PWM mode is different from other PWM modes in that it is a one-way operation. The counter accumulates from the BOTTOM to the TOP and then returns to the BOTTOM to recount. When the count value TCNT1 reaches TOP or BOTTOM, the output compare signal OC1x is set or cleared, depending on the compare output mode COM1 setting, as detailed in the register description. Due to the unidirectional operation, the fast PWM mode operates twice as often as the phase correction PWM mode with bidirectional operation. The high frequency feature makes Fast PWM mode suitable for power regulation, rectification, and DAC applications. The high-frequency signal reduces the size of external components (inductors, capacitors, etc.), thus reducing system cost.

When the count value reaches TOP, the timer counter overflow flag TOV1 will be set and the compare buffer value will be updated to the compare value. If the interrupt is enabled, the OCR1A register can be updated in the interrupt service program.

The waveform of the output comparison signal OC1x can be obtained only when the data direction register of the OC1x pin is set to output. The frequency of the waveform can be calculated by the following equation.

$f_{oc1xfpwm} = fsys/(N*(1+TOP))$

where N denotes the prescaling factor (1, 8, 64, 256 or 1024)

When TCNT1 and OCR1x are matched by comparison, the waveform generator sets (clears) the OC1x signal, and when TCNT1 is cleared, the waveform generator clears (sets) the OC1x signal to generate a PWM waveform. The resulting polar value of OCR1x will generate a special PWM waveform. When OCR1x is set to 0x00, the output PWM is a narrow spike pulse for every (1+TOP) count clock. When OCR1x is set to TOP, the output waveform is a continuous high or low level. If OCR1A is used as TOP and COM1A=1 is set, the output comparison signal OC1A will generate a PWM waveform with 50% duty cycle.

Phase Correction PWM Mode

When WGM0[3:0]=1, 2, 3, 10 or 11 is set, Timer 1 enters the phase correction PWM mode, and the maximum value of count TOP is 0xFF, 0x1FF, 0x3FF, ICR1 or OCR1A, respectively.The counter operates in both directions,incrementing from BOTTOM to TOP, then decrementing to BOTTOM, and then repeating this operation. The count changes direction when it reaches both TOP and BOTTOM, and the count value stays on TOP or BOTTOM for only one count clock. When the count value TCNT1 matches OCR1x during incrementing or decrementing, the output comparison signal OC1x will be cleared or set,depending on the setting of the comparison output mode COM1. Compared to unidirectional operation, the maximum frequency available for bidirectional operation is smaller, but its excellent symmetry is better suited for motor control.

The phase correction PWM mode sets the TOV1 flag when the count reaches BOTTOM and updates the comparison buffer value to the comparison value when the count reaches TOP. If the interrupt is enabled, the comparison buffer OCR1x memory can be updated in the interrupt service program.

The output comparison signal OC1x waveform is obtained only when the data direction register of OC1x pin is set to output. The frequency of the waveform can be calculated by the following formula.

$f_{oc1xcpcpwm} = fsys/(N*TOP*2)$

where N denotes the prescaling factor (1, 8, 64, 256 or 1024)

During incremental counting, the waveform generator clears (sets) the OC1x signal when TCNT1 matches OCR1x. During the

During decrement counting, the waveform generator sets (clears) the OC1x signal when TCNT1 matches OCR1x. The resulting polarity of OCR1x generates a special PWM waveform. When the OCR1x is set to TOP or BOTTOM, the OC1x signal output will always remain low or high. If OCR1A is used as TOP and COM1A=1 is set, the output comparison signal OC1A generates a PWM wave with a duty cycle of 50%.

To ensure the symmetry of the output PWM wave on both sides of the BOTTOM, there are two cases where the OC1x signal is also flipped when no comparison matching occurs. The first case is when the value of OCR1x changes from TOP to other data. When OCR1x is TOP and the count value reaches TOP, the output of OC1x is the same as the result of comparison matching during the previous descending count, i.e., OC1x is kept unchanged. At this point, the comparison value is updated to the new OCR1x value (not TOP) and the OC1x value is held until it is flipped when the comparison match occurs during ascending counting. At this point, the OC1x signal is not centered symmetrically on the minimum value, so it is necessary to flip the OC1x signal when TCNT1 reaches its maximum value, which is the first case of flipping the OC1x signal when no comparison match occurs. The second case is when TCNT1 starts counting from a value higher than OCR1x, thus losing a comparison match and causing an asymmetric situation. Again, the OC1x signal needs to be flipped to achieve symmetry on both sides of the minimum.

Phase Frequency Correction PWM Mode

When WGM0[3:0]=8 or 9 is set, Timer 1 enters phase frequency corrected PWM mode, and the maximum value of count TOP is ICR1 or OCR1A respectively. The counter operates in both directions, incrementing from BOTTOM to TOP, then decrementing to BOTTOM and repeating this operation. The count changes direction when it reaches both TOP and BOTTOM, and the count stays on TOP or BOTTOM for only one count clock. When the count value TCNT1 matches OCR1x during incrementing or decrementing, the output comparison signal OC1x will be cleared or set, depending on the setting of the comparison output mode COM1. Compared to unidirectional operation, the maximum frequency available for bidirectional operation is smaller, but its excellent symmetry is better suited for motor control.

In Phase Frequency Corrected PWM mode, the TOV1 flag is set when the count reaches BOTTOM and the comparison buffer value is updated to the comparison value. The time to update the comparison value is the biggest difference between Phase Frequency Corrected PWM mode and Phase Corrected PWM mode. If the interrupt is enabled, the comparison buffer OCR1x memory can be updated in the interrupt service program. When the CPU changes the TOP value, i.e. the value of ORC1A or ICR1, it must ensure that the new TOP value is not smaller than the TOP value already in use, otherwise the comparison match will not occur again.

The output comparison signal OC1x waveform is obtained only when the data direction register of OC1x pin is set to output. The frequency of the waveform can be calculated by the following formula.

$f_{oc1xcpfcpwm} = fsys/(N*TOP*2)$

where N denotes the prescaling factor (1, 8, 64, 256 or 1024)

During incremental counting, the waveform generator clears (sets) the OC1x signal when TCNT1 matches OCR1x. During decrement counting, the waveform generator sets (clears) the OC1x signal when TCNT1 matches OCR1x. The resulting polarity of OCR1x generates a special PWM waveform. When the OCR1x is set to TOP or BOTTOM, the OC1x signal output will always remain low or high.If OCR1A is used as TOP and COM1A=1 is set, the output comparison signal OC1A generates a PWM wave with a duty cycle of 50%.

Because the OCR1x register is updated at BOTTOM time, the count lengths for ascending and descending are the same on both sides of the TOP value, which also produces a symmetrical waveform with the correct frequency and phase.

When using a fixed TOP value,it is best to use the ICR1 register as the TOP value, i.e., set WGM1[3:0]=8, at which point OCR1A
The register is only required to generate the PWM output. If you want to generate PWM waves with varying frequency, you must change the TOP value by

The double buffering feature of the OCR1A would be better suited for this application.

Input Capture Mode

Input capture is used to capture an external event and assign a time stamp to it to indicate the moment when this event occurred, and can be done in the previous counting mode, although remove the waveform generation mode that uses the ICR1 value as the counting TOP value.

The trigger signal for the occurrence of an external event is input from pin ICP1 and can also be implemented through the analog comparator unit. When the logic level on pin ICP1 changes, or the output ACO level of the analog comparator changes, and this level change is captured by the input capture unit, the input capture is triggered, at which time the 16-bit count value TCNT1 data is copied to the input capture register ICR1, and the input capture flag ICF1 is set, and if the ICIE1 bit is "1", the input capture flag will generate an input capture interrupt.

Note that changing the trigger source may result in a single input capture, so ICF1 must be cleared once after changing the trigger source to avoid erroneous results.

The input capture signal is fed to the edge detector after passing through an optional noise suppressor to see if the detected edge meets the trigger conditions based on the configuration of the input capture selection control bit ICES1. The noise suppressor is a simple digital filter that samples the input signal four times and only feeds its output to the edge detector if all four samples are equal. The noise suppressor is controlled by the ICNC1 bit of the TCCR1B register to enable or disable it.

When using the input capture function, the value of the ICR1 register should be read as early as possible after ICF1 is set, as the value of ICR1 will be updated after the next capture event occurs. Enabling the input capture interrupt is recommended, and changing the count TOP value during operation is not recommended in any input capture operating mode.

The input captured time stamp can be used to calculate frequency, duty cycle, and other characteristics of the signal, as well as to create a log of trigger events. Measuring the duty cycle of an external signal requires that the trigger edge be changed after each capture, so the trigger signal edge must be changed as soon as possible after the ICR1 value is read.

Automatic shutdown and restart of PWM output

When the DOC1x bit of TCCR1C register is set high, the auto-off function of PWM output will be enabled, and when the trigger condition is met, the hardware will clear the corresponding COM1x bit, disconnect the PWM output signal OC1x from its output pin, and switch to the general-purpose IO output to realize the auto-off of PWM output. At this time, the state of the output pins can be controlled by the output of the general-purpose IO port.

After the PWM output auto-off is enabled, the trigger condition needs to be set, and the DSX1n bit of the TCCR1D register is used to select the trigger source. The trigger sources are analog comparator interrupt, external interrupt, pin level change interrupt, and timer overflow interrupt, as described in the TCCR1D register. When one or more trigger sources are selected as trigger conditions, the hardware will clear the COM1x bit to turn off the PWM output while these interrupt flag bits are set.

When a trigger event occurs to turn off the PWM output, the timer module does not have the corresponding interrupt flag bit, and the software needs to read the interrupt flag bit of the trigger source to know the trigger condition and the trigger event.

When the PWM output is automatically turned off and the output needs to be restarted again, the software simply resets the COM1x bit to toggle

The OC1x signal is output to the corresponding pin. Note that the timer does not stop operating after an automatic shutdown occurs, and the state of the OC1x signal is always updated. The software can set the COM1x bit to output the OC1x signal after a timer overflow or compare match has occurred, so that a clear PWM output state can be obtained.

Dead time control

When DTEN1 is set to "1", the function of inserting the dead time is enabled, and the output waveforms of OC1A and OC1B will insert the set dead time based on the waveform generated by the comparison output of channel B. The length of the time is the time value corresponding to the number of count clocks in the DTR1 register. As shown in the figure below, the dead time insertion of both OC1A and OC1B is based on the comparison output waveform of channel B. When COM1A and COM1B are both "2" or "3", the waveform polarity of OC1A is the same as that of OC1B, when COM1A and COM1B are "2" or "3" respectively, the waveform polarity of OC1A is the same as that of OC1B. When COM1A and COM1B are "2" or "3" respectively, the waveform polarity of OC1A is opposite to that of OC1B.

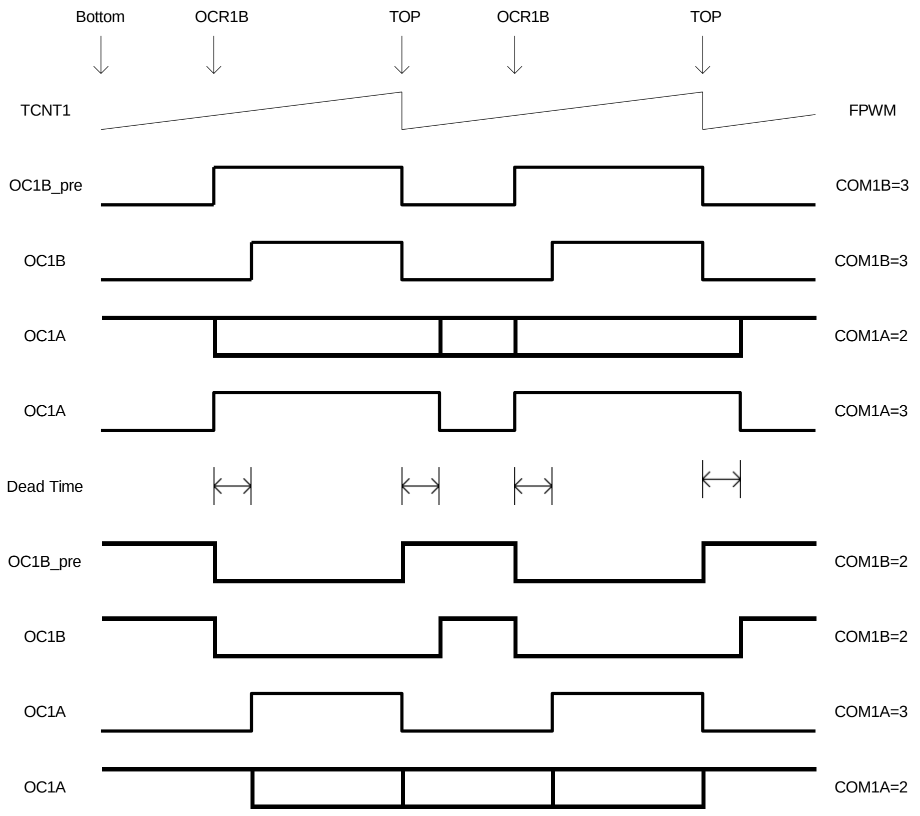


Figure 3 TC1 Dead Time Control in FPWM Mode

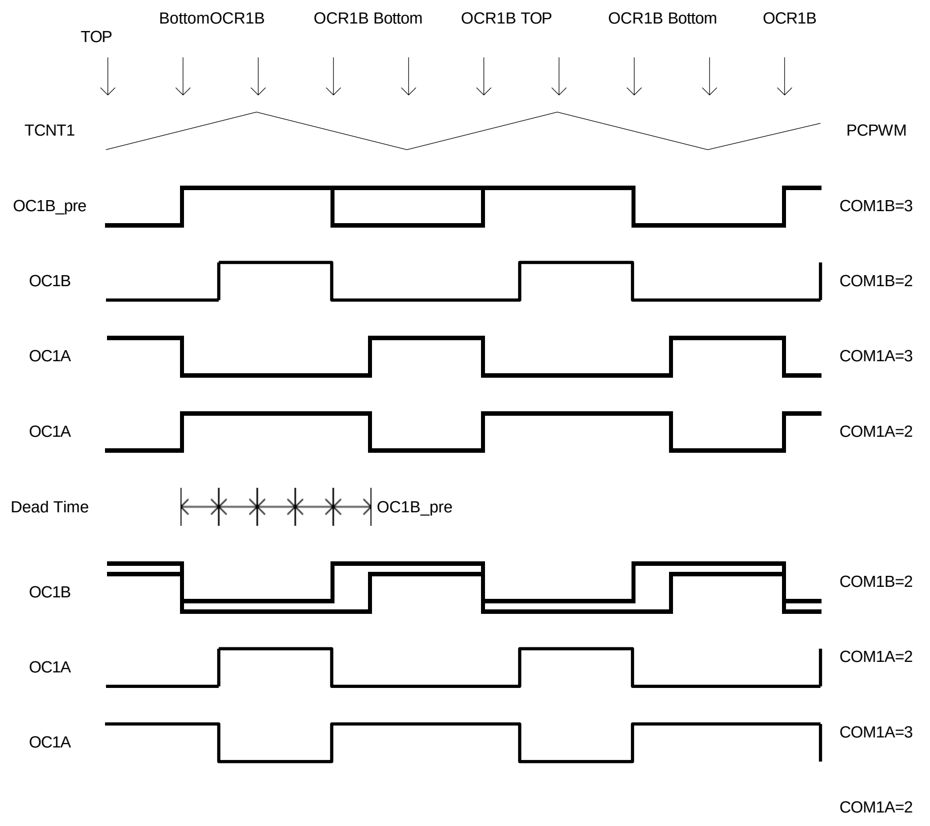


Figure 4 TC1 Dead Time Control in PCPWM Mode

When DTEN1 is set to "0", the function of inserting dead time is disabled, and the output waveforms of OC1A and OC1B are the waveforms generated by their respective comparison outputs.

High-speed counting mode

The high speed clock mode uses a higher frequency clock as the clock source for counting, which is used to generate higher speed and higher resolution PWM waveforms. This high-frequency clock is generated by doubling the output clock of the internal 32M RC oscillator, rc32m. Therefore, before entering the high frequency mode, the internal 32M RC oscillator needs to be enabled for frequency doubling, i.e., set the F2XEN bit of the TCKCSR register and wait for a certain time until the output of the doubled clock signal is stable. Then, the TC2XS1 bit of TCKCSR can be set to put the timer into high speed clock mode.

In this mode, the system clock is asynchronous to the high-speed clock, and some of the registers (see TC1 register list) are operating in the high-speed clock domain; therefore, the configuration and reading of such registers is also asynchronous, and care needs to be taken when operating them.

There are no special requirements for non-sequential read and write operations to registers under the high-speed clock domain, while for sequential read and write operations, you need to wait for a system clock, which can be done as follows.

5) Write register A.
6) Waiting for a system clock (NOP or register under the OS clock)
7) Read or write register A or B.
8) Wait for a system clock (NOP or register under OS clock)

When reading registers in the high-speed clock domain, all registers with a width of 8 bits can be read directly, while when reading the values of 16-bit registers (OCR1A, OCR1B, ICR1, TCNT1), the values of the lower registers are read first, waiting for a system clock

After reading the value of the high register, then read the value of the high register, while reading the value of TCNT1, while the counter is still counting, the value of TCNT1 will change with the high-speed clock, you can pause the counter (set CS1 to zero) and then read the value of TCNT1.

To read OCR1A, OCR1B and ICR1, follow these steps.

1） Reading OCR1AL/OCR1BL/ICR1L.
2） Waiting for a system clock（NOP）
3） Read OCR1AH/OCR1BH/ICR1H.

To read TCNT1, follow these steps.

1） (a) Set CS1 to zero.
2） Waiting for a system clock（NOP）
3） Read the value of TCNT1L.
4） Wait one system clock（NOP）read
the value of TCNT1H.

Register Definition

TC1 Register List

processor register	address	default value	description
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 Count value register low byte
TCNT1H*	0x85	0x00	TC1 Count 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 Output Compare 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	Timing counter interrupt mask register
TIFR1	0x36	0x00	Timing counter interrupt flag register
TCKCSR1	0xEC	0x00	TC1 Clock Control Status Register

[Note]

Registers with "*" work under the system clock and high-speed clock domain, registers without "*" work under the system clock domain only.

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	description
7	COM1A1	Compare Match Output A Mode Control High. COM1A1 and COM1A0 form COM1A[1:0] to control the output comparison waveform OC1A. if either bit 1 or bit 2 of COM1A is set, the output comparison waveform occupies the OC1A pin, but the data direction register of this pin must be set high to output this waveform. The COM1A controls the output compare waveform differently in different operating modes, as described in the comparison output mode control table.
6	COM1A0	Compare Match Output A Mode Control Low. COM1A1 and COM1A0 form COM1A[1:0] to control the output comparison waveform OC1A. if either bit 1 or bit 2 of COM1A is set, the output comparison waveform occupies the OC1A pin, but the data direction register of this pin must be set high to output this waveform. The COM1A controls the output compare waveform differently in different operating modes, as described in the comparison output mode control table.
5	COM1B1	Compare Match Output B Mode Control High. COM1B1 and COM1B0 form COM1B[1:0] to control the output compare waveform OC1B. if either bit 1 or bit 2 of COM1B is set, the output compare waveform occupies the OC1B pin, but the data direction register of this pin must be set high to output this waveform. COM1B controls the output compare waveform differently in different operating modes, as described in the comparison output mode control table.
4	COM1B0	Compare Match Output B Mode Control Low. COM1B1 and COM1B0 form COM1B[1:0] to control the output compare waveform OC1B. if either bit 1 or bit 2 of COM1B is set, the output compare waveform occupies the OC1B pin, but the data direction register of this pin must be set high to output this waveform. COM1B controls the output compare waveform differently in different operating modes, as described in the comparison output mode control table.
3:2	-	Retain the same

1	WGM11	The waveform generation mode controls the next lowest level. WGM11 and WGM13, WGM12, WGM10 together form the waveform generation mode control WGM1[3:0], which controls the counter counting mode and waveform generation mode, as described in the waveform generation mode table.
0	WGM10	The waveform generation mode controls the lowest level. WGM10 and WGM13, WGM12, WGM11 together form the waveform generation mode control The system WGM1[3:0], controls the counter counting mode and waveform generation mode, as described in the Waveform Generation Mode table.

The following table shows the control of the comparison output mode on the output comparison waveform in non- PWM modes (i.e. normal mode and CTC mode).

COM1x[1:0]	des crip tion
0	OC1x disconnected, general purpose IO port operation
1	Flip OC1x signal when comparing matches
2	Clear OC1x signal when comparing matches
3	Set OC1x signal when comparing matches

The following table shows the control of the comparison output mode on the output comparison waveform in fast PWM mode.

COM1x[1:0]	des crip tion
0	OC1x disconnected, general purpose IO port operation
1	When WGM1 is 15: Flip OC1A signal when comparing matches, OC1B disconnected When WGM1 is other values: OC1x disconnected, general purpose IO port operation
2	Clear OC1x signal for comparison match, set OC1x signal for maximum match
3	Set OC1x signal for comparison match, clear OC1x signal for maximum match

The following table shows the control of the output comparison waveform by the comparison output mode in phase correction mode.

COM1x[1:0]	des crip tion
0	OC1x disconnected, general purpose IO port operation
1	When WGM1 is 9 or 11: Flip OC1A signal when comparing matches, OC1B disconnected When WGM1 is other values: OC1x disconnected, general purpose IO port operation
2	Comparative match clear OC1x signal in ascending count, comparative match set OC1x signal in descending count
3	Comparative match set OC1x signal in ascending count, comparative match clear OC1x signal in descending count

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	description

7	ICNC1	Input Capture Noise Suppressor Enable Control bit. When the ICNC1 bit is set to "1", the input capture noise suppressor is enabled, and the input to the external pin ICP1 is filtered so that the input signal is valid for four consecutive samples of equal value, which delays the input capture by four clock cycles. When the ICNC1 bit is set to "0",input capture of the noise suppressor is disabled,and the external lead is then set to "0". The input to pin ICP1 is directly valid.
6	ICES1	Input capture trigger edge selection control bit. When the ICES1 bit is set to "1", the rising edge of the level is selected to trigger the input capture; when the ICES1 bit is set to "0", the falling edge of the level is selected to trigger the input capture. When an event is captured, the value of the counter is copied to the ICR1 register, the same If the interrupt is enabled, an input capture interrupt is generated.
5	-	Reserved.

4	WGM13	The waveform generation mode controls the high level. WGM13 and WGM12,WGM11,WGM10 together form the waveform generation mode control WGM1[3:0], which controls the counter counting mode and waveform generation mode, as described in the waveform generation mode table.
3	WGM12	The waveform generation mode controls the next highest level. WGM12 and WGM13, WGM11, WGM10 together form the waveform generation mode control WGM1[3:0], which controls the counter counting mode and waveform generation mode, as described in the waveform generation mode table.
2	CS12	Clock Select Control High Bit. Used to select the clock source for Timer Counter 1.
1	CS11	Clock Select Control Middle Bit. Used to select the clock source for Timer Counter 1.
0	CS10	Clock Select controls the low position. Used to select the clock source for Timer Counter 1.

CS1[2:0]	description
0	No clock source, stop counting
1	clksys
2	clksys/8, from prescaler
3	clksys/64, from prescaler
4	clksys/256, from prescaler
5	clksys/1024 from prescaler
6	External clock T1 pin, falling edge triggered
7	External clock T1 pin, rising edge triggered

The following table shows the waveform generation mode control.

WGM1 [3:0]	working mode	TOP Value	Update OCR0 Hour	Position TOV0 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	BOTTOM
9	PFCPWM	OCR1A	BOTTOM	BOTTOM
10	PCPWM	ICR1	TOP	BOTTOM
11	PCPWM	OCR1A	TOP	BOTTOM
12	CTC	ICR1	immediately	MAX
13	retain	-	-	-

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	W	W	R/W	R/W	R/W	-	-	-

Bit	Name	description
7	FOC1A	Forced output comparison A. When operating in non- PWM mode, a compare match can be generated by writing a "1" to the forced output compare bit FOC1A. The forced compare match will not set the OCF1A flag, nor will it reload or clear the timer, but the output pin OC1A will be updated accordingly to the COM1A setting, as if a compare match had actually occurred. When operating in PWM mode, clear the TCCR1A register to zero when writing it. The return value for reading FOC1A is always zero.
6	FOC1B	Forced output comparison B. When operating in non- PWM mode, a compare match can be generated by writing a "1" to the forced output compare bit FOC1B. The forced compare match will not set the OCF1B flag, nor will it reload or clear the timer, but the output pin OC1B will be updated accordingly to the COM1B setting,as if a compare match had actually occurred.When operating in PWM mode, the TCCR1A register is cleared to zero when writing it. The return value of reading FOC1B is always zero.
5	DOC1B	TC1 turns off the output compare enable control high. When the DOC1B bit is set to "1", the trigger source off output comparison signal OC1B is enabled. When a trigger event occurs, the hardware automatically clears the COM1B bit and turns off the waveform output of OC1B. The software can re-enable the PWM output by setting COM1B. When the DOC1B bit is set to "0", the trigger source is turned off and the output comparison signal OC1B is disabled. Stop.
4	DOC1A	TC1 turns off the output compare enable control low. When the DOC1A bit is set to "1", the trigger source off output comparison signal OC1A is enabled. When a trigger event occurs, the hardware automatically clears the COM1A bit and turns off the waveform output of OC1A. The software can re-enable the PWM output by setting COM1A. When the DOC1A bit is set to "0", the trigger source is turned off and the output comparison signal OC1A is disabled. Stop.

3	DTEN1	TC1 Dead time enable control bit. When DTEN1 is set to "1", dead time insertion is enabled. OC1A and OC1B both insert dead time on top of the waveform generated by the B channel comparison output, and the interval of the inserted dead time is determined by the count time corresponding to the DTR1 register. The polarity of the OC1A output waveform is determined by the correspondence between COM1A and COM1B, as shown in the table of waveform polarity after inserting the dead time in OC1A. When the DTEN1 bit is set to "0", dead time insertion is disabled and the OC1A and OC1B The waveform is the waveform generated by the respective comparison output.
2:0	-	retain

The following table shows the polarity control of the OC1A signal output waveform when the dead time is enabled.

Polarity Control of OC1A Signal Output Waveform in Dead Time Enable Mode

DTEN1	COM1A[1:0]	COM1B[1:0]	description
0	-	-	OC1A signal polarity is controlled by the OC1A compare output mode
1	0	-	OC1A disconnected, general purpose IO port operation
1	1	-	retain
1	2	2	OC1A signal has the same polarity as OC1B signal
		3	The OC1A signal is opposite in polarity to the OC1B signal
1	3	2	The OC1A signal is opposite in polarity to the OC1B signal
		3	OC1A signal has the same polarity as OC1B signal

[Attention].

The polarity of the OC1B signal output waveform is controlled by the OC1B compare output mode, which is the same as the unenabled dead time mode.

TCCR1D -TC1 Control Register D

TCCR1D -TC 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	description
7	DSX17	TC1 Trigger Source Select Control Enable Bit 7. When the DSX17 bit is set to "1", TC0 overflow is enabled as the trigger source for turning off the output comparison signal waveform OC1A/OC1B. When the DOC1A/DOC1B bit is "1", the rising edge of the interrupt flag register bit of the selected trigger source will automatically turn off the waveform output of OC1A/OC1B. When the DSX17 bit is set to "0", TC0 overflow is used to close the output comparison signal The trigger source of waveform OC1A/OC1B is disabled.
6	DSX16	TC1 Trigger Source Select Control Enable Bit 6. When the DSX16 bit is set to "1", TC2 overflow is enabled as the trigger source for turning off the output comparison signal waveform OC1A/OC1B. When the DOC1A/DOC1B bit is "1", the rising edge of the interrupt flag register bit of the selected trigger source will automatically turn off the waveform output of OC1A/OC1B. When the DSX16 bit is set to "0", TC2 overflow is used to close the output comparison signal The trigger source of waveform OC1A/OC1B is disabled.

5	DSX15	TC1 Trigger Source Select Control Enable Bit 5. When the DSX15 bit is set to "1", the pin level changes by 1 as the trigger source to turn off the output comparison signal waveform OC1A/OC1B is enabled. When the DOC1A/DOC1B bit is "1", the rising edge of the interrupt flag register bit of the selected trigger source will automatically turn off the waveform output of OC1A/OC1B. When the DSX15 bit is set to "0", the pin level changes by 1 as an off output ratio The trigger source of the more-signal waveform OC1A/OC1B is disabled.
4	DSX14	TC1 Trigger Source Select Control Enable Bit 4. When the DSX14 bit is set to "1", the external interrupt 1 is enabled as the trigger source to turn off the output comparison signal waveform OC1A/OC1B. When the DOC1A/DOC1B bits are

		"1", the rising edge of the selected trigger source's interrupt flag register bit will automatically turn off the waveform output of OC1A/OC1B. When the DSX14 bit is set to "0",external interrupt 1 is used to turn off the output compare signal. The trigger source of waveform OC1A/OC1B is disabled.
3:2	-	retain
1	DSX11	TC1 Trigger Source Select Control Enable Bit 1. When the DSX11 bit is set to "1",Analog Comparator 1 is enabled as the trigger source for turning off the output comparison signal waveform OC1A/OC1B. When the DOC1A/DOC1B bit is "1", the rising edge of the interrupt flag register bit of the selected trigger source will automatically turn off the waveform output of OC1A/OC1B. When the DSX11 bit is set to "0",analog comparator 1 is used as the off output comparison The trigger source of the signal waveform OC1A/OC1B is disabled.
0	DSX10	TC1 Trigger Source Select Control Enable Bit 0. When the DSX10 bit is set to "1",analog comparator 0 is enabled as the trigger source for turning off the output comparison signal waveform OC1A/OC1B. When the DOC1A/DOC1B bit is "1", the rising edge of the interrupt flag register bit of the selected trigger source will automatically turn off the waveform output of OC1A/OC1B. When the DSX10 bit is set to "0",analog comparator 0 is used as the off output comparison The trigger source of the signal waveform OC1A/OC1B is disabled.

The following table shows the selection control of the trigger source for the waveform output.

Off Trigger source selection control for OC1A/OC1B waveform output

DOC1x	DSX1n=1	trigger source	description
0	-	-	DOC1x bit is "0", the trigger source off waveform output function is disabled
1	0	Analog Comparator 0	The rising edge of ACIF0 will turn off the OC1x waveform output
1	1	Analog comparator 1	The rising edge of ACIF1 will turn off the OC1x waveform output
1	4	External interrupt 1	The rising edge of INTF1 will turn off the OC1x waveform output
1	5	Pin level change 1	The rising edge of PCIF1 will turn off the OC1x waveform output
1	6	TC2 Overflow	The rising edge of TOV2 will turn off the OC1x waveform output
1	7	TC0 Overflow	The rising edge of TOV0 will turn off the OC1x waveform output

[Attention].

DSX1n=1 means that when bit n of the DSX1 register is 1, each register bit can be set at the same time.

TCNT1L -TC1 Count Value Register Low Byte

TCNT1L -TC1 Count 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	description						
7:0	TCNT1	TC1 The low byte of the count value.						

		TCNT1H and TCNT1L are combined together to form TCNT1, which provides direct read and write access to the counter's 16-bit count value through the TCNT1 register. Two operations are required to read and write the 16-bit register. When writing 16-bit TCNT1, TCNT1H should be written first. when reading 16-bit TCNT1, TCNT1L should be read first. A CPU write operation to the TCNT1 register prevents a compare match from occurring on the next timer clock cycle, even if the timer has been stopped. This allows the initialization of the TCNT1 register to match the value of OCR1x without triggering an interrupt. If the value written to TCNT1 is equal to or bypasses the OCR1x value, the comparison match is lost, resulting in incorrect waveform generation results. The timer stops counting when no clock source is selected, but the CPU can still access TCNT1. CPU write counters have a higher priority than clear or add/drop operations.

TCNT1H -TC1 Counter Value Register High Byte

TCNT1H -TC1 Counter 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	description						
7:0	TCNT1H	TC1 The high byte of the count value. TCNT1H and TCNT1L are combined together to form TCNT1, which provides direct read and write access to the counter's 16-bit count value through the TCNT1 register. Two operations are required to read and write the 16-bit register. When writing 16-bit TCNT1, TCNT1H should be written first. when reading 16-bit TCNT1, TCNT1L should be read first. A CPU write operation to the TCNT1 register prevents a compare match from occurring on the next timer clock cycle, even if the timer has been stopped. This allows the initialization of the TCNT1 register to match the value of OCR1x without triggering an interrupt. If the value written to TCNT1 is equal to or bypasses the OCR1x value, the comparison match is lost, resulting in incorrect waveform generation results. The timer stops counting when no clock source is selected, but the CPU can still access TCNT1. CPU write counters have a higher priority than clear or add/drop operations.						

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	description
7:0	ICR1L	TC1 The low byte of the input capture value. ICR1H and ICR1L are combined together to form a 16-bit ICR1. reading and writing a 16-bit register requires two operations. When writing 16-bit ICR1, you should write ICR1H first. reading 16-bit

		For ICR1, ICR1L should be read first. when the input capture is triggered, the count value TCNT1 is updated and copied to the ICR1 register. ICR1 register can also be used to define the TOP value of the count.

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	description						
7:0	ICR1H	TC1 The high byte of the input capture value. ICR1H and ICR1L are combined to form the 16-bit ICR1. reading and writing the 16-bit register requires two operations. When writing 16-bit ICR1, ICR1H should be written first. when reading 16-bit ICR1, ICR1L should be read first. when the input capture is triggered, the count value TCNT1 is updated and copied to the ICR1 register. ICR1 register can also be used to define the TOP value of the count.						

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
	OCR1AL7	OCR1AL6	OCR1AL5	OCR1AL4	OCR1AL3	OCR1AL2	OCR1AL1	OCR1AL0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	Name	description						
7:0	OCR1AL	Output the low byte of compare register A. OCR1AL and OCR1AH are combined together to form the 16-bit OCR1A. reading and writing the 16-bit register requires two operations. When writing 16-bit OCR1A, OCR1AH should be written first. when reading 16-bit OCR1A, OCR1AL should be read first. The OCR1A is compared to the counter value TCNT1 without interruption. The compare match can be used to generate an output compare interrupt or to generate a waveform on the OC1A pin. When using PWM mode, the OCR1A register uses double-buffered registers. In contrast, the double-buffering function is disabled in normal operation mode and match clear mode. Double buffering synchronizes updating the OCR1A register with the count maximum or minimum moment, thus preventing the generation of asymmetrical PWM pulses and eliminating interference pulses. When using the double buffer function, the CPU accesses the OCR1A buffer register, disabling The CPU accesses the OCR1A itself during the double buffering function.						

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	description						
7:0	OCR1AH	Output the high byte of compare register A. OCR1AL and OCR1AH are combined together to form the 16-bit OCR1A. reading and writing the 16-bit register requires two operations. When writing 16-bit OCR1A, OCR1AH should be written first. when reading 16-bit OCR1A, OCR1AL should be read first. The OCR1A is compared to the counter value TCNT1 without interruption. The compare match can be used to generate an output compare interrupt or to generate a waveform on the OC1A pin. When using PWM mode, the OCR1A register uses double-buffered registers. In contrast, the double-buffering function is disabled in normal operation mode and match clear mode. Double buffering synchronizes updating the OCR1A register with the count maximum or minimum moment, thus preventing the generation of asymmetrical PWM pulses and eliminating interference pulses. When using the double buffer function, the CPU accesses the OCR1A buffer register, disabling The CPU accesses the OCR1A itself during the double buffering function.							

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	description						
7:0	OCR1BL	Output the low byte of compare register B. OCR1BL and OCR1BH are combined to form a 16-bit OCR1B. reading and writing a 16-bit register requires two operations.When writing 16-bit OCR1B,OCR1BH should be written first. when reading a 16-bit OCR1B, OCR1BL should be read first. OCR1B compares uninterruptedly with the counter value TCNT1.The compare match can be used to generate an output compare interrupt or to generate a waveform on the OC1B pin. When using PWM mode, the OCR1B register uses a double-buffer register. In contrast, the double buffer function is disabled in normal operation mode and match clear mode. Double buffering synchronizes updating the OCR1B register with the count maximum or minimum moment, thus preventing the							

		generation of asymmetric PWM pulses and eliminating interference pulses. When using the double buffering function, the CPU accesses the OCR1B buffer register and disables the double The CPU accesses the OCR1B itself during the buffering function.

OCR1BH -TC1 Output Compare Register B High Byte

OCR1BH -TC1 Output Compare Register B High Byte								
Address: 0x8B					Default value: 0x00			
Bit	7	6	5	4	3	2	1	0
	OCR1BH7	OCR1BH6	OCR1BH5	OCR1BH4	OCR1BH3	OCR1BH2	OCR1BH1	OCR1BH0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	description
7:0	OCR1BH	Output the high byte of compare register B. OCR1BL and OCR1BH are combined to form a 16-bit OCR1B. reading and writing the 16-bit register requires two operations. When writing 1 6 - b i t OCR1B, OCR1BH should be written first. when reading 16-bit OCR1B, OCR1BL should be read first. OCR1B compares uninterruptedly with the counter value TCNT1. The compare match can be used to generate an output compare interrupt or to generate a waveform on the OC1B pin. When using PWM mode, the OCR1B register uses double-buffered registers. In contrast, the double buffer function is disabled in normal operation mode and match clear mode. Double buffering synchronizes updating the OCR1B register with the count maximum or minimum moment, thus preventing the generation of asymmetric PWM pulses and eliminating interference pulses. When using the double buffer function, the CPU accesses the OCR1B buffer register and disables The CPU accesses the OCR1B itself during the double buffering function.

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	R/W

Bit	Name	description
7:6	-	Reserved.
5	TICIE1	TC1 Input capture interrupt enable control bit. When the ICIE1 bit is "1" and the global interrupt is set, the TC1 input capture interrupt is enabled. When the input capture is triggered, i.e., the ICF1 flag of TIFR1 is set, the interrupt occurs. When the ICIE1 bit is "0", the TC1 input capture interrupt is disabled.
4:3	-	Reserved.

2	OCIE1B	TC1 Output Compare B Match interrupt enable bit. When the OCIE1B bit is "1" and the global interrupt is set, TC1 outputs a compare B match interrupt enable. The interrupt is generated when a compare match occurs, i.e., when the OCF1B bit in the TIFR is set. When the OCIE1B bit is "0", the TC1 output compare B match interrupt is disabled.
1	OCIE1A	TC1 output compare A match interrupt enable bit. When OCIE1A bit is "1" and global interrupt is set, TC1 output compare A match in

		Interrupt Enable. The interrupt is generated when a compare match occurs, i.e., when the OCF1A bit in the TIFR is set. When the OCIE1A bit is "0", the TC1 output compare A match interrupt is disabled.
0	TOIE1	TC1 Overflow interrupt enable bit. When the TOIE1 bit is "1" and the global interrupt is set, TC1 overflow interrupt is enabled.WhenTC1 The interrupt is generated when an overflow occurs, i.e., when the TOV1 bit in the TIFR is set. When the TOIE1 bit is "0", the TC1 overflow interrupt is disabled.

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	R/W

Bit	Name	description
7:6	-	Reserved.
5	ICF1	Enter the capture flag bit. The ICF1 flag is set when an input capture event occurs. The ICF1 flag is set when ICR1 is used as the TOP value of the count and the count value reaches the TOP value. If ICIE1 is "1" and the global interrupt flag is set, an input capture interrupt will be generated. ICF1 will be automatically cleared when this interrupt service routine is executed, or the ICF1 bit can be cleared by writing a "1" to it.
4:3	-	Reserved.
2	OCF1B	The output compares the B match flag bit. When TCNT1 is equal to OCR1B, the compare unit will give a match signal and set the compare flag OCF1B. If the output compare interrupt enable OCIE1B is "1" and the global interrupt flag is set, an output compare interrupt will be generated.OCF1B will be cleared automatically when this interrupt service program is executed, or by writing "1" to the OCF1B bit.
1	OCF1A	The output compares the A match flag bit. When TCNT1 is equal to OCR1A, the compare unit gives a match signal and sets the compare flag OCF1A. If the output compare interrupt enable OCIE1A is "1" and the global interrupt flag is set, an output compare interrupt will be generated.OCF1A will be cleared automatically when this interrupt service program is executed, or by writing "1" to the OCF1A bit.

0	TOV1	Overflow flag bit. If the overflow interrupt enable TOIE1 is "1" and the global interrupt flag is set, an overflow interrupt will be generated. TOV1 will be cleared automatically when this interrupt service routine is executed, or by writing a "1" to the TOV1 bit.

DTR1L -TC1 Dead Time Register Low Byte

DTR1-TC1 Dead Time Register								
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	description						
7:0	DTR1L	Dead time register high byte. When the DTEN1 bit is high, OC1A and OC1B are complementary outputs and all outputs on the OC1A output The dead time of the insertion is determined by the DTR1L count clock.						

DTR1H -TC1 Dead Time Register High Byte

DTR1H-TC1 High Byte of Dead Time Register								
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	description						
7:0	DTR1H	Dead time register high byte. When the DTEN1 bit is high, OC1A and OC1B are complementary outputs and all outputs on the OC1B output The dead time of the insertion is determined by the DTR1H count clock.						

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/O	R/O	-	R/W	R/W	R/W
Bit	Name	description						
7	-	retain						
6	F2XEN	RC 32M Multiplier output enable control bit When the F2XEN bit is set to "1", the multiplier output of the 32M RC oscillator is enabled to output a 64M high-speed clock When the F2XEN bit is set to "1", the multiplier output of the 32M RC oscillator is disabled. 64M high-speed clock cannot be output						
5	TC2XF1	TC High-speed clock mode flag bit 1 When the TC2XF1 bit is read as "1", it indicates that Timer 1 is operating in high-speed clock mode, and when it is "0", it indicates that Timer 1 is operating in system clock mode						
4	TC2XF0	TC High-speed clock mode flag bit 0, refer to Timing Counter 0 Register Description						
3:2	-	retain						

1	TC2XS1	TC High-speed clock mode selection control bit 1 When the TC2XS1 bit is set to "1", Timer 1 is selected to operate in high-speed clock mode When the TC2XS1 bit is set to "0", Timer 1 is selected to operate on the system clock mode
0	TC2XS0	TC High-speed clock mode selection control bit 0, refer to Timer Counter 0 register description

TMR0/1/3 Prescaler

- 3 x 10-bit prescalers
- Multiplexing TC0, TC1 and TC3 in Multiplexed Mode Prescaler CPS310
- Standalone mode TC0 Exclusive Prescaler CPS310, TC1 Exclusive Prescaler CPS1, TC3 Exclusive Prescaler CPS3
- Software reset support

summarize

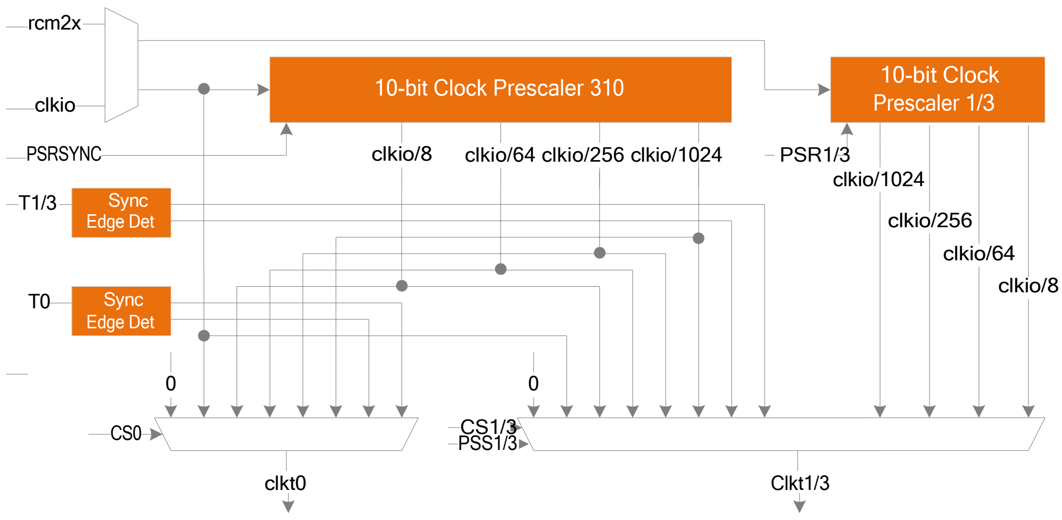
In multiplexed mode (PSS1=0 and PSS3=0) TC0, TC1 and TC3 share a 10-bit prescaler, CPS310, but they have different divider settings.

In single-use mode (PSS1=1 and PSS3=0) TC1 independently uses a 10-bit prescaler CPS1, TC0 and TC3 A 10-bit prescaler, CPS310, is used in common, but they have different crossover settings.

In single-use mode (PSS1=0 and PSS3=1) TC3 independently uses a 10-bit prescaler CPS3, TC0 and TC1 A 10-bit prescaler, CPS310, is used in common, but they have different crossover settings.

In independent mode (PSS1=1 and PSS3=1) TC0 uses a 10-bit prescaler CPS310 independently, TC1 uses a 10-bit prescaler CPS1 independently, and TC3 uses prescaler CPS3 independently.

The following descriptions are used for TC0, TC1 and TC3, where n represents 0, 1 or 3.



TC0/TC1 /TC3 Prescaler Structure Diagram

Internal clock source

When CSn[2:0]=1 is set, Timer 3 can be driven by system clock clkio only. Timer 0 or 1 can be driven directly by system clock clkio or high speed clock rcm2x (2x of internal 32M RC oscillator output clock). The prescaler can output 4 different clock frequencies, clkio/8, clkio/64, clkio/256 and clkio/1024.

Crossover reset

multiplexing mode

When the PSS1 bit is set to "0" and the PSS3 bit is "0", TC0, TC1 and TC3 share a prescaler CPS310.

The prescaler operates independently of the TC's clock selection logic, and it is shared by TC0, TC1, and TC3. Since it is not subject to clock selection control, the state of the prescaler has an effect on the application of the dividing clock. The effect occurs when the timer is enabled and the output of the prescaler is selected as the count clock source (6>CSn[2:0]> 1)may take from 1 to N+1 system clocks from timer enable to the first count, where N is the prescaler factor (8, 64, 256 or 1024)

It is possible to synchronize timer and program operation by resetting the prescaler. However, care must be taken whether another timer is using this prescaler, and resetting the prescaler will affect all timers connected to it.

single-use mode

When the PSS1 bit is set to "1", TC1 uses prescaler CPS1 independently, and the reset of the prescaler is controlled by the PSR1 bit. The respective resets work independently and do not affect other prescalers.

When the PSS3 bit is set to "1", the TC3 uses prescaler CPS3 independently and the reset of the prescaler is controlled by the PSR3 bit. The respective resets work independently and do not affect other prescalers.

When the PSS1 bit is set to "1" and the PSS3 bit is set to "1", TC0 uses prescaler CPS310 independently and the reset of the prescaler is controlled by the PSRSYNC bit, TC1 uses prescaler CPS1 independently and TC3 uses prescaler CPS3 independently. TC1 uses prescaler CPS1 independently, and TC3 uses prescaler CPS3 independently, with their respective resets acting independently and not affecting the other prescalers.

External clock source

The external clock source provided by the T0/T1/T3 pins can be used as the counter clock source. the signal from the T0/T1/T3 pins is used as the counter clock source after the synchronization logic and edge detector. Each rising edge (CSn[2:0]=7) or falling edge (CSn[2:0]=6) all generate a count pulse. The external clock source is not fed to the prescaler.

Due to the presence of synchronization and edge detection circuitry on the pins, a change in level on T0/T1/T3 requires a delay of 2.5 to 3.5 system clocks to cause the counter to update.

Disabling or enabling the clock input must be done only after T0/T1/T3 has remained stable for at least one system clock cycle, otherwise there is a possibility of generating incorrectly counted clock pulses.

To ensure correct sampling, the external clock pulse width must be greater than one system clock cycle and the external clock frequency must be less than half the system clock frequency at a duty cycle of 50%. Due to the difference in system clock frequency and duty cycle due to errors in the oscillator itself, it is recommended that the maximum frequency of the external clock not be greater than fsys/2.5.

Register Definition

GTCCR - General Timer Control Register

GTCCR - General Timer Control 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	description
7	TSM	Timing counter synchronization mode control bit. When the TSM bit is set to "1", the timer counter is in synchronous mode. In synchronous mode, the values written to the PSRASY bit and PSRSYNC bit are held,allowing the corresponding prescaler to be reset at all times. This ensures that the corresponding timer counters are aborted and configured to the same value. When the TSM bit is set to "0", the values of the PSRASY and PSRSYNC bits are cleared by hardware. Zero and the timer counter starts working at the same time.
6:2	-	Reserved.
1	PSRASY	See Timer TC2 register description.
0	PSRSYNC	Prescaler CPS310 reset control bit. When the PSRSYNC bit is set to "1", the prescaler CPS310 will be reset. When the TSM If the bit is not set, the hardware will clear the PSRSYNC bit after a reset. When the PSRSYNC bit is set to "0", the setting is invalid. In multiplexed mode, TC0/TC1/TC3 share the prescaler and a reset will affect all three timers. In standalone mode, the reset will only affect TC0. The value of this bit will always be read as "0".

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	description

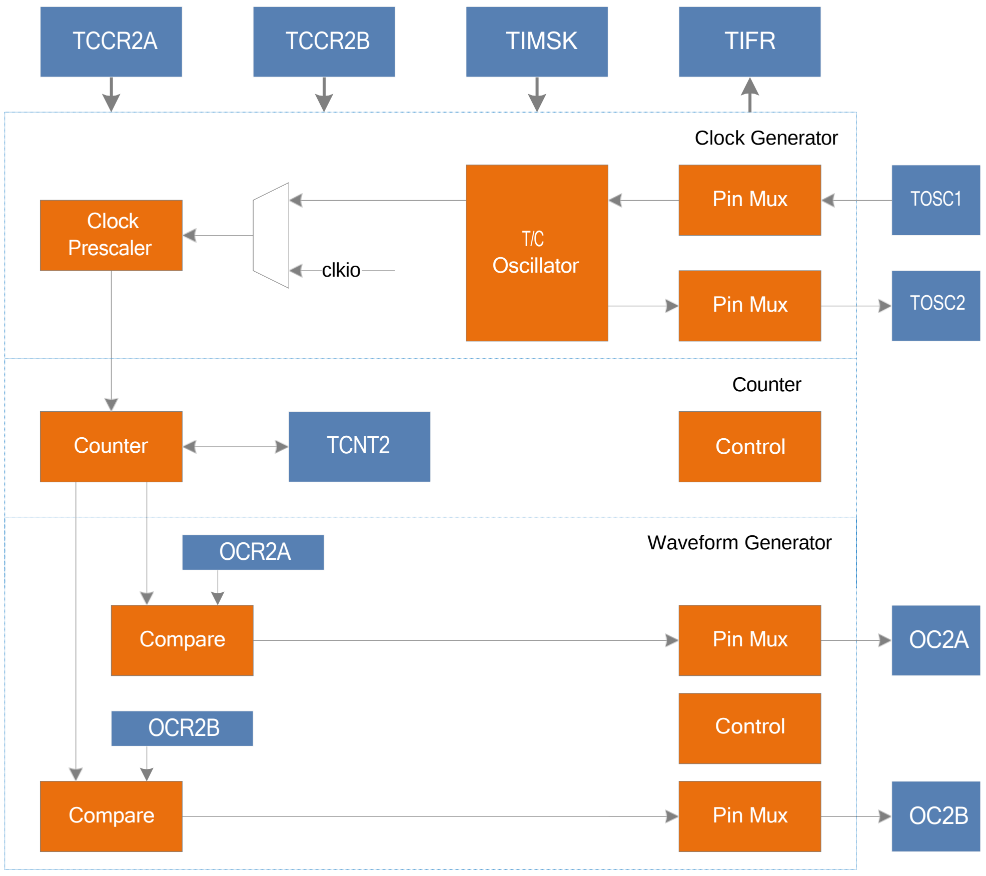
7	PSS1	Prescaler selection control bit. When the PSS1 bit is set to "1", TC1 uses the prescaler CPS1 alone. When PSS1 bit is set to "0", it is prescaler multiplexer mode. prescaler CPS310 is shared by TC0 and TC1,prescaler CPS1 is invalid and will be reset all the time.If the PSS3 bit is "0" at the same time, TC3 shares prescaler CPS310 with TC0 and TC1. Both CPS1 and CPS3 are invalid and will always be reset.
6	PSS3	Prescaler selection control bit. When the PSS3 bit is set to "1", TC3 uses the prescaler CPS3 alone.

		When PSS3 bit is set to "0", it is prescaler multiplexing mode. prescaler CPS310 is shared by TC0 and TC3,prescaler CPS3 is invalid and will be reset all the time. If the PSS1 bit is "0" at the same time, TC1 shares prescaler CPS310 with TC0 and TC3. prescaler CPS1 and CPS3 are invalid and will be reset all the time.
5:2	-	Reserved.
1	PSR3	Prescaler CPS3 reset control bit. The PSR3 bit is only valid in TC3 single use mode. When the PSR3 bit is set to "1", Prescaler CPS3 will be reset. The hardware will clear the PSR3 bit after the reset. When the PSR3 bit is set to "0", the setting is invalid. The value of this bit will always be read as "0".
0	PSR1	Prescaler CPS1 reset control bit. The PSR1 bit is only valid in TC1 single use mode. Prescaler CPS1 will be reset when the PSR1 bit is set to "1". The hardware will clear the PSR1 bit after the reset. When the PSR1 bit is set to "0", the setting is invalid. The value of this bit will always be read as "0".

Timer/Counter 2 (TMR2)

- 8-bit counter
- Two separate comparison units
- Automatically clears the counter and automatically loads it when a comparison match occurs
- Phase-corrected PWM output without interference pulses
- Frequency generator
- External event counter
- 10-bit clock prescaler
- Overflow and compare match interrupts
- Allows counting with an external 32.768KHz RTC crystal

summarize



TC2 Structure Diagram

TC2 is a general-purpose 8-bit timer counter module that supports PWM output for accurate waveform generation.TC2 contains one 8-bit counter, waveform generation mode control unit and two output comparison units. The waveform generation mode control unit controls the operation mode of the counter and the generation of the comparison output waveform. Depending on the operating mode, the counter achieves zero, plus one or minus one operation for each count clock Clkt2. Clkt2 can be generated by internal clock source or external clock source. TC2 can be used as an RTC counter when counting with an external 32.768KHz crystal. When the counter count value TCNT2

When the maximum value (equal to the maximum value 0xFF or the output comparison register OCR2A, defined as TOP, and the maximum value MAX to show the difference) is reached, the counter will be cleared or minus one. When the counter count value TCNT2 reaches the minimum value (equal to 0x00, defined as BOTTOM), the counter will perform a plus one operation. When the counter's count value TCNT2 reaches OCR2A/OCR2B, also known as when a comparison match occurs, it will clear or set the output comparison signal OC2A/OCR2B to generate the PWM waveform.

working mode

Timer 2 has four different operating modes, including Normal mode, Clear on Compare Match (CTC) mode, Fast Pulse Width Modulation (FPWM) mode and Phase Correction Pulse Width Modulation (PCPWM) mode, which are selected by the Waveform Generation Mode control bits WGM2[2:0]. These four modes are described in detail below. Since there are two independent output comparison units, denoted by "A" and "B", respectively, the two output comparison unit channels are denoted by lowercase "x".

normal mode

Normal mode is the simplest mode of operation of the timer counter, in which the waveform generation mode control bits WGM2[2:0]=0 and the maximum value of the count is MAX (0xFF). In this mode, the count is incremented by one for each count clock, and when the counter reaches TOP overflow, it returns to BOTTOM and starts accumulating again. The timer overflow flag TOV2 is set in the same count clock where the count value TCNT2 goes to zero; in this mode the TOV2 flag is like the 9th count bit, except that it is only set and not cleared. The overflow interrupt service routine automatically clears the TOV2 flag, which can be used by software to increase the resolution of the timer. There are no special circumstances to consider in normal mode, and a new count value can be written at any time.
The waveform of the output comparison signal OC2x can be obtained only when the data direction register of the OC2x pin is set to output. When COM2x=1
When a comparative match occurs, the OC2x signal is flipped and the frequency of the waveform in this case can be calculated using the following equation.

$f_{oc2xnormal} = fsys/(2*N*256)$

where N denotes the prescaling factor (1, 8, 64, 256 or 1024)
The output comparison unit can be used to generate interrupts, but interrupts are not recommended in normal mode, as they can take up too much CPU
of time.

CTC model

When WGM2[2:0]=2 is set, Timer 2 enters CTC mode, and the maximum value of count TOP is OCR2A. In this mode, the count is incremented by one for each count clock, and the counter is cleared when the counter value TCNT2 equals TOP. OCR2A defines the maximum value of count, which is also the resolution of the counter. This mode allows the user to easily control the frequency of the compare match output and also simplifies the operation of the external event count.
When the counter reaches the maximum value of the count, the output compare match flag OCF2 is set and an interrupt will be generated when the corresponding interrupt enable is set. The OCR2A register, the maximum value of the count, can be updated in the interrupt service program. In this mode the OCR2A does not use double buffering, so be careful when updating the maximum value to a value close to the minimum with the counter operating with no prescaler or a very low prescaler. If the value written to OCR2A is less than the then current TCNT2 value, the counter will lose a compare match. The counter has to count

to TOP before the next comparison match occurs, and then count to the OCR2A value starting from BOTTOM. As in normal mode, the count value returns to the BOTTOM with the TOV2 flag set in the count clock. The waveform of the output comparison signal OC2x can be obtained only when the data direction register of the OC2x pin is set to output. When COM2x=1, the OC2x signal will be flipped when a comparison match occurs, and the frequency of the waveform in this case can be calculated by the following formula.

$f_{oc2xctc}$ = fsys/(2*N*(1+OCR2A))

where N denotes the prescaling factor (1, 8, 64, 256 or 1024) From the formula, it can be seen that when setting OCR2x When 0x0 and no prescaler, an output waveform with a maximum frequency of fsys/2 can be obtained.

Fast *PWM* Mode

When WGM2[2:0]=3 or 7 is set, Timer 2 enters Fast PWM mode, which can be used to generate high frequency PWM waveform with the maximum count value TOP as MAX(0xFF) or OCR2A respectively. Fast PWM mode is different from other PWM modes in that it is a one-way operation. The counters accumulate from the minimum value of 0x00 to TOP and then return to BOTTOM to recount. When the count value TCNT2 reaches OCR2x or BOTTOM, the output compare signal OC2x is set or cleared, depending on the compare output mode COM2x setting, as detailed in the register description. Due to the unidirectional operation, the fast PWM mode operates at twice the frequency of the phase correction PWM mode with bidirectional operation. The high frequency feature makes the fast PWM mode suitable for power regulation, rectification, and DAC applications. The high-frequency signal reduces the size of external components (inductors, capacitors, etc.), thus reducing system cost.
When the count value reaches its maximum value, the timer counter overflow flag TOV2 will be set and the comparison buffer value will be updated to the comparison value. If the interrupt is enabled, the comparison buffer OCR2x register can be updated in the interrupt service program.
The waveform of the output comparison signal OC2x can be obtained only when the data direction register of the OC2x pin is set to output. The frequency of the waveform can be calculated by the following equation.

$f_{oc2xfpwm} = fsys/(N*(1+TOP))$

where N denotes the prescaling factor (1, 8, 64, 256 or 1024)
When TCNT2 and OCR2x are matched by comparison, the waveform generator sets (clears) the OC2x signal, and when TCNT2 is cleared, the waveform generator clears (sets) the OC2x signal to generate a PWM waveform. The resulting extreme value of OCR2x will generate a special PWM waveform. When OCR2x is set to 0x00, the output PWM is a narrow spike pulse for every (1+TOP) count clock. When OCR2x is set to the maximum value, the output waveform is a continuous high or low level.

Phase Correction *PWM* Mode

When WGM2[2:0]=1 or 5 is set, Timer 2 enters phase correction PWM mode, and the maximum value of count TOP is MAX(0xFF) or OCR2A, respectively. The counter operates in both directions, incrementing from BOTTOM to TOP, then decrementing to BOTTOM, and then repeating this operation. The count changes direction when it reaches both TOP and BOTTOM, and the count value stays on TOP or BOTTOM for only one count clock. When the count value TCNT2 matches OCR2x during incrementing or decrementing, the output comparison signal OC2x will be cleared or set, depending on the setting of the comparison output mode COM2x. Compared to unidirectional operation, the maximum frequency available for bidirectional operation is smaller, but its excellent symmetry is better suited for motor control.
The phase correction PWM mode sets the TOV2 flag when the count reaches BOTTOM and updates the comparison buffer value to the comparison value when the count reaches TOP. If the interrupt is enabled, the comparison buffer OCR2x register can be updated in the interrupt service program.
The waveform of the output comparison signal OC2x can be obtained only when the data direction register of the OC2x pin is set to output. The frequency of the waveform can be calculated by the following equation.

$f_{oc2xpcpwm} = fsys/(N*TOP*2)$

where N denotes the prescaling factor (1, 8, 64, 256 or 1024)
During incremental counting, the waveform generator clears (sets) the OC2x signal when TCNT2 matches OCR2x. During decrement counting, the waveform generator sets (clears) the OC2x signal when TCNT2 matches OCR2x. The resulting extreme value of OCR2x generates a special PWM waveform. When OCR2x is set to the maximum or minimum value, the OC2x signal output will remain low or high.

To ensure symmetry of the output PWM wave on both sides of the minimum, there are two cases where the OC2x signal is also flipped when no comparison matching occurs. The first case is when the value of OCR2x changes from the maximum value 0xFF to other data. When OCR2x is the maximum value and the count value reaches its maximum, the output of OC2x is the same as the result of the comparison match during the previous descending count, i.e., OC2x remains unchanged. At this point the comparison value is updated to the new OCR2x value (not 0xFF), and the OC2x value is kept until

The OC2x signal is flipped when TCNT2 reaches its maximum value. In this case, the OC2x signal is not symmetric around the minimum value, so it is necessary to flip the OC2x signal when TCNT2 reaches its maximum value, which is the first case of flipping the OC2x signal when no comparison match occurs. The second case is when TCNT2 starts counting from a value higher than OCR2x, thus losing a comparison match and causing an asymmetric situation. Again, the OC2x signal needs to be flipped to achieve symmetry on both sides of the minimum.

TC2's asynchronous operation method

When the AS2 bit in the ASSR register is "1", TC2 operates in asynchronous mode and the counter is clocked from the external timer oscillator. The operation of TC2 in asynchronous mode should consider the following points.

- Transitions between synchronous and asynchronous modes have the potential to cause corruption of TCNT2, OCR2A, OCR2B, TCCR2A, and TCCR2B data. Safe operating procedures are shown below.
  1. Clear the OCIE2A, TOIE2 and OCIE2B register bits to turn off interrupts for TC2.
  2. (a) Set the AS2 bit to select the appropriate clock source.
  3. Write new data to the TCNT2, OCR2A, TCCR2A, OCR2B and TCCR2B registers.
  4. Waiting for the TCN2UB, OCR2AUB, TCR2AUB, OCR2BUB and TCR2BUB bits to clear when switching to asynchronous mode.
  5. Clear the interrupt flag bit of TC2.
  6. Enables interrupts that need to be used.
- The oscillator should preferably use a 32.768KHz watch crystal. The system clock frequency must be at least 4 times higher than the crystal frequency.
- When the CPU writes TCNT2, OCR2A, TCCR2A, OCR2B, and TCCR2B, the hardware puts the data into the staging register first, and the two TOSC1 clocks latch into the corresponding registers only after the rising edge. A new data write operation cannot be performed until the data is latched from the staging register to the destination register. Each register has its own independent staging register, so writing TCNT2 does not interfere with writing OCR2. The Asynchronous Status Register ASSR is used to check if data has been written to the destination register.
- If TC2 is used as the wake-up condition for MCU hibernation mode, you cannot enter hibernation mode until each register update is completed, otherwise the MCU may enter hibernation mode before the TC2 setting takes effect, and thus TC2 cannot wake up the system.
- If TC2 is used as the wake-up condition for MCU hibernation mode, care must be taken to re-enter hibernation mode. The interrupt logic requires one TOSC1 clock cycle to reset, and if the time from wake-up to re-entry into hibernation is less than one TOSC1 clock cycle, the interrupt will no longer occur and the device will not wake up. The following method of operation is recommended.
  1. Writing the appropriate data to the individual registers.
  2. Waiting for the corresponding update busy flag bit of the ASSR to clear.
  3. Enter hibernation mode.
- If the asynchronous operating mode is selected, the TC2's oscillator will always operate unless it is put into power-down mode. The user must be aware that this oscillator may take up to 1 second to stabilize, so it is recommended that the user wait at least 1 second after enabling the TC2's oscillator before using the TC2's asynchronous mode.
- The process of waking up in hibernation mode during asynchronous operation mode: after the interrupt condition is met, the wake-up process is initiated at the next timer clock. That is, the counter accumulates at least one more clock before the processor can read the counter value. After waking up, the MCU executes the interrupt service program, after which the program following the SLEEP statement begins to execute.
- Reading the value of TCNT2 a short time after waking up from sleep mode may return incorrect data.

Because TCNT2 is driven by the asynchronous TOSC1 clock, reading TCNT2 must be done through a register synchronized by the internal system clock, with synchronization occurring on the rising edge of each TOSC1. When the system clock is reactivated upon waking from sleep mode, the TCNT2 value read is the value prior to entering sleep mode and is not updated until the arrival of the next TOSC1 rising edge. The phase of TOSC1 when waking up from sleep mode is completely unpredictable and is dependent on the wake-up time. Therefore, the recommended sequence for reading the TCNT2 value is

1. Write an arbitrary value to the OCR2A or TCCR2A.

2. Waiting for the corresponding update busy flag bit to be cleared.
3. Read TCNT2.

- In asynchronous mode, synchronization of the interrupt flag bits requires 3 system clock cycles plus 1 timer cycle. At least one more clock is accumulated by the counter before the MCU can read the counter value that caused the interrupt flag to be set. The change in the output compare signal is synchronized with the timer clock, not the system clock.

Prescaler for *TC2*

The input clock to the TC2 prescaler is called clkt2s and is selected by the AS2 bit located in the ASSR register for either the internal system clock clkio or the external TOSC1 clock source, which is connected to the system clock clkio by default. If AS2 is set, TC2 will be driven by TOSC1 asynchronously. TC2 can be used as an RTC counter when a 32.768KHz clock crystal is connected to the TOSC1 pin and TOSC2 pin. It is not recommended to apply an external clock signal directly to the TOSC1 pin.

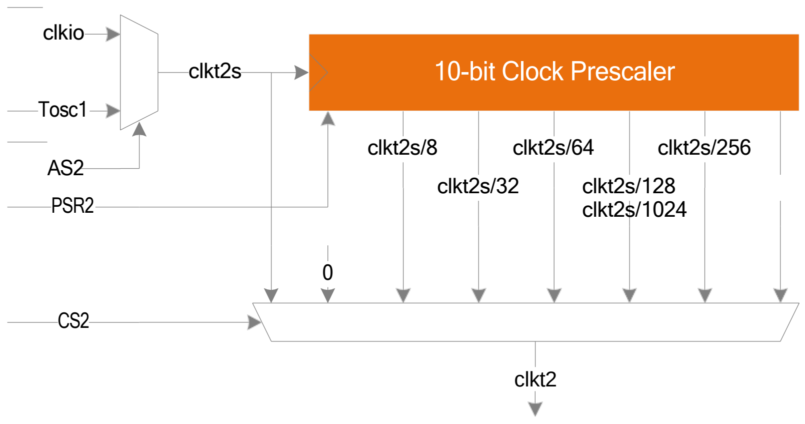


Figure 5 TC2 Prescaler Structure Diagram

The above diagram shows the TC2 prescaler, as shown, with the following possible prescaler options: clkt2s/8, clkt2s/32, clkt2s/64, clkt2s/128, clkt2s/256, and clkt2s/1024. In addition clkt2s and 0 (stop counting)can be selected. Setting the PSR2 bit of the SFIOR register will reset the prescaler, allowing the user to start from a predictable prescaler.

Register Definition

TC2 Register List

processor register	addres s	default value	descri ption
TCCR2A	0xB0	0x00	TC2 control register A
TCCR2B	0xB1	0x00	TC2 Control Register B
TCNT2	0xB2	0x00	TC2 Count Value Register
OCR2A	0xB3	0x00	TC2 Output Comparison Register A
OCR2B	0xB4	0x00	TC2 Output Comparison Register B
ASSR	0xB6	0x00	TC2 Asynchronous Status Register
TIMSK2	0x70	0x00	Timing counter interrupt mask register
TIFR2	0x37	0x00	Timing counter interrupt flag register

TCCR2A-TC2 Control Register A

TCCR2 A-TC2 Control Register A								
Address: 0xB0					Default value: 0x00			
Bit	7	6	5	4	3	2	1	0
	COM2A1	COM2A0	COM2B1	COM2B0	-	-	WGM21	WGM20
R/W	W	R/W	R/W	R/W	-	-	R/W	R/W

Bit	Name	description
7	COM2A1	TC2 Compare Match Output A Mode Control High. COM2A1 and COM2A0 together form the output compare mode control COM2A[1:0], which controls the output waveform of OC2A. If either bit 1 or bit 2 of COM2A is set, the output compare waveform occupies the OC2A pin, although the data direction register of this pin must be set high to output this waveform. The COM2A controls the output comparison waveform differently in different operating modes, as described in the comparison output mode control table.
6	COM2A0	TC2 Compare Match Output A Mode Control Low. COM2A0 and COM2A1 together form the output compare mode control COM2A[1:0], which controls the output waveform of OC2A. If either bit 1 or bit 2 of COM2A is set, the output compare waveform occupies the OC2A pin, though the data direction register of this pin must be set high to output this waveform. The COM2A controls the output comparison waveform differently in different operating modes, as described in the comparison output mode control table.
5	COM2B1	TC2 Compare Match Output B Mode Control High. COM2B1 and COM2B0 together form the output compare mode control COM2B[1:0], which controls the output waveform of OC2B. If either bit 1 or bit 2 of COM2B is set, the output compare waveform occupies the OC2B pin, although the data direction register of this pin must be set high to output this waveform. The COM2B controls the output compare waveform differently in different operating modes, as described in the compare output mode control table.
4	COM2B0	TC2 Compare Match Output B Mode Control Low. COM2B0 and COM2B1 together form the output compare mode control COM2B[1:0], which controls the output waveform of OC2B. If either bit 1 or bit 2 of COM2B is set, the output compare waveform occupies the OC2B pin, although the data direction register of this pin must be set high to output this waveform. The COM2B controls the output compare waveform differently in different operating modes, as described in the compare output mode control table.
3:2	-	Reserved.

1	WGM21	TC2 Waveform Generation Mode Control High. WGM20 together with WGM21 and WGM22 form the waveform generation mode control WGM2[2:0], which controls the counter counting mode and waveform generation mode, as described in the waveform generation mode table.
0	WGM20	TC2 Waveform Generation Mode Control Low. WGM21 together with WGM20 and WGM22 form the waveform generation mode control WGM2[2:0], controls how the counter counts and how the waveform is generated, as described in the Waveform Generation Mode table.

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

Bit	Name	description
7	FOC2A	TC2 forces the output to compare the A control bit. When operating in non- PWM mode, a compare match can be generated by writing a "1" to the forced output compare bit FOC2A. The forced compare match will not set the OCF2A flag, nor will it reload or clear the timer, but the output pin OC2A will be updated accordingly to the COM2A setting, just as if a compare match had actually occurred. The return value for reading FOC2A is always zero.
6	FOC2B	TC2 forces the output to compare the B control bit. When operating in non- PWM mode, a compare match can be generated by writing a "1" to the forced output compare bit FOC2B. The forced compare match will not set the OCF2B flag, nor will it reload or clear the timer, but the output pin OC2B will be updated accordingly to the COM2B setting, just as if a compare match had actually occurred. The return value for reading FOC2B is always zero.
5:4	-	Reserved.
3	WGM22	TC2 Waveform Generation Mode Control High. WGM22, together w i t h WGM20 、WGM21, form Waveform Generation Mode Control WGM2[2:0], which controls how the counter counts and how the waveform is generated, as described in the Waveform Generation Mode table.
2	CS22	TC2 Clock Select Control High. Used to select the clock source for Timer Counter 2.
1	CS21	TC2 Clock Select Control Neutral. Used to select the clock source for Timer Counter 2.
0	CS20	TC2 Clock Select Control Low. Used to select the clock source for Timer Counter 2.

CS2 [2:0]	description
0	No clock source, stop counting
1	clkt2s
2	clkt2s/8, from prescaler
3	clkt2s/32, from prescaler
4	clkt2s/64, from prescaler
5	clkt2s/128, from prescaler
6	clkt2s/256, from prescaler
7	clkt2s/1024 from prescaler

The following table shows the control of the comparison output mode on the output comparison waveform in non-PWM modes (i.e. normal mode and CTC mode).

Table 1 OC2x Comparative Output Mode Control in Non-PWM Mode

COM2x[1:0]	description
0	OC2x disconnected, general purpose IO port operation
1	Flip OC2x signal when comparing matches
2	Clear OC2x signal when comparing matches
3	Set OC2x signal when comparing matches

The following table shows the control of the comparison output mode on the output comparison waveform in fast PWM mode.

Table 2 OC2x Comparative Output Mode Control in Fast PWM Mode

COM2x[1:0]	description
0	OC2x disconnected, general purpose IO port operation
1	retain
2	Clear OC2x signal for comparison match, set OC2x signal for maximum match
3	Set OC2x signal for comparison match, clear OC2x signal for maximum match

The following table shows the control of the output comparison waveform by the comparison output mode in phase correction mode.

Table 3 OC2x Comparative Output Mode Control in Phase Correction PWM Mode

COM2x[1:0]	description
0	OC2x disconnected, general purpose IO port operation
1	retain
2	Clear OC2x signal when comparing matches in ascending count, set OC2x signal when comparing matches in descending count
3	Set the OC2x signal when comparing matches in ascending count, and clear the OC2x signal when comparing matches in descending count

The following table shows the waveform generation mode control.

Table 4 Waveform Generation Mode Control

WGM2 [2:0]	working mode	TOP Value	Update OCR2x time	Position TOV2 Time
0	Normal	0xFF	immediately	MAX
1	PCPWM	0xFF	TOP	BOTTOM
2	CTC	OCR2A	immediately	MAX
3	FPWM	0xFF	TOP	MAX
4	retain	-	-	-
5	PCPWM	OCR2A	TOP	BOTTOM
6	retain	-	-	-
7	FPWM	OCR2A	TOP	TOP

TCNT2 -TC2 Counter Value Register

TCNT2 -TC2 Counter Value 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	description
7:0	TCNT2	TC2 Count value register. The TCNT2 register provides direct read and write access to the 8-count value of the counter. A CPU write operation to the TCNT2 register prevents a compare match from occurring at the next timer clock cycle, even if the timer has been stopped. This allows the initialization of the TCNT2 register to match the value of OCR2 without triggering an interrupt. If the value written to TCNT2 is equal to or bypasses the OCR2 value, the comparison match is lost, resulting in incorrect waveform generation results. The timer stops counting when no clock source is selected, but the CPU can still access TCNT2. CPU Write counters have higher priority than clear or add/drop operations.

OCR2A - TC2 Output Comparison Register A

OCR2A - TC2 Output Comparison 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	description
7:0	OCR2A	TC2 output comparison register A. The OCR2A contains an 8-bit data that is compared uninterruptedly to the counter value TCNT2. The compare match can be used to generate an output compare interrupt or to generate a waveform on the OC2A pin. When using PWM mode, the OCR2A registers use double-buffered registers. In contrast, the double-buffering function is disabled in normal operation mode and match clear mode. Double buffering synchronizes updating the OCR2A register with the count maximum or minimum moment, thus preventing the generation of asymmetrical PWM pulses and eliminating interference pulses. When using the double buffer function, the CPU accesses the OCR2A buffer register and disables the double buffer function. The CPU accesses the OCR2A itself when it can.

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	descri ption
7:0	OCR2B	TC2 output compares the B register. The OCR2B contains an 8-bit data that is compared uninterrupted to the counter value TCNT2. The compare match can be used to generate an output compare interrupt or to generate a waveform on the OC2B pin. When using PWM mode, the OCR2B register uses double-buffered registers. In contrast, the double-buffering function is disabled in normal operation mode and match clear mode. Double buffering synchronizes updating the OCR2B register with the count maximum or minimum moment, thus preventing the generation of asymmetric PWM pulses and eliminating interference pulses. When the double buffering function is used, the CPU accesses the OCR2B buffer register, and when the double buffering function is disabled the CPU accesses the OCR2B per se.

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	descri ption
7:3		Reserved.
2	OCIE2B	TC2 output compare B match interrupt enable bit. When the OCIE2B bit is "1" and the global interrupt is set, TC2 outputs a compare B match interrupt enable. The interrupt is generated when a compare match occurs, i.e., when the OCF2B bit in TIFR2 is set. When the OCIE2B bit is "0", the TC2 output compare B match interrupt is disabled.
1	OCIE2A	TC2 output compare A match interrupt enable bit. When the OCIE2A bit is "1" and the global interrupt is set, TC2 outputs a compare A match interrupt enable. The interrupt is generated when the compare match occurs, i.e., when the OCF2A bit in TIFR2 is set. When the OCIE2A bit is "0", the TC2 output compare A match interrupt is disabled.
0	TOIE2	TC2 Overflow interrupt enable bit. The TC2 overflow interrupt is enabled when the TOIE2 bit is "1" and the global interrupt is set. The interrupt is generated when TC2 overflows, i.e., when the TOV2 bit in TIFR2 is set. When the TOIE2 bit is "0", the 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	descri ption
7:3	-	Reserved.
2	OCF2B	TC2 output compares the B match flag bits.

		When TCNT2 is equal to OCR2B, the compare unit gives a match signal and sets the compare flag OCF2B. If the output compare B interrupt enable OCIE2B is "1" and the global interrupt flag is set, the output compare B interrupt will be generated. OCF2B will be cleared automatically when this interrupt service routine is executed, or by writing a "1" to the OCF2B bit.
1	OCF2A	TC2 output compares the A match flag bits. When TCNT2 is equal to OCR2A, the comparison unit gives a match signal and sets the comparison flag OCF2A. If the output comparison A interrupt enable OCIE2A is "1" and the global interrupt flag is set, the output comparison A interrupt is generated. OCF2A will be cleared automatically when this interrupt service routine is executed, or by writing a "1" to the OCF2A bit.
0	TOV2	TC2 Overflow flag bit. If the overflow interrupt enable TOIE2 is "1" and the global interrupt flag is set, an overflow interrupt will be generated. TOV2 will be cleared automatically when this interrupt service routine is executed, or by writing a "1" to the TOV2 bit.

ASSR - Asynchronous Interface Status Register

ASSR- TC2 Asynchronous Interface 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	descri ption
7	INTCK	Asynchronous clock selection control bit. When the INTCK bit is set to 1, the internal RC32K is selected as the asynchronous clock source. When the INTCK bit is set to 0, the external crystal clock is selected as the asynchronous clock source.
6	-	Reserved.
5	AS2	Timer 2 Asynchronous mode selection control bit. When the AS2 bit is set to 1, Timer 2 operates in asynchronous mode with its clock source from INTCK bit to choose from. When the AS2 bit is set to 0, Timer 2 operates in synchronous mode and its clock source is Clkio. when the value of AS2 is changed, TCNT2, OCR2A, OCR2B, TCCR2A and TCCR2B send The value of the memory may be incorrect and will need to be reconfigured.
4	TCN2UB	TCNT2 register update flag bit. When Timer 2 is operating in asynchronous mode and a write operation is performed to TCNT2, the TCN2UB bit will be set. When the value of TCNT2 is updated, the hardware will clear the TCN2UB bit. TCNT2 can be updated only when the TCN2UB bit is 0.

3	OCR2AUB	OCR2A register update flag bit. When Timer 2 is operating in asynchronous mode, the OCR2AUB bit is set when a write operation is performed to the OCR2A. When the value of OCR2A is updated, the hardware will clear the OCR2AUB bit. The OCR2A can be updated only when the OCR2AUB bit is 0.
2	OCR2BUB	OCR2B Register update flag bit. When Timer 2 is operating in asynchronous mode and a write operation is performed to OCR2B, the OCR2BUB bit will be set. When the value of OCR2B is updated, the hardware will clear the OCR2BUB bit. The OCR2B can be updated only when the OCR2BUB bit is 0.

1	TCR2AUB	TCCR2A register update flag bit. When Timer 2 is operating in asynchronous mode, the TCR2AUB bit is set when a write operation is performed to TCCR2A. When the value of TCCR2A is updated, the hardware will clear the TCR2AUB bit. The TCCR2A can be updated only when the TCR2AUB bit is 0.
0	TCR2BUB	TCCR2B Register update flag bit. When Timer 2 is operating in asynchronous mode and a write operation is performed to TCCR2B, the TCR2BUB bit will be set. When the value of TCCR2B is updated, the hardware will clear the TCR2BUB bit. The TCCR2B can only be updated when the TCR2BUB bit is 0.

Timer/Counter 3 *(TMR3)*

- True 16-bit design, allowing 16-bit PWM
- 3 independent output comparison units
- Double-buffered output comparison register
- 1 input capture unit
- Input Capture Noise Suppressor
- Automatically clears the counter and automatically loads it when comparing matches
- PWM with phase correction without interference pulses
- Variable PWM Cycle
- Frequency generator
- External event counter
- 5 independent interrupt sources
- With dead time control
- 6 selectable trigger sources automatically turn off the PWM output

summarize

The TC3 is a general-purpose 16-bit timer counter module that supports PWM output for accurate waveform generation.The TC3 contains a 16-bit counter, waveform generation mode control unit, two independent output comparison units and an input capture unit. The waveform generation mode control unit controls the counter's operating mode and the generation of the comparison output waveform. Depending on the operating mode, the counter achieves zero, plus one or minus one operation for each count clock Clkt3, which can be generated by the internal clock source or external clock source. When the counter count value TCNT3 reaches the maximum value (equal to the maximum value 0xFFFF or a fixed value or the output comparison register OCR3A or the input capture register ICR3, defined as TOP, and the maximum value MAX to show the difference), the counter will perform a zero or minus one operation. When the counter count value TCNT3 reaches the minimum value (equal to 0x0000, defined as BOTTOM), the counter will perform a plus one operation. When the counter's count value TCNT3 reaches OCR3A or OCR3B or OCR3C, also known as when a comparison match occurs, it will clear or set the output comparison signal OC3A or OC3B or OC3C to generate a PWM waveform. When the input capture function is enabled, the counter is triggered to start or stop counting, and the ICR3 register records the count value during the trigger period of the capture signal.

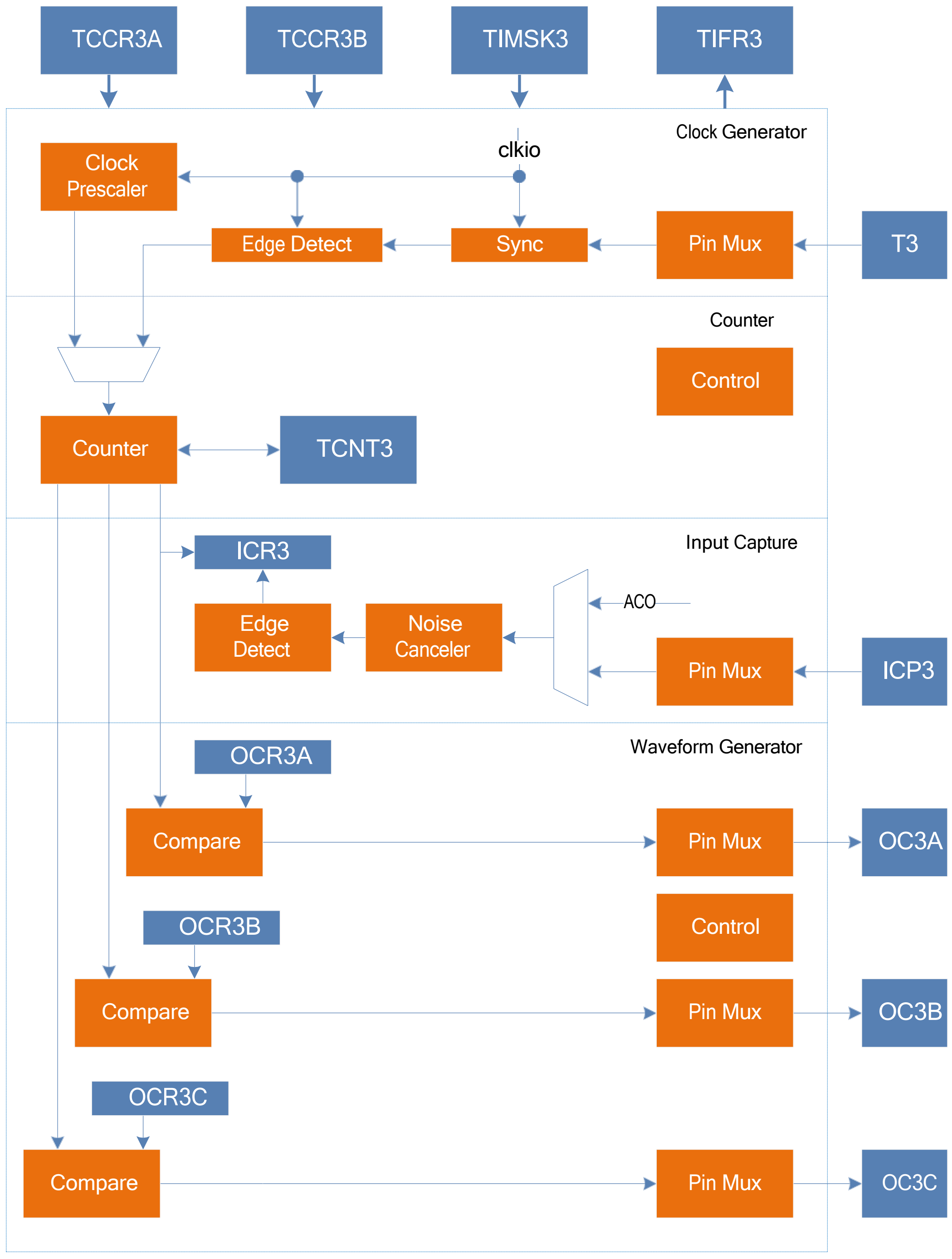


Figure 6 TC3 Structure Diagram

workin

g mode

Timing Counter 1 has six different operating modes, including Normal mode (Normal)Clear on Compare Match (CTC) mode, Fast Pulse Width Modulation (FPWM) mode, Phase Correction Pulse Width Modulation (PCPWM) mode, Phase Frequency Correction Pulse Width Modulation (PFCPWM) mode, and Input Capture (ICP) mode. The selection is made by the waveform generation mode control bits WGM3[3:0]. These six modes are described in detail below. Since there are three independent output comparison units, denoted by "A", "B", and "C", respectively, with a lowercase "x " to denote these two output comparison unit channels.

normal mode

Normal mode is the simplest mode of operation of the timer counter, in which the waveform generation mode control bits WGM3[3:0]=0 and the maximum value of the count is MAX (0xFFFF) In this mode, the count is incremented by one for each count clock, and when the counter reaches TOP overflow, it returns to BOTTOM and starts accumulating again. The timer overflow flag TOV3 is set in the same count clock where the count value TCNT3 goes to zero; in this mode the TOV3 flag is like the 17th count bit, except that it is only set and not cleared. The overflow interrupt service routine automatically clears the TOV3 flag, which can be used by software to increase the resolution of the timer. There are no special circumstances to consider in normal mode, and a new count value can be written at any time.

The waveform of the output comparison signal OC3x can be obtained only when the data direction register of the OC3x pin is set to output. When COM3x=1, the comparison match will flip the OC3x signal, and the frequency of the waveform in this case can be calculated by the following formula.

$f_{OC3xnormal} = fsys/(2*N*65536)$

where N denotes the prescaling factor (1, 8, 64, 256 or 1024)

The output comparison unit can be used to generate interrupts, but interrupts are not recommended in normal mode, as they can take up too much CPU
of time.

CTC model

When WGM3[3:0]=4 or 12 is set, Timer 1 enters CTC mode. When WGM3[3]=0, the maximum count value TOP is OCR3A, when WGM3[3]=1, the maximum count value TOP is ICR3. The following is an example of CTC mode with WGM3[3:0]=4. In this mode, the count mode is incremented by one for each count clock, and the counter is cleared when the counter value TCNT3 equals TOP. This mode allows the user to easily control the frequency of the compare match output and also simplifies the operation of external event counting.

The output compare match flag OCF3A is set when the counter reaches TOP=OCR3A, and the output compare match flag ICF3 is set when the counter reaches TOP=ICR3, and an interrupt will be generated when the corresponding interrupt enable is set. The OCR3A register can be updated in the interrupt service program. In this mode OCR3A does not use double buffering, so be careful when updating the maximum value to a value close to the minimum with the counter operating with no prescaler or a very low prescaler. If the value written to OCR3A is less than the then current TCNT3 value, the counter will lose a compare match. The counter has to count to MAX and then to OCR3A from BOTTOM before the next compare match occurs. as in normal mode, the count value returns to the count clock at 0x0 to set the TOV3 flag.

The waveform of the output comparison signal OC3x can be obtained only when the data direction register of the OC3x pin is set to output. The frequency of the waveform can be calculated using the following equation.

$f_{OC3xctc} = fsys/(2*N*(1+OCR3A))$

where N denotes the prescaling factor (1, 8, 64, 256 or 1024)
From the equation, it can be seen that when setting OCR3A to 0x0 and no prescaler, an output waveform with a maximum frequency of fsys/2 can be obtained.

When WGM3[3:0]=12 is similar to WGM3[3:0]=4, just replace the one associated with OCR3A

with ICR3. Fast PWM Mode

When WGM3[3:0]=5, 6, 7, 14 or 15 is set, Timer 1 enters fast PWM mode and counts the maximum value TOP is 0xFF, 0x1FF, 0x3FF, ICR3 or OCR3A respectively, which can be used to generate high frequency PWM waveforms. Fast PWM

The mode differs from other PWM modes in that it is a one-way operation. The counter accumulates from BOTTOM to TOP and then returns to BOTTOM to recount. When the count value TCNT3 reaches TOP or BOTTOM, the output comparison signal OC3x is set or cleared, depending on the setting of the comparison output mode COM3, as detailed in the register description. Due to the unidirectional operation, the fast PWM mode operates at twice the frequency of the phase correction PWM mode with bidirectional operation. The high frequency feature makes the Fast PWM mode suitable for power regulation, rectification, and DAC applications. The high-frequency signal reduces the size of external components (inductors, capacitors, etc.), thus reducing system cost.

When the count value reaches TOP, the timer counter overflow flag TOV3 will be set and the compare buffer value will be updated to the compare value. If the interrupt is enabled, the OCR3A register can be updated in the interrupt service program.

The waveform of the output comparison signal OC3x can be obtained only when the data direction register of the OC3x pin is set to output. The frequency of the waveform can be calculated by the following equation.

$f_{OC3xfpwm}$ = fsys/(N*(1+TOP))

where N denotes the prescaling factor (1, 8, 64, 256 or 1024)

When TCNT3 and OCR3x are matched by comparison, the waveform generator sets (clears) the OC3x signal, and when TCNT3 is cleared, the waveform generator clears (sets) the OC3x signal to generate a PWM waveform. The resulting polar value of OCR3x will generate a special PWM waveform. When OCR3x is set to 0x00, the output PWM is a narrow spike pulse for every (1+TOP) count clock. When OCR3x is set to TOP, the output waveform is a continuous high or low level. If OCR3A is used as TOP and COM3A=1 is set, the output comparison signal OC3A will generate a PWM waveform with 50% duty cycle.

Phase Correction PWM Mode

When WGM3[3:0]=1, 2, 3, 10 or 11 is set, Timer 1 enters the phase correction PWM mode, and the maximum value of count TOP is 0xFF, 0x1FF, 0x3FF, ICR3 or OCR3A respectively. The counter operates in both directions, incrementing from BOTTOM to TOP, then decrementing to BOTTOM, and then repeating this operation. The count changes direction when it reaches both TOP and BOTTOM, and the count value stays on TOP or BOTTOM for only one count clock. When the count value TCNT3 matches OCR3x during incrementing or decrementing, the output comparison signal OC3x will be cleared or set,depending on the setting of the comparison output mode COM3. The maximum frequency obtainable for bidirectional operation is smaller than for unidirectional operation, but its excellent symmetry is better suited for motor control.

The phase correction PWM mode sets the TOV3 flag when the count reaches BOTTOM and updates the comparison buffer value to the comparison value when the count reaches TOP. If the interrupt is enabled, the comparison buffer OCR3x memory can be updated in the interrupt service program.

The output comparison signal OC3x waveform is obtained only when the data direction register of OC3x pin is set to output. The frequency of the waveform can be calculated by the following formula.

$f_{OC3xcpcpwm}$ = fsys/(N*TOP*2)

where N denotes the prescaling factor (1, 8, 64, 256 or 1024)

During incremental counting, the waveform generator clears (sets) the OC3x signal when TCNT3 matches

OCR3x. During decrement counting, the waveform generator sets (clears) the OC3x signal when TCNT3 matches the OCR3x. The resulting polarity of OCR3x generates a special PWM waveform. When OCR3x is set to TOP or BOTTOM, the OC3x signal loses

The output will always be low or high. If OCR3A is used as TOP and COM3A=1 is set, the output comparison signal OC3A will generate a PWM wave with a duty cycle of 50%.

To ensure the symmetry of the output PWM wave on both sides of the BOTTOM, there are two cases where the OC3x signal is also flipped when no comparison matching occurs. The first case is when the value of OCR3x changes from TOP to other data. When OCR3x is TOP and the count value reaches TOP, the output of OC3x is the same as the result of comparison matching during the previous descending count, i.e., OC3x remains unchanged. At this point, the comparison value is updated to the new OCR3x value (not TOP), and the OC3x value is held until it is flipped when the comparison match occurs during ascending counting. At this point, the OC3x signal is not centered symmetrically on the minimum value, so it is necessary to flip the OC3x signal when TCNT3 reaches its maximum value, which is the first case of flipping the OC3x signal when no comparison match occurs. The second case is when TCNT3 starts counting from a value higher than OCR3x, thus losing a comparison match and causing an asymmetric situation. Again, the OC3x signal needs to be flipped to achieve symmetry on both sides of the minimum.

Phase Frequency Correction PWM Mode

When WGM3[3:0]=8 or 9 is set, Timer 1 enters the phase frequency correction PWM mode, and the maximum value of count TOP is ICR3 or OCR3A respectively. The counter operates in both directions, incrementing from BOTTOM to TOP, then decrementing to BOTTOM and repeating this operation. The count changes direction when it reaches both TOP and BOTTOM, and the count stays on TOP or BOTTOM for only one count clock. When the count value TCNT3 matches OCR3x during incrementing or decrementing, the output comparison signal OC3x will be cleared or set, depending on the setting of the comparison output mode COM3. Compared to unidirectional operation, the maximum frequency available for bidirectional operation is smaller, but its excellent symmetry is better suited for motor control.

In Phase Frequency Corrected PWM mode, the TOV3 flag is set when the count reaches BOTTOM and the comparison buffer value is updated to the comparison value. The time to update the comparison value is the biggest difference between Phase Frequency Corrected PWM mode and Phase Corrected PWM mode. If the interrupt is enabled, the comparison buffer OCR3x memory can be updated in the interrupt service program. When the CPU changes the TOP value, i.e. the value of OCR3A or ICR3, it must ensure that the new TOP value is not smaller than the TOP value already in use, otherwise the comparison match will not occur again.

The output comparison signal OC3x waveform is obtained only when the data direction register of OC3x pin is set to output. The frequency of the waveform can be calculated by the following formula.

$f_{OC3xcpfcpwm} = f_{sys}/(N*TOP*2)$

where N denotes the prescaling factor (1, 8, 64, 256 or 1024)

During incremental counting, the waveform generator clears (sets) the OC3x signal when TCNT3 matches OCR3x. During decrement counting, the waveform generator sets (clears) the OC3x signal when TCNT3 matches the OCR3x. The resulting polarity of OCR3x generates a special PWM waveform. When the OCR3x is set to TOP or BOTTOM, the OC3x signal output will always remain low or high. If OCR3A is used as TOP and COM3A=1 is set, the output comparison signal OC3A generates a PWM wave with a duty cycle of 50%.

Because the OCR3x register is updated at BOTTOM time, the count lengths for ascending and descending are the same on both sides of the TOP value, which also produces a symmetrical waveform with the correct frequency and phase.

When using a fixed TOP value, it is best to use the ICR3 register as the TOP value, i.e. set WGM3[3:0]=8, when the OCR3A register is only needed to generate the PWM output. If you want to generate a PWM wave with varying frequency, you have to change the TOP value by changing it, the double buffering feature of OCR3A will be more suitable for this application.

Input Capture Mode

Input capture is used to capture an external event and assign a time stamp to it to indicate the moment when this event occurred, and can be done in the previous counting mode, although remove the waveform generation mode that uses the ICR3 value as the counting TOP value.

The trigger signal for the occurrence of an external event is input from pin ICP3 and can also be implemented through the analog comparator unit. When the logic level on pin ICP3 changes, or the output ACO level of the analog comparator changes, and this level change is captured by the input capture unit, the input capture is triggered, then the 16-bit count value TCNT3 data is copied to the input capture register ICR3, and the input capture flag ICF3 is set, if the ICIE1 bit is "1", the input capture flag will generate an input capture interrupt.

The input capture trigger source ICP3 or ACO is selected by setting the Analog Compare Input Capture Control bit ACIC in the Analog Compare Control and Status Register ACSR. It should be noted that changing the trigger source may result in a single input capture, so ICF3 must be cleared once after changing the trigger source to avoid erroneous results.

The input capture signal is fed to the edge detector after passing through an optional noise suppressor to see if the detected edge meets the trigger conditions based on the configuration of the input capture selection control bit ICES1. The noise suppressor is a simple digital filter that samples the input signal four times and only feeds its output to the edge detector if all four sample values are equal. The noise suppressor is controlled by the ICNC1 bit of the TCCR3B register to enable or disable it.

When using the input capture function, the value of the ICR3 register should be read as early as possible after ICF3 is set, as the value of ICR3 will be updated after the next capture event occurs. Enabling the input capture interrupt is recommended, and changing the count TOP value during operation is not recommended in any input capture operating mode.

The input captured time stamp can be used to calculate frequency, duty cycle, and other characteristics of the signal, as well as to create a log of trigger events. Measuring the duty cycle of an external signal requires that the trigger edge be changed after each capture, so the trigger signal edge must be changed as soon as possible after the ICR3 value is read.

Automatic shutdown and restart of PWM output

When the DOC3x bit of TCCR3C register is set high, the auto-off function of PWM output will be enabled, and when the trigger condition is met, the hardware will clear the corresponding COM3x bit, disconnect the PWM output signal OC3x from its output pin, and switch to the general-purpose IO output to realize the auto-off of PWM output. At this time, the state of the output pins can be controlled by the output of the general-purpose IO port.

After the PWM output auto-off is enabled, the trigger condition needs to be set, and the DSX3n bit of the TCCR3D register is used to select the trigger source. The trigger sources are analog comparator interrupt, external interrupt, pin level change interrupt, and timer overflow interrupt, as described in the TCCR3D register. When one or more trigger sources are selected as trigger conditions, the hardware will clear the COM3x bit to turn off the PWM output while these interrupt flag bits are set.

When a trigger event occurs to turn off the PWM output, the timer module does not have the corresponding interrupt flag bit, and the software needs to read the interrupt flag bit of the trigger source to know the trigger condition and the trigger event.

When the PWM output is turned off automatically and the output needs to be restarted again, the software simply resets the COM3x bit to switch the OC3x signal output to the appropriate pin. Note that the timer does not stop working after an automatic shutdown occurs, and the state of the OC3x signal is always updated. The software can set the COM3x bit again to output the OC3x signal after a timer overflow or comparison match occurs, so that a clear PWM output state can be obtained.

Dead time control

When DTEN3 is set to "1", the function of inserting the dead time is enabled, and the output waveforms of OC3A and OC3B will insert the set dead time based on the waveform generated by the comparison output of channel B. The length of the time is the time value corresponding to the number of count clocks in the DTR3 register. As shown in the figure below, the dead time insertion of both OC3A and OC3B is based on the comparison output waveform of channel B. When COM3A and COM3B are both "2" or "3", the waveform polarity of OC3A is the same as that of OC3B; when COM3A and COM3B are "2" or "3" respectively, the waveform polarity of OC3A is the same as that of OC3B. When COM3A and COM3B are "2" or "3" respectively, the waveform polarity of OC3A is opposite to that of OC3B.

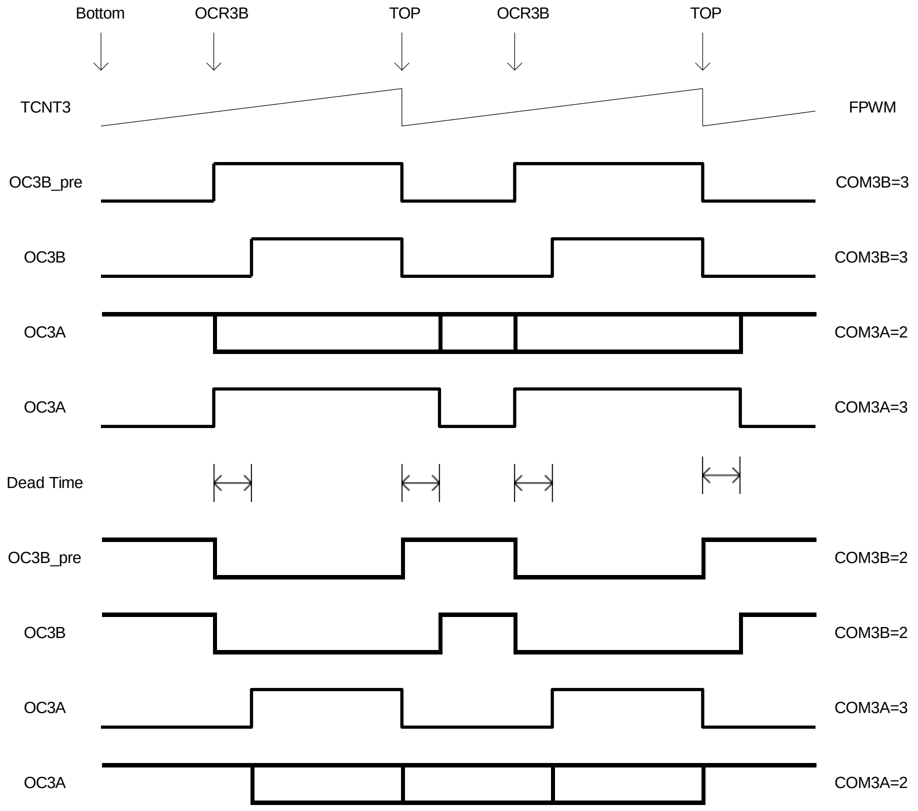


Figure 7 TC3 Dead Time Control in FPWM Mode

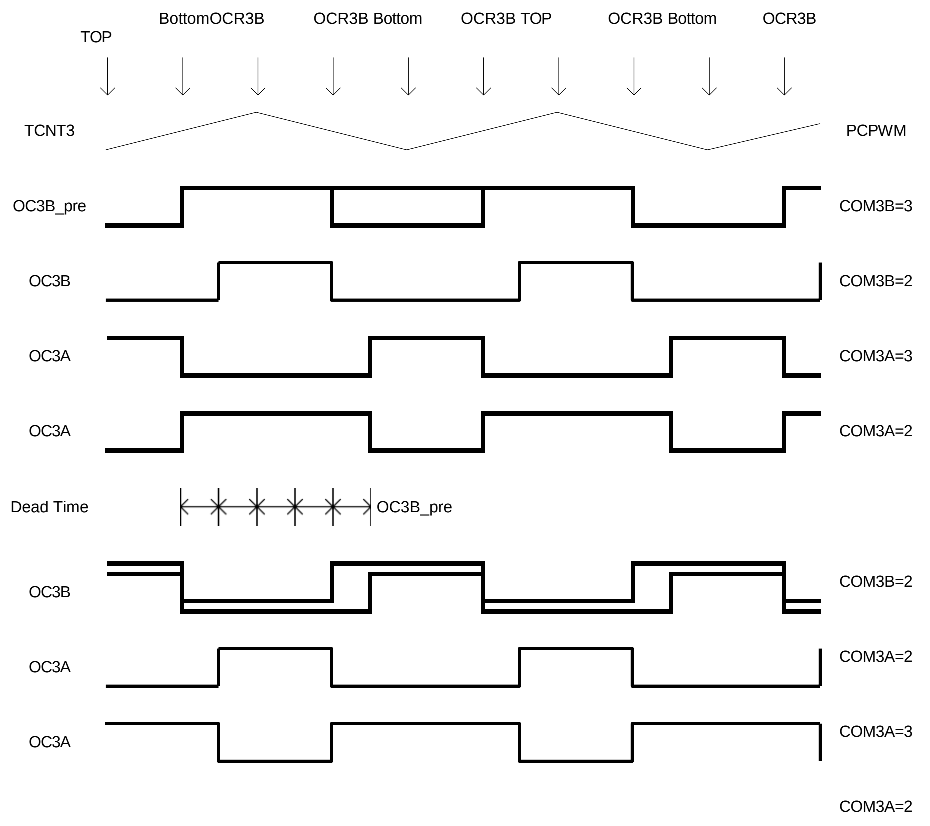


Figure 8 TC3 Dead Time Control in PCPWM Mode

When DTEN3 is set to "0", the function of inserting dead time is disabled, and the output waveforms of OC3A and OC3B are the waveforms generated by their respective comparison outputs.

Register Definition

TC3 Register List

processor register	address	default value	description
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 Count value register low byte
TCNT3H	0x95	0x00	TC3 Count Value 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 Dead Time Register Low Byte
DTR3H	0x9D	0x00	TC3 High Byte of Dead Time Register
OCR3CL	0x9E	0x00	TC3 Output Compare Register C Low Byte

OCR3CH	0x9F	0x00	TC3 Output Compare Register C High Byte
TIMSK3	0x71	0x00	Timing counter interrupt mask register
TIFR3	0x38	0x00	Timing counter interrupt flag register

TCCR3A-TC3 Control Register A

TCCR3A -TC3 Control Register A								
Address: 0x90					Default value: 0x00			
Bit	7	6	5	4	3	2	1	0
Name	COM3A1	COM3A0	COM3B1	COM3B0	COM3C1	COM3C0	WGM31	WGM30
R/W	R/W	R/W	R/W	R/W	W	W	R/W	R/W

Bit	Name	description
7	COM3A1	Compare Match Output A Mode Control High. COM3A1 and COM3A0 form COM3A[1:0] to control the output compare waveform OC3A. if either bit 1 or bit 2 of COM3A is set, the output compare waveform occupies the OC3A pin, but the data direction register of this pin must be set high to output this waveform. The COM3A controls the output compare waveform differently in different operating modes, as described in the comparison output mode control table.
6	COM3A0	Compare Match Output A Mode Control Low. COM3A1 and COM3A0 form COM3A[1:0] to control the output comparison waveform OC3A. if either bit 1 or bit 2 of COM3A is set, the output comparison waveform occupies the OC3A pin, but the data direction register of this pin must be set high to output this waveform. The COM3A controls the output comparison waveform differently in different operating modes, see the comparison output mode control table for details Grid Description.
5	COM3B1	Compare Match Output B Mode Control High. COM3B1 and COM3B0 form COM3B[1:0] to control the output compare waveform OC3B. if either bit 1 or bit 2 of COM3B is set, the output compare waveform occupies the OC3B pin, but the data direction register of this pin must be set high to output this waveform. The COM3B controls the output compare waveform differently in different operating modes, as described in the compare output mode control table.
4	COM3B0	Compare Match Output B Mode Control Low. COM3B1 and COM3B0 form COM3B[1:0] to control the output compare waveform OC3B. if either bit 1 or bit 2 of COM3B is set, the output compare waveform occupies the OC3B pin, but the data direction register of this pin must be set high to output this waveform. The COM3B controls the output compare waveform differently in different operating modes, as described in the comparison output mode control table.

3	COM3C1	Compare Match Output C Mode Control High. COM3C1 and COM3C0 form COM3C[1:0] to control the output compare waveform OC3C. if either bit 1 or bit 2 of COM3C is set, the output compare waveform occupies the OC3C pin, but the data direction register of this pin must be set high to output this waveform. The COM3C controls the output compare waveform differently in different operating modes, as shown in the compare output mode control table Grid Description.
2	COM3C0	Compare Match Output C Mode Control Low. COM3C1 and COM3C0 form COM3C[1:0] to control the output comparison waveform OC3C. if

		With either bit 1 or bit 2 of COM3C set, the output compare waveform occupies the OC3C pin, although the data direction register of this pin must be set high to output this waveform. The COM3C controls the output compare waveform differently in different operating modes, as described in the compare output mode control table.
1	WGM31	The waveform generation mode controls the next lowest level. WGM31 and WGM33,WGM32,WGM30 together form the waveform generation mode control WGM3[3:0], which controls the counter counting mode and waveform generation mode, as described in the waveform generation mode table.
0	WGM30	The waveform generation mode controls the lowest level. WGM30 and WGM33,WGM32,WGM31 together form the waveform generation mode control WGM3[3:0], which controls the counter counting mode and waveform generation mode, as described in the waveform generation mode table.

The following table shows the control of the comparison output mode on the output comparison waveform in non-PWM modes (i.e. normal mode and CTC mode).

Comparative output mode control in non- PWM mode

COM3x[1:0]	description
0	OC3x disconnected, general purpose IO port operation
1	Flip OC3x signal when comparing matches
2	Clear OC3x signal when comparing matches
3	Set OC3x signal when comparing matches

The following table shows the control of the comparison output mode on the output comparison waveform in fast PWM mode.

Fast PWM Mode Comparative Output Mode Control

COM3x[1:0]	description
0	OC3x disconnected, general purpose IO port operation
1	When WGM3 is 15: Flip OC3A signal when comparing matches, OC3B disconnected When WGM3 is other values: OC3x disconnected, general purpose IO port operation
2	Clear OC3x signal for comparison match, set OC3x signal for maximum match
3	Set OC3x signal for comparison match, clear OC3x signal for maximum match

The following table shows the control of the output comparison waveform by the comparison output mode in phase correction mode.

Phase Correction and Phase Frequency Correction PWM Mode Comparative Output Mode Control

COM3x[1:0]	description
0	OC3x disconnected, general purpose IO port operation
1	When WGM3 is 9 or 11: Flip OC3A signal when comparing matches, OC3B disconnected When WGM3 is other values: OC3x disconnected, general purpose IO port operation

2	Comparative match clear OC3x signal in ascending count, comparative match set in descending count OC3x signal
3	Comparative match set OC3x signal in ascending count, comparative match clear in descending count OC3x signal

TCCR3B-TC3 Control Register B

TCCR3B -TC3 Control Register B								
Address: 0x91					Default value: 0x00			
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	description
7	ICNC3	Input Capture Noise Suppressor Enable Control bit. When the ICNC3 bit is set to "1", the input capture noise suppressor is enabled, and the external pin ICP3 The input is filtered so that the input signal is valid only when 4 consecutive samples are equal. The incoming capture is delayed by 4 clock cycles. When the ICNC3 bit is set to "0", input capture of the noise suppressor is disabled, and the external pin ICP3 The input is straightforward and effective.
6	ICES3	Input capture trigger edge selection control bit. When the ICES3 bit is set to "1", the rising edge of the selected level triggers input capture; when ICES3 is set When the bit is "0", the falling edge of the selected level triggers the input capture. When an event is captured, the counter value is copied to the ICR3 register and the input capture flag ICF3 is set. if the interrupt is enabled, an input capture interrupt is generated.
5	-	Reserved.
4	WGM33	The waveform generation mode controls the high level. WGM33, together with WGM32, WGM31 and WGM30, form the waveform generation mode control WGM3[3:0], which controls the counter counting mode and waveform generation mode, as described in the waveform generation mode table.
3	WGM32	The waveform generation mode controls the next highest level. WGM32 and WGM33,WGM31,WGM30 together form the waveform generation mode control WGM3[3:0], which controls the counter counting mode and waveform generation mode, as described in the waveform generation mode table.
2	CS32	Clock Select controls the high position. Used to select the clock source for Timer Counter 3.
1	CS31	The clock selects the control median. Used to select the clock source for Timer Counter 3.
0	CS30	Clock Select controls the low position. Used to select the clock source for Timer Counter 3.

CS3[2:0]	description
0	No clock source, stop counting
1	clksys
2	clksys/8, from prescaler

		3	clksys/64, from prescaler
		4	clksys/256, from prescaler
		5	clksys/1024 from prescaler
		6	External clock T3 pin, falling edge triggered
		7	External clock T3 pin, rising edge triggered

The following table shows the waveform generation mode control.

Table 5 Waveform Generation Mode Control

WGM3 [3:0]	working mode	TOP Value	Update OCR1A Hour	Position 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	OCR3A	BOTTOM	BOTTOM
10	PCPWM	ICR3	TOP	BOTTOM
11	PCPWM	OCR3A	TOP	BOTTOM
12	CTC	ICR3	immediately	MAX
13	retain	-	-	-
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
Name	FOC3A	FOC3B	DOC3B	DOC3A	DTEN3	-	DOC3C	FOC3C
R/W	W	W	-	-	-	-	-	-

Bit	Name	description
7	FOC3A	Forced output comparison A. When operating in non- PWM mode, a compare match can be generated by writing a "1" to the forced output compare bit FOC3A. The forced compare match will not set the OCF3A flag, nor will it reload or clear the timer, but the output pin OC3A will be updated accordingly to the COM3A setting, as if a compare match had actually occurred. When operating in PWM mode, clear the TCCR3A register to zero when writing it. The return value for reading FOC3A is always zero.
6	FOC3B	Forced output comparison B. When operating in non- PWM mode, a compare match can be generated by writing a "1" to the forced output compare bit FOC3B. The forced compare match will not set the OCF3B flag, nor will it reload or clear the timer, but the output pin OC3B will be updated accordingly to the COM3B setting, just as if a compare match had actually occurred. When operating in PWM mode, clear the TCCR3A register to zero when writing it. The return value for reading FOC3B is always zero.

5	DOC3B	Disable output compare B Enable control bit.

		When the DOC3B bit is high, hardware disable output compare B is enabled and the disable output condition is met when After the COM3B bit is cleared, the output pin OC3B is disconnected and the pin becomes a general purpose IO operation. When the DOC3B bit is low, the hardware disable output compare B function is disabled.
4	DOC3A	Disable the Output Compare A enable control bit. When the DOC3A bit is high, hardware disable output compare A is enabled, and after the disable output condition is met, the COM3A bit is cleared, output pin OC3A is disconnected, and the pin becomes a general purpose IO operation. When the DOC3A bit is low, the hardware disable output compare A function is disabled.
3	DTEN3	Dead time enable control bit. When the DTEN3 bit is high, the dead time is enabled, OC3A and OC3B become complementary outputs, and the dead time is inserted as set by DTR3L and DTR3H. When the DTEN3 bit is low, dead time is disabled.OC3A and OC3B are both single outputs.
2	-	
1	DOC3C	Disable output compare C Enable control bit. When the DOC3C bit is high, hardware disable output compare C is enabled, and after the disable output condition is met, the COM3C bit is cleared, output pin OC3C is disconnected, and the pin becomes a general purpose IO operation. When the DOC3C bit is low, the hardware disable output compare C function is disabled.
0	FOC3C	Forced output comparison C. When operating in non- PWM mode, a compare match can be generated by writing a "1" to the forced output compare bit FOC3C. The forced compare match will not set the OCF3C flag, nor will it reload or clear the timer, but the output pin OC3C will be updated accordingly to the COM3C setting, as if a compare match had actually occurred. When operating in PWM mode, clear the TCCR3A register to zero when writing it. The return value for reading FOC3C is always 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
Name	DSX37	DSX36	DSX35	DSX34	-	-	DSX31	DSX30
R/W	R/W	R/W	R/W	R/W	-	-	R/W	R/W

Bit	Name	description
7	DSX37	TC3 Trigger Source Select Control Enable Bit 7. When the DSX37 bit is set to "1", TC0 overflow is enabled as the trigger source to turn off the output comparison signal waveform OC3x. When the DOC3x bit is "1", the rising edge of the interrupt flag register bit of the selected trigger source will automatically turn off the waveform output of OC3x. When the DSX37 bit is set to "0", TC0 overflows as the off output comparison signal waveform OC3x of the trigger source is disabled.

6	DSX36	TC3 Trigger Source Select Control Enable Bit 6. When the DSX36 bit is set to "1", TC2 overflow is enabled as the trigger source to turn off the output comparison signal waveform OC3x. When the DOC3x bit is "1", the rising edge of the interrupt flag register bit of the selected trigger source will automatically turn off the waveform output of OC3x. When the DSX36 bit is set to "0", TC2 overflows as the off output comparison signal waveform OC3x of the trigger source is disabled.

5	DSX35	TC3 Trigger Source Select Control Enable Bit 5. When the DSX35 bit is set to "1", the pin level changes by 1 as the trigger source to turn off the output comparison signal waveform OC3x is enabled. When the DOC3x bit is "1", the rising edge of the interrupt flag register bit of the selected trigger source will automatically turn off the waveform output of OC3x. When the DSX35 bit is set to "0", the pin level is changed by 1 as an off output comparison signal wave The trigger source of the shaped OC3x is disabled.
4	DSX34	TC3 Trigger Source Select Control Enable Bit 4. When the DSX34 bit is set to "1", external interrupt 1 is enabled as the trigger source to turn off the output comparison signal waveform OC3x. When the DOC3x bit is "1", the rising edge of the interrupt flag register bit of the selected trigger source will automatically turn off the waveform output of OC3x. When the DSX34 bit is set to "0", the external interrupt 1 is used to turn off the output comparison signal waveform. The OC3x's trigger source is disabled.
3:2	-	Reserved.
1	DSX31	TC3 Trigger Source Select Control Enable Bit 1. When the DSX31 bit is set to "1", Analog Comparator 1 is enabled as the trigger source to turn off the output comparison signal waveform OC3x. When the DOC3x bit is "1", the rising edge of the interrupt flag register bit of the selected trigger source will automatically turn off the waveform output of OC3x. When DSX31 bit is set to "0", analog comparator 1 is used to turn off the output comparison signal waveform. The OC3x's trigger source is disabled.
0	DSX30	TC3 Trigger Source Select Control Enable Bit 0. When the DSX30 bit is set to "1", analog comparator 0 is enabled as the trigger source to turn off the output comparison signal waveform OC3x. When the DOC3x bit is "1", the rising edge of the interrupt flag register bit of the selected trigger source will automatically turn off the waveform output of OC3x. When the DSX30 bit is set to "0", the analog comparator 0 is used to turn off the output comparison signal waveform. The OC3x's trigger source is disabled.

The following table shows the selection control of the trigger source for the waveform output.

Turn off trigger source selection control for OC3x waveform output

DOC3x	DSX3n=1	trigger source	description
0	-	-	DOC3x bit is "0", the trigger source off waveform output function is disabled
1	0	Analog Comparator 0	The rising edge of ACIF0 will turn off the OC3x waveform output
1	1	Analog comparator 1	The rising edge of ACIF1 will turn off the OC3x waveform output
1	4	External interrupt 1	The rising edge of INTF1 will turn off the OC3x waveform output
1	5	Pin level change 1	The rising edge of PCIF1 will turn off the OC3x waveform output

1	6	TC2 Overflow	The rising edge of TOV2 will turn off the OC3x waveform output
1	7	TC0 Overflow	The rising edge of TOV0 will turn off the OC3x waveform output

Caution.

2) DSX3n=1 means that when bit n of the TCCR1D register is 1, each register bit can be set at the same time.

TCNT3L-TC3 Counter Register Low Byte

TCNT3L -TC3 Count Value Register Low Byte								
Address: 0x94					Default value: 0x00			
Bit	7	6	5	4	3	2	1	0

Name	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	description
7:0	TCNT3L	TC3 The low byte of the count value. TCNT3H and TCNT3L are combined together to form TCNT3, which provides direct read and write access to the counter's 16-bit count value through the TCNT3 register. Two operations are required to read and write the 16-bit register. When writing 1 6 - b i t TCNT3, TCNT3H should be written first. when reading 16-bit TCNT3, TCNT3L should be read first. A CPU write operation to the TCNT3 register prevents a compare match from occurring on the next timer clock cycle, even if the timer has been stopped. This allows the initialization of the TCNT3 register to match the value of OCR3x without triggering an interrupt. If the value written to TCNT3 is equal to or bypasses the OCR3x value, the comparison match is lost, resulting in incorrect waveform generation results. The timer stops counting when no clock source is selected, but the CPU can still access TCNT3. The CPU writes the counter than clears or adds The minus operation has high priority.

TCNT3H-TC3 Counter Register High Byte

TCNT3H -TC3 Counter Value Register High Byte								
Address: 0x95					Default value: 0x00			
Bit	7	6	5	4	3	2	1	0
Name	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	description
7:0	TCNT3H	TC3 The high byte of the count value. TCNT3H and TCNT3L are combined to form TCNT3, which provides direct read and write access to the counter's 16-bit count value through the TCNT3 registerTwo operations are required to read and write the 16-bit register.When writing 16-bit TCNT3,TCNT3H should be written first. when reading 16-bit TCNT3, TCNT3L should be read first. A CPU write operation to the TCNT3 register prevents a compare match from occurring at the next timer clock cycle, even if the timer has been stopped. This allows the initialization of the TCNT3 register to match the value of OCR3x without triggering an interrupt. If the value written to TCNT3 is equal to or bypasses the OCR3x value, the comparison match is lost, resulting in incorrect waveform generation results. The timer stops counting when no clock source is selected, but the CPU can still access TCNT3. The CPU writes the counter than clears or adds The minus operation has high priority.

ICR3L-TC3 Capture Register Low Byte

ICR3L -TC3 Input Capture Register Low Byte								
Address: 0x96					Default value: 0x00			
Bit	7	6	5	4	3	2	1	0
Name	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	description

7:0	ICR3L	TC3 The low byte of the input capture value. ICR3H and ICR3L are combined to form the 16-bit ICR3. reading and writing 16-bit registers requires two operations. Writing 16-bit For ICR3, ICR3H should be written first. for reading 16-bit ICR3, ICR3L should be read first.

		When an input capture is triggered, the count value TCNT3 is updated and copied to the ICR3 register.The ICR3 register can also be used to Defines the TOP value of the count.

ICR3H-TC3 Capture Register High Byte

ICR3H -TC3 Input Capture Register High Byte								
Address: 0x97					Default value: 0x00			
Bit	7	6	5	4	3	2	1	0
Name	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	description
7:0	ICR3H	TC3 The high byte of the input capture value. ICR3H and ICR3L are combined to form the 16-bit ICR3. reading and writing 16-bit registers requires two operations. Writing 16-bit For ICR3, ICR3H should be written first. for reading 16-bit ICR3, ICR3L should be read first. When an input capture is triggered, the count value TCNT3 is updated and copied to the ICR3 register.The ICR3 register can also be used to define the TOP value of the count.

OCR3AL-TC3 Output Compare Register A Low Byte

OCR3AL -TC3 Output Compare R e g i s t e r A Low Byte								
Address: 0x98					Default value: 0x00			
Bit	7	6	5	4	3	2	1	0
Name	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	description
7:0	OCR3AL	Output the low byte of compare register A. OCR3AL and OCR3AH are combined together to form the 16-bit OCR3A. reading and writing 16-bit registers requires two operations. Write For 16-bit OCR3A, OCR3AH should be written first. for reading 16-bit OCR3A, OCR3AL should be read first. The OCR3A compares uninterruptedly with the counter value TCNT3. The compare match can be used to generate an output compare interrupt or to generate a waveform on the OC3A pin. When using PWM mode, the OCR3A registers use double-buffered registers. In contrast, the double buffer function is disabled in normal operation mode and match clear mode. Double buffering synchronizes updating the OCR3A register with the count maximum or minimum moment, thus preventing the generation of asymmetrical PWM pulses and eliminating interference pulses. When using the double buffer function, the CPU accesses the OCR3A buffer register, and when disabling the double buffer function the CPU accesses the OCR3A per se.

OCR3AH-TC3 Output Compare Register A High Byte

OCR3AH -TC3 Output Compare R e g i s t e r 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	description
7:0	OCR3AH	Output the high byte of compare register A.

		OCR3AL and OCR3AH are combined together to form the 16-bit OCR3A. reading and writing 16-bit registers requires two operations. Write For 16-bit OCR3A, OCR3AH should be written first. for reading 16-bit OCR3A, OCR3AL should be read first. The OCR3A compares uninterruptedly with the counter value TCNT3. The compare match can be used to generate an output compare interrupt or to generate a waveform on the OC3A pin. When using PWM mode, the OCR3A registers use double-buffered registers. In contrast, the double buffer function is disabled in normal operation mode and match clear mode. Double buffering synchronizes updating the OCR3A register with the count maximum or minimum moment, thus preventing the generation of asymmetrical PWM pulses and eliminating interference pulses. When using the double buffer function, the CPU accesses the OCR3A buffer register, and when disabling the double buffer function the CPU accesses the OCR3A per se.

OCR3BL-TC3 Output Compare Register B Low Byte

OCR3BL -TC3 Output Compare R e g i s t e r B Low Byte								
Address: 0x9A				Default value: 0x00				
Bit	7	6	5	4	3	2	1	0
Name	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	description						
7:0	OCR3BL	Output the low byte of compare register B. OCR3BL and OCR3BH are combined together to form a 16-bit OCR3B. reading and writing 16-bit registers requires two operations. Write For 16-bit OCR3B, OCR3BH should be written first. for reading 16-bit OCR3B, OCR3BL should be read first. The OCR3B compares uninterruptedly with the counter value TCNT3. The compare match can be used to generate an output compare interrupt or to generate a waveform on the OC3B pin. When using PWM mode, the OCR3B registers use double-buffered registers. In contrast, the double-buffering function is disabled in normal operation mode and match clear mode. Double buffering synchronizes updating the OCR3B register with the count maximum or minimum moment, thus preventing the generation of asymmetric PWM pulses and eliminating interference pulses. When using the double buffering feature, the CPU accesses the OCR3B buffer register, and when disabling the double buffering feature, the CPU accesses the OCR3B itself.						

OCR3BH-TC3 Output Compare Register B High Byte

OCR3BH -TC3 Output Compare R e g i s t e r B High Byte								
Address: 0x9B				Default value: 0x00				
Bit	7	6	5	4	3	2	1	0
Name	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	description						

7:0	OCR3BH	Output the high byte of compare register B. OCR3BL and OCR3BH are combined to form the 16-bit OCR3B. reading and writing 16-bit registers requires two operations. Write For 16-bit OCR3B, OCR3BH should be written first. for reading 16-bit OCR3B, OCR3BL should be read first. The OCR3B compares uninterruptedly with the counter value TCNT3. The compare match can be used to generate an output compare interrupt or to generate a waveform on the OC3B pin. When using PWM mode, the OCR3B registers use double-buffered registers. In contrast, the double buffer function is disabled in normal operation mode and match clear mode. Double buffering can update the OCR3B register with the count maximum or minimum value when This prevents t h e generation of asymmetrical PWM pulses and eliminates interference pulses.

		When using the double buffering feature, the CPU accesses the OCR3B buffer register, and when disabling the double buffering feature, the CPU accesses the 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
Name	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	description
7:0	OCR3CL	Output the low byte of compare register C. OCR3CL and OCR3CH are combined together to form the 16-bit OCR3C. reading and writing 16-bit registers requires two operations. Write 16 When reading a 16-bit OCR3C, OCR3CH should be written first. when reading a 16-bit OCR3C, OCR3CL should be read first. The OCR3C compares uninterruptedly with the counter value TCNT3. The compare match can be used to generate an output compare interrupt or to generate a waveform on the OC3C pin. When using PWM mode, the OCR3C registers use double-buffered registers. In contrast, the double-buffering function is disabled in normal operation mode and match clear mode. Double buffering synchronizes updating the OCR3C register with the count maximum or minimum moment, thus preventing the generation of asymmetric PWM pulses and eliminating interference pulses. When using the double buffering function, the CPU accesses the OCR3C buffer register, and when disabling the double buffering function, the CPU accesses the OCR3C Per se.

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
Name	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	description
7:0	OCR3CH	Outputs the high byte of compare register C. OCR3CL and OCR3CH are combined together to form the 16-bit OCR3C. reading and writing 16-bit registers requires two operations. Write For 16-bit OCR3C, OCR3CH should be written first. for reading 16-bit OCR3C, OCR3CL should be read first. The OCR3C compares uninterruptedly with the counter value TCNT3. The compare match can be used to generate an output compare interrupt or to generate a waveform on the OC3C pin. When using PWM mode, the OCR3C registers use double-buffered registers. In contrast, the double-buffering function is disabled in normal operation mode and match clear mode. Double buffering synchronizes updating the OCR3C register with the count maximum or minimum moment, thus preventing the generation of asymmetric PWM pulses and eliminating interference pulses. When using the double buffering function, the CPU accesses the OCR3C buffer register, and when disabling the double buffering function, the CPU accesses the 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
Name	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	description						
7:0	DTR3L	Dead time register low byte. When the DTEN3 bit is high, OC3A and OC3B are complementary outputs and the dead time inserted on the OC3A output is determined by DTR3L The number of count clocks is determined.						

DTR3H-TC3 Dead Time Register High Byte

DTR3H -TC3 Dead Time Register High Byte								
Address: 0x9D					Default value: 0x00			
Bit	7	6	5	4	3	2	1	0
Name	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	description						
7:0	DTR3H	Dead time register high byte. When the DTEN3 bit is high, OC3A and OC3B are complementary outputs and the dead time inserted on the OC3B output is determined by DTR3H The number of count clocks is determined.						

TIMSK3-TC3 Interrupt Mask Register

TIMSK3 - TC3 Interrupt Mask Register								
Address: 0x71					Default value: 0x00			
Bit	7	6	5	4	3	2	1	0
Name	-	-	ICIE3	-	OCIE3C	OCIE3B	OCIE3A	TOIE3
R/W	-	-	R/W	-	R/W	R/W	R/W	R/W
Bit	Name	description						
7:6	-	Reserved.						
5	ICIE3	TC3 Input capture interrupt enable control bit. When the ICIE3 bit is "1" and the global interrupt is set, the TC3 input capture interrupt is enabled. When the input capture is triggered, i.e., the ICF3 flag of TIFR3 is set, the interrupt is generated. When the ICIE3 bit is "0", the TC3 input capture interrupt is disabled.						
4	-	Reserved.						
3	OCIE3C	TC3 Output Compare C Match interrupt enable bit. When the OCIE3C bit is "1" and the global interrupt is set, TC3 outputs a compare C match interrupt enable. The interrupt is generated when a compare match occurs, i.e., when the OCF3C bit in TIFR3 is set. When the OCIE3C bit is "0", the TC3 output compare C match interrupt is disabled.						
2	OCIE3B	TC3 output compare B match interrupt enable bit. When the OCIE3B bit is "1" and the global interrupt is set, TC3 outputs a compare B match interrupt enable. The interrupt is generated when the compare match occurs, i.e., when the OCF3B bit in TIFR3 is set. When the OCIE3B bit is "0", the TC3 output compare B match interrupt is disabled.						

1	OCIE3A	TC3 output compare A match interrupt enable bit. When the OCIE3A bit is "1" and the global interrupt is set, TC3 outputs a compare A match interrupt enable. When the compare match occurs

		The interrupt is generated when the OCF3A bit in TIFR3 is set. When the OCIE3A bit is "0", TC3 outputs a comparison A Matching interrupts are disabled.
0	TOIE3	TC3 Overflow interrupt enable bit. When the TOIE3 bit is "1" and the global interrupt is set, TC3 overflow interrupt is enabled. When an overflow occurs in TC3, i.e., when the global interrupt in TIFR3 The interrupt is generated when the TOV3 bit is set. When the TOIE3 bit is "0", the 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
Name	-	-	ICF3	-	-	OCF3B	OCF3A	TOV3
R/W	-	-	R/W	-	-	R/W	R/W	R/W

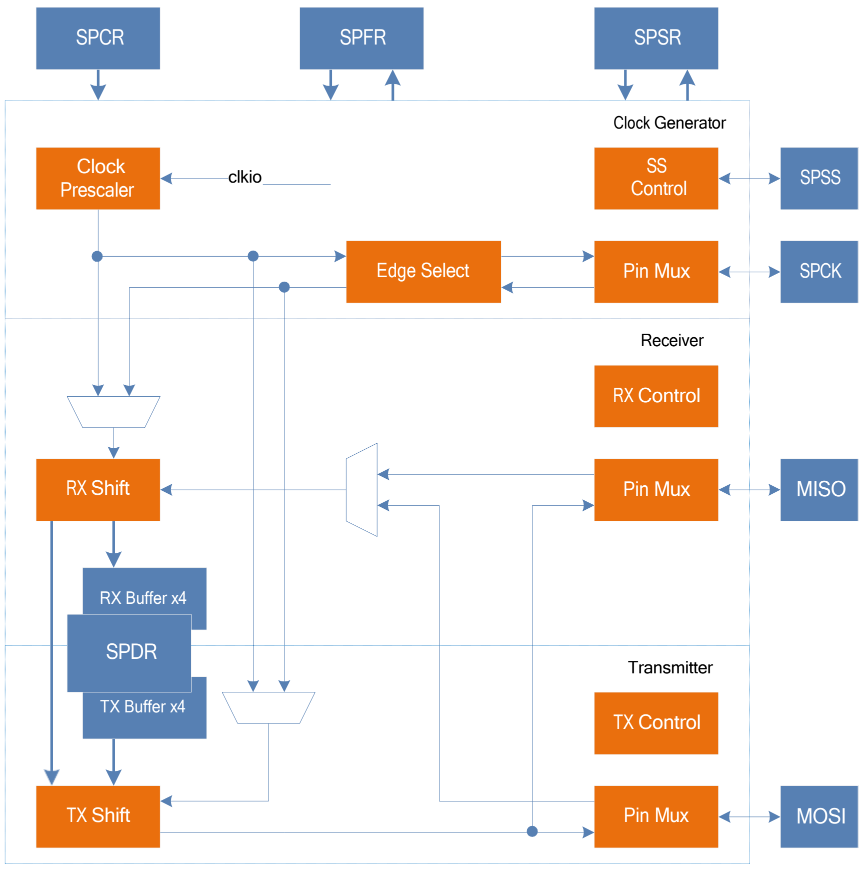
Bit	Name	description
7:6	-	Reserved.
5	ICF3	Enter the capture flag bit. The ICF3 flag is set when an input capture event occurs. The ICF3 flag is set when ICR3 is used as the TOP value of the count and the count value reaches the TOP value. If ICIE1 is "1" and the global interrupt flag is set, an input capture interrupt is generated. the ICF3 flag bit is not automatically cleared and requires software to write a "1" to the ICF3 bit to clear it.
4	-	Reserved.
3	OCF3C	The output c o m p a r e s t h e C match flag bits. When TCNT3 is equal to OCR3C, the compare unit gives a match signal and sets the compare flag OCF3C. if the output compare interrupt enable OCIE3C is "1" and the global interrupt flag is set, an output compare interrupt is generated. the OCF3 flag bit is not automatically cleared and requires the software OCF3C The OCF3 flag bit is not automatically cleared, but requires a software OCF3C bit write "1" to clear the bit.
2	OCF3B	The output c o m p a r e s t h e B match flag bit. When TCNT3 is equal to OCR3B, the compare unit gives a match signal and sets the compare flag OCF3B. If the output compare interrupt enable OCIE3B is "1" and the global interrupt flag is set, an output compare interrupt will be generated. The OCF3B flag bit is not automatically cleared, but needs to be cleared by software by writing "1" to the OCF3B bit.
1	OCF3A	The output c o m p a r e s t h e A match flag bit. When TCNT3 is equal to OCR3A, the compare unit gives a match signal and sets the compare flag OCF3A. If the output compare interrupt enable OCIE3A is "1" and the global interrupt flag is set, an output compare interrupt will be generated. The OCF3A flag bit is not automatically cleared, but needs to be cleared by writing "1" to the OCF3A bit.
0	TOV3	Overflow flag bit. If the overflow interrupt enable TOIE3 is "1" and the global interrupt flag is set, an overflow interrupt will be generated. The TOV3 flag bit is not automatically cleared and requires the software to write a "1" t o t h e TOV3 bit to clear it.

Synchronous Serial Peripheral Interface *(SPI)*

- Full duplex, three-wire synchronous data transmission
- Host or slave operation
- Lowest or highest bit priority transmission
- 7 programmable bit rates
- Send end of interrupt flag
- Writing in conflict flag protection mechanisms
- Can be woken up from idle mode
- Host operation with multiplier mode
- Supports host dual line input mode
- 4 cache registers for both input/output

a general narrative

The SPI consists of three main parts: a clock prescaler, a clock detector, a slave selection detector, a transmitter and a receiver.



SPI Structure Diagram

The control and status registers are shared by these three sections. The clock prescaler operates in the host operation mode only and is used by the bit rate control bits to select the dividing factor, which generates the corresponding dividing clock that is output to the SPCK pin. The clock detector operates in the slave operation mode only and detects the clock edge input from the SPCK pin to perform shift operations on the transmit and receive shift registers according to the SPI data transfer mode. The slave select detector detects the slave select signal SPSS and obtains

The state of the transmission is used to control the operation of the transmitter and receiver. The transmitter consists of a shift register and transmit control logic. The receiver consists of a shift register, four receive buffers, and receive control logic.

clock generation

The clock generation logic is divided into a host clock prescaler and a slave clock detector, operating in the host and slave operation modes, respectively. The clock prescaler selects the dividing factor by the bit rate control bit and the multiplier control bit to generate the corresponding dividing clock (there are seven selectable dividing factors, see register description for details) which is output to the SPCK pin to provide a clock for communication and a shift clock for the internal transmit and receive shift registers. The clock detector performs edge detection of the input clock SPCK and shifts the transmitter and receiver according to the SPI's data transfer mode. To ensure proper sampling of the clock signal, the width of both the high and low levels of the SPCK clock must be greater than two system clock cycles.

Sending and receiving

The SPI module supports simultaneous transmit and receive in single-wire mode, and host-only dual-wire receive in dual-wire mode.

Single line send and receive

The host of the SPI starts a transfer process by pulling the slave selection signal SPSS low that needs to be communicated with. The host and slave prepare the data to be transferred, and the host generates a clock pulse on the clock signal SPCK to exchange the data, moving the data out of MOSI and in from MISO for the host and out of MISO and in from MOSI for the slave.

When configured as a master, the SPI module does not control the SPSS pin and must be handled by the user software. Software pulls the SPSS pin low, selects the slave to communicate with, and initiates the transfer. By writing the data to be transferred to the SPDR register, the software starts the clock generator, the hardware generates the clock for communication, and shifts the 8-bit data out to the slave and the slave's data in at the same time. After shifting one byte of data, the clock generator is stopped and the transfer completion flag SPIF is set. software can write data to the SPDR register again to continue the next byte transfer, or pull up the SPSS signal to end the current transfer. The last incoming data will be stored in the receive buffer.

When configured as a slave, the SPI module will remain asleep and keep the MISO pin tri-stated as long as the SPSS signal remains high. At this point the software can update the contents of the SPDR register. Even if there is an input clock pulse on the SPCK pin at this time, the data in SPDR will not be shifted out until the SPSS signal is pulled low. When a byte of data has been transferred, the hardware sets the transfer completion flag SPIF, at which point software can continue to write data to the SPDR register before reading the shifted-in data, and the last incoming data is stored in the receive buffer.

The SPI module has only four buffers in the transmit direction, and four buffers in the receive direction. When sending data, the SPDR register can be written when the transmit buffer is in a non-full state (i.e., the transmit buffer full flag bit WRFULL bit is low). And when receiving data, when the receive buffer is in a non-empty state (i.e., the receive buffer empty flag bit RDEMPT bit is low), the characters already received can be read by accessing the SPDR register.

Host dual-line reception

The SPI module's two-wire mode is only valid in host operation mode and differs from the single-wire mode in that both MOSI and MISO are used to receive data from the host, with each SPCK clock pulse receiving 2 bits of data at the same time (data on the MISO line is in

After receiving two bytes of data, the hardware sets the transfer completion flag SPIF and saves the data in the receive buffer and shift register. At this point, the software must read the SPDR register twice to get the two bytes of data received. Note that although the host does not send data to the slave in dual-line mode, the software still needs to write data to the SPDR register to start the clock generator to generate the communication clock, and write the SPDR register once to receive two bytes of data.

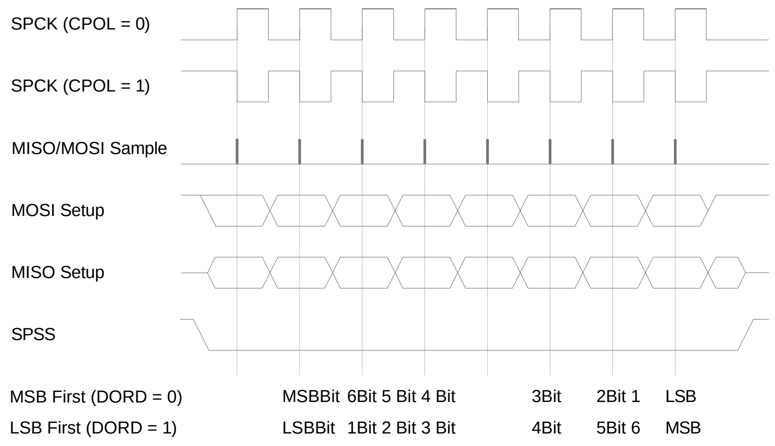
Data model

In single-wire mode, SPI has four combinations of SPCK phase and polarity relative to serial data, as defined by CPHA and CPOL
to control, as shown in the table below.

CPHA and CPOL Select Data Transfer Mode

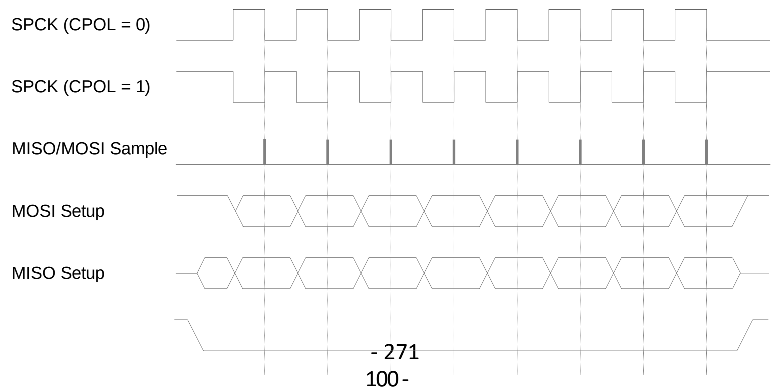
CPOL	CPHA	starting edge	end up along	SPI mode
0	0	Sampling (rising edge)	Setting (falling edge)	0
0	1	Setting (rising edge)	Sampling (falling edge)	1
1	0	Sampling (falling edge)	Setting (rising edge)	2
1	1	Setting (falling edge)	Sampling (rising edge)	3

When CPHA = 0, the data is sampled and set with the clock edge as shown below.



SPI data transmission mode when CPHA is "0"

When CPHA = 1, the data is sampled and set with the clock edge as shown below.

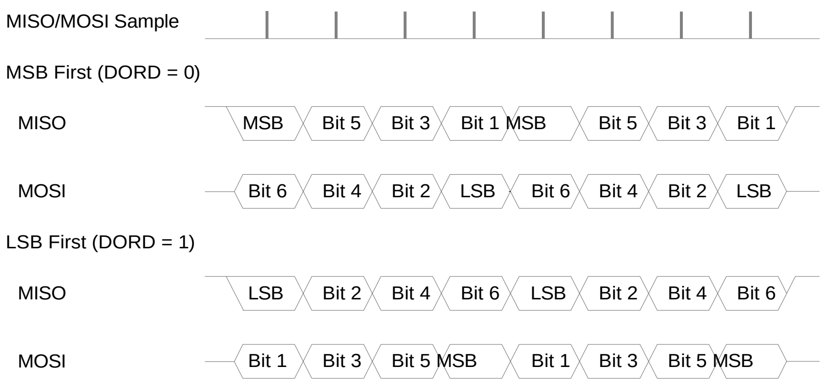


SPSS

MSB First (DORD = 0)	MSBBit	6Bit 5 Bit 4 Bit	3Bit	2Bit 1	LSB
LSB First (DORD = 1)	LSBBit	1Bit 2 Bit 3 Bit	4Bit	5Bit 6	MSB

SPI data transmission mode when CPHA is "1"

In the two-wire mode, both MISO and MISO are used as inputs to the host, and the moment of data sampling is still determined by the data transfer mode, which is sampled as shown in the following figure.



SPI data sampling mode when DUAL is "1" in host mode

SPSS Pin Function

When configured as a slave, the Slave Select Signal SPSS pin is always used as an input. When the SPSS pin is held low, the SPI interface is activated, the MISO pin becomes an output (if software configures the port accordingly), all other pins are inputs. When the SPSS pin is held high, the SPI module is reset and no more data is received.The SPSS pin is useful for packet/byte synchronization, synchronizing the slave's bit counter with the host's clock generator. When SPSS is pulled high, the SPI slave immediately resets the receive and transmit logic and discards incomplete data in the shift register.

When configured as a host, user software can determine the orientation of the SPSS pins.
If SPSS is configured as an output, it can be used to drive the slave's SPSS pins. If SPSS is configured as an input, it must be held high for proper operation of the master. When configured as a master and the SPSS pin is an input,and an external circuit pulls the SPSS pin low,the SPI module assumes that another master has selected itself as a slave and begins transferring data.To prevent bus conflicts, the SPI module will perform the following actions.

1. clear the MSTR bit located in the SPCR register to convert to slave, so that MOSI and SPCK become inputs.
2. Set the SPIF bit located in the SPSR register to generate an SPI interrupt if the interrupt is enabled.

Therefore, when using interrupts to handle data transfers from the SPI host and there is a possibility that the SPSS is pulled low, the interrupt service program should check that the MSTR bit is "1".If it is cleared, the software must set it to re-enable SPI host mode.

SPI Initialization

The SPI must first be initialized before communication can take place. The initialization process usually includes the selection of the host-slave operation, the setting of the data transfer mode, the selection of the bit rate, and the control of the direction of each pin. The control of the pin direction varies between host and slave operation, as shown in the following table.

Pin Orientation Control

pins	Orientation in host mode	Direction in slave mode
MOSI	user software definition	importation
MISO	importation	user software definition

SPCK	user software definition	importation
SPSS	user software definition	importation

SPI host initialization

The SPI host mode is initialized as follows.

1. (a) Set the MSTR bit to set the bit rate selection control bit, data transfer mode, data transfer order, interrupt enable or not, and dual line enable or not.
2. Setting the MOSI and SPCK pins as outputs.
3. Set the SPE bit.

TeSPSS pin can be set as an output in host mode when you do not want the SPI module to be selected for use as a slave by another host.

SPI slave initialization

The SPI slave mode initialization process is as follows.

1. Clear the MSTR bit, set the data transfer mode, data transfer order, interrupt enable or not.
2. Setting the MISO pin as an output.
3. Set the SPE bit.

SPI interrupt

The SPI's interrupt flag bit SPIF will be set when one or more of the following events occur.

1. When configured as host and the SPSS pin is an input, an external circuit pulls the SPSS pin low.
2. When t h e transmit buffer status is full, the software continues to write data to the SPDR register.
3. When the receive buffer status is full.
4. When all data written to the send buffer has been sent, the send buffer status is empty.

An SPI interrupt is generated when the SPIF bit is set and both the SPI interrupt enable bit SPIE and the global interrupt enable bit are high. Upon entering the interrupt service routine, the hardware will clear SPIF. If the SPIF bit is set by events 1 and 2 above, SPIF will be cleared; if the SPIF bit is set by events 3 and 4 above, SPIF will not be cleared because the SPIF bit will still be set if the receive or transmit buffer state has not changed, and will need to be cleared by software operation.

The sequence of operations for software clearing of the SPIF bit in the SPI interrupt service program is as follows.

1） Read the status of the SPIF bit, if it is low, it means that the SPIF bit has been cleared by hardware and does not need to be cleared again by software; if it is high, continue with the next operation.
2） Read the SPFR register, if the RDFULL bit is high, it indicates that the current receive buffer status is full, read the SPDR register to get the receive data, the RDFULL bit will become low and the software can continue to read the SPDR register to get the receive data until the RDEMPT bit is high.
3） Read the SPFR register, if the RDFULL bit is low and the WREMPT bit is high, indicating that the current receive buffer status is non-full and the transmit buffer status is empty, the software can read the SPDR register to obtain the receive data until the RDEMPT bit is high.
4） The software acquires the received data and then performs the clearing of the SPIF bit. Since the SPIF bit is a read-only bit, the SPIF bit cannot be cleared directly, but the SPIF bit needs to be cleared by reading the SPSR register first and then accessing the SPDR (reading or writing the SPDR register).

Register Definition

SPI Register List

processor register	address	default value	description
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	description
7	SPIE	SPI interrupt enable bit. The SPI interrupt is enabled when the SPIE bit is set to "1". When the SPIF in the SPSR register is set to "1", the SPI interrupt is enabled. When the bit is set and the global interrupt is enabled, an SPI interrupt is generated. When the SPIE bit is set to "0", SPI interrupts are disabled.
6	SPE	SPI enable bit. The SPI module is enabled when the SPE bit is set to "1". SPE must be set before any SPI operation can be performed. When the SPE bit is set to "0", the SPI module is disabled.
5	DORD	Data order control bit. When the DORD bit is set to "1", the LSB of the data is sent first. When the DORD bit is set to "0", the MSB of the data is sent first.
4	MSTR	The host slave selects the control bit. When the MSTR bit is set to "1", the master mode is selected. When the MSTR bit is set to "0", the slave mode is selected. When the SPSS pin is configured as an input and pulled low in host mode, the MSTR bit will be cleared and the bit The SPIF in the SPSR register is set 1. User must reset the MSTR to enter host mode.
3	CPOL	Clock polarity control bit. When the CPOL bit is set to "1", the SPCK is high in the idle state. When the CPOL bit is set to "0", the SPCK is low in the idle state.

		CPOL	starting edge	end up along
		0	upside down	edge of drop
		1	edge of drop	upside down
2	CPHA	Clock phase control bits. When the CPHA bit is set to "1", the data is set at the start edge and sampled at the end edge. When the CPHA b i t i s set to "0", the data is sampled at the start edge and the data is set at the end edge.		
		CPHA	starting edge	end up along

		0	sample	set up
		1	set up	sampl e
1	SPR1	Clock rate select bit 1. SPR1 and SPR0 are used to select the clock rate of the SPI transmission. See SPCK and Table of relationships for system clocks.		
0	SPR0	Clock rate select bit 0. SPR1 and SPR0 are used to select the clock rate for SPI transfers. See the table for the relationship between SPCK and the system clock for details on how to control this.		

SPSR - SPI Status Register

SPSR - SPI Status Register								
Address: 0x4D				Default value: 0x00				
Bit	7	6	5	4	3	2	1	0
Name	SPIF	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	description
7	SPIF	SPI interrupt flag bit. The SPIF flag is set at the end of the serial transfer, and SPIF is also set when the SPSS pin is configured as an input and pulled low in host mode.If both the SPIE bit and the global interrupt enable bit of the SPCR register are set at this time, the SPI interrupt is generated. The SPIF bit is automatically cleared after entering the interrupt service program Zero, or clear the SPIF bit by reading the SPSR register first and then accessing the SPDR register.
6	WCOL	Write conflict flag bit. Writing the SPDR register during a data transfer will set the WCOL bit. the WCOL bit can be cleared by reading the SPSR register before accessing the SPDR register.
5	-	Reserved.
4	-	Reserved.
3	-	Reserved.
2	DUAL	Two-wire mode control bit. When the DUAL bit is set to "1", the SPI two-wire transfer mode is enabled. When the DUAL bit is set to "0", SPI two-wire transfer mode is disabled. The two-wire transfer mode is only valid in SPI host mode, and both MISO and MOSI are used as host numbers Data is input, and the data is transmitted as described in the Host Dual Line Receive and Data Mode sections.

1	-	Reserved.
0	SPI2X	SPI multiplier control bit. When the SPI2X bit is set to "1", the SPI transmission speed is doubled. When the SPI2X bit is set to "0", the SPI transmission speed is not doubled. See the table for the relationship between SPCK and the system clock for details on how to control it.

The following table shows the relationship between the SPCK and the system clock.

Relationship between SPCK and system clock

SPI2X	SPR1	SPR0	Frequency of SPCK
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	description
7:0	SPDR	Data sent and received by SPI. SPI send data and receive data share the SPI data register SPDR. writing data to SPDR is writing to the send data shift register and reading data from SPDR is reading the receive data buffer.

SPFR - SPI buffer register

SPFR - SPI buffer register								
Address: 0x39					Default value: 0x00			
Bit	7	6	5	4	3	2	1	0
Name	RDFULl	RDEMPT	RDPTR1	RDPTR0	WRFULL	WREMPT	WRPTR1	WRPTR0
R/W	R	R/W	R	R	R	R/W	R	R

Bit	Name	description
7	RDFULL	Receive buffer full flag bit. When the data in the receive buffer reaches four bytes, the RDFULL bit is high, indicating that the receive buffer is full, and the interrupt flag bit will be set. If the software does not read away the data in the receive buffer in time, the receive buffer overflows when the data is received again, and the previous data is overwritten by the new data. When there is less than four bytes of data in the receive buffer, the RDFULL bit is low, indicating that the receive buffer is non-full and data can still be received. When the RDEMPT bit and WREMPT bit are set at the same time, the receive and transmit buffers The address and the SPI shift register pointer will be zeroed and the RDFULL bit will be low.

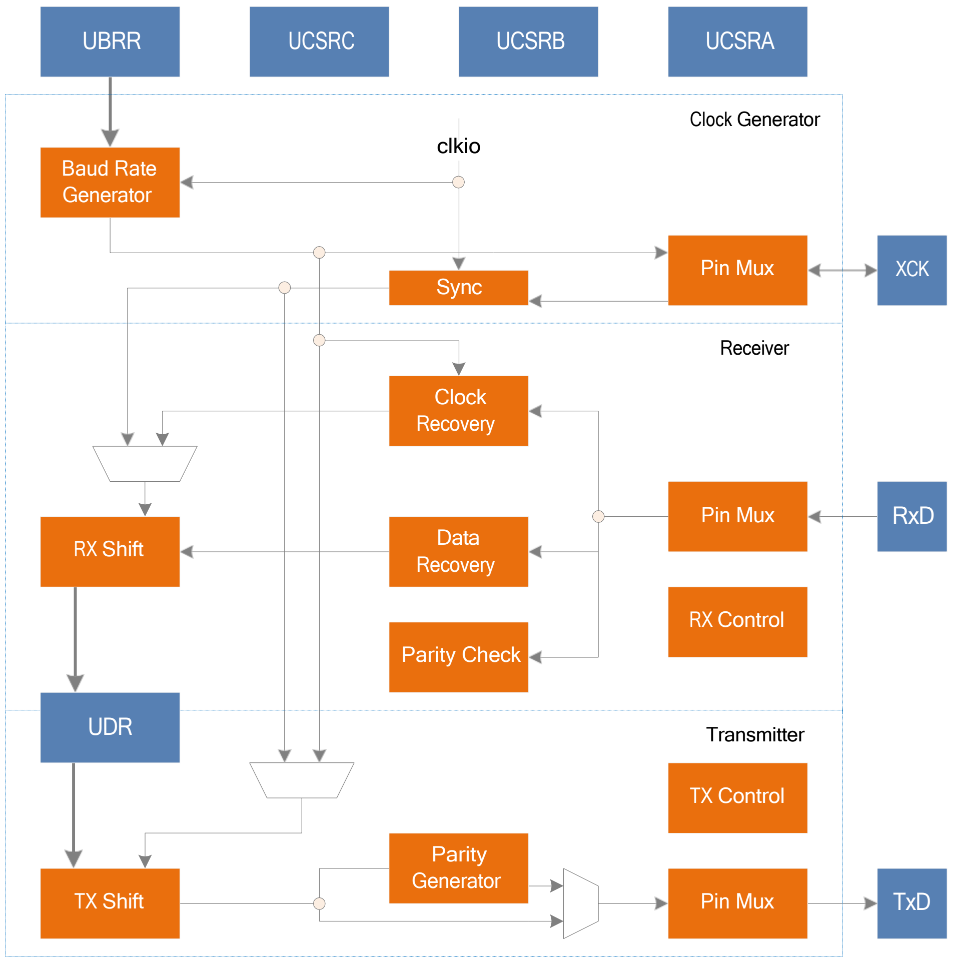
6	RDEMPT	Receive buffer empty flag bit. When no data is received, the RDEMPT bit is high, indicating that the receive buffer is empty. When there is received data, it will be stored in the receive buffer, and the RDEMPT bit is low, indicating that the receive buffer is non-empty, then the MCU can read the number in the receive buffer by accessing the SPDR register

		Data. To ensure that the received data is not lost, the software can read away the data in the receive buffer when the receive buffer is non-empty, i.e., the RDEMPT bit is low. When a set operation (write 1) is performed on the RDEMPT bit, the receive buffer address will go to zero. When the RDEMPT bit and WREMPT bit are set at the same time, the receive and transmit buffers The address, as well as the SPI shift register pointer, will be zeroed and the RDEMPT bit will be high.
5	RDPTR1	Receive buffer address high.
4	RDPTR0	Receive buffer address low bit. When a read operation is performed on the SPDR register, the MCU will read the received data from the receive buffer while the receive buffer address is accumulated. When a set operation (write 1) is performed on the RDEMPT bit, the receive buffer address will go to zero.
3	WRFULL	Send buffer full flag bit. When the data in the transmit buffer reaches four bytes, the WRFULL bit is high, indicating that the transmit buffer is full. When there is less than four bytes of data in the transmit buffer, the WRFULL bit is low,indicating that the transmit buffer is non-full. If you want to increase the transmission speed, the software can set the WRFULL bit low when the transmit buffer is non-full, i.e. The data is written when the WRFULL bit is low, and the SPI controller sends the data out in sequence.
2	WREMPT	Send buffer empty flag bit. When all data written to the transmit buffer has been sent, the WREMPT bit is high, indicating that the transmit buffer is empty, and the interrupt flag bit SPIF is set. When a write operation is performed to the SPDR register, the send buffer address is accumulated, the data written to the send buffer is not all sent, there is at least one byte of data in the receive buffer, and the WREMPT bit is low, indicating that the send buffer is non-empty. When a set operation (write 1) is performed on the WREMPT bit, the transmit buffer address is zeroed. When both the RDEMPT bit and WREMPT bit are set, the receive and transmit buffers The address, as well as the SPI shift register pointer, will be zeroed and the WREMPT bit will be high.
1	WRPTR1	Send buffer address high.
0	WRPTR0	Send buffer address low bit. When a write operation is performed to the SPDR register,the data in the SPDR will be written to the transmit buffer, and the transmit buffer address will be accumulated. When the WREMPT bit is set (write 1), the transmit buffer address will go to zero.

USART0 - Universal synchronous/asynchronous serial transceiver

- Full duplex operation (separate serial receive and transmit registers)
- Asynchronous or synchronous operation
- Host or slave operation
- High-precision baud rate generator
- Supports 5, 6, 7, 8, or 9 data bits and 1, or 2 stop bits
- Hardware-supported parity generation and parity check mechanism
- Data overspeed detection
- frame error detection
- Noise filtering, including false start bit detection and digital low-pass filter
- Three independent interrupts: end-of-send interrupt, end-of-send data register air interrupt, and end-of-receive interrupt
- Multi-processor communication mode
- Multiplier asynchronous communication mode

a general narrative



USART Structure Diagram

USART consists of three main parts: the clock generator, the transmitter and the receiver. The control and status registers are shared by these three parts. The clock generator consists of a baud rate generator and synchronization logic for the external input clock in synchronous slave operation mode. The XCK pin is used only in synchronous transmission mode. The transmitter consists of a write data buffer, serial shift registers, parity generators, and the control logic required to handle the different frame formats. The write data buffer allows data to be sent continuously without introducing delays between data frames. The receiver has a clock and data recovery unit for asynchronous data reception. In addition to the recovery unit, the receiver includes parity, control logic, serial shift registers and a two-stage receive buffer UDR. the receiver supports the same frame format as the transmitter and can detect frame errors, data overspeed and parity errors.

clock generation

The clock generation logic generates the base clock for the transmitter and receiver. the USART supports four modes of clocking: normal asynchronous mode, multiplied asynchronous mode, host synchronous mode, and slave synchronous mode. the UMSEL bit of the USCRC is used to select synchronous or asynchronous mode. the U2X bit of the USCRA controls the multiplied enable in asynchronous mode. The data direction register of the XCK pin (multiplexed with IO), which is valid only in synchronous mode, determines whether the clock source is generated internally (master mode) or externally (slave mode).

Baud Rate Generator

The baud rate register UBRR and the descending counter are connected together as a programmable prescaler or baud rate generator for the USART. The descending counter operates at the system clock (fsys) and automatically loads the value of the UBRR register when it counts to zero or when the UBRRL register is written. A clock is generated when the count reaches zero, and this clock is used as the output clock of the baud rate generator at fsys/(UBRR+1).

The following table gives the formulas for calculating the baud rate (bits/sec) and the UBRR values for the various operating modes.

working mode	Baud rate calculation formula (1)	UBRR value calculation formula
Asynchronous normal mode	BAUD = fsys/(16*(UBRR+1))	UBRR = fsys/(16*BAUD) - 1
asynchronous doubling mode	BAUD = fsys/(8*(UBRR+1))	UBRR = fsys/(8*BAUD) - 1
Synchronous host mode	BAUD = fsys/(2*(UBRR+1))	UBRR = fsys/(2*BAUD) - 1

Description.

1. Baud rate is defined as the rate of bit transmission per second (bps)
2. BUAD is the baud rate, fsys is the system clock, and UBRR is the combined value of the baud rate registers UBRRH and UBRRL.

Multiplier working mode

The transmission rate can be doubled by setting the U2X bit of the UCSRA register, which is only valid in asynchronous operation mode and is set to "0" in synchronous operation mode.
Setting this bit will cut the baud rate divider divider value in half, effectively doubling the transmission rate of asynchronous communications. In this case, the receiver uses only half the number of samples to sample the data and clock recovery, so a more precise baud rate setting and system clock are required. The

transmitter, on the other hand, remains unchanged.

external clock

The synchronous slave operation mode is driven by an external clock. The external clock passes through a synchronization register and an edge detector before being sent by the transmitter and receiver use, a process that introduces a delay between the two system clocks, so the maximum clock frequency of the external XCK is limited by the following equation.

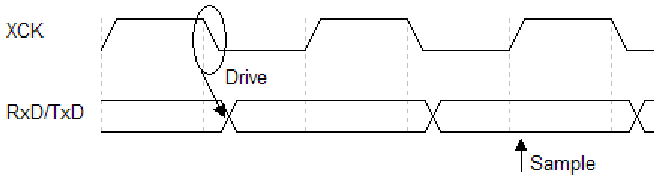
$$f_{XCK} < fsys/4$$

Be aware that fsys has the stability of the system clock to determine, and it is recommended to keep enough margin to prevent data loss due to frequency drift.

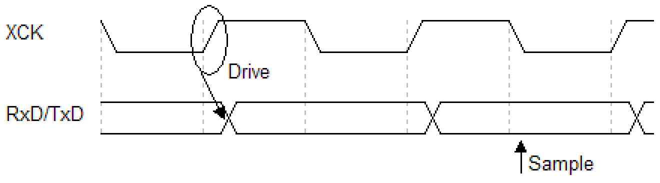
Synchronized clock operation

The XCK pin is used in synchronous mode for either the clock input (slave mode)or the clock output (master mode) The basic rule of clock edges in relation to data sampling and data changes is that the clock edge used to sample the data input (RxD) is opposite to the clock edge used to change the data output.

UCPOL = 1



UCPOL = 0



XCK Timing in Synchronous Mode

As shown above, when the UCPOL value is "1", the data output is changed on the falling edge of XCK and the data is sampled on the rising edge of XCK; when the UCPOL value is "0", the data output is changed on the rising edge of XCK and the data is sampled on the rising edge of XCK. When the UCPOL value is "0", the data output is changed on the rising edge of XCK and the data is sampled on the falling edge of XCK.

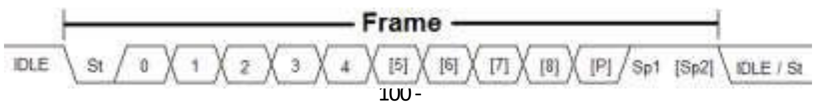
frame format

A serial data frame consists of a data word plus synchronization bits (start and stop bits) and parity bits for error correction.

USART accepts the following 30 combinations of data frame formats.

- 1 starting position
- 5, 6, 7, 8 or 9 data bits
- No parity bit, odd parity bit or even parity bit
- 1 or 2 stop bits

The data frame starts with the start bit, followed by the lowest bit of the data word, followed by the other data bits, and ends with the highest bit of the data word, with a maximum of 9 bits of data successfully transferred. If checksum is enabled, the checksum bit will follow the data word and finally the stop bit. When a complete data frame is transmitted, the next new data frame can be transmitted immediately, or the transmission line can be left idle (high). The diagram below shows a possible data frame structure, with the bits in square brackets being optional.



USART frame structure diagram

Description.

1) No data transmission on the IDLE communication line (RxD or TxD), line must be high when idle

2) St Start bit, always low

3) 0-8 Data bits

4) P Parity bit, odd or even parity

5) Sp Stop bit, always high

The structure of the data frame is set by UCSZ[2:0], UPM[1:0] and USBS in the UCSRB and UCSRC registers. The same settings are used for receive as for transmit. Any change in the settings may corrupt the ongoing data transmission. In particular, UCSZ[2:0] determines the number of data bits in the data frame, UPM[1:0] is used to enable and determine the type of checksum, and USBS sets the frame to have one or two end bits. The receiver ignores the second stop bit, so frame errors are only detected when the first end bit is "0".

Check digit calculation

The checksum bit is calculated by performing an iso-or operation on the individual bits of the data. If an odd checksum is selected, the result of the iso-or needs to be inverted as well. The relationship between the parity bits and the data bits is as follows.

P_{even} = dn−

1⊕. . .⊕d3⊕d2⊕d1⊕d0⊕0 P_{odd} =

dn−1⊕. ..⊕d3⊕d2⊕d1⊕d0⊕1

Description.

1) Peven even

check result 2) Podd odd

check result 3) dn nth

data bit

USART initialization

The USART must first be initialized before communication can take place. The initialization process typically includes setting the baud rate, setting the frame structure, and enabling the receiver or transmitter as needed. For interrupt-driven USART operation, the global interrupt flag is cleared and all interrupts to the USART are disabled during initialization.

When performing a reinitialization such as changing the baud rate or frame structure, you must ensure that no data is being transmitted.The TXC flag bit can be used to detect if the transmitter has completed all transmissions and the RXC flag bit can be used to detect if there is data left in the receive buffer that has not been read out. If the TXC flag bit is used for this purpose, the TXC flag bit must be cleared before each transmission of data (before writing the UDR register).

transmitter

Placing the TXEN bit of the UCSRB register will enable data transmission from the USART. When enabled, the general purpose IO function of the TxD pin is replaced by the USART function, which becomes the serial output of the transmitter. The baud rate, operating mode and frame format should be set before sending data. If the synchronous transmit mode is used, the clock signal applied to the XCK pin is the clock for data transmission.

Sending frames of 5 to 8 data

Data sending is initiated by loading the data to be sent into the send buffer.The CPU loads the data by writing to the UDR register. When the transmit shift register is ready to send a new frame of data, the data in the buffer is transferred to the shift register. When the shift register is idle (no data transfer in progress) or when the last stop bit of the previous frame of data has been sent, it will load new data. Once the shift register is loaded with new data, it will follow the established settings to transfer

Enter a complete frame.

Sending a frame with 9-bit data

If a frame with 9 bits of data is sent, the 9th bit of the data should be written to the TXB8 bit of register UCSRB first, and then the lower 8 bits of data should be written to transmit data register UDR. The 9th bit of data is used to represent address frames in multi-machine communication and can be used for protocol processing in synchronous communication.

Send parity bit

The parity generation circuit generates the appropriate parity bits for the serial data frame. When the parity bit is enabled (UPM1 = 1) the transmit control logic circuitry inserts a parity bit between the last bit of the data word and the first stop bit.

Send flag bits and interrupt handling

The USART transmitter has two flag bits: the USART data register empty flag UDRE and the end-of-transmission flag TXC, both of which can generate interrupts.
The data register empty flag UDRE is used to indicate whether the transmit buffer is ready to write a new data. This bit is set to "1" when the transmit buffer is empty and to "0" when it is full. When the UDRE bit is "1", the CPU can write new data to the data register UDR, but not vice versa.
When the data register air break enable bit UDRIE in the UCSRB register is "1", a USART data register air break request will be generated whenever UDRE is set (and the global interrupt is enabled)When transferring data by interrupt, a new data must be written to UDR in the data register air break service routine to clear UDRE or to disable the data register air break. Otherwise a new interrupt will be generated again once this interrupt service routine is finished.
When the entire data frame is shifted out of the transmit shift register and there is no new data in the transmit register, the end-of-send flag TXC will be set. When TXCIE, the end-of-send interrupt enable bit on UCSRB (and global interrupt enable), is set to "1", the USART end-of-send interrupt will be executed with the TXC flag bit set. Once the interrupt service program is entered, the TXC flag bit is automatically cleared, or the CPU can write a "1" to this bit to clear it.

Prohibition of transmitters

When TXEN is cleared to zero, the transmitter can only be truly disabled after all data has been sent, i.e., there is no data to be transmitted in the transmit shift register and the transmit buffer register. After the transmitter is disabled, the TxD pin resumes its general purpose IO function.

receivers

The USART receiver is enabled by setting the Receive Allow bit (RXEN) of the UCSRB register. When enabled, the general purpose IO function of the RxD pin is replaced by the USART function, which becomes the serial input port of the receiver. Before performing data reception, first set the baud rate, operation mode and frame format. If the synchronous reception mode is used, the clock on the XCK pin is used as the transmission clock.

Receive frames of *5* to *8* bits of data

Once the receiver detects a valid start bit, it will start receiving data. Each bit of data after the start bit will be received at the set baud rate or XCK clock until the first stop bit of a frame of data is received, and the second stop bit will be

The receiver ignores it. Each bit of data received is fed into the receive shift register, and after the first stop bit is received, the receiver sets the RXC bit of the receive data completion flag located in the UCSRA register and transfers the complete data frame in the shift register to the receive buffer, and the CPU can obtain the received data by reading the UDR register.

Receive frames with 9-bit data

If a data frame with 9 bits of data is set, the RXB8 bit of register UCSRB must first be read to obtain the 9th bit of data before the lower 8 bits of data can be read from the UDR. This rule also applies to the status flag bits FE, DOR, and PE. reading the UDR memory cell changes the state of the receive buffer, which in turn changes the TXB8, FE, DOR, and PE bits also stored in the buffer.

End of reception flag and interrupt handling

The USART receiver has a flag bit, the end-of-receive flag RXC, which indicates whether there is unread data in the receive buffer. This bit is "1" when there is unread data in the receive buffer, and "0" vice versa. If the receiver is disabled, the receive buffer will be flushed and RXC will be cleared.
After setting the UCSRB end-of-receive interrupt enable bit RXCIE, the USART end-of-receive interrupt is generated as soon as the RXC flag is set (and the global interrupt is enabled) When using the interrupt method for data reception, the data receive end interrupt service program must read data from the UDR to clear the RXC flag, otherwise a new interrupt will be generated as soon as the interrupt handler is finished.

Receive error flag

The USART receiver has three error flags: frame error FE, data overflow DOR, and parity error PE, all of which are located in the UCSRA register. The error flags are stored in the receive buffer along with the data frame. None of the error flags can generate an interrupt.
The frame error flag FE indicates the status of the first stop bit of the next readable frame stored in the receive buffer. If the stop bit is correct (value "1") the FE flag is "0", otherwise the FE flag is "1". This flag can be used to detect loss of synchronization, transmission interruptions, and also for protocol processing.
The data overflow flag, DOR, indicates that data has been lost due to a full receive buffer. When the receive buffer is full and data is already in the receive shift register, a data overflow is generated if a new start bit is detected at this time.The DOR flag being set indicates that one or more data frames were lost between the last read of the UDR and the next read of the UDR. The DOR flag is cleared when the data frame has been successfully transferred from the shift register into the receive buffer.
Parity error flag PE indicates that the next frame of data in the receive buffer is received with a parity error. If parity is not enabled, PE is cleared to zero.

parity checker

Setting the parity mode bit UPM1 will start the parity checker. The mode of parity (even or odd) is determined by UPM0. When parity is enabled, the checker will calculate the parity of the input data and compare the result with the parity bits of the data frame. The result is stored in the receive buffer along with the data and stop bits, and the CPU checks for parity errors in the received frame by reading the PE bits. If the next data read from the receive buffer has a parity error and parity is enabled, UPE is set and remains in effect until the receive buffer UDR is read.

Prohibition of receivers

In contrast to the transmitter, disabling the receiver works immediately. The data being received will be lost. After disabling the receiver (RXEN is cleared), the receiver will no longer occupy the RxD pin and the receive buffer will be flushed.

Asynchronous data reception

The USART has a clock recovery unit and a data recovery unit to handle asynchronous data reception. The clock recovery logic is used to synchronize the asynchronous serial data input from the RxD pin with the internal baud rate clock. The data recovery logic is used to capture the data and filter each bit of the input data through a low-pass filter, thereby improving the receiver's immunity to interference. The operating range of asynchronous reception is dependent on the accuracy of the internal baud rate clock, the rate of the frame input, and the number of data bits contained in a frame.

Asynchronous working range

The operating range of the receiver is dependent on the degree of mismatch between the received data rate and the internal baud rate. If the transmitter transmits data at too fast or too slow a bit rate, or if the receiver does not have the same internally generated baud rate, then the receiver will not be able to synchronize with the start bit. To ensure that the receiver does not miss sampling the start bit of the next frame, the data input rate and the internal receiver baud rate must not differ too much, and the ratio between them is used to describe the error range of the baud rate. The following two tables give the maximum baud rate error range allowed in normal mode and in multiplier mode, respectively.

Maximum receiver baud rate error range in normal mode

Data bits + parity bit length and	Maximum error range (%)	Recommended error range (%)
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

Maximum receiver baud rate error range in multiplier mode

Data bits + parity bit length and	Maximum error range (%)	Recommended error range (%)
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

As can be seen from the table, a wider range of variation in baud rate is allowed in normal mode. The above recommended baud rate error ranges are derived assuming that the receiver and transmitter contribute equally to the maximum total error. There are two possible reasons for the receiver baud rate error. First, the stability of the receiver system clock is related to the operating voltage and temperature. This is generally not a problem when using a crystal to generate the system clock, but when using an internal oscillator, the system clock may be off. The second reason is that the baud rate generator may not always be able to get exactly the desired baud rate by dividing the system clock. In this case, the UBRR value can be adjusted to make the error low enough to be acceptable.

Baud rate setting and introduction error

For standard crystal and resonator frequencies, the actual baud rate of communication in asynchronous mode can be obtained by the baud rate calculation formula, and the error between it and the commonly used communication baud rate can be calculated by the following formula.

Error[%] = (Baudreal/Baud - 1)*100%

Baud is the common communication baud rate, Baudreal is the baud rate calculated by the formula, and the baud rate error is related to the system clock fsys and the baud rate register UBRR value as follows.

Normal mode.

Error[%] = (fsys/(16*(UBRR+1))/Baud - 1)*100%

Multiplier mode.

Error[%] = (fsys/(8*(UBRR+1))/Baud - 1)*100%

The baud rate error UBRR is obtained when the clock errors on both sides of the communication are not considered, i.e., the system clock fsys is the standard clock

The relationship between the values. The following table shows the baud rate error for different UBRR value settings at 16MHz system clock.

Error in setting UBRR value at 16MHz system clock

baud rate (bps)	fsys = 16.000MHz			
	Normal mode (U2X = 0)		Multiplier mode (U2X = 1)	
	UBRR	inaccuracies	UBRR	inaccuracies
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%

Multi-processor communication mode

Placing the Multi-Processor Communication Mode (MPCM) bit of the UCSRA allows filtering of data frames received by the USART receiver. Those frames without address information will be ignored and will not be deposited into the receive buffer. In a multiprocessor system, where the processors communicate over the same serial bus, this filtering effectively reduces the number of data frames that need to be processed by the CPU. The setting of the MPCM bit does not affect the operation of the transmitter, but its use will vary in systems with multiprocessor communication.

If the receiver receives a data frame of 5 to 8 bits in length, the first stop bit is used to indicate the current frame

contains
Whether it is data or address information. If the length of the data frame received by the receiver is 9 bits, then the 9th bit determines whether it is data or address information. If the frame type flag bit is "1", then it is an address frame, otherwise it is a data frame.

In multi-processor communication mode, multiple slave processors are allowed to receive data from a master processor. It is first determined which slave processor is being addressed by decoding the address frames. The addressed slave processor will receive subsequent data normally, while the other slave processors will ignore the data frames until the next address frame is received.

For a processor acting as a host, it can use the 9-bit data frame format and identify the frame with the ninth bit of data format. In this communication mode, the slave processor must also operate in
the 9-bit data frame format. The following are the steps for data exchange in
multiprocessor communication mode.

1. All slave processors operate in multiprocessor communication mode (set MPCM)
2. The master processor sends an address frame, which is received by all slave processors. The RXC bit of the slave processor's UCSRA register is normally set.
3. Each slave processor reads the contents of the UDR register and decodes the address frame to determine if it is selected. If selected, the MPCM bit of the UCSRA register is cleared, and if not selected, the MPCM is held at "1" and the next address frame is awaited.
4. Addressed slave processors receive all data frames until a new address frame is received. Slave processors that are not addressed ignore these data frames.
5. The slave processor being addressed receives the last data frame, sets the MPCM bit, and waits for the next address frame to arrive. The process is then repeated from the second step.

Using a frame format with 5 to 8 bits of data is possible, but impractical because the receiver must switch between using n and n+1 frame formats. Since the receiver and transmitter use the same character length setting, this setup makes full duplex operation difficult. If a frame format of 5 to 8 bits of data is used, the transmitter should set two stop bits, the first of which is used to determine 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	description
7	RXC	Receive end flag bit. When the value of RXC is "1", it indicates that there is unread data in the receive buffer. When the value of RXC is "0", it indicates that there is no unread data in the receive buffer. When the receiver is disabled, the receive buffer is flushed, causing RXC to be cleared to zero. RXC can be used to generate an end-of-receive interrupt when the end-of-receive interrupt enable bit RXCIE is "1".
6	TXC	Send end flag bit. TXC is set when the data in the transmit shift register is sent out and the transmit buffer is empty. TXC is automatically cleared when the end-of-send interrupt is executedor can be cleared by writing a "1" to TXC. TXC can be used to

		generate an end-of-send interrupt when the end-of-send interrupt enable bit TXCIE is "1".

5	UDRE	Data register empty flag bit. When UDRE is "1", it indicates that the USART transmit data buffer is empty and data can be written. When UDRE is "0", it indicates that the USART transmit data buffer is full and no data can be written. When the data register air-break enable bit UDRIE is "1", UDRE can be used to generate data to send data. The depository is broken in the air.
4	FE	Frame error flag bit. When FE is "1", the data received by the receive data buffer has a framing error, i.e., the first stop bit is "0". When FE is "0", the data received by the receive data buffer has no frame error, i.e., the first stop bit is "1." FE is set and remains valid until the UDR is read. When writing to UCSRA, the FE bit is written "0".
3	DOR	Data overflow flag bit. When the receive buffer is full (contains two data) and data in the receive shift register, if a new start bit is detected at this time, a data overflow is generated and DOR is set and remains valid until UDR is read. When writing to UCSRA, the DOR bit is written "0".
2	PE	Parity error flag bit. When parity is enabled (UPM1 is "1") and the data frame received in the receive buffer has a parity error, PE is set and remains valid until the UDR is read. When writing to UCSRA, write "0" to this bit of PE.
1	U2X	Multiplier send enable bit. When U2X is "1", the transmission rate of the asynchronous communication mode is doubled. When U2X is "0", the transmission rate of the asynchronous communication mode is the normal rate. This bit is only valid in asynchronous operation mode; clear this bit to zero when using synchronous operation mode.
0	MPCM	Multiprocessor communication mode enable bit. Setting the MPCM bit will initiate multiprocessor communication mode. with MPCM set, those input frames received by the USART receiver that do not contain address information will be ignored. Transmitter is unaffected by the MPCM setting.

UCSRB - USART Control and Status Register B

UCSRB - 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	description
7	RXCIE	Receive end interrupt enable bit. The RXC interrupt is enabled when it is set, and disabled when it is cleared. When RXCIE is "1", the global interrupt is enabled, and when RXC of UCSRA register is "1", the USART receive end interrupt can be generated.

6	TXCIE	Send end interrupt enable bit. The TXC interrupt is enabled when set and disabled when cleared. When TXCIE is "1", the global interrupt Enable, when TXC of UCSRA register is "1", USART send end interrupt can be generated.
5	UDRIE	Data register air break enable bit. The UDRE interrupt is enabled when it is set and disabled when it is cleared. When UDRIE is "1", global interrupt is enabled and UDRE of UCSRA register is "1", USART data register empty can be generated.

		Interruption.
4	RXEN	Receive enable bit. The USART receiver is activated when set. the general purpose IO function of the RxD pin is replaced by USART receive. Disabling the receiver will flush the receive buffer and invalidate the FE, DOR and PE flags.
3	TXEN	Send enable bit. The general purpose IO function of the TxD pin is replaced by USART transmit. after TXEN is cleared to zero, USART transmit can only be truly disabled until all data has been sent.
2	UCSZ2	Character length control bit 2. UCSZ2 is combined with UCSZ1:0 of the UCSRC register to set the number of data bits contained in the data frame.
1	RXB8	Receive data bit 8. When the data frame length is 9 bits, RXB8 is the highest bit of the received data. RXB8 is read before the lower 8 bits of data contained in the UDR are read.
0	TXB8	Send data bit 8. When the data frame length is 9 bits, TXB8 is the highest bit of the transmitted data. TXB8 is written before the lower 8 bits of data contained in the UDR are written.

UCSRC– USART Control and Status Register C

UCSRC– 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

Bit	Name	description	
7:6	UMSEL1:0	USART mode select bit. UMSEL Selects synchronous or asynchronous operation mode.	
		UMSEL	mode
		0 1 2 3	USART Asynchronous operation mode USART Synchronous operation mode SPI Slave operation mode SPI Host operation mode
		Parity mode selection bit. The high bit UPM1 selects to enable or disable parity and low bit UPM0 selects parity or even parity.	
		UPM1:0	mode

5:4	UPM1:0	0 1 2 3	Prohibit parity reservation Enable Even Check enable odd-check
3	USBS	Stop bit selection bit. Selects the number of bits for the stop bit.	
		USBS	Number of Stop Bits
		0	1

		1	2
2:1	UCSZ1:0	Data frame character length selection bit. UCSZ1:0 combines with UCSZ2 of the UCSRB register to set the number of data bits contained in the data frame.	
		UCSZ2:0	Data frame length
		0	5 places
		1	6 places
		2	7 places
		3	8 bits
		4	retain
		5	retain
		6	retain
		7	9 places
0	UCPOL	Clock polarity selection bit. In USART synchronous operation mode, UCPOL sets the relationship between the change of output data and the sampling of input data and the synchronous clock XCK. Use the asynchronous operating mode with no relation to UCPOL, clear this bit to zero	

UCPOL	Send 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 low byte

UBRRL - USART baud rate register low byte								
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	description
7:0	UBRR[7:0]	The low byte portion of the USART baud rate register. The USART baud rate register contains two parts, UBRRL and UBRRH, which are combined to set the baud rate for communication.

UBRRH - USART baud rate register high byte

UBRRH - USART baud rate register high byte								
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	description
7:4	-	Reserved.

3:0	UBRR[11:8]	The high byte portion of the USART baud rate register. The USART baud rate register contains two parts, UBRRL and UBRRH, which are combined to set the baud rate for communication. UBRR = {UBRR[11:8], UBRRL}	
		working mode	Baud rate calculation formula
		Asynchronous normal mode	BAUD = fsys/(16*(UBRR+1))
		asynchronous doubling mode	BAUD = fsys/(8*(UBRR+1))
		Synchronous host 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

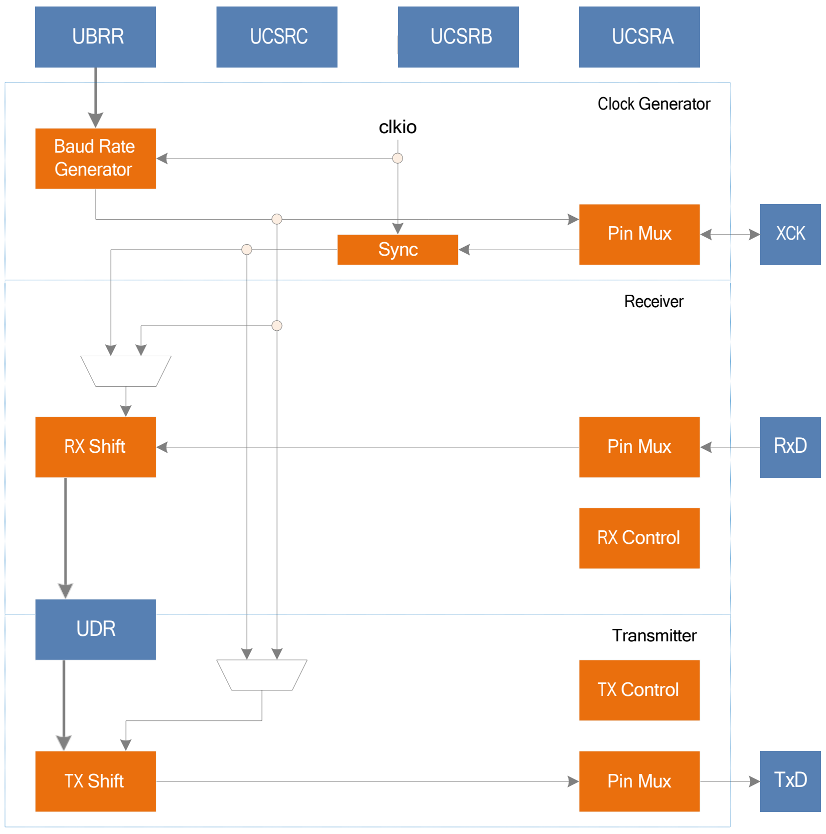
Bit	Name	description
7:0	UDR	The data sent and received by the USART. The USART send data buffer and receive data buffer share the USART data register UDR. writing data to UDR is writing to the send data buffer, and reading data from UDR is reading to the receive data buffer. In 5 to 8 bit data frame mode, unused bits 9 are ignored by the transmitter and the receiver sets them to 0. A write operation to the transmit buffer can only be performed when the UDRE flag of the UCSRA register is "1", otherwise the transmitter will operate with an error. When the transmit shift register is empty, the transmitter loads the data in the transmit buffer into the transmit shift register, and then the data is output serially from the TxD pin. The receive buffer contains a two-stage FIFO, and once the receive buffer is read, the FIFO is changed Change its state.

USART0 - SPI operating mode

- Full-duplex operation, three-wire synchronous data transmission
- Host or slave operation
- Supports all four operating modes (modes 0, 1, 2 and 3)
- Low or high bit transmitted first (configurable data transmission order)
- Queue operation (double buffer)
- High resolution baud rate generator

a general narrative

When the UMSEL1 bit of USCRC is set to "1", the SPI mode of operation is enabled and is represented by USPI. This SPI module is a three-wire SPI operating mode, missing the slave select line compared to the four-wire SPI mode, and the other three lines are identical. uspi occupies the resources of the USART, including the transmit and receive shift registers and buffers, and the baud rate generator. The parity generation and check logic, data and clock recovery logic are disabled. The addresses of the control and status registers are the same, although the function of the register bits changes as required by the SPI operating mode.



USART in SPI Structure Diagram

clock generation

When the SPI is operating in host mode, a clock for communication needs to be provided, and the USART's baud rate generator is used to generate this clock. This clock is output from the XCK pin, so the data direction register (DDR_XCK) on the XCK pin must be set to

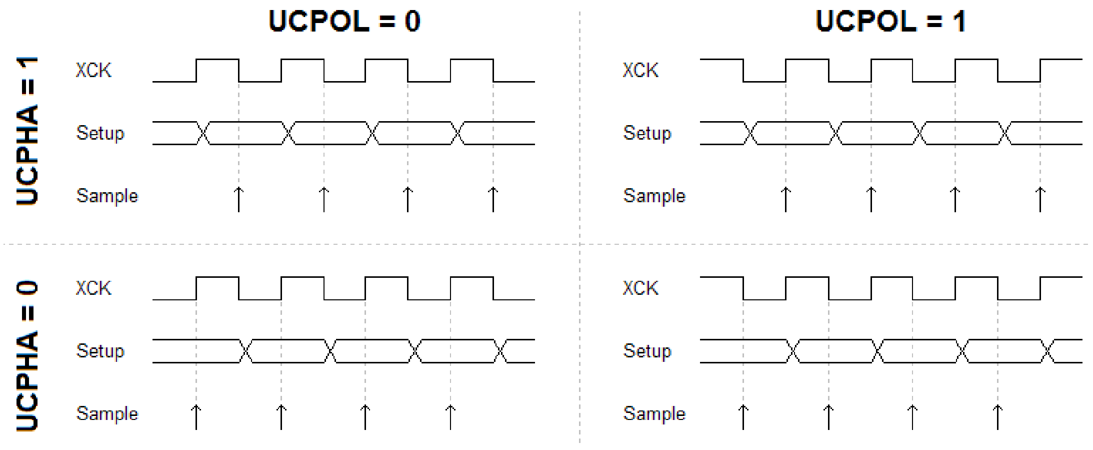
"1".
The clock frequency is determined by the following formula.
BAUD = fsys/(2*(UBRR+1))
When the SPI is operating in slave mode, the communication clock is provided by the external host and is input from the XCK pin, so the data direction register (DDR_XCK) on the XCK pin must be set to "0".

SPI Data Mode and Timing

The SPI has four combinations of clock phase and polarity, determined by the control bits UCPHA and UCPOL, as shown in the following table and figure below.

SPI Operating Modes

SPI mode	UCPOL	UCPHA	starting edge	end up along
0	0	0	rising edge sampling	Falling edge setting
1	0	1	Rising edge setting	sampling along the descent
2	1	0	sampling along the descent	Rising edge setting
3	1	1	Falling edge setting	rising edge sampling



SPI Operating Mode Diagram

frame format

A serial frame in SPI can start with the lowest or highest bit and end with the highest or lowest bit, for a total of 8 bits of data. A new frame can be transmitted immediately after the end of a frame, and the data line can be pulled up to idle at the end of the transmission.

data transmission

The SPI sets the TXEN bit of the UCSRB register to "1" to enable the transmitter, and the TxD pin is occupied by the transmitter to send serial output data. The receiver can be unenabled at this time.
The SPI sets the RXEN bit of the UCSRB register to "1" to enable the receiver, and the RxD pin is occupied by the receiver to receive serial input data. The transmitter must be enabled at this time.
Both SPI transmit and receive use XCK as the transmit clock.

The SPI must first be initialized before communication can take place. The initialization process typically includes setting the baud rate, setting the frame data bit transfer order, and enabling the receiver or transmitter as needed. For interrupt-driven SPI operation, the initialization

To clear the global interrupt flag and disable all interrupts for the SPI.
When performing a reinitialization such as changing the baud rate or frame structure, you must ensure that no data is being transmitted.The TXC flag bit can be used to detect if the transmitter has completed all transmissions and the RXC flag bit can be used to detect if there is data left in the receive buffer that has not been read out. If the TXC flag bit is used for this purpose, the TXC flag bit must be cleared before each transmission of data (before writing the UDR register).

After initializing the SPI, write data to the UDR register to start data transfer. Since the transmitter controls the transmission clock, both sending and receiving data operate as such. When the transmit shift register is ready to send a new frame of data, the transmitter moves the data written to the UDR register from the transmit buffer to the transmit shift register and sends it out. To keep the input buffer and transmit data synchronized, the UDR register must be read once after each byte of data is sent. When a data overflow occurs, the most recently received data will be lost, not the earliest received data.

Send Flag Bits and Interrupts

The SPI transmitter has two flag bits: the SPI data register empty flag UDRE and the end-of-transmission flag TXC, both of which can generate interrupts.

The data register empty flag UDRE is used to indicate whether the transmit buffer is ready to write a new data. This bit is set to "1" when the transmit buffer is empty and to "0" when it is full. When the UDRE bit is "1", the CPU can write new data to the data register UDR, but not vice versa.

When the data register empty interrupt enable bit UDRIE in the UCSRB register is "1", an SPI data register empty interrupt request will be generated whenever UDRE is set (and the global interrupt is enabled)Performing a write operation to register UDR will clear zero
UDRE. when transferring data by interrupt, a new data must be written to UDR in the data register air break service program to clear UDRE, or to disable the data register air break. Otherwise a new interrupt will be generated again once this interrupt service routine is finished.

When the entire data frame is shifted out of the transmit shift register and there is no new data in the transmit register, the end-of-send flag TXC will be set. When the end-of-send interrupt enable bit TXCIE (and global interrupt enable) on the UCSRB is set to "1", the SPI end-of-send interrupt will be executed with the TXC flag bit set. The TXC flag bit is automatically cleared once the interrupt service program is entered, or the CPU can write a "1" to this bit to clear it.

Prohibition of transmitters

When TXEN is cleared to zero, the transmitter can only be truly disabled after all data has been sent, i.e., there is no data to be transmitted in the transmit shift register and the transmit buffer register. After the transmitter is disabled, the TxD pin resumes its general purpose IO function.

End of reception flag and interrupt

The SPI receiver has a flag bit, the end-of-receive flag RXC, which indicates whether there is unread data in the receive buffer. This bit is "1" when there is unread data in the receive buffer, and "0" vice versa. If the receiver is disabled, the receive buffer will be flushed and RXC will be cleared. When the RXCIE end-of-receive interrupt enable bit of the UCSRB is set, the SPI end-of-receive interrupt is generated whenever the RXC flag is set (and the global interrupt is enabled)When using the interrupt method for data reception, the data receive end interrupt service program must read data from the UDR to clear the RXC flag, otherwise the RXC flag is cleared as long as

As soon as the interrupt handler is finished, a new interrupt is generated.

Prohibition of receivers

In contrast to the transmitter, disabling the receiver works immediately. The data being received will be lost. After disabling the receiver (RXEN cleared), the receiver will no longer occupy the RxD pin and the receive buffer will be flushed.

Register Definition

USART Register List

processor register	address	default value	description
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 byte
UBRRH	0xC5	0x0	USPI Baud Rate Register High Byte
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	description
7	RXC	Receive end flag bit. When the value of RXC is "1", it indicates that there is unread data in the receive buffer. When the value of RXC is "0", it indicates that there is no unread data in the receive buffer. When the receiver is disabled, the receive buffer is flushed, causing RXC to be cleared to zero. RXC can be used to generate an end-of-receive interrupt when the end-of-receive interrupt enable bit RXCIE is "1".
6	TXC	Send end flag bit. TXC is set when the data in the transmit shift register is sent out and the transmit buffer is empty. TXC is automatically cleared when the end-of-send interrupt is executed, or can be cleared by writing a "1" to TXC. TXC can be used to generate

		an end-of-send interrupt when the end-of-send interrupt enable bit TXCIE is "1".
5	UDRE	Data register empty flag bit. When UDRE is "1", the USPI transmit data buffer is empty and data can be written. When UDRE is "0", the USPI transmit data buffer is full and no data can be written. When the data register empty enable bit UDRIE is "1", UDRE can be used to generate the data register empty Interruption.
4:0	-	Reserved under USPI.

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	description
7	RXCIE	Receive end interrupt enable bit. The RXC interrupt is enabled when it is set and disabled when it is cleared. When RXCIE is "1", the global interrupt is enabled, and the RXC of UCSRA register is "1", the USPI receive end interrupt can be generated.
6	TXCIE	Send end interrupt enable bit. The TXC interrupt is enabled when it is set and disabled when it is cleared. When TXCIE is "1", the global interrupt is enabled, and when TXC of UCSRA register is "1", the USPI end-of-send interrupt can be generated.
5	UDRIE	Data register air break enable bit. The UDRE interrupt is enabled when it is set and disabled when it is cleared. When UDRIE is "1", the global interrupt is enabled and the UDRE of UCSRA register is "1", the USPI data register can be interrupted in the air.
4	RXEN	Receive enable bit. When set, the USPI receiver is activatedthegeneral purpose IO function of the RxD pin is replaced by USPI receive. Disabling the receiver will flush the receive buffer.
3	TXEN	Send enable bit. The general purpose IO function of the TxD pin is replaced by USPI transmit when set. After TXEN is cleared to zero, USART transmission can only be truly disabled until all data transmission is complete.
2:0	-	Reserved under USPI.

UCSRC– USART Control and Status Register C

UCSRC– USART Control and Status Register C								
Address: 0xC2					Default value: 0x86			
Bit	7	6	5	4	3	2	1	0
Name	UMSEL1	UMSEL0	-	-	-	DORD	UCPHA	UCPOL
R/W	R/W	R/W	-	-	-	R/W	R/W	R/W

Bit	Name	description	
		USART mode select bit. UMSEL Selects synchronous or asynchronous operation mode.	
		UMSEL	mode

7:6	UMSEL1:0	0 1 2 3	USART Asynchronous operation mode USART Synchronous operation mode SPI Slave operation mode SPI host operation mode
5:3	-	Reserved under USPI.	

2	DORD	Data transfer order selection bit.		
		DORD	data order	
		0	high level first transmission	
		1	low first transmission	
1	UCPHA	Clock phase selection. UCPHA Selects whether data sampling occurs at the start or end edge.		
		UCPHA	Sampling moment	
		0	starting edge	
		1	end up along	
0	UCPOL	Clock polarity selection. UCPOL selects whether the data change and sampling occurs on the rising or falling edge.		
		UCPOL	Change in sending data	Sampling of received data
		0	Rising edge of XCK	Falling edge of XCK
		1	Falling edge of XCK	Rising edge of XCK

UBRRL - USPI baud rate register low byte

UBRRL - USPI baud rate register low byte								
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	description
7:0	UBRR[7:0]	The low byte portion of the USPI baud rate register. The USPI baud rate register contains the UBRRL and UBRRH parts, combined to set the baud rate for communication.

UBRRH - USPI Baud Rate Register High Byte

UBRRH - USPI Baud Rate Register High Byte								
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	description
7:4	-	Reserved under USPI.

3:0	UBRR [11:8]	The high byte portion of the USPI baud rate register. The USPI baud rate register contains two parts, UBRRL and UBRRH, which are combined to set the baud rate for communication. UBRR = {UBRR[11:8], UBRRL}

		working mode	Baud rate calculation formula
		slave mode	Baud rate is determined by the external host
		Host 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	description
7:0	UDR	Data sent and received by USPI. The USPI send data buffer and the receive data buffer share the USPI data register UDR. writing data to UDR is writing to the send data buffer, and reading data from UDR is reading to the receive data buffer. In 5 to 8 bit data frame mode, unused bits 9 are ignored by the transmitter and the receiver sets them to 0. A write operation to the transmit buffer can only be performed when the UDRE flag of the UCSRA register is "1", otherwise the transmitter will operate with an error. When the transmit shift register is empty, the transmitter loads the data in the transmit buffer into the transmit shift register, and then the data is output serially from the TxD pin. The receive buffer contains a two-stage FIFO, and once the receive buffer is read, the FIFO is changed Change its state.

TWI - Two Wire Serial Bus *(I2C)*

- Simple yet powerful and flexible communication interface, requiring only 2 wires
- Supports host and slave operation
- The device can operate in either transmitter mode or receiver mode
- 7-bit address space allows 128 slaves
- Support for multi-host arbitration
- Up to 400Kbps data transfer rate
- Fully programmable slave addresses as well as public addresses
- Wakeup is possible when the address is matched in sleep mode

TWI Bus Introduction

The two-wire serial interface TWI is well suited for typical processor applications.The TWI protocol allows the system designer to interconnect 128 different devices together using only two bi-directional transmission lines. The two lines are the clock SCL and the data SDA. external hardware is only required to connect two pull-up resistors to each line. All devices connected to the bus have their own address. the TWI protocol solves the problem of bus arbitration.

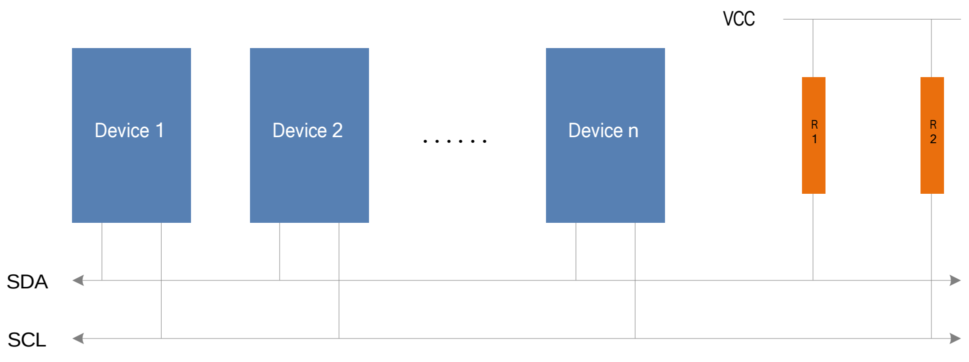
TWI Terminology

The terms defined below will appear frequently in this section.

terminologies	description
host computer	The device that starts and stops the transmission. The host is also responsible for generating the SCL clock.
aircraft from	Devices addressed by the host
transmitter	Devices that put data on the bus
receivers	Devices that receive data from the bus

Electrical connections

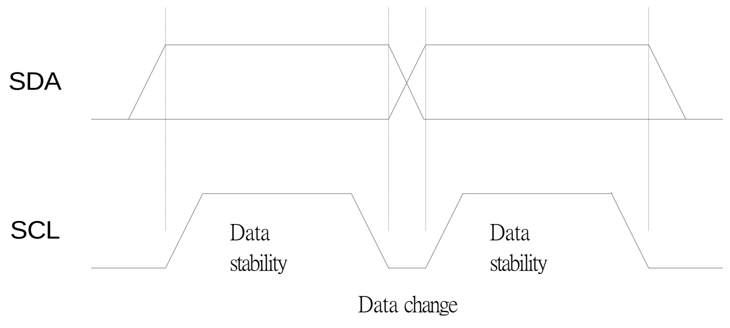
As shown in the figure below, both wires of the TWI interface are connected to the positive supply through pull-up resistors. All TWI-compatible devices are bus driven open-drain or open-collector, which enables line and function for interface operation. When the TWI device output is "0", the TWI bus goes low. When all TWI device outputs are tri-state, the bus allows the pull-up resistor to pull the voltage high. For all bus operation, all devices connected to the TWI bus must be powered up.



TWI Bus Interconnection Diagram

Data transmission and frame structure

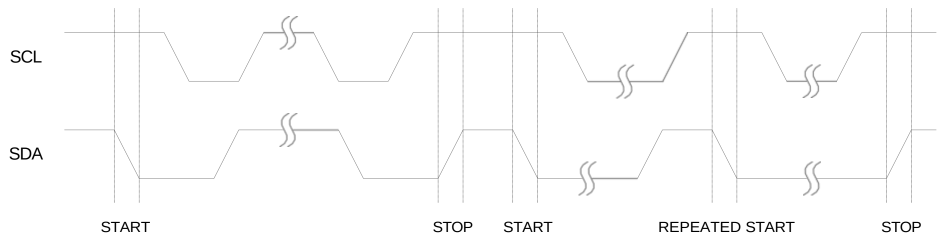
Each bit of data transmitted on the TWI bus is synchronized with the clock. When the clock line is high, the level on the data line must remain stable unless it is to produce a start or stop state.



TWI Data Validity Chart

Start and stop states

The TWI transmission is started and stopped by the host. The host issues a START state on the bus to send data transfers and a STOP state to stop data transfers. Between the START and STOP states, the bus is considered busy and no other host is allowed to attempt to take control of the bus. A special case is allowed to occur only when a new START state is generated between the START and STOP states; this is called the REPEATED START state and applies when the current host starts a new transmission without giving up control of the bus. after REPEATED START the bus is still considered busy until the next STOP.This is consistent with START,so throughout this document START is used to refer to both START and REPEATED START if not otherwise specified. as shown below, the START and STOP conditions change the level state of the SDA line when the SCL line is high.



START, REPEATED START a n d STOP Status Charts

address packet format

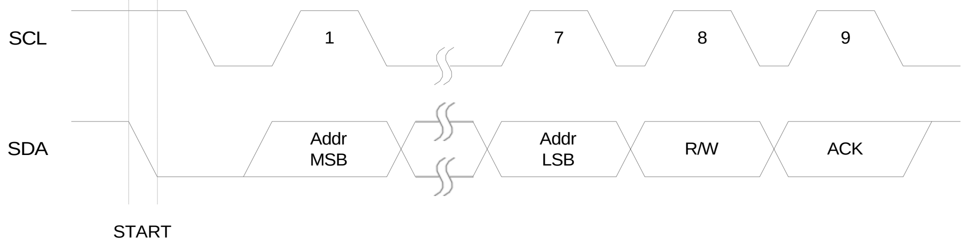
All address packets transmitted on the TWI bus are 9 bits long and consist of 7 bits of address, 1 READ/WRITE control bit, and 1 answer bit. When the READ/WRITE bit is "1", a read operation is performed; when the READ/WRITE bit is "0", a write operation is performed. After the slave is addressed, it must answer on the 9th SCL (ACK) cycle by pulling the SDA line low. If the slave is busy or otherwise unable to respond to the host, the SDA line shall be held high during the ACK cycle. The host may then issue a STOP status or REPEATED START status to restart transmission.

The address packet consists of a slave address and a read or write control bit, denoted by SLA+R or SLA+W, respectively.

The MSB bit of the address byte occurs first. Except for the reserved address "00000000" which is reserved for broadcast calls and all addresses shaped like
Other than the addresses in the "1111xxxx" format, which need to be reserved for future use, the other slave addresses can be freely assigned by the designer.

When a broadcast call occurs, all slaves shall answer by pulling down the SDA line during the ACK cycle. The broadcast function can be used when the master needs to send the same message to multiple slaves. After the broadcast call address plus the WRITE bit is sent to the bus, all slaves that need to respond to the broadcast call will pull down the SDA line during the ACK cycle. All slaves that respond to the broadcast call will receive the immediately following packet. Note that it does not make sense to send the broadcast call address plus the READ bit, because it will cause a bus conflict if several slaves send different data at the same time.

The address packet format is shown below.

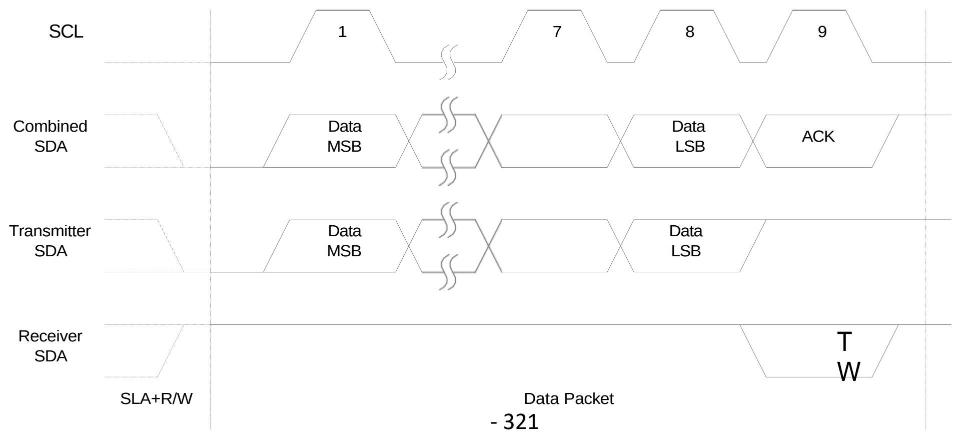


TWI Address Packet Format Diagram

Packet format

All packets transmitted on the TWI bus are 9-bit data length, consisting of 1 data byte and 1 answer bit. During data transmission, the host is responsible for generating the transmission clock SCL and START and STOP states, the transmitter sends the byte of data to be transmitted, and the receiver generates the receive response. The acknowledge signal ACK is generated by the receiver on the 9th SCL (ACK) cycle by pulling the SDA line low. If the receiver keeps the SDA line high during the ACK cycle, the unacknowledged signal NACK is sent.When the receiver has received the last byte, or for some reason cannot receive any more data, it should inform the transmitter by sending NACK after receiving the last byte. The MSB bit of the data byte is transmitted first.

The packet format is shown in the following figure.



T W I packet format diagram

ACK

N
e
x
t

D
a
t
a

P
a
c
k
e
t
,

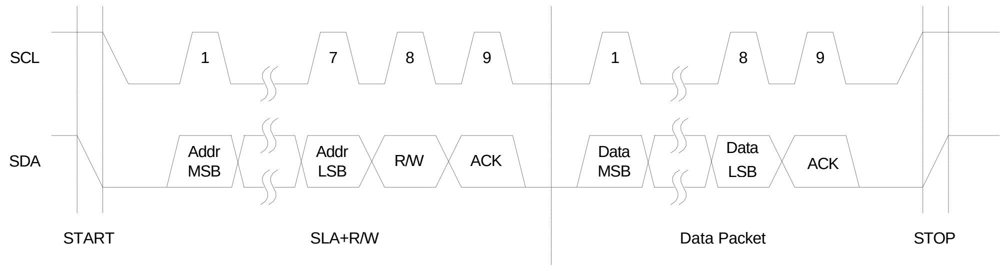
S
T
O
P

o
r

R
E
P
E
A
T
E
D
S
T
A
R
T

Combined address and packet transmission, one transmission basically consists of 1 START, 1 SLA+R/W, 1 or more packet and 1 STOP. Only null messages for START and STOP are illegal. The line and function of the SCL line can be used to implement a handshake between the master and the slave. The slave can extend the ground level period of SCL by pulling down the SCL line. This feature is useful when the master is set to clock much faster than the slave, or when the slave needs extra time to process data. The slave extending the low cycle of SCL does not affect the high cycle of SCL, which is still determined by the master. It follows that the slave can reduce the data transfer speed of the TWI by changing the duty cycle of SCL.

A typical data transfer is shown in the figure below. Note that multiple bytes can be transferred between SLA+R/W and STOP, depending on the application software implementation protocol.



Typical TWI Transmission

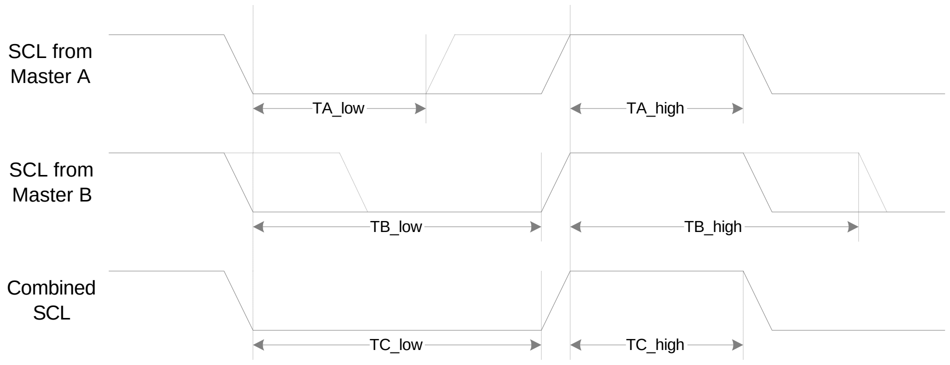
Multi-host systems and their arbitration and synchronization

The TWI protocol allows multiple hosts on the bus and employs special measures to ensure that even if two or more hosts initiate a transmission at the same time it can be handled as a normal transmission. Two problems arise with multi-host systems.

1. The implemented algorithm allows only one of the multiple hosts to complete the transmission. The other hosts must stop their transmissions when they discover that they have lost their selection rights. This process of selection is called arbitration. When a competing host finds that its arbitration has failed, it should immediately switch to slave mode to detect whether it is being addressed by the host that has gained control of the bus. In fact multiple hosts should not be detected by a slave when they start transmitting at the same time, i.e., they are not allowed to destroy the data being transmitted on the bus.
2. Different hosts may use different SCL frequencies. To ensure consistent transmission, a scheme to synchronize the host serial clocks must be devised. This will simplify the arbitration process.

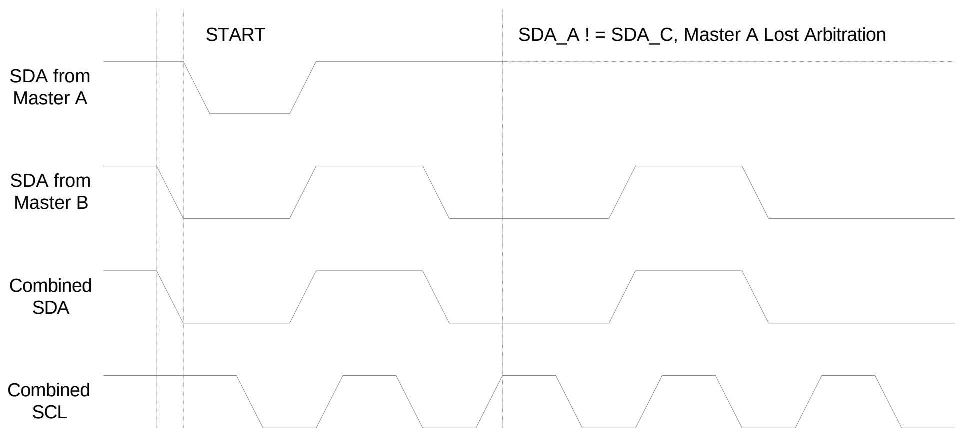
The line-and-speak function of the bus is used to solve the above problem. The serial clocks of all hosts are wired together to produce a combined clock whose high time is equal to the shortest of all the host clocks and whose low is equal to the longest of all the host clocks. All hosts listen to the SCLs, and when the combined SCL clock goes high or low, they can effectively start counting their respective SCL high and low overflow periods, respectively.

The SCL clock synchronization mechanism for multiple hosts is shown in the following figure.



Multi-Host SCL Clock Synchronization Timing Diagram

After outputting data all hosts continuously listen to the SDA line for arbitration. If the value read back from the SDA does not match the value output by the host, that host loses arbitration. Note that arbitration is lost when a host outputs a high SDA and another host outputs a low SDA. A host that loses arbitration should immediately switch to slave mode and test to see if it is addressed. A host that loses arbitration must set the SDA line high, but may still generate a clock signal until the current data or address packet ends. Arbitration will continue until there is only one master left in the system, which may take up multiple bits. If multiple hosts address the same slave, arbitration will continue until the packet.



Arbitration between two hosts

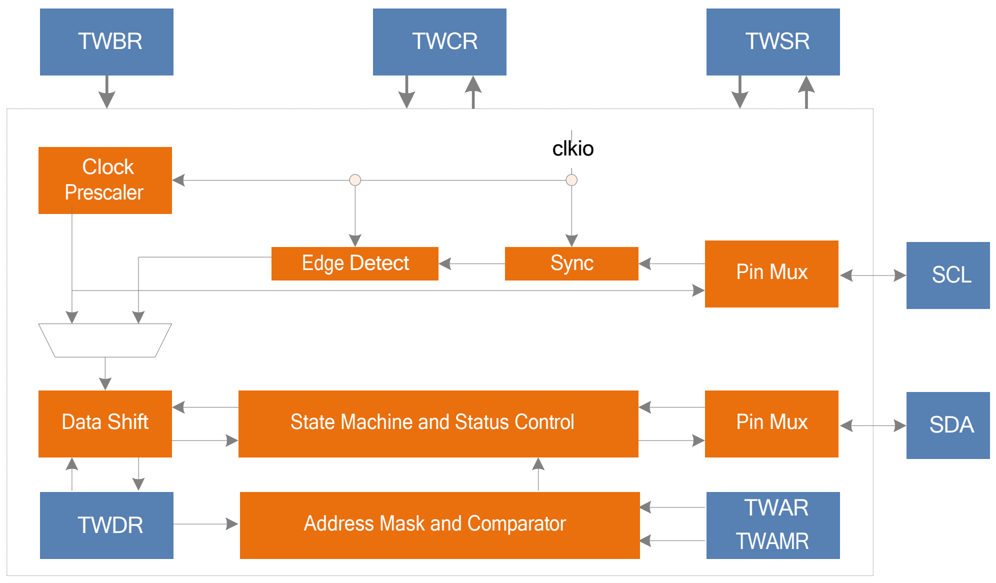
Note that arbitration is not permitted in the following circumstances.

- between a REPEATED START state and a data bit.
- between a STOP state and a data bit.
- between a REPEATED START state and a STOP state.

The application software must take the above into account to ensure that these illegal arbitration scenarios do not occur. This means that in a multi-host system, all data transmissions must consist of the same SLA+R/W with packets. In other words, all transmissions must contain the same number of packets, otherwise the arbitration result cannot be defined.

TWI Module Overview

The structure diagram of the TWI module is shown in the figure below.



TWI Block Structure Diagram

The TWI module consists mainly of a bit rate generator, bus interface unit, address comparator and control unit. See detailed description below.

Bitrate generator unit

The bit rate generator unit primarily controls the SCL clock period in master mode.The SCL clock period is determined by the prescaler control bits in both the TWI bit rate register, TWBR, and the TWI status register, TWSR. Slave operation is not affected by the bit rate or prescaler settings, but ensure that the slave's operating clock is at least 16 times the SCL frequency. Note that the slave may extend the low period of SCL, thereby reducing the average clock frequency of the TWI bus.The SCL clock frequency is generated with the following formula.

$f_{scl} = fsys/(16 + {}^{2*TWBR*4TWPS})$

where TWBR is the value of the TWI Bit Rate Register and TWPS is the prescaler control bit in the TWI Status Register.

Bus Interface Unit

The bus interface unit includes the data and address shift registers TWDR, START/STOP controller and arbitration determination hardware circuitry.

The TWDR contains the address or data byte to be sent, or the address or data byte that has been received. In addition to containing the 8-bit TWDR, the Bus Interface Unit also includes the ACK/NACK register for transmitting or receiving. This ACK/NACK register is not directly accessible by the application software. When data is received, it can be set or cleared by the TWI control register TWCR. When data is sent, the received ACK/NACK value is reflected by the TWS value in the TWI Status Register TWSR.

The START/STOP controller is responsible for generating and detecting the START, REPEATED START, and STOP states. When the MCU is in certain sleep modes, the START/STOP controller can still detect START and STOP states and wake up the MCU from sleep mode when addressed by the host on the TWI bus.

If the TWI initiates a data transfer in host mode, the arbitration detection circuitry will continuously listen to the bus to determine if it still has bus control. When the TWI module loses bus control, the control unit will perform the correct action and generate the appropriate status code to notify the MCU.

address matching unit

The Address Match Unit is used to check that the received address byte matches the 7-bit address in the TWI Address Register. When the TWI Broadcast Call Recognition Enable bit (TWGCE) in the TWAR register is set, the address received from the bus is also compared with the broadcast address. Once the address match is successful, the control unit will perform the correct action.The TWI module may or may not respond to host addressing, depending on the TWCR register setting. Even in sleep mode, the address matching unit can compare addresses and wake the MCU from sleep mode if it is addressed by a host on the bus.

control unit

The Control Unit is responsible for listening to the bus and generating the appropriate response based on the TWCR setting. The TWI interrupt flag bit TWINT will be set when an event occurs on the TWI bus that requires application software participation. During the next clock cycle, the TWI status register TWSR will be updated with a status code indicating the event. While TWINT is set, TWSR contains the exact status information. At other times, TWSR is a special status code indicating that no exact status information is available. Once the TWINT flag bit is set, the SCL line remains low, suspending TWI transmission on the bus and allowing the application software to process the event.

The TWINT flag bit will be set in the following cases.

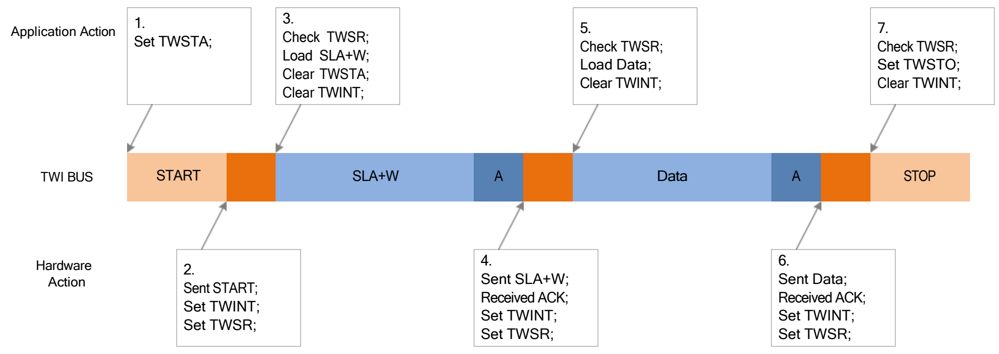
- After the TWI transmits the START/REPEATED START status
- After TWI transmits SLA+R/W
- TWI After transmitting an address byte
- After TWI bus arbitration failure
- TWI after being addressed by the host (slave address matching or broadcast method)
- After receiving STOP or REPEATED START when being addressed as a slave operation
- When a bus error is caused by an illegal START or STOP state

Use of *TWI*

The TWI interface is byte-oriented and interrupt-based.All bus events,such as a byte being received or a START signal being sent,will generate a TWI interrupt.Since TWI is interrupt-based, the application software is free to perform other operations during the TWI byte transfer.The TWI interrupt enable bit TWIE in the TWCR register controls, together with the global interrupt enable bit, whether a TWI interrupt is generated when the TWINT flag is in position. If the TWIE bit is cleared, the application software must detect the action on the TWI bus by querying the TWINT flag bit.

When the TWINT flag bit is set, it indicates that the TWI interface has completed the current operation and is waiting for a response from the application software. In this case, the TWI status register TWSR contains a status code that reflects the current bus status. The TWCR and TWDR registers can be set by the application software to determine how the TWI interface should operate during the next TWI bus cycle.

The following figure gives an example of an application connected to the TWI interface. In this example, the host expects to send one byte of data to the slave. The description here is simple and will be shown in more detail in the next sections.


Application Action
1. Set TWSTA;
3. Check TWSR; Load SLA+W; Clear TWSTA; Clear TWINT;
5. Check TWSR; Load Data; Clear TWINT;
7. Check TWSR; Set TWSTO; Clear TWINT;
TWI BUS
START
SLA+W
A
Data
A
STOP
Hardware Action
2. Sent START; Set TWINT; Set TWSR;
4. Sent SLA+W; Received ACK; Set TWINT; Set TWSR;
6. Sent Data; Received ACK; Set TWINT; Set TWSR;

TWI Typical Transmission Process Diagram

The TWI transfer process shown in the figure is

1. The first step in the TWI transmission is to send a START. the TWI hardware is instructed to send a START signal by writing a specific value to the TWCR register. The value to be written is described in detail later. It is important to set TWINT in the value written, as writing a "1" to the TWINT bit will clear the bit.the TWI will not initiate any operation while TWINT is set in the TWCR register. As soon as the TWINT bit is cleared by software, the TWI module initiates the transmission of the START signal.
2. When the START status is sent, the TWINT flag bit of the TWCR is set and the TWSR is updated to a new status code, indicating that the START signal was successfully sent.
3. The application looks at the value of the TWSR to determine that the START status has been successfully sent. If the TWSR shows another value, the application can perform some special operations, such as calling an error handler. After determining that the status code is as expected, the program loads the value of SLA+W into the TWDR register, which can be used in both address and data. The software then writes a specific value to the TWCR register, instructing the TWI hardware to send the value of SLA+W in the TWDR. The written values are described in detail later. The TWINT flag bit is cleared by setting TWINT in the written value. the TWI will not initiate any operation while TWINT is set in the TWCR register. As soon as the TWINT bit is cleared by software, the TWI module initiates transmission of the address packet.
4. When the address packet is sent, the TWINT flag bit of the TWCR is set and the TWSR is updated to a new status code indicating that the address packet was successfully sent. The status code will also reflect whether the slave responded to the address packet.
5. The application looks at the value of TWSR to determine that the address packet was successfully sent and that the ACK received was the desired value. If TWSR shows another value, the application can perform some special action, such as invoking an error handler. When it is determined that the status code is as expected, the program loads the value of Data into the TWDR register. The software then writes a specific value to the TWCR register, instructing the TWI hardware to send the value of Data in the TWDR. The values written are described in detail later. The TWINT flag bit is cleared by setting the TWINT bit in the written value, and the TWI will not initiate any operation while the TWINT bit in the TWCR register is set. As soon as the TWINT bit is cleared by software, the TWI module initiates packet transmission.
6. When the packet is sent, the TWINT flag bit of the TWCR is set and the TWSR is updated to a new status code indicating that the packet was successfully sent. The status code will also reflect whether the slave responded to the packet.
7. The application looks at the value of TWSR to determine that the packet was successfully sent and that the ACK received was the desired value. If the TWSR shows another value, the application can perform some special action, such as invoking an error handler. When it is determined that the status code is as expected, the software writes a specific value to the TWCR register, instructing the TWI hardware to send a STOP signal. The values written are described in detail later. The TWINT flag bit

is cleared by setting TWINT in the written value. the TWI will not initiate any operation while TWINT is set in the TWCR register. As soon as the TWINT bit is cleared by software, the TWI module initiates the transmission of the STOP signal. Note that TWINT will not be set after the STOP signal is sent.

Although the example is relatively simple, it contains all the rules for the TWI data transfer process. It is summarized as follows.

- The TWINT flag is set when the TWI completes an operation and waits for feedback from the application.The SCL clock line is always

Pull down until TWINT is cleared.

- When the TWINT flag is set, the user must update the values of all TWI registers to the values associated with the next TWI bus cycle. For example, the TWDR register must be loaded with the value to be sent for the next bus cycle.
- After all registers have been updated and other necessary operations have been completed, the application writes the TWCR register. When the TWCR is written, the TWINT bit must be set to clear the TWINT flag.After TWINT is cleared, the TWI begins to perform the operation set by the TWCR.

transfer mode

TWI can operate in the following 4 main modes: Master Transmitter (MT) Master Receiver (MR) Slave Transmitter (ST) and Slave Receiver (SR) Multiple modes can be used in the same application. For example, the TWI may use MT mode to write data to the TWI EEPROM and MR mode to read data from the EEPROM. If there are other hosts on the system, some of which may also send data to the TWI, the SR mode will be used. It is up to the application software to decide which mode to use.

These modes will be described in detail below. In each mode, the data transmission is combined with pictures to describe the possible status codes. These pictures contain the following abbreviations.

S. Start Status
Rs. REPEATED START Status
R: Read operation flag bit (SDA is high)
W: Write operation flag bit (SDA is low)
A: Answer bit (SDA is low)
NA: No answer bit (SDA is high)
Data: 8-bit data byte
P. STOP status
SLA: Slave address

The circles in the image are used to indicate that the TWINT flag is set and the numbers in the circles indicate the status codes in the TWSR registers, where the prescaler control bits are masked to "0". The TWI transfer will be suspended until the TWINT flag bit is cleared.

When the TWINT flag is set, the status codes in the TWSR are used to determine the appropriate software operation. The tables give details of the required software operation and subsequent serial transfers at each status code. Note that the prescaler control bit in the TWSR is masked to "0" in the tables.

Host send mode

In the host transmit mode, the TWI sends a certain number of data bytes to the slave receiver. In order to enter master mode, the START signal must be sent. The format of the address packet that follows determines whether the TWI enters host transmitter mode or master receiver mode. If SLA+W is sent, the TWI enters host transmitter mode. If SLA+R is sent, the TWI enters host receiver mode. The status codes mentioned in this section assume that the prescaler control bit is "0".

The START signal is issued by writing the following values to the TWCR register.

TWINT	TWEA	TWSTA	TWSTO	TWWC	TWEN	-	TWIE
1	control	1	0	control	1	0	control

The TWEN bit must be set to "1" to enable the TWI interface, TWSTA to "1" to send the START signal, and TWINT to

"1" to clear

TWINT flag bit To TWI module detects the bus status and sends the START signal as soon as the bus is free.When the START After that, the hardware sets the TWINT flag bit and updates the status code of TWSR to 0x08.

In order to enter the host transmit mode, SLA+W must be sent, which can be done by the following operation. First write SLA+W to the TWDR register, then write a "1" to the TWINT bit to clear the TWINT flag bit to continue transmission, i.e. write the following value to the TWCR register to send SLA+W.

TWINT	TWEA	TWSTA	TWSTO	TWWC	TWEN	-	TWIE
1	control	0	0	control	1	0	control

When the SLA+W transmission is completed and an answer signal is received, TWINT is set again and the status code of the TWSR is updated. The possible status codes are 0x18, 0x20 or 0x38. The appropriate response under each status code is described in detail in the status code table.

When SLA+W has been sent successfully, the packet can be started. This can be done by writing data to the TWDR register. TWDR can only be written when the TWINT flag bit is high.Otherwise,access is ignored and the write conflict flag bit TWWC is set. After updating TWDR, write a "1" to the TWINT bit to clear the TWINT flag bit to continue the transfer. That is, the following values are written to the TWCR register to send data.

TWINT	TWEA	TWSTA	TWSTO	TWWC	TWEN	-	TWIE
1	control	0	0	control	1	0	control

When the packet is sent and an answer signal is received, TWINT is set again and the status code of TWSR is updated. The possible status codes are 0x28 or 0x30. The appropriate response under each status code is described in detail in the status code table.

When the data has been sent successfully, the packet can continue to be sent. This process is repeated until the last byte is sent. The host generates a STOP signal or REPEATED START signal before the entire transmission ends.

The STOP signal is issued by writing the following values to the TWCR register.

TWINT	TWEA	TWSTA	TWSTO	TWWC	TWEN	-	TWIE
1	control	0	1	control	1	0	control

The REPEATED START signal is issued by writing the following values to the TWCR register.

TWINT	TWEA	TWSTA	TWSTO	TWWC	TWEN	-	TWIE
1	control	1	0	control	1	0	control

After sending REPEATED START (status code 0x10), the TWI interface can access the same slave again, or access a new slave without sending the STOP signal.REPEATED START enables the host to switch between different slaves and between host transmitter and host receiver modes without losing control of the bus.

The status codes and corresponding operations in host transmit mode are shown in the following table.

Status code table for host transmit mode

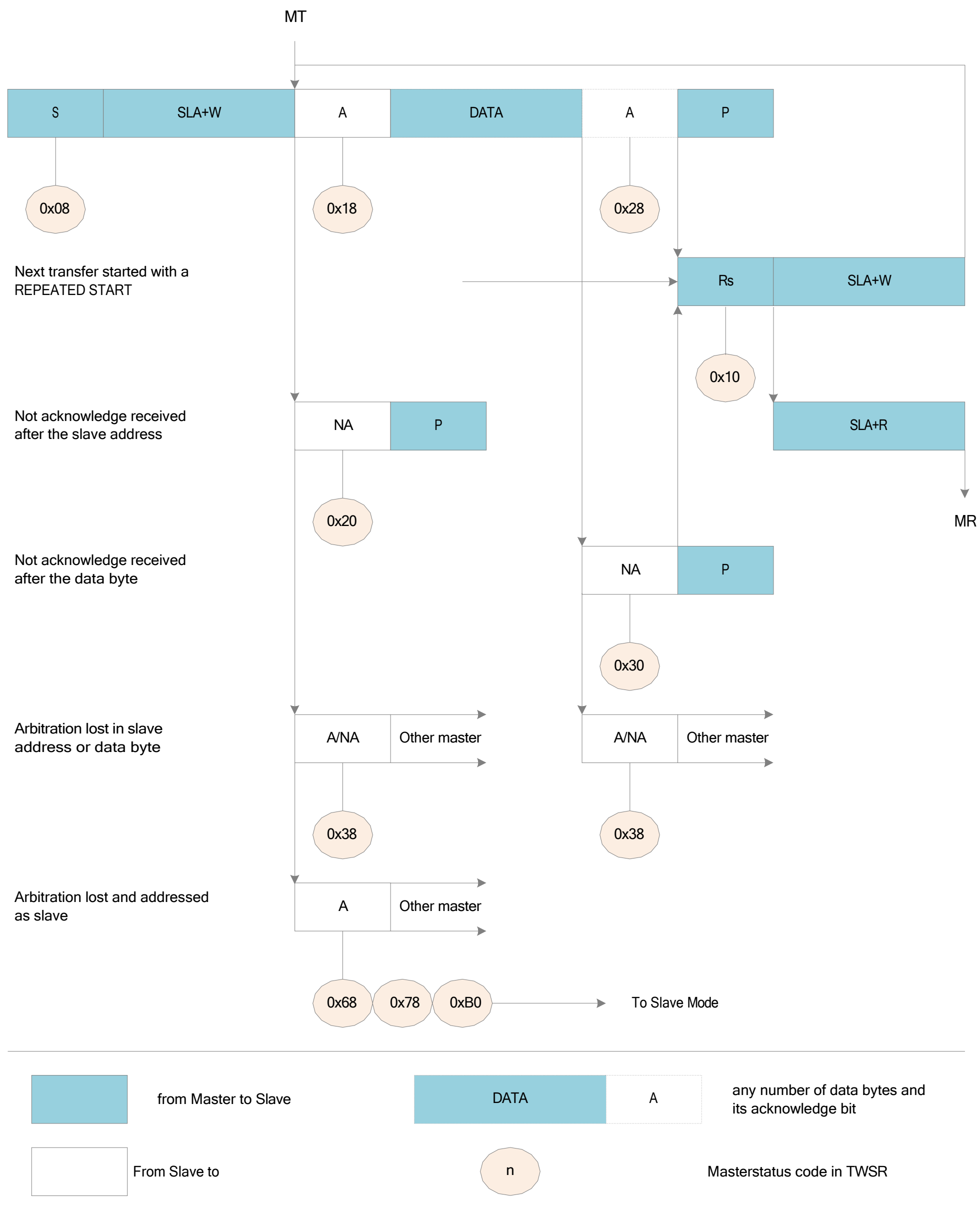
Status Code	Bus and hardware status	Application software response					Next steps for hardware
		Read/write TWDR	Operation of TWCR				
			STA	STO	TWINT	TWEA	

0x08	START Issued give (as a present)	Loading SLA+W	0	0	1	control	SLA+W will be sent. ACK or NACK will be received
0x10	REPEATED START Issued	Loading SLA+W	0	0	1	control	SLA+W will be sent. ACK or NACK will be received

	give (as a present)	Loading SLA+R	0	0	1	control	SLA+R will be sent. ACK or NACK will be received; Will switch to MR mode
0x18	SLA+W has Sending. Received ACK	Loading data	0	0	1	control	Data will be sent. ACK or NACK will be received
		No Operation	1	0	1	control	Will send REPEATED START
		No Operation	0	1	1	control	STOP will be sent. Will reset the TWSTO flag
		No Operation	1	1	1	control	STOP will be sent. The TWSTO flag will be reset; the START will be sent
0x20	SLA+W has Sending. Received from NACK	Loading data	0	0	1	control	Data will be sent. ACK or NACK will be received
		No Operation	1	0	1	control	Will send REPEATED START
		No Operation	0	1	1	control	STOP will be sent. The TWSTO flag will be reset
		No Operation	1	1	1	control	STOP will be sent. The TWSTO flag will be reset; the START will be sent
0x28	Data byte sent; ACK received	Loading data	0	0	1	control	Data will be sent. ACK or NACK will be received
		No Operation	1	0	1	control	Will send REPEATED START
		No Operation	0	1	1	control	STOP will be sent. Will reset the TWSTO flag
		No Operation	1	1	1	control	STOP will be sent. The TWSTO flag will be reset; the START will be sent
0x30	Data byte sent; NACK	Loading	0	0	1	control	Data will be sent. ACK or NACK will be received

	received	data					
		No Operation	1	0	1	control	Will send REPEATED START
		No Operation	0	1	1	control	STOP will be sent. Will reset the TWSTO flag
		No Operation	1	1	1	control	STOP will be sent. The TWSTO flag will be reset; the START will be sent
0x38	SLA+W or Data arbitration Failure	No Operation	0	0	1	control	will release the bus. Will enter unaddressed slave mode
		No Operation	1	0	1	control	Will be sent in free time START

The format and status of the host transmit mode is shown below.


MT
S
SLA+W
A
DATA
A
P
0x08
0x18
0x28
Next transfer started with a REPEATED START
Rs
SLA+W
0x10
Not acknowledge received after the slave address
NA
P
SLA+R
0x20
MR
Not acknowledge received after the data byte
NA
P
0x30
Arbitration lost in slave address or data byte
A/NA
Other master
A/NA
Other master
0x38
0x38
Arbitration lost and addressed as slave
A
Other master
0x68
0x78
0xB0
To Slave Mode
from Master to Slave
DATA
A
any number of data bytes and its acknowledge bit
From Slave to
n
Masterstatus code in TWSR

Format and status diagram for host send mode

Host receive mode

In the host receive mode, the TWI receives a certain number of data bytes from the slave transmitter. In order to enter host mode, the START signal must be sent. The format of the address packet that follows determines whether the TWI enters the host transmitter mode or the host receiver mode. If SLA+W is sent, it enters host transmitter mode. If SLA+R is sent, the TWI enters host receiver mode. The status codes mentioned in this section assume that the prescaler control bit is "0".

The START signal is issued by writing the following values to the TWCR register.

TWINT	TWEA	TWSTA	TWSTO	TWWC	TWEN	-	TWIE
1	control	1	0	control	1	0	control

The TWEN bit must be set to "1" to enable the TWI interface, TWSTA to "1" to send the START signal, and TWINT to "1" to clear the TWINT flag bit. The TWI module detects the bus status and sends the START signal as soon as the bus is free.When the START is sent, the hardware sets the TWINT flag bit and updates the status code of TWSR to 0x08.

In order to enter host receive mode, SLA+R must be sent, which can be done by the following operation. First write SLA+R to the TWDR register, then write a "1" to the TWINT bit to clear the TWINT flag bit to continue transmission, i.e. write the following value to the TWCR register to send SLA+R.

TWINT	TWEA	TWSTA	TWSTO	TWWC	TWEN	-	TWIE
1	control	0	0	control	1	0	control

When the SLA+R transmission is completed and an answer signal is received, TWINT is set again and the status code of TWSR is updated. The possible status codes are 0x38, 0x40 or 0x48. The appropriate response under each status code is described in detail in the status code table.

After SLA+R has been successfully sent,packet reception can begin.Continue receiving by writing a "1" to the TWINT bit to clear the TWINT flag bit. That is, write the following values to the TWCR register to initiate reception.

TWINT	TWEA	TWSTA	TWSTO	TWWC	TWEN	-	TWIE
1	control	0	0	control	1	0	control

When the packet reception is complete and the answer signal is sent, TWINT is set again and the status code of TWSR is updated. The possible status codes are 0x50 or 0x58. The appropriate response under each status code is described in detail in the status code table.

When the data is received successfully, the packet can continue to be received. This process is repeated until the last byte is received. After the host receives the last byte, it must send a NACK answer signal to the slave transmitter. The host generates a STOP signal or REPEATED START signal for the entire reception to end.

The STOP signal is issued by writing the following values to the TWCR register.

TWINT	TWEA	TWSTA	TWSTO	TWWC	TWEN	-	TWIE
1	control	0	1	control	1	0	control

The REPEATED START signal is issued by writing the following values to the TWCR register.

TWINT	TWEA	TWSTA	TWSTO	TWWC	TWEN	-	TWIE
1	control	1	0	control	1	0	control

After sending REPEATED START (status code 0x10), the TWI interface can access the same host again, or access a new host without sending the STOP signal.REPEATED START enables the host to switch between different slaves, and between host transmitter and host receiver modes, without losing control of the bus.

The status codes and corresponding operations in host receive mode are shown in the following table.

Status code table for host receive mode

status code	Bus and hardware status	Application software response					Next steps for hardware
		Read/write TWDR	Operation of TWCR				
			STA	STO	TWINT	TWEA	

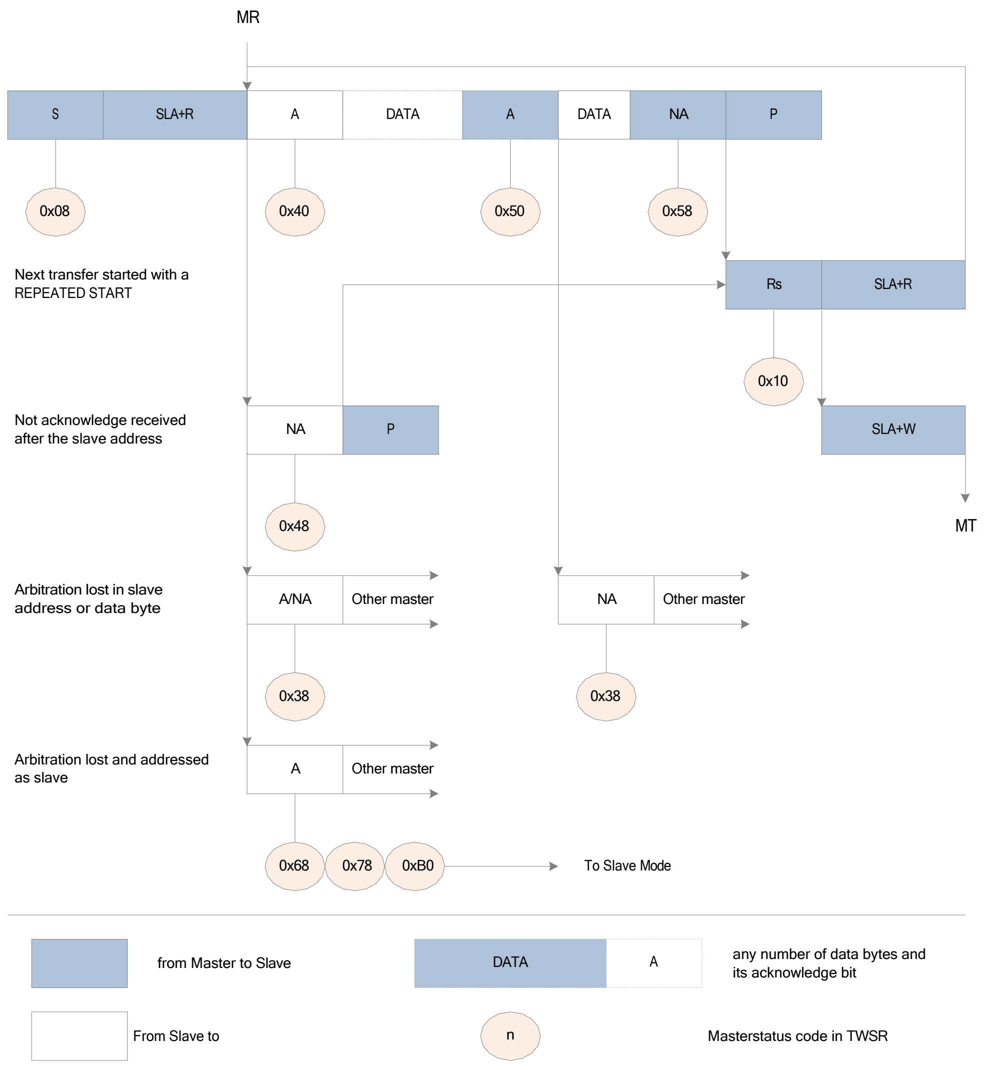
0x08	START has sending	Loadin g SLA+R	0	0	1	control	SLA+R will be sent. ACK or NACK will be received
0x10	REPEATED START has	Loadin g SLA+R	0	0	1	control	SLA+R will be sent. ACK or NACK will be received

	sending	Loading SLA+W	0	0	1	control	SLA+W will be sent. ACK or NACK will be received; Will switch to MT mode
0x38	SLA+R or Data arbitration failure	unaffected (i.e. behaving naturally) operate	0	0	1	control	will release the bus. Will enter unaddressed slave mode
		unaffected (i.e. behaving naturally) operate	1	0	1	control	Will be sent in free time START
0x40	SLA+R has Send; Receive To ACK	unaffected (i.e. behaving naturally) operate	0	0	1	0	Data will be received; NACK will be sent

		unaffected (i.e. behaving naturally) operate	0	0	1	1	Data will be received; ACK will be sent
0x48	SLA+R has Send; Receive to NACK	unaffected (i.e. behaving naturally) operate	1	0	1	control	send out START give REPEATED (as a present)
		unaffected (i.e. behaving naturally) operate	0	1	1	control	STOP will be sent. Will reset the TWSTO flag
		unaffected (i.e operate	1	1	1	control	STOP will be sent. The TWSTO flag will be reset; the START will be sent

		. behavingnaturally)					
0x50	Data byte received; ACK sent give (as a present)	Read Data	0	0	1	0	Data will be received; NACK will be sent
		Read Data	0	0	1	1	Data will be received; ACK will be sent
0x58	Data byte received; NACK received sending	Read Data	1	0	1	control	send out START give (as a present) REPEATED
		Read Data	0	1	1	control	STOP will be sent. Will reset the TWSTO flag
		Read Data	1	1	1	control	STOP will be sent. The TWSTO flag will be reset; the START will be sent

The format and status of the host receive mode is shown in the following figure.


MR
S
SLA+R
A
DATA
A
DATA
NA
P
0x08
0x40
0x50
0x58
Next transfer started with a REPEATED START
Rs
SLA+R
0x10
SLA+W
MT
Not acknowledge received after the slave address
NA
P
0x48
Arbitration lost in slave address or data byte
A/NA
Other master
NA
Other master
0x38
0x38
Arbitration lost and addressed as slave
A
Other master
0x68
0x78
0xB0
To Slave Mode
from Master to Slave
DATA
A
any number of data bytes and its acknowledge bit
From Slave to
n
Masterstatus code in TWSR

Format and status diagram for host receive mode

Slave receive mode

In slave receive mode, a certain number of data bytes can be received from the host transmitter. The status codes mentioned in this section assume that the prescaler control bit is "0".

To start the slave receive mode, set the TWAR and TWCR registers.
TWAR needs to be set as follows.

TWA6	TWA5	TWA4	TWA3	TWA2	TWA1	TWA0	TWGCE
Device Slave Address							

The high 7 bits of TWAR are the slave address to which the TWI interface will respond when addressed by the host. If LSB is set, the TWI will respond to the broadcast call address (0x00) otherwise the broadcast call address is ignored.
TWCR needs to be set as follows.

TWINT	TWEA	TWSTA	TWSTO	TWWC	TWEN	-	TWIE
0	1	0	0	0	1	0	control

TWEN must be set to enable the TWI interface and TWEA must be set to return an acknowledgement message ACK when the host addresses (slave address or broadcast call) to itself.TWSTA and TWSTO must be cleared to zero.

After initializing the TWAR and TWCR, the TWI interface begins to wait until its own slave address (or broadcast address) is addressed. When the data direction bit immediately following the slave address is a "0" (indicating a write operation), the TWI enters slave receive mode. When the data direction bit is "1" (indicating a read operation), the TWI enters the slave transmit mode. After receiving its own slave address and write operation flag bit, the TWINT flag bit is set and the valid status codes are updated to the TWSR. The appropriate response under each status code is described in detail in the status code table. Note that the slave receive mode can also be entered when TWI arbitration in master mode has failed (see status codes 0x68 and 0x78)

If the TWEA bit is reset during a transmission, the TWI will return NACK (high) to the SDA line after a byte is received. This can be used to indicate that the slave cannot receive more data. The TWI will also not respond to its own slave address when the TWEA bit is "0". However, the TWI will still listen to the bus and can resume address recognition and respond once TWEA is set. This means that TWEA can be used to temporarily isolate the TWI interface from the bus.

The TWI interface clock can be turned off when in a sleep mode other than idle mode. If the slave receive mode is enabled, the interface will continue to respond to the slave address or broadcast address using the bus clock. An address match will wake up the MCU and during the wakeup period, the TWI interface will hold SCL low until the TWINT flag is cleared. More data can be received when the TWI interface clock returns to normal.

The status codes for the slave receive mode are shown in the following table.

Status code table for slave receive mode

status code	Bus and hardware status	Application software response					Next steps for hardware
		Read/write TWDR	Operation of TWCR				
			STA	STO	TWINT	TWEA	
0x60	SLA+W Received. ACK sent	unaffected (i.e. be having naturally) operate	control	0	1	0	Data will be received; NACK will be sent
		unaffected (i.e operate	control	0	1	1	Data will be received; ACK will be sent

		. behaving naturally)					
0x68	Arbitration failed when sending SLA+R/W; SLA+W was received; ACK was sent	unaffected (i.e. behaving naturally) operate	control	0	1	0	Data will be received; NACK will be sent
		unaffected (i.e. behaving naturally) operate	control	0	1	1	Data will be received; ACK will be sent
0x70	Broadcast addresses are received. ACK sent	unaffected (i.e. behavi operate	control	0	1	0	Data will be received; NACK will be sent

		ng natur ally)						
		unaffected (i.e. behaving naturally)	operate	control	0	1	1	Data will be received; ACK will be sent
0x78	Arbitration failed when sending SLA+R/W; SLA+W was received; ACK was sent	unaffected (i.e. behaving naturally)	operate	control	0	1	0	Data will be received; NACK will be sent
		unaffected (i.e. behaving naturall	operate	control	0	1	1	Data will be received; ACK will be sent

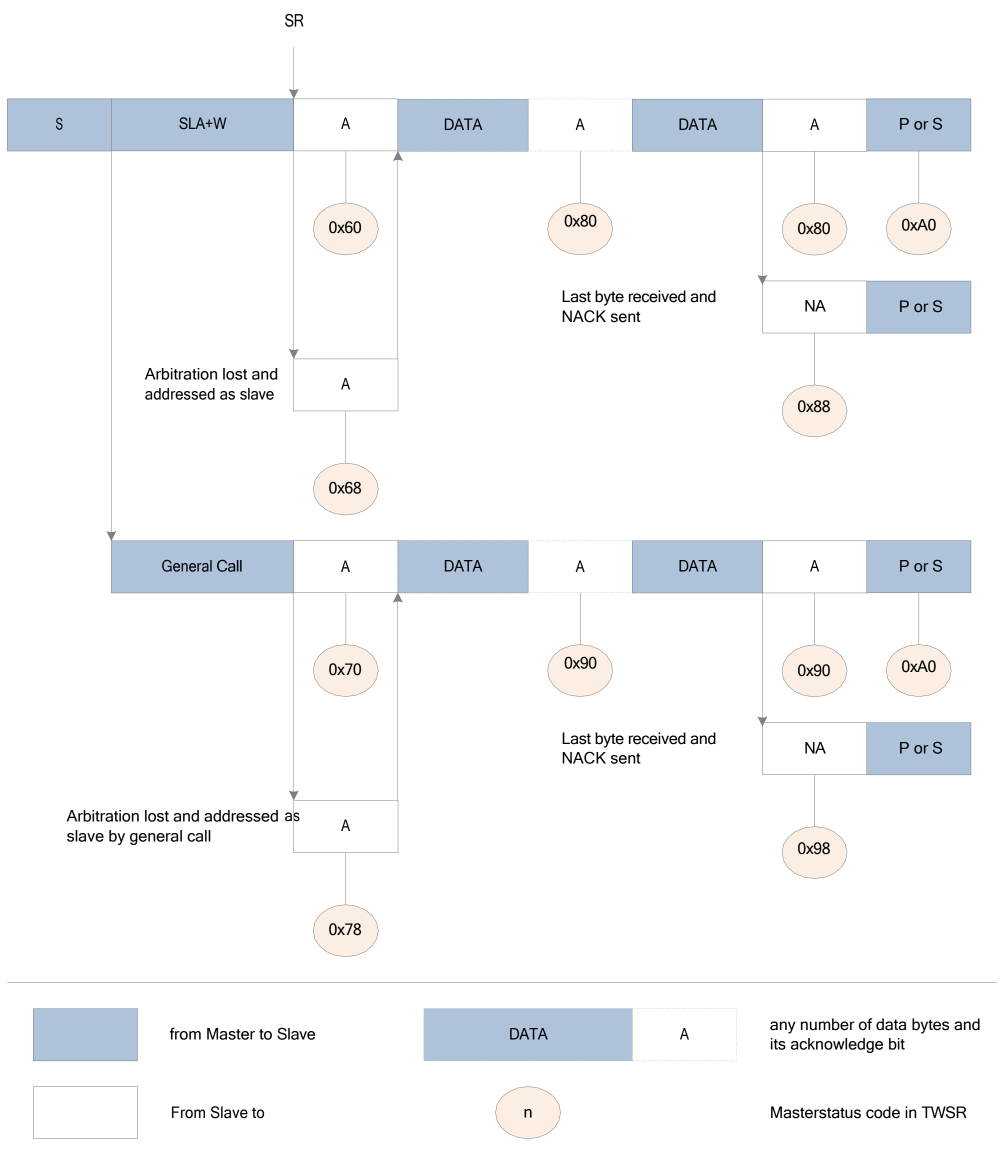
		y)					
0x80	Self-data received. ACK sent	Read Data	control	0	1	0	Data will be received; NACK will be sent
		Read Data	control	0	1	1	Data will be received; ACK will be sent
0x88	Self-data received.	rea fetch d	0	0	1	0	Will switch to unaddressed

	NACK has been sent	data					Slave mode. Will not respond to slave addresses and broadcasts
		Read Data	0	0	1	1	will switch to unaddressed slave mode. will respond to the slave address. TWGCE=1 will sound respond to a broadcast
		Read Data	1	0	1	0	will switch to unaddressed slave mode. will not respond to slave addresses and broadcasts. When the bus is idle it will send Send START
		Read Data	1	0	1	1	will switch to unaddressed slave mode. will respond to the slave address. will respond to the broadcast when TWGCE = 1. When the bus is idle it will send Send START
0x90	Broadcast data received. ACK sent	Read Data	control	0	1	0	Data will be received; NACK will be sent
		Read Data	control	0	1	1	Data will be received; ACK will be sent
0x98	Broadcast data received. NACK has been sent	Read Data	0	0	1	0	will switch to unaddressed slave mode. will not respond to the slave ground Addresses and Broadcasts

		Read Data	0	0	1	1	will switch to unaddressed slave mode. will respond to the slave address. TWGCE=1 will sound respond to a broadcast
		Read Data	1	0	1	0	will switch to unaddressed slave mode. will not respond to slave addresses and broadcasts. When the bus is idle it will send Send START
		Read and take	1	0	1	1	Will switch to unaddressed

		data					Slave mode. will respond to the slave address. will respond to the broadcast when TWGCE = 1. When the bus is idle it will send Send START
0xA0	STOP or REPEATED START is received while the slave is operating	No Operation	0	0	1	0	will switch to unaddressed slave mode. will not respond to the slave ground Addresses and Broadcasts
		No Operation	0	0	1	1	will switch to unaddressed slave mode. will respond to the slave address. TWGCE=1 will sound respond to a broadcast
		No Operation	1	0	1	0	will switch to unaddressed slave mode. will not respond to slave addresses and broadcasts. When the bus is idle it will send Send START
		No Operation	1	0	1	1	will switch to unaddressed slave mode. will respond to the slave address. will respond to the broadcast when TWGCE = 1. When the bus is idle it will send Send START

The format and status diagram for the slave receive mode is shown below.


SR
S
SLA+W
A
DATA
A
DATA
A
P or S
0x60
0x80
0x80
0xA0
Last byte received and NACK sent
NA
P or S
0x88
Arbitration lost and addressed as slave
A
0x68
General Call
A
DATA
A
DATA
A
P or S
0x70
0x90
0x90
0xA0
Last byte received and NACK sent
NA
P or S
0x98
Arbitration lost and addressed as slave by general call
A
0x78
from Master to Slave
DATA
A
any number of data bytes and its acknowledge bit
From Slave to
n
Masterstatus code in TWSR

Format and status diagram of the slave receive mode

Slave transmit mode

In slave transmit mode, a certain number of data bytes can be sent to the host receiver. The status codes mentioned in this section assume that the prescaler control bit is "0".

To start the slave receive mode, set the TWAR and TWCR registers.
TWAR needs to be set as follows.

TWA6	TWA5	TWA4	TWA3	TWA2	TWA1	TWA0	TWGCE
Device Slave Address							

The high 7 bits of TWAR are the slave address to which the TWI interface will respond when addressed by the host. If LSB is set, the TWI will respond to the broadcast call address (0x00) otherwise the broadcast call address is ignored.
TWCR needs to be set as follows.

TWINT	TWEA	TWSTA	TWSTO	TWWC	TWEN	-	TWIE
0	1	0	0	0	1	0	control

TWEN must be set to enable the TWI interface and TWEA must be set to return an acknowledgement message ACK when the host addresses (slave address or broadcast call) to itself.TWSTA and TWSTO must be cleared to zero.

After initializing the TWAR and TWCR, the TWI interface begins to wait until its own slave address (or broadcast address) is addressed. When the data direction bit immediately following the slave address is a "0" (indicating a write operation), the TWI enters slave receive mode. When the data direction bit is "1" (indicating a read operation), the TWI enters the slave transmit mode. After receiving its slave address and the read operation flag bit, the TWINT flag bit is set and the valid status codes are updated to the TWSR. The appropriate response under each status code is described in detail in the status code table. Note that the slave transmit mode can also be entered when TWI arbitration in master mode has failed (see status code 0xB0)

If the TWEA bit is reset during a transmission, the TWI will switch to unaddressed slave mode after the last byte is transmitted. After the host receiver gives a NACK or ACK for the last byte transmitted, the status code in the TWSR register will be updated to 0xC0 or 0xC8. If the host receiver continues the transmission operation, the slave transmitter will not respond and the host will receive data with an all "1" (i.e., 0xFF) When the slave has sent the last byte of data (TWEA is cleared) and expects a NACK response, and the host wants to receive more data and sends an ACK in response, the TWSR will be updated to 0xC8.

When the TWEA bit is "0", the TWI will not respond to its own slave address either. However, the TWI will still listen to the bus and can resume address recognition and respond once TWEA is set. This means that TWEA can be used to temporarily isolate the TWI interface from the bus.

The TWI interface clock can be turned off when in a sleep mode other than idle mode. If the slave receive mode is enabled, the interface will continue to respond to the slave address or broadcast address using the bus clock. An address match will wake up the MCU and during the wakeup period, the TWI interface will hold SCL low until the TWINT flag is cleared. More data can be received when the TWI interface clock returns to normal.

The status codes for the slave transmit mode are shown in the following table.

Status code table for slave transmit mode

status code	Bus and hardware status	Application software response					Next steps for hardware
		Read/write TWDR	Operation of TWCR				
			STA	STO	TWINT	TWEA	
0xA8	SLA+R has Received; ACK sent	Loading data	control	0	1	0	The last data will be sent. Expect to receive NACK
		Loading data	control	0	1	1	Data will be sent; ACK will be received
0xB0	send out send SLA+R/W	Loading data	control	0	1	0	The last data will be sent. Expect to receive NACK

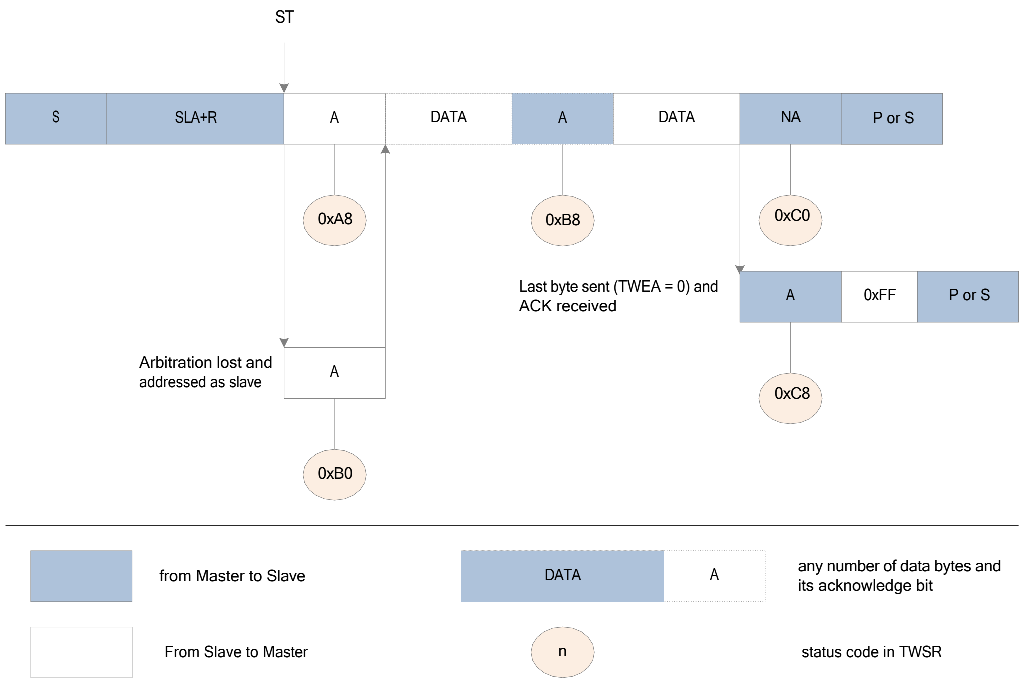
	When arbitration fails; SLA+R has Receiving. ACK sent	Loading data	contr ol	0	1	1	Data will be sent; ACK will be received

0xB8	Data sent; ACK received	Loading data	control	0	1	0	The last data will be sent. Expect to receive NACK
		Loading data	control	0	1	1	Data will be sent; ACK will be received
0xC0	Data has been sent; NACK has receive	no operation	0	0	1	0	will switch to unaddressed slave mode. will not respond to the slave address and (formal) propagate
		no operation	0	0	1	1	will switch to unaddressed slave mode. will respond to the slave address. TWGCE=1 will respond to the broadcast
		no operation	1	0	1	0	will switch to unaddressed slave mode. will not respond to slave addresses and broadcasts. When the bus is idle it will send START
		no operation	1	0	1	1	will switch to unaddressed slave mode. will respond to the slave address; TWGCE=1 will respond to the broadcast. When the bus is idle it will send START
0xC8	(a) The last data has been sent. ACK Received	no operation	0	0	1	0	will switch to unaddressed slave mode. will not respond to the slave address and (formal) propagate
		no operation	0	0	1	1	will switch to unaddressed slave mode. will respond to the slave address.

							TWGCE=1 will respond to the broadcast
		no operation	1	0	1	0	will switch to unaddressed slave mode. will not respond to slave addresses and broadcasts. When the bus is idle it will send START
		no operation	1	0	1	1	will switch to unaddressed slave mode.

							will respond to the slave address; TWGCE=1 will respond to the broadcast. When the bus is idle it will send START

The format and status of the slave transmit mode is shown in the following figure.


ST
S
SLA+R
A
DATA
A
DATA
NA
P or S
0xA8
0xB8
0xC0
Last byte sent (TWEA = 0) and ACK received
A
0xFF
P or S
0xC8
Arbitration lost and addressed as slave
A
0xB0
from Master to Slave
DATA
A
any number of data bytes and its acknowledge bit
From Slave to Master
n
status code in TWSR

Format and status diagram for slave transmit mode

Other states

There are two status codes that do not have corresponding TWI status definitions, as shown in the following table.

Other status code tables

status code	Bus and hardware status	Application software response					Next steps for hardware
		Read/write TWDR	Operation of TWCR				
			STA	STO	TWINT	TWEA	
0xF8	No status information. TWINT= 0	no operation	No operation TWCR				Waiting or performing the current operation
0x00	Bus error caused by an illegal START or STOP	no operation	0	1	1	control	Affects internal hardware only; no STOP is sent to the bus; the bus is released and the TWSTO bit is cleared

]

The status code 0xF8 indicates that there is currently no relevant information because the TWINT flag is "0". This status may occur when the TWI interface is not participating in a serial transfer or when the current transfer has not yet completed.

Status 0x00 indicates that a bus error occurred during a serial transfer. A bus error occurs when an illegal START or STOP occurs. To recover from the error, TWSTO must be set and TWINT cleared by writing a "1". This will put the TWI interface into unaddressed slave mode without generating a STOP, as well as release SCL and SDA and clear the TWSTO bit.

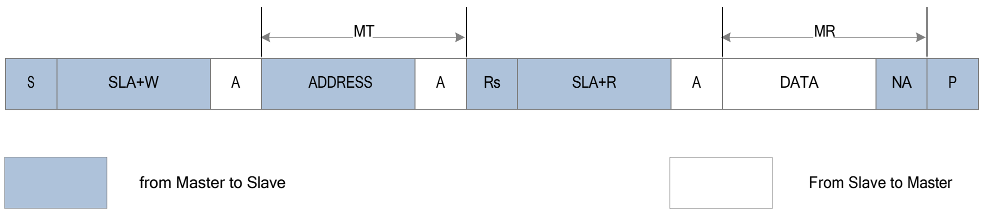
Combination mode

In some cases, several TWI modes must be combined in order to accomplish the desired job. For example, from a serial EEPROM To read the data, a typical transfer consists of the following steps.

1. Transmission must be initiated.
2. The EEPROM must be told where the data should be read.
3. (a) The read operation must be completed.
4. The transmission must end.

Note that data can be transmitted from the master to the slave and vice versa. The host tells the slave where to read the data, using the host send mode. Next, the data is read from the slave, and the host receive mode is used. The direction of the transmission changes. The host must maintain control of the bus at all stages, and all steps are uninterrupted operations. If another host changes the location of the read data between steps 2 and 3 in a multi-host system, this principle is broken and the host reads the data in the wrong location. Changing the direction of the data transfer is accomplished by sending a REPEATED START between the transmitted address byte and the received data. After sending REPEATED START, the host still has bus control.

The following diagram depicts this transmission process.



Combining Multiple TWI Modes to Access the Serial EEPROM Figure

Multi-host systems and arbitration

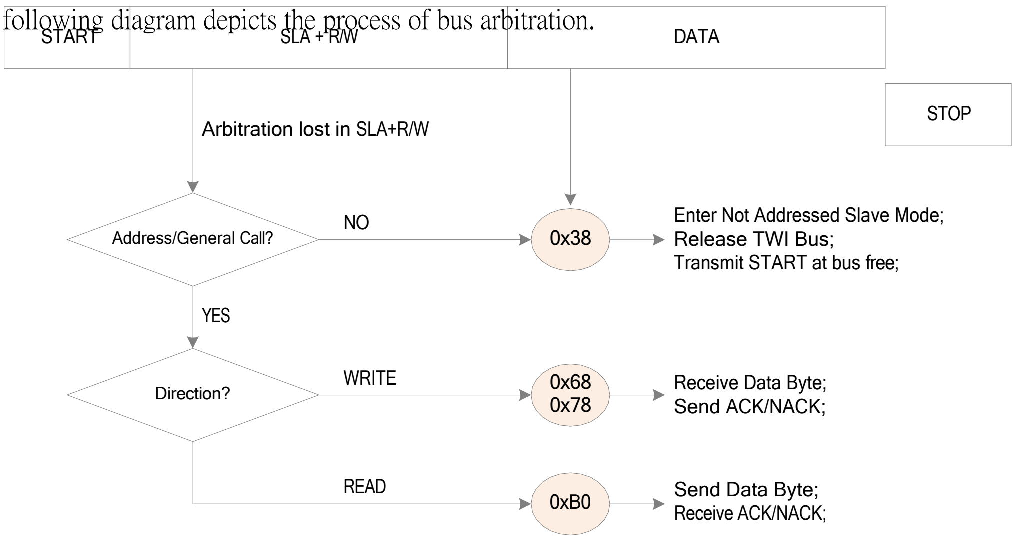
If there are multiple hosts connected to the same TWI bus, one or more of them may start data transmission at the same time. The TWI protocol ensures that in such a case, no data is lost through an arbitration process that allows one of the hosts to make the transfer. The following is an example of two hosts trying to send data to a slave to describe the process of bus arbitration.

There are several different scenarios that give rise to the bus arbitration process.

- Two or more hosts are communicating with a slave at the same time. In this case, neither the master nor the slave is aware that there is competition on the bus.
- Two or more hosts are simultaneously accessing the same slave in different data or operation directions. Arbitration occurs in this case, either in the READ/WRITE bits or in the data bits.When another host sends a "0" to the SDA line, the host sending a "1" to the SDA line will fail to arbitrate. The failed host will either switch to unaddressed slave mode or wait for the bus to become free to send a new START signal, depending on the operation of the application software.
- Two or more hosts access different slaves. In this case, bus arbitration occurs during the SLA phase. When another host sends a "0" to the SDA line, the host sending a "1" to the SDA line will fail to arbitrate. A host that fails during SLA bus arbitration will switch to slave mode and check if it is addressed by a host that has gained control of the bus. If it is addressed, it will enter SR or ST mode,

depending on the READ/WRITE bit following the SLA. If not addressed address, it will switch to the unaddressed slave mode or wait for a new START signal to be sent when the bus is free, depending on the operation of the application software.

The following diagram depicts the process of bus arbitration.



Bus Arbitration Process Diagram

Register Definition

TWI Register List

processor register	address	default value	description
TWBR	0x B8	0x00	TWI Bit Rate Register
TWSR	0xB9	0x00	TWI Status Register
TWAR	0xBA	0x00	TWI Address Register
TWDR	0xBB	0x00	TWI Data Register
TWCR	0xBC	0x00	TWI control register
TWAMR	0xBD	0x00	TWI Address Mask Register

TWBR - TWI Bit Rate Register

TWBR - TWI Bit Rate Register								
Address: 0xB8					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	R/W

Bit	Name	description
7:0	TWBR[7:0]	TWI Bit rate selection control bit. TWBR is the bit rate generator crossover factor. The bit rate generator is a divider used to generate the SCL clock in host mode. The formula for calculating the bit rate is shown below. $f_{scl} = fsys/(16 + 2*TWBR*4^{TWPS})$.

TWSR - TWI Status Register

TWSR - TWI Status Register								
Address: 0xB9					Default value: 0xF8			
Bit	7	6	5	4	3	2	1	0
Name	TWS7	TWS6	TWS5	TWS4	TWS3	-	TWPS1	TWPS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	description	
7:3	TWS [7:3]	TWI status flag bit. The 5-bit TWS reflects the state of the TWI logic and the bus. The different status values have different meanings, as described in the TWI operating mode. The value read from the TWSR consists of the 5-bit status value and the 2-bit prescaler control bit, which should be masked to "0" when detecting status. This is a status detection independent of the prescaler setting.	
2	-	Reserved.	
1	TWPS1	TWI Prescaler Control High. TWPS1 and TWPS0 together form TWPS[1:0], which is used to control the bitrate prescaler factor and, together with TWBR, the bitrate.	
0	TWPS0	TWI Prescaler Control Low. TWPS0 and TWPS1 together form TWPS[1:0], which is used to control the bitrate prescaler factor and, together with TWBR, the bitrate.	
		TWPS[1:0]	prescaling factor
		0	1
		1	4
		2	16
		3	64

TWAR - TWI Address Register

TWAR - TWI Address Register								
Address: 0xBA					Default value: 0x00			
Bit	7	6	5	4	3	2	1	0
Name	TWAR6	TWAR5	TWAR4	TWAR3	TWAR2	TWAR1	TWAR0	TWGCE
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	description
7:1	TWA [6:0]	TWI Slave address bit. TWA is the TWI slave address. When the TWI is operating in slave mode, the TWI will respond based on this address. This address is not required for master mode. However, in a multi-master system, it is also necessary to set the slave address for access by other masters.

0	TWGCE	TWI Broadcast Identification Enable Control Bit. When the TWGCE bit is set to "1", TWI bus broadcast recognition is enabled. When the TWGCE bit is set to "0", TWI bus broadcast recognition is disabled. When TWGCE is set and the received address frame is 0x00, the TWI module responds to this Bus Broadcast.

TWDR - TWI Data Register

TWDR - TWI Data Register								
Address: 0xBB					Default value: 0xFF			
Bit	7	6	5	4	3	2	1	0
Name	TWD7	TWD6	TWD5	TWD4	TWD3	TWD2	TWD1	TWD0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	description
7:0	TWD[7:0]	TWI Data Register. TWD is the next byte to be transmitted on the bus, or the previous byte just received from the bus.

TWCR - TWI Control Register

TWCR - TWI Control Register								
Address: 0xBC					Default value: 0x00			
Bit	7	6	5	4	3	2	1	0
Name	TWINT	TWEA	TWSTA	TWSTO	TWWC	TWEN	-	TWIE
R/W	R/W	R/W	R/W	R/W	R	R/W	-	R/W

Bit	Name	description
7	TWINT	TWI interrupt flag bit. Hardware will set the TWINT bit when the TWI has finished its current job ad wants the application software to intervene. If the global interrupt is set and the TWIE bit is in place, a TWI interrupt will be generated and the MCU will execute the TWI interrupt service routine. When the TWINT flag is set, the low level of the SCL signal will be extended. The TWINT flag bit can only be cleared by writing a "1" to the bit. The hardware will not automatically clear this bit, even if an interrupt service routine is executed. Also note that clearing this bit will immediately turn on TWI operation.Therefore,before clearing the TWINT bit,first complete the TWAR, TWAMR, TWSR and TWDR register accesses.
6	TWEA	TWI Enables the answer control bit. The TWEA bit controls the generation of the answer pulse. When the TWEA bit is set to "1" and one of the following conditions is met, an answer pulse will be generated on the TWI bus. 1) The slave address of the received device. 2) Receiving a broadcast call when TWGCE is in position. 3) One byte of data is received in host receive or slave receive mode. When the TWEA bit is set to "0", the device is temporarily disconnected from the TWI bus. The device regains address recognition after the bit is set.

5	TWSTA	TWI Start State Control Bit. The TWSTA bit needs to be set when the CPU wants to be the host on the TWI bus itself. The hardware will check if the bus is available and generate a start state on the bus when the bus is idle. When the bus is not idle, the TWI will wait until a stop state is detected and then generate a start state to declare that it wishes to be the host. After sending the start state the software The TWSTA bit must be cleared.

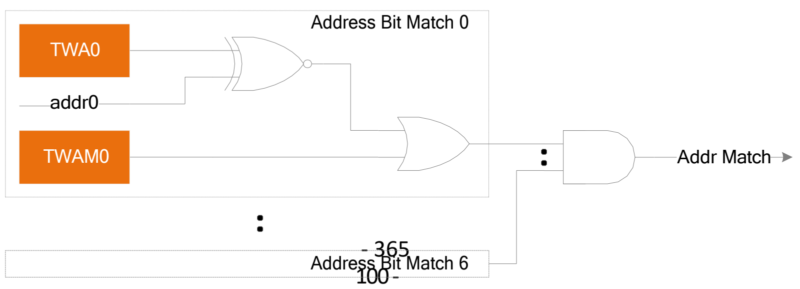
4	TWSTO	TWI Stop state control bit. When the TWSTO bit is "1" in master mode, the TWI will generate a stop state on the bus and then automatically clear the TWSTO bit. In slave mode, setting the TWSTO bit will allow the TWI to recover from the error state. This will not generate a stop state, but will simply return the TWI to a defined unaddressed slave mode, while releasing the SCL and SDA signal lines to High resistance state.
3	TWWC	TWI Write conflict flag bit. When the TWINT flag bit is low, writing the TWDR register will set the TWWC flag bit. When the TWINT flag bit is high, writing the TWDR register will clear the TWWC flag bit.
2	TWEN	TWI enable control bit. The TWEN bit enables TWI operation and activates the TWI interface. When the TWEN bit is set to "1", the TWI control IO pins are connected to the SCL and SDA pins. When the TWEN bit is set to "0", the TWI interface module is shut down and all transfers are terminated, including ongoing operations.
1	-	Reserved.
0	TWIE	TWI interrupt enable control bit. When the TWIE bit is set to "1" and the global interrupt is set, the TWI interrupt request will be activated whenever the TWINT flag bit is high.

TWAMR - TWI Address Mask Register

TWAMR - TWI Address Mask Register								
Address: 0xBD					Default value: 0x00			
Bit	7	6	5	4	3	2	1	0
Name	TWAR6	TWAR5	TWAR4	TWAR3	TWAR2	TWAR1	TWAR0	TWGCE
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	Name	description						
7:1	TWAM [6:0]	TWI Address Mask Control Bit. TWAM is a 7-bit TWI slave address mask control. Each bit of TWAM is used to mask (disable) the corresponding address bit in TWAR. When the location bit is masked, the address matching logic ignores the result of the comparison of the received address bit with the corresponding bit in the TWA. The following figure gives details of the address matching logic.						
0	-	Reserved.						

TWI Address Matching Logic

The following figure shows the TWI address matching logic block diagram.



Analog Comparator *0 (AC0)*

- Comparison accuracy of 10mV
- Factory out-of-tune calibration
- Supports 3 off-chip analog inputs
- Multiplexed inputs for ADC support (ADMUX)
- Internal differential amplifier input (DFFO) support
- Supports internal 8-bit DAC input (DAO)
- Programmable output digital filter control

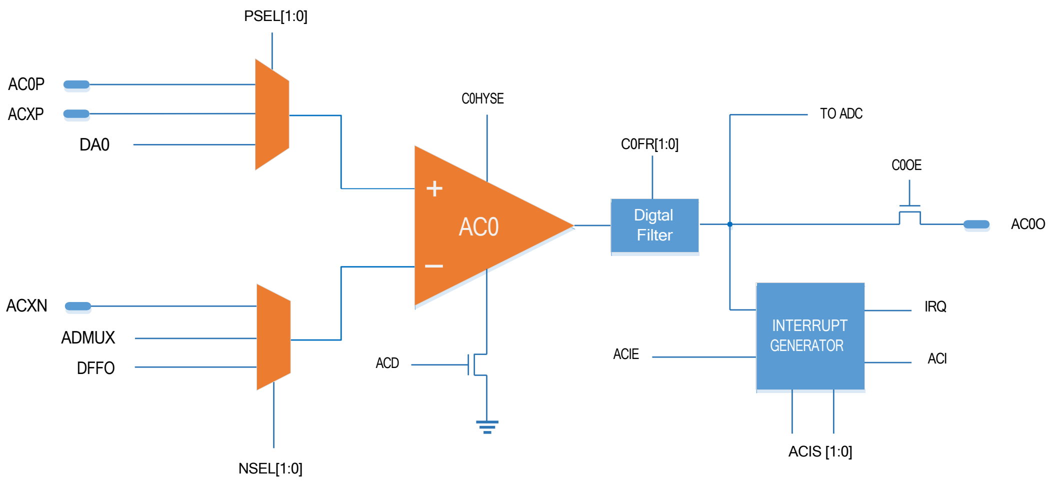
a general narrative

The analog comparator compares the levels of the positive and negative terminals of the input comparator, and the output ACO of the analog comparator is set when the voltage at the positive terminal is higher than the voltage at the negative terminal. When the level of ACO changes, the edge of the signal can be used to trigger an interrupt. The output signal ACO can also be used to trigger the input capture of Timer Counter 1 and to control the PWM output generated by the timer.

The LGT8FX8P integrated analog comparator, AC0, includes a multiple analog input selector, and the comparator positive and negative input sources can be selected from an external port or from a variety of internally generated reference sources. The analog comparator itself supports out-of-tune calibration, which ensures consistent comparator operation. The comparator supports an optional hardware hysteresis function to improve the stability of the comparator output. A hardware programmable digital filter is integrated into the comparator output, allowing you to select the appropriate filter setting for a more stable comparison output, depending on the application requirements.

The comparator output state can be read directly from registers, or interrupt requests can be generated for more efficient real-time event capture. The comparator output can also be output directly to an external IO port.

The structure of the op-amp/analog comparator 0 is shown in the figure below.



Analog Comparator 0
Function Schematic

Analog comparator input

Both inputs of the analog comparator support a variety of selectable input sources. The positive end has three selectable inputs.

1. External independent analog input AC0P
2. Analog Comparator 0/1 Common Analog Input ACXP
3. Output DAO for internal 8-bit DAC

The selection of the input source is controlled by the C0BG bit in the Control Status Register C0SR and the C0PS0 bit in the C0XR register, as described in the Register Description section of this chapter.

AC0P is a dedicated positive mode input channel for AC0. Note that there is a slight difference in the pinout of AC0P on the different package chips.

The ACXP is a common positive input for comparator 0/1. The LGT8FX8P has two internal analog comparators, and the ACXP is connected to the positive multiplexer of both comparators at the same time to facilitate the cooperative operation of both comparators.

The DAO is derived from the output of the internal 8-bit DAC.The reference source for the DAC can be selected from the system power supply, an internal reference, or an input from an external reference.Refer to the DAC-related section for DAC configuration.

C0BG	C0PS0	AC0 Positive input source
0	0	AC0P
0	1	ACXP
1	0	DAO
1	1	Close the positive input channel of the comparator

The negative input can also be selected from three different analog inputs:

1. Comparator 0/1 Common Analog Input ACXN
2. ADC Multiplexer Output ADMUX
3. Internal differential amplifier output DFFO

The comparator negative input channel selection is controlled by bit CME00/01 in the ADCSRB register from the ADC module. When the comparator negative input selection is ADMUX, the analog input channel selection is required via the CHMUX bit of the ADC module's ADMUX register; this mode allows for more flexible expansion of the comparator inputs.

ACXN is a common negative input for comparator 0/1, facilitating the implementation of comparator 0/1 co operation.

The DFFO comes from the internal differential amplifier outputDifferential amplifier with optional x1/x8/x16/x32 gain control enables detection and measurement of small signals.

CME01	CME00	AC0 Negative input source
0	0	ACXN
0	1	ADMUX
1	0	DFFO
1	1	Close the negative input channel of the comparator

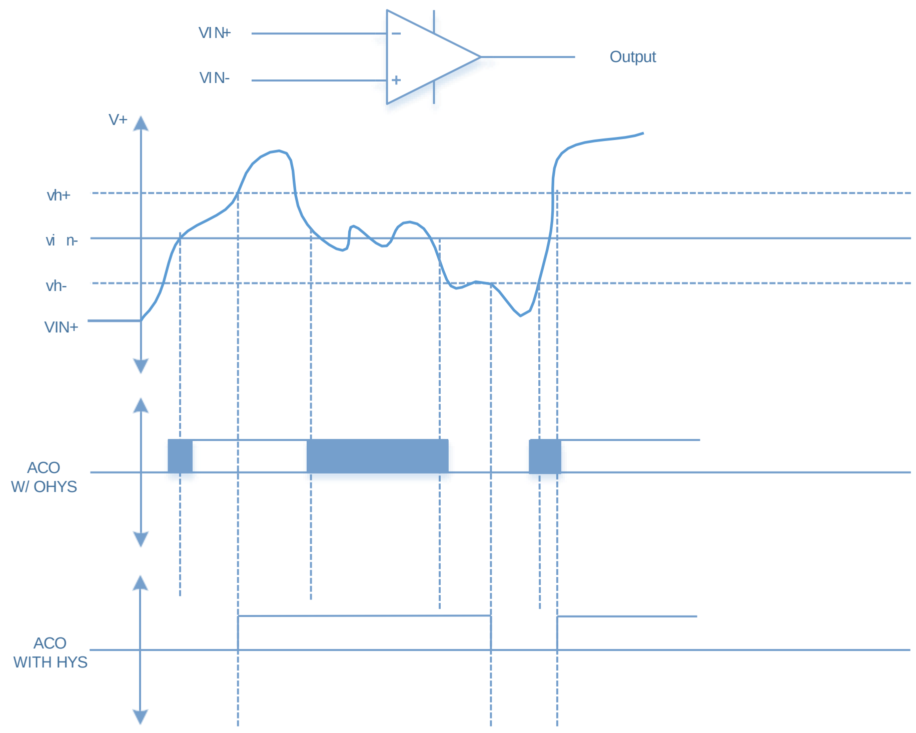
Comparator output filtering

The comparator outputs support a controllable hysteresis internally. The user can enable the hysteresis circuit via the C0HYSE bit of the C0XR register. The hysteresis circuit can eliminate the instability of the comparator state change process and achieve the output filtering function.

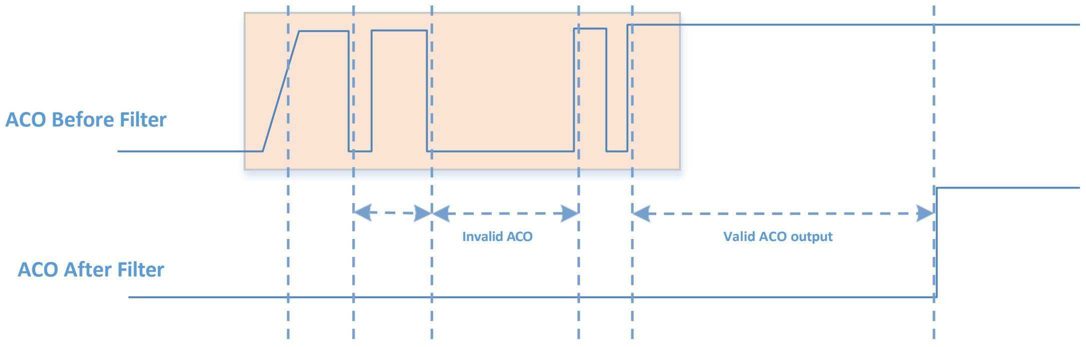
It is recommended that the user turn on the hysteresis circuit when using the comparator to obtain a stable comparator output.

As shown in the figure below, the hysteresis circuit is located between the analog output and the digital output of the comparator. When the input voltage at the positive end of the comparator, VIN+, is greater than (VIN- + VH+), the comparator COUT output is high; when the VIN+ voltage is less than (VIN- - VH-), the comparator output is low. The hysteresis circuit avoids the jitter introduced by the circuit itself when the voltage at the positive end of the comparator is close to the voltage at the negative end.

Diagram of comparator hysteresis voltage versus comparator output.



Although hysteresis circuits are very effective in suppressing voltage ripples close to the comparator threshold, practical applications are subject to varying intensities of interference on the input signal. Stronger disturbances may cause the input level to momentarily raise beyond the threshold range of the hysteresis circuit and cannot be effectively suppressed. the LGT8FX8P incorporates a programmable digital filter at the comparator output to filter out the effects of transient disturbances on the comparator output. The digital filter can select a suitable filter time width according to the application requirements. Only when the comparator output steadily and continuously meets the filter time limit, the filter circuit updates the comparator output. Thus, a more stable output result is achieved.



Comparator Output Filter Timing

The digital filtering of AC0 is controlled by the C0FEN and C0FS bits of the C0XR registeras set in the Register Definition section of this chapter.

Comparator Outputs and *PWM* Control

The LGT8FX8P supports multi-channel PWM outputs, and the PWM signals can be used in conjunction with a comparator module. The output of the comparator can be used to directly turn off the PWM signal, thus enabling a more flexible PWM protection scheme.

For controls related to PWM output, refer to the relevant section in the Timer chapter.

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	R/W	R/W	R/W	R/W	R/W

Bit	Name	description
7	C0D	Analog comparator disable bit. When the C0D bit is set to "1", the analog comparator is turned off. When the C0D b i t i s set to "0", the analog comparator is turned on.
6	C0BG	The C0BG sets the AC0 positive input source in conjunction with the C0PS0 bit of the C0XR register, {C0BG, C0PS0} = 00 = AC0P as positive input 01 = ACXP as positive input 10 = Output of internal DAC as positive input 11 = Turn off the positive input source of AC0
5	C0O	Analog comparator output status bits. The output of the analog comparator is synchronized and connected directly to the C0O bit. The software can read C0O The value of the bit to get the output value of the analog comparator.
4	C0I	Analog comparator interrupt flag bit. The C0I bit is set when an analog comparator output event triggers the interrupt mode defined by the C0IS bit. The interrupt is generated when the interrupt enable bit C0IE is "1" and the global interrupt is set. C0I will be cleared automatically when the analog comparator interrupt service routine is executed, or by writing a "1" to the C0I bit.
3	C0IE	Analog comparator's interrupt enable bit. When the C0IE bit is set to 1 and the global interrupt is enabled, the interrupt of AC0 is enabled. When the C0IE bit is set to 0, interrupts for AC0 are disabled.
2	C0IC	Analog comparator input capture enable bit C0IC = 1, the input capture source for Timing Counter 1 is from the output of the analog comparator. C0IC = 0, the input capture source for Timer 1 is from the external pin ICP1.
1	C0IS1	Analog comparator interrupt mode control high.
0	C0IS0	The analog comparator interrupt mode control low. c0is0 and c0is1 together form c0is[1:0], which is used to control the analog comparator interrupt trigger mode.

C0IS[1:0]	interrupt mode
00	Rising or falling edge triggering of the ACO
01	Reserved.

		10	Falling edge triggering of the ACO
		11	Rising edge triggering of the ACO

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	CME01	CME00	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	description
7	CME01	AC0 Negative input selection, CME0 = {CME01, CME00} 00: External port ACXN as AC0 negative input 01: ADC multiplexed output as AC0 negative input 10: Differential amplifier output as AC0 negative input 11: Turn off the negative input source of AC0
6	CME00	
5	CME11	AC1 Negative input selection, CME1 = {CME11, CME10} 00: External port ACXN as AC1 negative input 01: External port AC1N as AC1 negative input 10: ADC internal 1/5 divider as AC1 negative input 11: The output of the differential op-amp is used as the negative input of AC1
4	CME10	
3	ACHS	AC Trigger Source Channel Selection 0 - AC0 output as ADC auto-conversion trigger source 1 - AC1 output as ADC auto-conversion trigger source
2:0	ADTS	See ADC register description.

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	description
7	-	retain
6	C0OE	AC0 Comparator output to external port enable control C0OE = 1, Comparator output from AC0 to external port PD2 C0OE = 0, Disable comparator output to external port
5	C0HYSE	AC0 Output Hysteresis Function Enable Control. 1 = Enable output hysteresis 0 = Disable output hysteresis
4	C0PS0	AC0 Positive input source select low. C0PS0 and C0BG together control the positive input source of AC0, please refer to C0SR register definition

3	COWKE	AC0 Enable control for hibernation wake-up. 1 = Enable wake-up for comparator output

		0 = Turn off wake-up for comparator output
2	C0FEN	Comparator digital filter enable control. 1 = Enables digital filter 0 = Disable digital filter
1:0	C0FS[1:0]	Comparator digital filter width setting 00 = Closed 01 = 32us 10 = 64us 11 = 96us

Analog Comparator 1 (AC1)

- Comparison accuracy of 10mV
- Factory out-of-tune calibration
- Supports 4 off-chip analog inputs
- Supports internal 1/5 voltage divider input (VDO)
- Internal differential amplifier input (DFFO) support
- Supports internal 8-bit DAC input (DAO)
- Programmable output filtering control

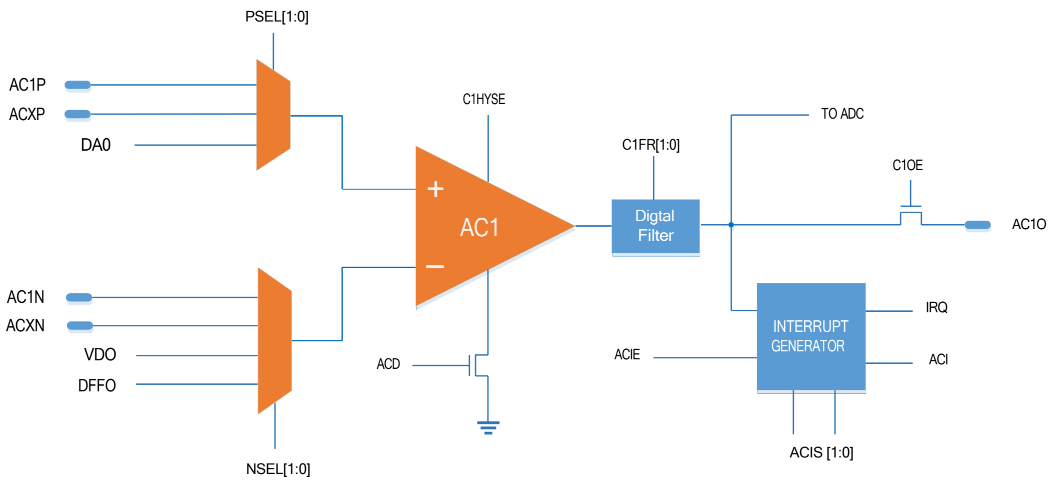
a general narrative

The analog comparator compares the levels of the positive and negative terminals of the input comparator, and the output ACO of the analog comparator is set when the voltage at the positive terminal is higher than the voltage at the negative terminal. When the level of ACO changes, the edge of the signal can be used to trigger an interrupt. The output signal ACO can also be used to trigger the input capture of Timer Counter 1 and to control the PWM output generated by the timer.

The LGT8FX8P integrated analog comparator, AC1, includes a multiple analog input selector, and the comparator positive and negative input sources can be selected from an external port or from a variety of internally generated reference sources. The analog comparator itself supports out-of-tune calibration, which ensures consistent comparator operation. The comparator supports an optional hardware hysteresis function to improve the stability of the comparator output. A hardware programmable digital filter is integrated into the comparator output, allowing you to select the appropriate filter setting for a more stable comparison output, depending on the application requirements.

The comparator output state can be read directly from registers, or interrupt requests can be generated for more efficient real-time event capture. The comparator output can also be output directly to an external IO port.

The structure of analog comparator 1 is shown in the figure below.



Analog Comparator 1 Module Structure Schematic

Analog comparator input

Both inputs of the analog comparator support a variety of selectable input sources. The positive end has three selectable inputs.

1. External independent analog input AC1P
2. Analog Comparator 0/1 Common Analog Input ACXP

3. Output DAO for internal 8-bit DAC

The selection of the input source is controlled by the C1BG bit in the Control Status Register C1SR and the C1PS0 bit in the C1XR register, as described in the Register Description section of this chapter.

AC1P is the dedicated positive mode input channel for AC1.

The ACXP is a common positive input for comparator 0/1. The LGT8FX8P has two internal analog comparators, and the ACXP is connected to the positive multiplexer of both comparators at the same time to facilitate the cooperative operation of both comparators.

The DAO is derived from the output of the internal 8-bit DAC.The reference source for the DAC can be selected from the system power supply, an internal reference, or an input from an external reference.Refer to the DAC related section for DAC configuration.

C1BG	C1PS0	AC1 Positive input
0	0	ACXP
0	1	AC1P
1	0	DAO
1	1	Close the positive input channel of the comparator

The negative input can also be selected from 4 different analog inputs:

1. External analog input AC1N as AC1 negative input
2. Comparator 0/1 Common Negative Input ACXN
3. ADC internal 1/5 divider output as negative input to AC1
4. Internal differential amplifier output DFFO as negative input to AC1

Comparator negative input channel selection is controlled by bit CME11/10 in the ADCSRB register from the ADC module. When the comparator negative input is selected as the ADC internal multiplexer output, the input reference source for the multiplexer needs to be selected via the ADCSRC register VDS bit of the ADC module.

ACXN is a common negative input for comparator 0/1, facilitating the implementation of comparator 0/1 co operation.

The DFFO comes from the internal differential amplifier outputDifferential amplifier with optional x1/x8/x16/x32 gain control enables detection and measurement of small signals.

CME11	CME10	AC1 Negative input
0	0	ACXN
0	1	AC1N
1	0	VDO
1	1	DFFO

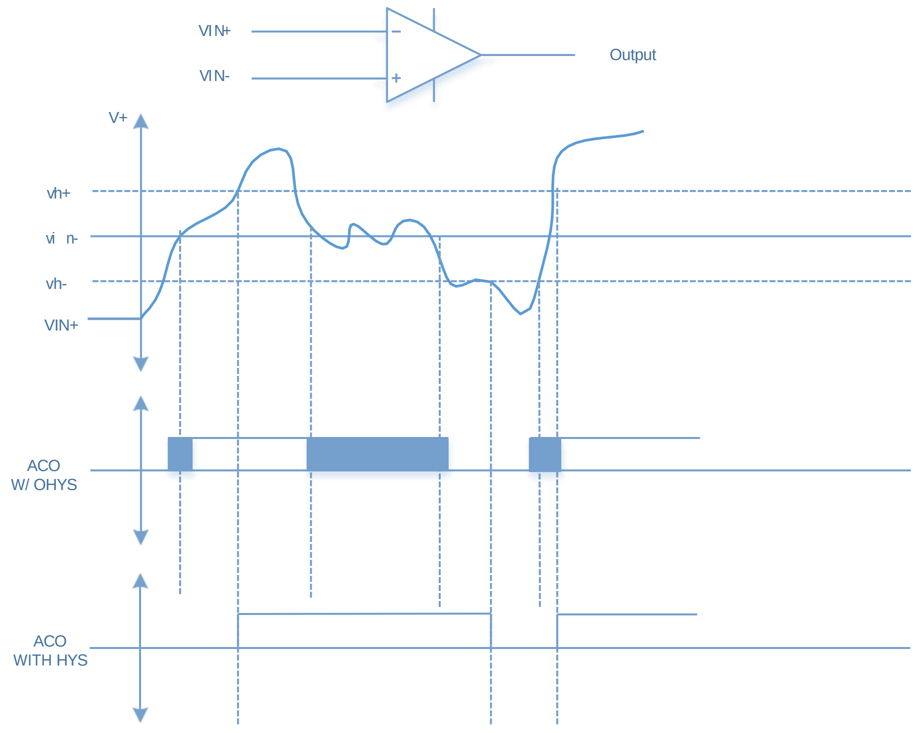
Comparator output filtering

A controlled hysteresis is supported internally at the comparator output. The user can enable the hysteresis circuit via the C1HYSE bit of the C1XR register. The hysteresis circuit can eliminate the instability of the comparator state change process and achieve the output filtering function.

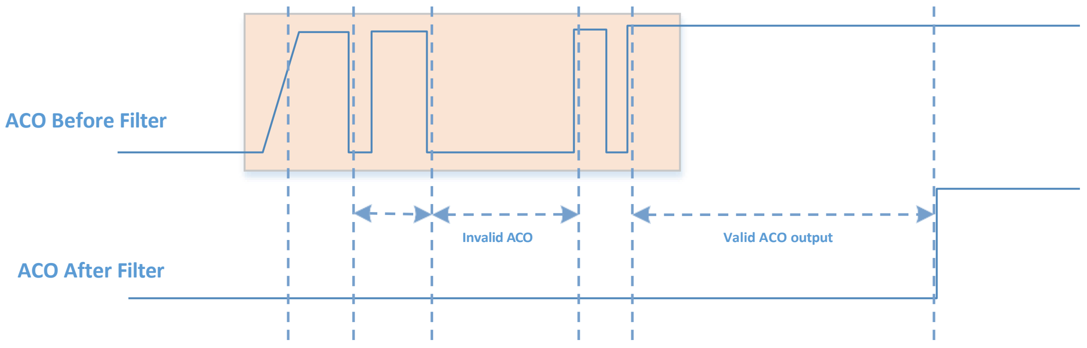
It is recommended that the user turn on the hysteresis circuit when using the comparator to obtain a stable comparator output.

As shown in the figure below, the hysteresis circuit is located between the analog output and the digital output of the comparator. When the input voltage at the positive end of the comparator, VIN+, is greater than (VIN- + VH+), the comparator COUT output is high; when the VIN+ voltage is less than (VIN- - VH-), the comparator output is low. The hysteresis circuit avoids the jitter introduced by the circuit itself when the voltage at the positive end of the comparator is close to the voltage at the negative end.

Diagram of comparator hysteresis voltage versus comparator output.



Although hysteresis circuits are very effective in suppressing voltage ripples close to the comparator threshold, practical applications are subject to varying intensities of interference on the input signal. Stronger disturbances may cause the input level to momentarily raise beyond the threshold range of the hysteresis circuit and cannot be effectively suppressed. the LGT8FX8P incorporates a programmable digital filter at the comparator output to filter out the effects of transient disturbances on the comparator output. The digital filter can select a suitable filter time width according to the application requirements. Only when the comparator output steadily and continuously meets the filter time limit, the filter circuit updates the comparator output. Thus, a more stable output result is achieved.



Comparator Output Filter Timing

The digital filtering of AC1 is controlled by the C0FEN and C1FS bits of the C1XR registeras set in the Register Definitions section of this chapter.

Comparator Outputs and *PWM* Control

The LGT8FX8P supports multi-channel PWM outputs, and the PWM signals can be used in conjunction with a comparator module. The output of the comparator can be used to directly turn off the PWM signal, thus enabling a more flexible PWM protection scheme.

For controls related to PWM output, refer to the relevant section in the Timer chapter.

Register Definition

C1SR - 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	R/W	R/W	R/W	R/W	R/W

Bit	Name	description
7	C1D	Analog comparator disable bit. When the C1D bit is set to "1", the analog comparator is turned off. When the C1D bit is set to "0", the analog comparator is turned on.
6	C1BG	C1BG sets the AC1 positive input source in conjunction with the C1PS0 bit of the C1XR register, {C1BG, C1PS0} = 00 = ACXP as positive input 01 = AC1P as positive input 10 = Output of internal DAC as positive input 11 = Turn off the positive input source of AC1
5	C1O	Analog comparator output status bits. The output of the analog comparator is synchronized and connected directly to the C1O bit. The software can read C1O The value of the bit to get the output value of the analog comparator.
4	C1I	Analog comparator interrupt flag bit. The C1I bit is set when an analog comparator output event triggers the interrupt mode defined by the C1IS bit. The interrupt is generated when the interrupt enable bit C1IE is "1" and the global interrupt is set. C1I will be cleared automatically when the analog comparator interrupt service routine is executed, or by writing a "1" to the C1I bit.
3	C1IE	Analog comparator's interrupt enable bit. When the C1IE bit is set to 1 and global interrupts are enabled, AC1's interrupts are enabled. When the C1IE bit is set to 0, interrupts for AC1 are disabled.
2	C1IC	Analog comparator input capture enable bit C1IC = 1, the input capture source for Timing Counter 1 is from the output of the analog comparator. C1IC = 0, the input capture source for Timer 1 is from the external pin ICP1.
1	C1IS1	Analog comparator interrupt mode control high.
0	C1IS0	Analog comparator interrupt mode control low.C1IS0 and C1IS1 together form C1PS[1:0], which is used to control the interrupt triggering mode of the analog comparator.

C1IS[1:0]	interrupt mode
00	Triggered by rising or falling edge of AC1
01	Reserved.

		10	Falling edge trigger of AC1
		11	Rising edge trigger 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	CME01	CME00	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	description
7	CME01	AC0 Negative input selection, CME0 = {CME01, CME00} 00: External port ACXN as AC0 negative input 01: ADC multiplexed output as AC0 negative input 10: Differential amplifier output as AC0 negative input 11: Turn off the negative input source of AC0
6	CME00	
5	CME11	AC1 Negative input selection, CME1 = {CME11, CME10} 00: External port ACXN as AC1 negative input 01: External port AC1N as AC1 negative input 10: ADC internal 1/5 divider as AC1 negative input 11: The output of the differential op-amp is used as the negative input of AC1
4	CME10	
3	ACHS	AC Trigger Source Channel Selection 0 - AC0 output as ADC auto-conversion trigger source 1 - AC1 output as ADC auto-conversion trigger source
2:0	ADTS	See ADC register description.

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	description
7	-	retain
6	C1OE	AC1 Comparator output to external port enable control C1OE = 1, Comparator output of AC1 to external port PE5 C1OE = 0, Disable comparator output to external port
5	C1HYSE	AC1 Output Hysteresis Function Enable Control. 1 = Enable output hysteresis 0 = Disable output hysteresis
4	C1PS0	AC1 Positive input source select low. C1PS0 and C1BG together control the positive input source of AC1, please refer to C1SR register definition

3	C1WKE	AC1 Enable control for hibernation wake-up. 1 = Enable wake-up for comparator output

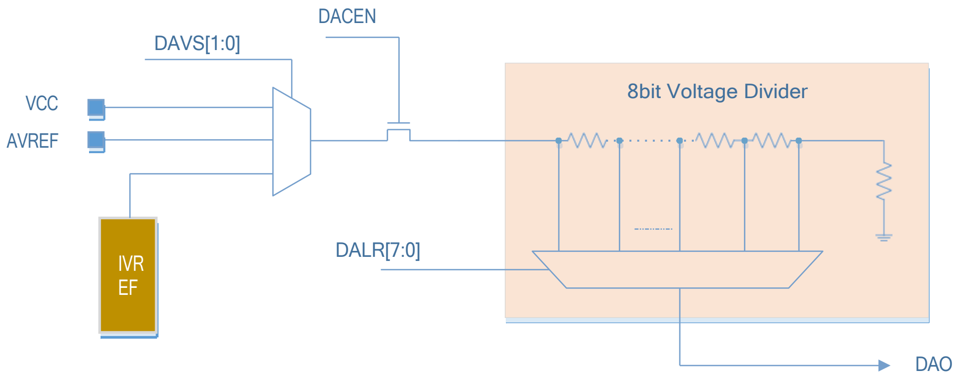
		0 = Turn off wake-up for comparator output
2	C1FEN	Comparator digital filter enable control. 1 = Enables digital filter 0 = Disable digital filter
1:0	C1FS[1:0]	Comparator digital filter width setting 00 = Closed 01 = 32us 10 = 64us 11 = 96us

Digital-to-analog converters *(DACs)*

- 8-bit digital-to-analog output
- DAC output can be used as an analog comparator reference input
- Supports DAC output to external port (DAO)
- Optional VCC/AVREF/IVREF voltage divider power supply

a general narrative

The DAC's reference power input can be selected from the system operating power supply, the internal reference voltage source or from the chip's external port AVREF input.The DAC's output can optionally be used as the input source for the internal comparator AC0/1 or directly to the chip's external pins for external reference use. When the DAC is output to an external pin, it cannot be used to drive the load directly and needs to be driven through a voltage follower or other similar drive circuit.The DAC internal structure is shown in the following figure.



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	W/R	R/W	W/R
Bit	Name	description						
7:4	-	retain						
3	DACEN	DAC Enable Control Bit 1 = Enables the DAC module 0 = Disable DAC module						
2	DAOE	DAC output to external port enable control 1 = Enables DAC output to external PD4 0 = Disable DAC output to external port						
1	DAVS1	DAC Reference voltage source selection bit 1						
0	DAVS0	DAC reference voltage source select bit 0. [DVS1, DVS0] =						

		00 : Voltage source selection system operating voltage VCC 01 : Voltage source selected as external input AVREF 10 : Voltage source selected as internal reference voltage 11 : Turning off the DAC reference source will also turn off the DAC module

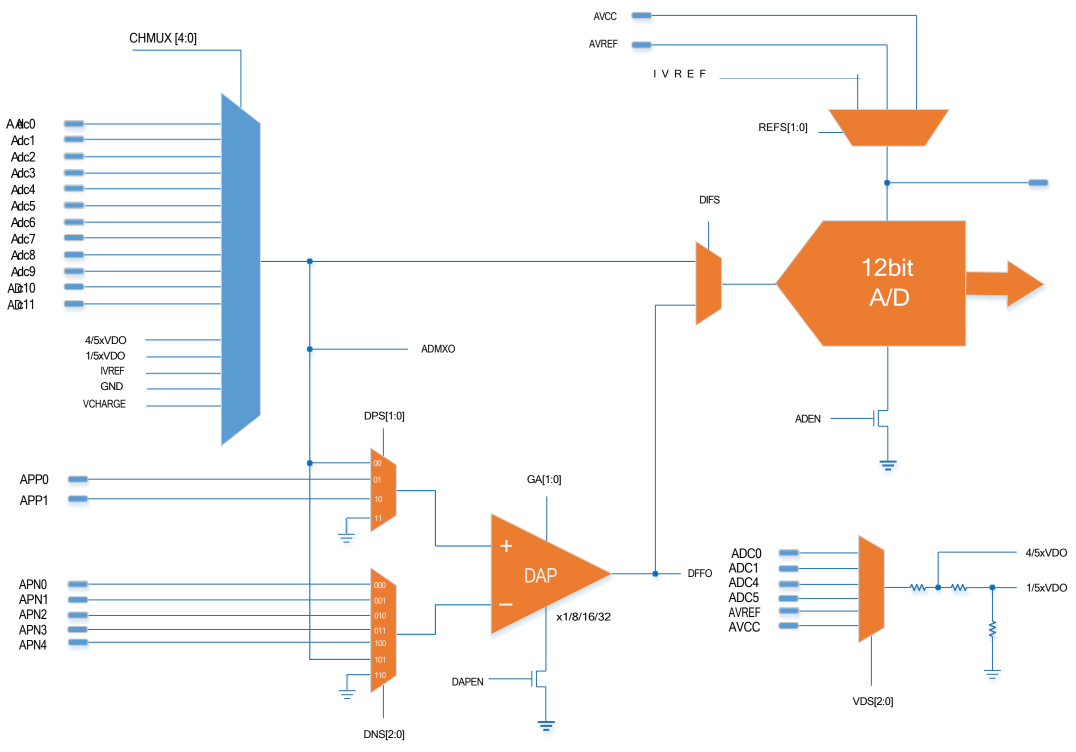
DALR - DAC Data Register

VRCON1- DAC1 Control Register								
Address: 0xA1					0000_0000			
Bit	7	6	5	4	3	2	1	0
	DALR[7:0]							
R/W	W/R							
Bit	Name	description						
7:0	DALR	DAC data register, set DAC mode output voltage level DAC output voltage versus DALR. VDAO = VREF*(DALR + 1)/256 Of which. VDAO is the DAC output analog voltage VREF is the DAC reference voltage source, selected by the DAVS bit of the DACON register						

12-bit analog-to-digital converter *(ADC)*

- 12-bit resolution, ±1LSB for DNL, ±1.5LSB for INL
- Sample rates up to 500KSPS at highest resolution
- 12 multiplexed single-ended input channels
- Multiple input programmable gain differential amplifier channels
- Input voltage range 0-VCC
- Internal 1.024V/2.048V/4.096V reference voltage
- Supports AVCC and external reference voltage inputs
- Internal multi-input 1/5, 4/5 voltage divider circuit
- Support for out-of-phase calibration in positive and negative directions
- Interrupt source-based automatic start transition trigger mode
- Automatic channel monitoring with up/down overflow support
- Conversion results support optional alignment modes
- End of conversion interrupt request

summarize



ADC Structure Diagram

The ADC is a 12-bit successive approximation ADC connected to a 17-channel analog multiplexer that samples and converts 12 analog inputs from the chip's external port and 5 internal voltage sources. The amplifier input can be derived from an external port or from the output of an ADC multiplexer. The result of the differential op-amp can be used as an analog input to the ADC.

The ADC's internal analog input sources include a multiplexed input divider from within the ADC; an internal reference voltage source; an internal analog reference ground; and an analog output from the Touch Button Module. The internal multi-input divider outputs both 4/5, 1/5 voltage; the input to the voltage divider can be selected from the level of an external port or from the system power supply.

The ADC supports out-of-tune calibration. The process of out-of-tune calibration is controlled by software. The out-of-tune calibration includes calibration amounts in both positive and negative directions. When out-of-tune calibration is enabled, the ADC controller will automatically calibrate the ADC sample results using both positive and negative calibration values.

Refer to the relevant section of this chapter for the out-of-tune calibration method.

ADC operation

The ADC converts the input analog voltage to a 12-bit digital quantity by successive approximation. The minimum value represents GND and the maximum value represents the reference voltage minus 1LSB. the reference voltage source can be the ADC's supply voltage AVCC, the external reference voltage AVREF or the internal 1.024V/2.048V reference voltage, selected by writing the REFS bit of the ADMUX register.

The analog input channel can be selected by writing the CHMUX bit of the ADMUX register. Any of the ADC's input pins, external reference voltage pins, and internal reference voltage sources can be used as single-ended inputs to the ADC. Input channel of the ADC can be switched to the internal differential amplifier by setting the DIFS of the ADTMR register. The differential amplifier-related input source and gain can be set via the DAPCR register.

The ADC can be started by setting the ADEN bit in the ADCSRA register. The ADC does not consume power when ADEN is cleared, so it is recommended that the ADC be turned off before entering sleep mode.

The ADC conversion result is 12 bits and is stored in the ADC data registers ADCH and ADCL. By default, the conversion result is right-aligned, but can be changed to left-aligned by setting the ADLAR bit of the ADMUX register.

If the conversion result is set to be left-aligned and only 8 bits of conversion accuracy is required at most, then simply reading ADCH is sufficient. Otherwise, read ADCL first and then ADCH to ensure that the contents of the data registers are the result of the same conversion. Once ADCL is read, the data registers ADCL and ADCH are latched and the conversion result can be updated to the data registers ADCL and ADCH after ADCH is read.

An interrupt can be triggered by the end of an ADC conversion. The interrupt will be triggered even if the end of conversion occurs between reading ADCL and ADCH.

Initiate a conversion

WRITING A "1" TO THE ADC START CONVERSION BIT, ADSC, INITIATES A SINGLE CONVERSION.THIS BIT REMAINS HIGH DURING THE CONVERSION UNTIL IT IS CLEARED BY HARDWARE AT THE END OF THE CONVERSION. IF A CHANNEL IS CHANGED DURING THE CONVERSION, THE ADC WILL COMPLETE THE CONVERSION BEFORE THE CHANNEL IS CHANGED.

ADC conversions have different trigger sources. Setting the ADC Auto Trigger Allow bit ADATE of the ADCSRA register enables auto trigger. Setting the ADC Trigger Select bit ADTS of the ADCSRB register selects the trigger source. When the selected trigger signal generates a rising edge, the ADC prescaler resets and starts conversion. This provides a way to start conversion at a fixed time interval. A new conversion will not be initiated after the conversion is completed even if the trigger signal is still present. If the trigger signal generates another rising edge during the conversion, this rising edge will also be ignored. Even if a specific interrupt is disabled or the global interrupt enable bit is "0", its interrupt flag will still be set. This allows a transition to be triggered without generating an interrupt. However, in order to trigger a new transition on the next interrupt event, the interrupt flag must be cleared to zero.

Using the ADC interrupt flag as a trigger source, the next ADC conversion can be started as soon as the current conversion is completed. The ADC then operates in continuous conversion mode, continuously sampling and updating the ADC data registers. The first conversion

The conversion is initiated by writing a "1" to the ADSC bit of the ADCSRA register. In this mode, subsequent ADC conversions do not depend on the
Whether the ADC interrupt flag ADIF is set.

The ADSC flag can also be used to detect if a conversion is in progress. Regardless of how the conversion is initiated, ADSC remains "1" during the conversion.

Prescaling and *ADC* Conversion Timing

Under default conditions, the successive approximation circuit requires an input clock from 300KHz to 3MHz for maximum accuracy. If the required conversion accuracy is less than 12 bits, the input clock frequency can be higher than 3MHz to achieve higher sampling rates.

The ADC module includes a prescaler that generates an acceptable ADC input clock from the system clock. The prescaler is set via the ADPS bit of the ADCSRA register. Setting the ADEN of the ADCSRA register will enable the ADC and the prescaler will begin counting. As long as the ADEN bit is "1", the prescaler will continue to count until ADEN is cleared.

When ADSC in the ADCSRA register is set, single-ended conversion starts on the rising edge of the next ADC clock cycle. Normal conversion requires 15 ADC clock cycles. 50 ADC input clock cycles are required to initialize the analog circuitry after ADC enable (ADEN of the ADCSRA register is set) before the first conversion is valid.

During ADC conversion, sample hold begins 1.5 ADC input clocks after conversion start,and the first ADC conversion result output occurs 14.5 ADC input clocks after start.At the end of the conversion, the ADC result is fed into the ADC data register and the ADIF flag bit is set, while ADSC is cleared to zero. The software can then reset the ADSC flag again or trigger it automatically to start a new conversion.

Sampling Channel and Reference Voltage

The MUX and REFS in the ADMUX register are single-buffered via temporary registers. the CPU can perform random access to the temporary registers. The CPU can configure the channel and reference source selection at any time before the conversion starts. To ensure that the ADC has sufficient sampling time, configuration of the channel and reference source selections is not allowed once the conversion has started. The channel and reference source selections are not updated until after the conversion is complete (ADIF in the ADCSRA register is set). The start of conversion is at the rising edge of the next ADC input clock after ADSC is set. Therefore, it is recommended that the user does not operate ADMUX to select the new channel and reference source for one ADC input clock cycle after setting ADSC.

When using auto-trigger, the timing of the trigger event is uncertain. To control the effect of the new setting on the conversion, special care should be taken when updating the ADMUX register. If both ADATE and ADEN are set, the interrupt time can occur at any point, thus automatically triggering and starting the ADC conversion. If the contents of the ADMUX register are changed during this time, then the user will not be able to tell if the next conversion is based on the old or new configuration. It is recommended that the user update the ADMUX at the following safe times.

1) (a) The ADATE or ADEN bit is "0".
2) during conversion, but at least one ADC input clock cycle after the trigger event occurs.
3) After the conversion is completed, but before the interrupt flag of the trigger source is cleared.

If you update ADMUX in either of the cases mentioned above, the new configuration will take effect before the next

conversion.
Care must be taken when selecting the ADC input channel, by selecting the channel before starting the conversion, a new analog input channel can be selected after one ADC clock cycle after the ADSC is set, but it is easiest to wait until the conversion is complete before changing the channel.

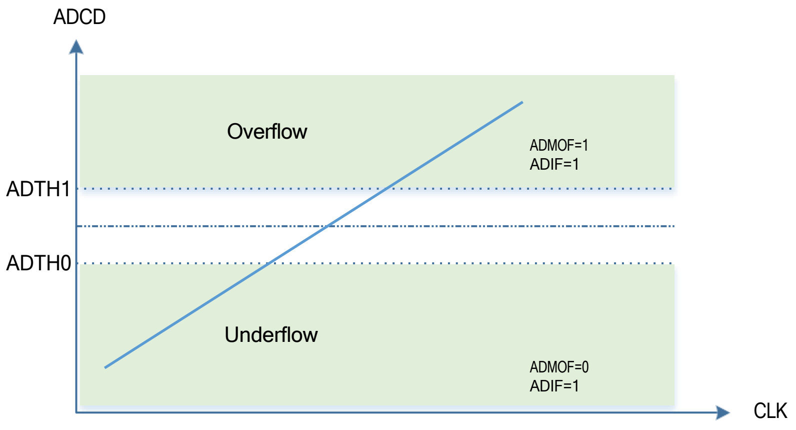
The ADC's reference voltage source, Vref, reflects the ADC's conversion range. If the single-ended channel level exceeds Vref, the conversion result will be close to the maximum value of 0xFFF. Vref can be AVCC, the voltage of the external AREF pin, the internal reference voltage source.

Using the internal reference (1.024V/2.048V/4.096V) Caution.

When the chip is powered on, the internal reference is calibrated to 1.024V by default, so if the user uses the 1.024V internal reference, he can use it directly without any other operation. The calibration value of 2.048V/4.096V is loaded into VCAL2/3 (0xCE/0xCC) after power-on, and the value of VCAL2/3 is read and written to VCAL (0XC8) during program initialization to complete the calibrationCalibration.

Automatic channel monitoring

Automatic channel monitoring mode is used to monitor the voltage change of the selected ADC input channel in real time. The software enables automatic channel monitoring by setting the AMEN bit in the ADCSRC register. The ADC automatically converts the voltage of the selected channel, and when the result is outside the given overflow range, the ADC interrupt flag bit (ADIF) is set and automatic monitoring is stopped. The AMOF bit in the ADMSC register is used to indicate the type of overflow event.The ADIF flag bit is automatically cleared by hardware after a reset in response to an interrupt; in query mode, it can be cleared by a software write 1.The automatic monitoring mode can be re-enabled only when ADIF is cleared and by setting the AMEN bit of the ADCSRC register.



To overcome the instability of single ADC conversion results, auto-detection supports a configurable digital filtering feature. Digital filtering is performed by detecting successive conversion results and triggering an overflow event only if a consistent result is obtained for a limited number of successive conversions. The number of successive conversions can be set via the AMFC[3:0] bits of the ADMSC register.

The automatic channel monitoring function is controlled through the AMEN bit of the ADCSRC register. Register ADT0 is used to set the lower overflow threshold; ADT1 is used to set the upper overflow threshold; ADT0/1 is a 16-bit register. When the AMEN bit is set by the software, the current conversion of the ADC will be stopped immediately and the ADC control state will be reset, after which the ADC will enter the automatic conversion mode.

Before starting the automatic channel detection mode, the channels to be detected and other related configurations need to be set. The software can disable the automatic detection mode at any time by clearing the AMEN register.

Multiple Input Voltage Divider Circuit *(VDS)*

The ADC contains an internal voltage divider module with multiple inputs. The divider input voltage source can be optionally sourced from an external ADC input channel

(ADC0/1/4/5), an external reference AVREF, or an analog operating power supply. The voltage divider module outputs both 4/5 and 1/5 voltages to the internal 12 and 13 input channels of the ADC respectively. The 4/5 is mostly used for ADC out-of-range calibration; the 1/5 is used for supply voltage detection and similar applications in addition to internal out-of-range calibration. The voltage divider circuit is mainly controlled by the ADCSRD register.

ADC Out-of-Sync Calibration

Due to manufacturing process deviations and the inherent characteristics of the circuit structure, the comparator circuitry within the ADC can have varying degrees of detuning errors. Therefore, compensation of the detuned voltage is critical to produce a highly accurate ADC conversion structure. The ADC inside the LGT8FX8P chip supports an interface for the offset voltage test, which can be used to measure and calibrate the offset with the cooperation of software.

Principles of out-of-phase calibration.

Out-of-tune calibration is mainly done by changing the input polarity of the internal comparator and testing the ADC conversion results in both positive and negative directions. Since the out-of-tune voltages in both directions are also expressed as two polarities, an intermediate out-of-tune error value can be obtained by subtracting the results of these two conversions. In normal applications, the conversion result can be adjusted accordingly according to this detuning voltage.

Out of tune calibration process.

1. Configure the VDS module to select the VDS input source as the analog power supply (AVCC)
2. The reference voltage for the ADC is selected as the analog power supply (AVCC)
3. ADCSRC[SPN] = 0, ADC reads 4/5VDO channels, conversion value recorded as PVAL
4. ADCSRC[SPN] = 1, ADC reads 4/5VDO channels, conversion value recording bit NVAL
5. Store the value (NVAL - PVAL) >> 1 to the OFR0 register
6. ADCSRC[SPN] = 1, ADC reads 1/5VDO channel, conversion result is recorded as NVAL
7. ADCSRC[SPN] = 0, ADC reads 1/5VDO channel, conversion result recorded bit PVAL
8. Store the value (NVAL - PVAL) >> 1 to the OFR1 register
9. Set ADCSRC[OFEN]=1 to enable the out-of-tune compensation function

Special note: Since the misalignment error has positive and negative directions, the above data and operations are signed operations.

The out-of-tune calibration process requires changes to the ADC-related configuration, so it is recommended that the out-of-tune calibration be completed prior to the normally used configuration. To improve calibration accuracy, it is recommended that the ADC samples multiple filters when reading channel conversions.

After configuration of the out-of-tune calibration OFR0/1 is completed, automatic out-of-tune compensation is enabled via the OFEN bit. After subsequent normal conversions, the ADC control will automatically compensate using OFR0/1 based on ADC conversion results.

ADC Dynamic Calibration

The out-of-tune calibration method described above is based on out-of-tune in a test environment and with test inputs. When the system environment changes, the ADC's detuning will also change. Therefore, if real-time calibration compensation can be implemented, it is very important to overcome the performance difference of the device with the change of operating environment and improve the ADC measurement accuracy.

A suggested algorithm is provided here, based on the principles of the out-of-tune calibration algorithm, which enables dynamic compensation of the out-of-tune error due to the operating environment and obtains consistent and accurate test results.

This method does not require the calculation of the detuning voltage and does not enable the detuning compensation (OFEN). The algorithm only needs to control the polarity of the ADC conversion by SPN and sample two measurements at different SPNs.

We assume that the test error introduced by the detuning is VOFS when the ADC is converted, and therefore control the SPN to perform two consecutive ADC conversion, the resulting ADC conversion result can be expressed as

When SPN = 1, $V_{ADC1} = V_{REL} + V_{OFS1}$

When SPN = 0, $V_{ADC0} = V_{REL} - V_{OFS0}$

By adding the two measurements, we can eliminate the effect of VOFS on the actual sampled input VREL. Due to the matching characteristics of the circuit, VOFS1 and VOFS0 may not be identical,but the overall effect of compensating for the misalignment error is still achieved.

<u>Flow of dynamic dissonance compensation algorithm.</u>

1. Initialize ADC conversion parameters as required by the application

2. Set SPN=1 to start ADC sampling and record the ADC sampling result as VADC1
3. Set SPN=0 to start ADC sampling and record the ADC sampling result as VADC2
4. (VADC1 + VADC2) >> 1 is the result of this ADC conversion

In practice, this algorithm can be combined with the sampling average algorithm, which can give more desirable results.

Register Definition

ADC Register List

processor register	address	default value	description
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 multiplexer control register
ADCSRC	0x7D	0x01	ADC Control and Status Registers C
DIDR0	0x7E	0x00	Digital input disable control register 0
DIDR1	0x7F	0x00	Digital input disable control register 0
DAPCR	0xDC	0x00	Differential Amplifier Control Register
OFR0	0xA3	0x00	Out-of-tune compensation register 0
OFR1	0xA4	0x00	Out-of-tune compensation register 1
ADT0L	0xA5	0x00	Automatic monitoring of the lower 8 positions of the relief valve value
ADT0H	0xA6	0x00	Automatic monitoring of underflow valve value 8 digits higher
ADT1L	0xAA	0x00	Automatic monitoring of the upper relief valve value 8 positions lower
ADT1H	0xAB	0x00	Automatic monitoring of the upper relief valve value 8 positions higher
ADMSC	0xAC	0x01	Automatic monitoring of status and control registers
ADCSRD	0xAD	0x00	ADC control and status registers 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	description
7:0	ADC[7:0]/ ADC[3:0]	ADC data low byte register. When the ADLAR bit is "0", the ADC output data is stored in the register according to the low bit alignment, i.e. ADCL is ADC[7:0], as shown in Name0. When the ADLAR bit is "1", the ADC output data is stored in the register according to the high bit alignment, i.e. the high 4 bits of ADCL is ADC[3:0], and the low 4 bits are meaningless as shown in Name1.

ADCH - ADC Data High Byte Register

ADCH - 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	description
7:0	ADC[11:8]/ ADC[11:4]	ADC data low byte register. When the ADLAR bit is "0", the ADC output data is stored in the register according to the low alignment, i.e. the low 4 bits of ADCH are ADC[11:8] and the high 4 bits are meaningless, as shown in Name0. When the ADLAR bit is "1", the ADC output data in the register is aligned with high bits, i.e. ADCH is ADC[11:4], as shown in Name1.

ADCSRA - ADC Control and Status Register A

ADCSRA - ADC Control and Status Register A								
Address: 0x7A					Default value: 0x05			
Bit	7	6	5	4	3	2	1	0
Name	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	1	0

Bit	Name	description
7	ADEN	ADC enable control bit. When the ADEN bit is set to "1", the ADC is enabled. When the ADEN bit is set to "0", the ADC is disabled.
6	ADSC	The ADC starts the conversion. In single conversion mode, the ADSC bit will initiate one conversion. In continuous conversion mode, the ADSC bit will initiate the first conversion.
5	ADATE	ADC auto trigger enable control bit. When the ADATE bit is set to "1", the auto trigger function is enabled. The rising edge of the selected trigger signal turns on a conversion. The selection of the trigger source is controlled by the ADTS of the ADCSRB register. When the ADATE bit is set to "0", the automatic trigger function is disabled.

4	ADIF	ADC interrupt flag bit. If the ADC interrupt enable bit ADIE is "1" and the global interrupt is set, the ADC interrupt is generated. Execute the ADC The interrupt will clear the ADIF bit, or you can write a "1" to the bit to clear it.
3	ADIE	ADC interrupt enable control bit. When the ADIE bit is set to "1" and the global interrupt is set, the ADC interrupt is enabled. When the ADIE bit is set to "0", the ADC interrupt is disabled.

2:0	ADPS [2:0]	ADC prescaler selection control bit. ADPS selects the prescaling factor for the system clock to generate the ADC clock.	
		ADPS [2:0]	**prescaling 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
Name	ACME01	ACME00	ACME11	ACME10	ACTS	ADTS2	ADTS1	ADTS0
R/W	R/W	R/W	R/W	R/W	W/O	R/W	R/W	R/W
Initial	0	0	0	0	0	0	0	0

Bit	Name	description
7 6	ACME01 ACME00	Comparator 0 Negative input selection 00: Negative end select external input ACIN0 01: Negative terminal selection ADC multiplexed output 1X: Negative side selects the output of op-amp 0
5 4	ACME11 ACME10	Comparator 1 Negative Input Selection 00: Negative end select external input ACIN2 01: Negative terminal selection ADC multiplexed output 1X: Negative terminal selects the output of op amp 1
3	ACTS	AC Trigger Source Channel Selection 0 - AC0 output as ADC auto-conversion trigger source 1 - AC1 output as ADC auto-conversion trigger source
2:0	ADTS[2:0]	ADC auto trigger source selection control bit. When the ADATE bit is set to "1", the auto trigger function is enabled and selection of the trigger source is controlled by ADTS. When the ADATE bit is set to "0", the setting of ADTS is disabled. The rising edge of the selected trigger signal interrupt flag turns on a transition. When switching from a trigger source with the interrupt flag cleared to a trigger source with the interrupt flag set, the trigger signal generates a rising edge, and if ADEN is set, the ADC will also turn on a conversion. When switching to continuous conversion mode (ADTS=0), the auto-trigger function is disabled.
		ADTS[2:0] | **trigger source**

		0	Continuous conversion mode
		1	Comparator 0/1

		2	External interrupt 0
		3	Timing Counter 0 Compare Match
		4	Timing counter 0 Overflow
		5	Timing counter 1 Compare match B
		6	Timing counter 1 overflow
		7	Timing Counter 1 Input Capture Event

ADMUX - ADC Multiplexer Control Register

ADMUX - ADC Multiplexer Control Register								
Address: 0x7C				Default value: 0x00				
Bit	7	6	5	4	3	2	1	0
Name	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	description		
7:6	REFS[1:0]	Used in conjunction with REFS2 of the ADCSRD register to select the ADC's reference voltage source The reference voltage is selected by setting the REFS control bit, and if the REFS is changed during the conversion setting, the change will only take effect until after the current conversion has finished.		
		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	The conversion results in a left-aligned enable control bit. When the ADLAR bit is set to "1", the conversion result is left-aligned in the ADC data register. When the ADLAR bit is set to "0", the result of the conversion is the right pair in the ADC data register. Qi.		
4:0	CHMUX [4:0]	ADC input source selection control bit.		
		CHMUX [4:0]	Single-ended input source	description
		0_0000	PC0	External Port Input
		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 partial voltage circuit
		0_1000	1/5 VDO	
		0_1101	IVREF	Internal reference
		0_1111	AGND	simulated land
		1_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
Name	OFEN	-	SPN	AMEN	-	SPD	DIFS	ADTM
R/W	R/W	-	R/W	R/W	-	R/W	R/W	R/W

Bit	Name	description
7	OFEN	1=enables detuning compensation; 0=disables detuning compensation
6	-	Unimplemented
5	SPN	ADC conversion input polarity control, used only for the out-of-tune calibration process. Must be cleared during normal
4	AMEN	Automatic channel monitoring enablement. 1: Enables automatic channel monitoring 0: Disable channel auto monitoring function
3	-	Unimplemented
2	SPD	0=ADC low speed conversion mode 1=ADC high-speed conversion mode for low-impedance analog inputs only
1	DIFS	0 = ADC conversion from ADC multiplexer 1 = ADC conversion from internal differential amplifier
0	ADTM	Test mode, internal reference voltage 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
Name	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	description
7	PE3D	1=Disable PE3 digital input function
6	PE1D	1=Disable PE1 digital input function
5	PC5D	1=Disable PC5 digital input function

4	PC4D	1=Disable PC4 digital input function

3	PC3D	1=Disable PC3 digital input function
2	PC2D	1=Disable PC2 digital input function
1	PC1D	1=Disable PC1 digital input function
0	PC0D	1=Disable PC0 digital input function

DIDR1 - Digital Input Disable Control Register 1

DIDR1 - Digital Input Disable Control Register 1								
Address: 0x7F					Default value: 0x00			
Bit	7	6	5	4	3	2	1	0
Name	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	description
0	PD6D	1=Disable PD6 digital input function
1	PD7D	1=Disable PD7 digital input function
2	PC7D	1=Disable PC7 digital input function
3	PF0D	1=Disable PF0 digital input function
4	C0PD	1=Disable AC0P digital input function (LQFP48)
5	PE0D	1=Disable PE0 digital input function
6	PE6D	1=Disable PE6 digital input function
7	PE7D	1=Disable 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
Name	BGEN	REFS2	IVSEL1	IVSEL0	-	VDS2	VDS1	VDS0
R/W	R/W	R/W	R/W	R/W	-	R/W	R/W	R/W

Bit	Name	description
7	BGEN	Internal reference global enable control, 1=enable
6	REFS2	Refer to the definition of REFS in the ADMUX register for the reference voltage used to select the ADC conversion in combination with the REFS of the ADMUX register
5:4	IVSEL	When the ADC's reference voltage is selected as VCC or AVREF, IVSEL is used to control the output voltage of the internal reference. 00 = 1.024V 01 = 2.048V 1x = 4.096V
3	-	retain

2:0	VDS[2:0]	Voltage divider circuit input source selection 000/111 = Switch off the voltage divider circuit module 001 = ADC0 010 = ADC1 011 = ADC4

		100 = ADC5 101 = External reference input (AVREF) 110 = System power

DAPCR - Differential Op Amp Control Register

DAPCR - Differential Op Amp Control Register								
Address: 0xDC					Default value: 0x00			
Bit	7	6	5	4	3	2	1	0
Name	DAPEN	GA1	GA0	DNS2	DNS1	DNS0	DPS1	DPS0
R/W	W/R	W/R	W/R	W/R	W/R	R/W	R/W	R/W

Bit	Name	description
7	DAPEN	1 = Enables differential amplifier; 0 = Disables differential amplifier
6:5	GA[1:0]	Differential Amplifier Gain Control 00 = x1 01 = x8 10 = x16 11 = x32
4:2	DNS[2:0]	Input source selection bit for the reverse input of the differential amplifier 000 = ADC2/APN0 001 = ADC3/APN1 010 = ADC8/APN2 011 = ADC9/APN3 100 = PE0/APN4 101 = ADC multiplexing 110 = AGND 111 = Turn off differential amplifier reverse input
1:0	DPS[1:0]	Input source selection bit at the forward input of the differential amplifier 00 = ADC multiplexing 01 = ADC0/APP0 10 = ADC1/APP1 11 = AGND

OFR0 - Out-of-tune compensation register 0

OFR0 – Out-of-tune compensation register 0								
Address: 0xA3					Default value: 0x00			
Bit	7	6	5	4	3	2	1	0
Name	OFR0[7:0]							
R/W	W/R							

Bit	Name	description
7:0	OFR0	Out-of-tune compensation register 0; OFR0 is a signed number. Stored in binary complement format

OFR1 - Out-of-tune compensation register 1

OFR1 – Out-of-tune compensation register 1								
Address: 0xA4					Default value: 0x00			
Bit	7	6	5	4	3	2	1	0
Name	OFR1[7:0]							
R/W	W/R							

Bit	Name	description
7:0	OFR1	Out-of-tune compensation register 1; OFR1 is a signed number. Stored in binary complement format

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	description
7	AMOF	Automatic monitoring of overflow event type flag bit; 1=up overflow, 0=down overflow
6:4	-	Unimplemented
3:0	AMFC	Automatic monitoring of digital filtering control bits. 0000 = Disable configuration 0001 = One conversion, no filtering 0010 = two consecutive concordances 0011 = three consecutive concordances .... 1110 = 14 consecutive concordances 1111 = 15 consecutive concordances

ADT0L - Automatic monitoring of lower relief valve value 8 positions lower

ADT0L - Automatic monitoring of lower relief valve value 8 positions lower								
Address: 0xA5					Default value: 0x00			
Bit	7	6	5	4	3	2	1	0
Name	ADT0L[7:0]							
R/W	W/R							

Bit	Name	description
7:0	ADT0L	Automatic monitoring of the lower 8 bits of the Overflow Threshold Register

ADT0H - Automatic monitoring of underflow valve value 8 digits higher

ADT0H - Automatic monitoring of underflow valve value 8 digits higher								
Address: 0xA6					Default value: 0x00			
Bit	7	6	5	4	3	2	1	0
Name	ADT0H[7:0]							
R/W	W/R							

Bit	Name	description
7:0	ADT0H	Automatic monitoring of the upper 8 bits of the Lower Overflow Threshold Register

ADT1L - Automatic monitoring of upper relief valve value 8 positions lower

ADT0L - Automatic monitoring of upper relief valve value 8 positions lower								
Address: 0xAA					Default value: 0x00			
Bit	7	6	5	4	3	2	1	0
Name	ADT1L[7:0]							
R/W	W/R							

Bit	Name	description
7:0	ADT1L	Automatic monitoring of the lower 8 bits of the upper overflow threshold register

ADT1H - Automatic monitoring of the upper relief valve value 8 digits higher

ADT1H - Automatic monitoring of the upper relief valve value 8 digits higher								
Address: 0xAB					Default value: 0x00			
Bit	7	6	5	4	3	2	1	0
Name	ADT1H[7:0]							
R/W	W/R							

Bit	Name	description
7:0	ADT1H	Automatic monitoring of the upper 8 bits of the overflow threshold register

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	description

7:0	VCAL	Internal reference calibration register. A calibration value of 1.024V is loaded by default after power-up. By writing the calibration value of other reference voltages into this register, the calibration of the relevant reference can be achieved. For example, if the reference configuration is 2.048V, write VCAL2 to the change register to complete the calibration of 2.048V Calibration of the internal reference.

VCAL1 - 1.024V Reference Calibration Register

VCAL1 - 1.024V Internal Reference Calibration Register								
Address: 0xCD					Default value: 0x00			
Bit	7	6	5	4	3	2	1	0
Name	VCAL1[7:0]							
R/W	R/O							

Bit	Name	description
7:0	VCAL1	1.024V Internal Reference Calibration Factor

VCAL2 - 2.048V Reference Calibration Register

VCAL2 - 2.048V Internal Reference Calibration Register								
Address: 0xCE					Default value: 0x00			
Bit	7	6	5	4	3	2	1	0
Name	VCAL2[7:0]							
R/W	R/O							

Bit	Name	description
7:0	VCAL2	2.048V Internal reference calibration factor

VCAL3- 4.096V Reference Calibration Register

VCAL1 - 4.096V Internal Reference Calibration Register								
Address: 0xCC					Default value: 0x00			
Bit	7	6	5	4	3	2	1	0
Name	VCAL3[7:0]							
R/W	R/O							

Bit	Name	description
7:0	VCAL3	4.096V Internal Reference Calibration Factor

Register Quick Checklist

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	PRR/0	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

command set speed table

command	number of operations	description	operations	marker position	cyclicality
Arithmetic logic operation instructions					
ADD	Rd, Rr	Register summation	Rd ← Rd + Rr	Z,C,N,V,H	1
ADC	Rd, Rr	Register summation with rounding	Rd ← Rd + Rr + C	Z,C,N,V,H	1
ADIW	Rdl, K	Immediate number and word summation	Rdh:Rdl ← Rdh:Rdl + K	Z,C,N,V,S	1
SUB	Rd, Rr	Register summation and subtraction	Rd ← Rd - Rr	Z,C,N,V,H	1
SUBI	Rd, K	Register minus constants	Rd ← Rd - K	Z,C,N,V,H	1
SBC	Rd, Rr	Register summing and subtracting with debit	Rd ← Rd - Rr - C	Z,C,N,V,H	1
SBCI	Rd, K	Register with borrowed bits minus constants	Rd ← Rd - K - C	Z,C,N,V,H	1
SBIW	Rdl, K	Immediate number and word subtraction	Rdh:Rdl ← Rdh:Rdl - K	Z,C,N,V,S	1
AND	Rd, Rr	logic and	Rd ← Rd & Rr	Z,N,V	1
ANDI	Rd, K	Register Logic and Constants	Rd ← Rd & K	Z,N,V	1
OR	Rd, Rr	logical or	Rd ← Rd | Rr	Z,N,V	1
ORI	Rd, K	register logic or constant	Rd ← Rd | K	Z,N,V	1
EOR	Rd, Rr	register iso-or	Rd ← Rd ⊕ Rr	Z,N,V	1
COM	Rd	inverse code	Rd ← $FF - Rd	Z,C,N,V	1
NEG	Rd	2 Prohibition of complementary codes	Rd ← $00 - Rd	Z,C,N,V,H	1
SBR	Rd, K	Set the bits in the register	Rd ← Rd v K	Z,N,V	1
CBR	Rd, K	Clear the bits in the register	Rd ← Rd v ($FF - K)	Z,N,V	1
INC	Rd	progressive	Rd ← Rd + 1	Z,N,V	1
DEC	Rd	in descending order	Rd ← Rd - 1	Z,N,V	1
TST	Rd	Test is 0 or negative	Rd ← Rd & Rd	Z,N,V	1
CLR	Rd	clear register	Rd ← Rd⊕Rd	Z,N,V	1
SER	Rd	All registers set to 1	Rd ← $FF	None	1
MUL	Rd, Rr	unsigned multiplication	R1: R0 ← Rd X Rr	Z,C	1
MULS	Rd, Rr	signed multiplication	R1: R0 ← Rd X Rr	Z,C	1
MULSU	Rd, Rr	Multiplying signed numbers by unsigned numbers	R1: R0 ← Rd X Rr	Z,C	1
FMUL	Rd, Rr	Unsigned Multiplication, Shift	R1: R0 ← (Rd X Rr) << 1	Z,C	1
FMULS	Rd, Rr	Signed Multiplication, Shift	R1: R0 ← (Rd X Rr) << 1	Z,C	1
FMULSU	Rd, Rr	Multiplying a signed number by an unsigned number, shifting	R1: R0 ← (Rd X Rr) << 1	Z,C	1
Jump Instructions					
RJMP	K	relative jump	PC ← PC + K + 1	None	1
IJMP		Indirect jump (to Z-pointing address)	PC ← Z	None	2
JMP	K	Direct Jump	PC ← K	None	2
RCALL	K	Relative address subroutine calls	PC ← PC + K + 1	None	1

ICALL		Indirect subroutine call (Z pointing to address)	PC ← Z	None	2
CALL	K	Direct subroutine calls	PC ← K	None	2
RET		The subroutine returns	PC ← Stack	None	2
RETI		Interrupt return	PC ← Stack	I	2

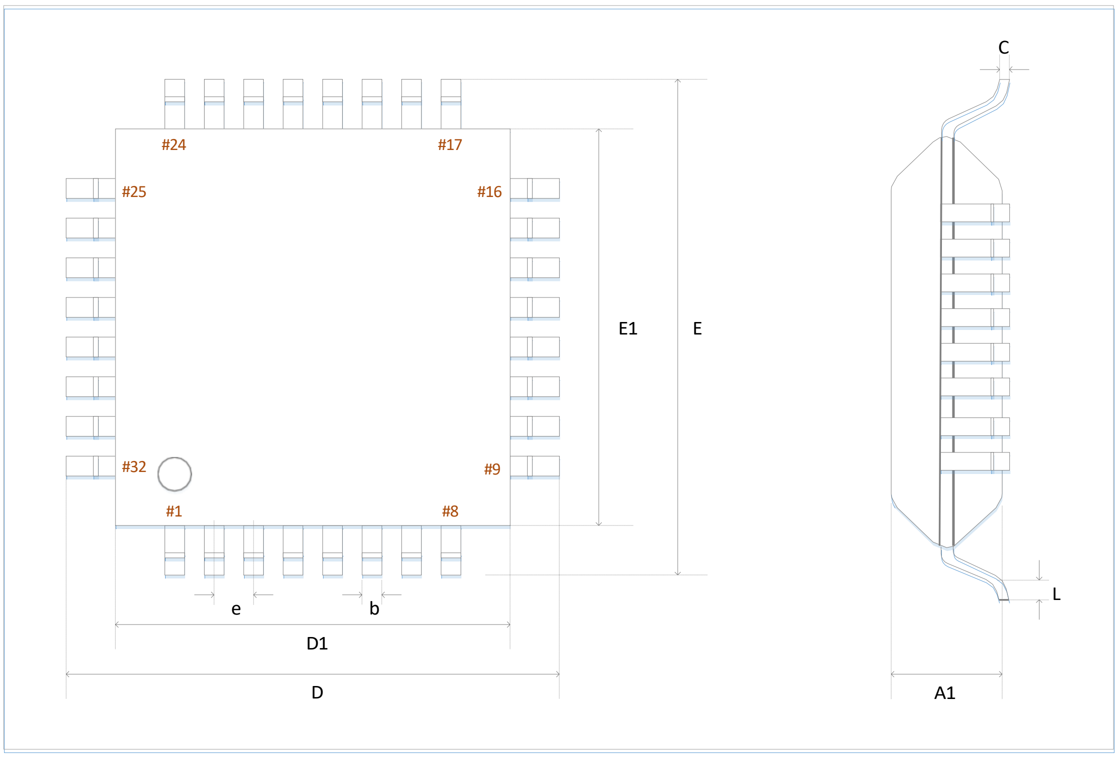
command	number of operations	description	operations	marker position	cyclicality
Jump instructions (continued)					
CPSE	Rd, Rr	Equivalent i.e. jump	If(Rd=Rr) PC ← PC + 2 or 3	None	1/2
CP	Rd, Rr	comparisons	Rd - Rr	Z,N,V,C,H	1
CPC	Rd, Rr	with progressive comparison	Rd - Rr - C	Z,N,V,C,H	1
CPI	Rd, K	Comparison with immediate numbers	Rd - K	Z,N,V,C,H	1
SBRC	Rr, b	A bit of 0 means that the next instruction is skipped	If(Rr(b)=0) PC ← PC + 2 or 3	None	1/2
SBRS	Rr, b	A bit of 1 skips the next instruction	If(Rr(b)=1) PC ← PC + 2 or 3	None	1/2
SBIC	P, b	An I/O bit of 0 means that the next instruction is skipped	If(P(b)=0) PC ← PC + 2 or 3	None	1/2
SBIS	P, b	I/O bit is 1 to skip the next instruction	If(P(b)=1) PC ← PC + 2 or 3	None	1/2
BRBS	s, k	Status marker is 1 to jump	If(SREG(S)=1) PC ← PC + K + 1	None	1/2
BRBC	s, k	The status marker is 0 which means jump	If(SREG(S)=0) PC ← PC + K + 1	None	1/2
BREQ	k	Equivalent i.e. jump	if (Z = 1) then PC ← PC + k + 1	None	1/2
BRNE	k	Jump without waiting	if (Z = 0) then PC ← PC + k + 1	None	1/2
BRCS	k	A rounding will jump	if (C = 1) then PC ← PC + k + 1	None	1/2
BRCC	k	Skip if no feed	if (C = 0) then PC ← PC + k + 1	None	1/2
BRSH	k	Less than then skip	if (C = 0) then PC ← PC + k + 1	None	1/2
BRLO	k	Less than then jump	if (C = 1) then PC ← PC + k + 1	None	1/2
BRMI	k	Negative then jump	if (N = 1) then PC ← PC + k + 1	None	1/2
BRPL	k	for regular jumps	if (N = 0) then PC ← PC + k + 1	None	1/2
BRGE	k	Signed not less than i.e. jump	if (N ⊕ V= 0) then PC ← PC + k + 1	None	1/2
BRLT	k	Signed less than 0 to jump	if (N ⊕ V= 1) then PC ← PC + k + 1	None	1/2
BRHS	k	A half-entry of 1 will jump	if (H = 1) then PC ← PC + k + 1	None	1/2
BRHC	k	A half-entry of 0 will jump	if (H = 0) then PC ← PC + k + 1	None	1/2
BRTS	k	T is set to jump	if (T = 1) then PC ← PC + k + 1	None	1/2
BRTC	k	T Clear then jump	if (T = 0) then PC ← PC + k + 1	None	1/2
BRVS	k	Overflow then jump	f (V = 1) then PC ← PC + k + 1	None	1/2
BRVC	k	Jump if not overflowing	f (V = 0) then PC ← PC + k + 1	None	1/2
BRIE	k	Global interrupt enable then jumps	f (I = 1) then PC ← PC + k + 1	None	1/2
BRID	k	Jump if global interrupt is disabled	f (I = 0) then PC ← PC + k + 1	None	1/2
Data transmission instructions					
MOV	Rd, Rr	Moving data between registers	Rd ← Rr	None	1
MOVW	Rd, Rr	Move one word of data	Rd+1:Rd ← Rr+1:Rr	None	1
LDI	Rd, K	Loading immediate count	Rd ← K	None	1
LD	Rd, X	Indirect loading	Rd ← (X)	None	1/2
LD	Rd, X+	Indirect loading, address increment	Rd ← (X), X ← X + 1	None	1/2
LD	Rd, -X	Address decrement, indirect loading	X ← X - 1, Rd ← (X)	None	1/2
LD	Rd, Y	Indirect loading	Rd ← (Y)	None	1/2

LD	Rd, Y+	Indirect loading, address increment	Rd ← (Y), Y ← Y + 1	None	1/2
LD	Rd, -Y	Address decrement, indirect loading	Y ← Y - 1, Rd ← (Y)	None	1/2
LDD	Rd, Y+q	Indirect loading with offsets	Rd ← (Y + q)	None	1/2
LD	Rd, Z	Indirect loading	Rd ← (Z)	None	1/2

LD	Rd, Z+	Indirect loading, address increment	Rd ← (Z), Z ← Z+1	None	1/2
LD	Rd, -Z	Address decrement, indirect loading	Z ← Z - 1, Rd ← (Z)	None	1/2
LDD	Rd, Z+q	Indirect loading with offsets	Rd ← (Z + q)	None	1/2
LDS	Rd, k	Load directly from SRAM	Rd ← (k)	None	2
ST	X, Rr	Indirect storage	(X) ← Rr	None	1
ST	X+, Rr	Indirect storage, address increment	(X) ← Rr, X ← X + 1	None	1
ST	-X, Rr	Address Decrement, Indirect Storage	X ← X - 1, (X) ← Rr	None	1
ST	Y, Rr	Indirect storage	(Y) ← Rr	None	1
ST	Y+, Rr	Indirect storage, address increment	(Y) ← Rr, Y ← Y + 1	None	1
ST	-Y, Rr	Address Decrement, Indirect Storage	Y ← Y - 1, (Y) ← Rr	None	1
STD	Y+q, Rr	Indirect storage with offsets	(Y + q) ← Rr	None	1
ST	Z, Rr	Indirect storage	(Z) ← Rr	None	1
ST	Z+, Rr	Indirect storage, address increment	(Z) ← Rr, Z ← Z + 1	None	1
ST	-Z, Rr	Address Decrement, Indirect Storage	Z ← Z - 1, (Z) ← Rr	None	1
STD	Z+q, Rr	Indirect storage with offsets	(Z + q) ← Rr	None	1
STS	k, Rr	Direct storage into SRAM	(k) ← Rr	None	2
LPM		Loading program space data	R0 ← (Z)	None	2
LPM	Rd, Z	Loading program space data	Rd ← (Z)	None	2
LPM	Rd, Z+	Load program data, address increment	Rd ← (Z), Z ← Z+1	None	2
LD	Rd, Z+	Indirect loading, address increment	Rd ← (Z), Z ← Z+1	None	1
LD	Rd, -Z	Address decrement, indirect loading	Z ← Z - 1, Rd ← (Z)	None	1
LDD	Rd, Z+q	Indirect loading with offsets	Rd ← (Z + q)	None	1
LDS	Rd, k	Load directly from SRAM	Rd ← (k)	None	2
IN	Rd, P	read port	Rd ← P	None	1
OUT	P, Rr	write port	P ← Rr	None	1
PUSH	Rr	pressure stack	STACK ← Rr	None	1
POP	Rd	leave the warehouse	Rd ← STACK	None	1/2
SBI	P, b	Setting IO registers	I/O(P, b) ← 1	None	1
CBI	P, b	Clear IO registers	I/O(P, b) ← 0	None	1
LSL	Rd	Logical left shift	Rd(n+1) ← Rd(n), Rd(0) ← 0	Z,C,N,V	1
LSR	Rd	Logical right shift	Rd(n) ← Rd(n+1), Rd(7) ← 0	Z	1
ROL	Rd	Cyclic left shifts that include rounding	Rd(0)←C, Rd(n+1) ← Rd(n), C←Rd(7)	Z	1
ROR	Rd	Cyclic right shift including rounding	Rd(7)←C, Rd(n) ← Rd(n+1), C←Rd(0)	Z	1
ASR	Rd	Arithmetic right shift	Rd(n) ← Rd(n+1), n=0:6	Z	1
SWAP	Rd	bit-swap	Rd(3:0) ← Rd(7:4), Rd(7:4) ← Rd(3:0)	None	1
BSET	s	Setting the status bit	SREG(s) ← 1	SREG(s)	1
BCLR	s	Clear status bits	SREG(s) ← 0	SREG(s)	1
BST	Rr, b	Store to T bit	T ← Rr(b)	T	1
BLD	Rd, b	Read T bit to register	Rd(b) ← T	None	1
SEC		Setting the feed flag	C ← 1	C	1
CLC		Clear feed flag	C ← 0	C	1

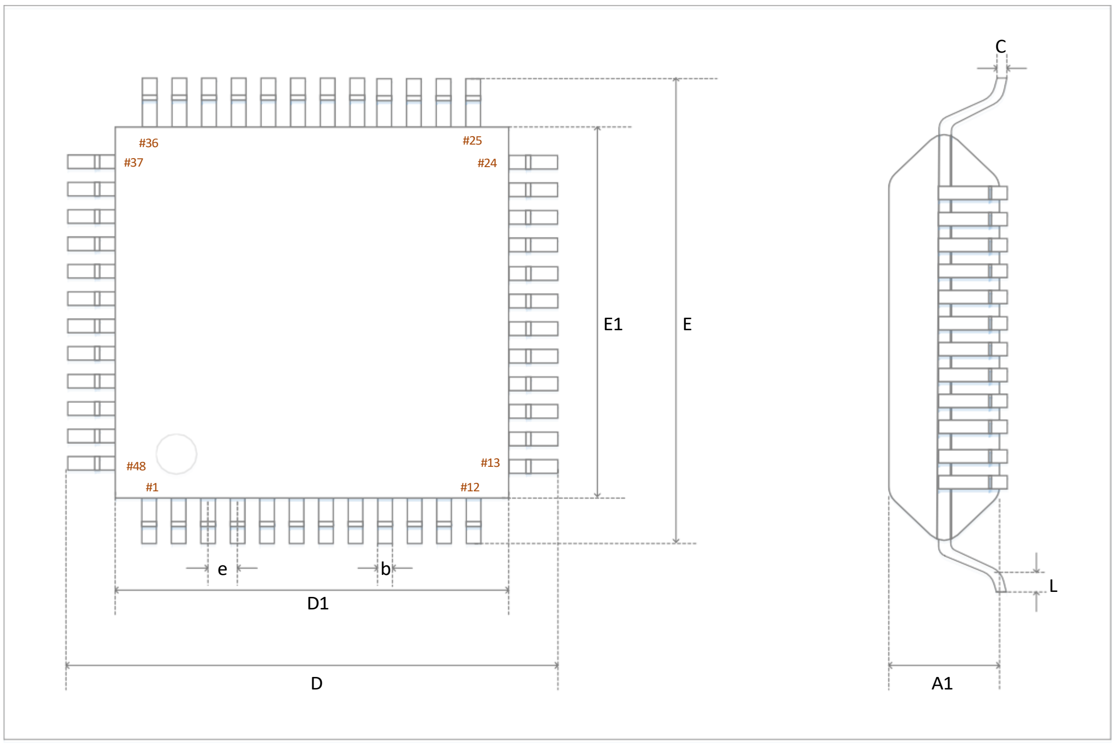
SEN		Setting the negative flag	N ← 1	N	1
CLN		Clear the negative flag	N ← 0	N	1
SEZ		Setting the zero mark	Z ← 1	Z	1
CLZ		Clear the zero flag	Z ← 0	Z	1
SEI		Enabling global interrupts	I ← 1	I	1
CLI		Disable global interruptions	I ← 0	I	1
SES		Setting symbol test flags	S ← 1	S	1
CLS		Clear symbol test flags	S ← 0	S	1
SEV		Set the binary complement overflow flag	V ← 1	V	1
CLV		Clear the binary complement overflow flag	V ← 0	V	1
SET		Set T bit (SREG)	T ← 1	T	1
CLT		Clear T-bit (SREG)	T ← 0	T	1
MCU control commands					
NOP		null instruction		None	1
SLEEP		Entering hibernation mode		None	1
WDR		Watchdog reset		None	1
BREAK		soft breakpoint	For debugging purposes only	None	N/A
NOP		null instruction		None	1
SLEEP		Entering hibernation mode		None	1

Package Parameters



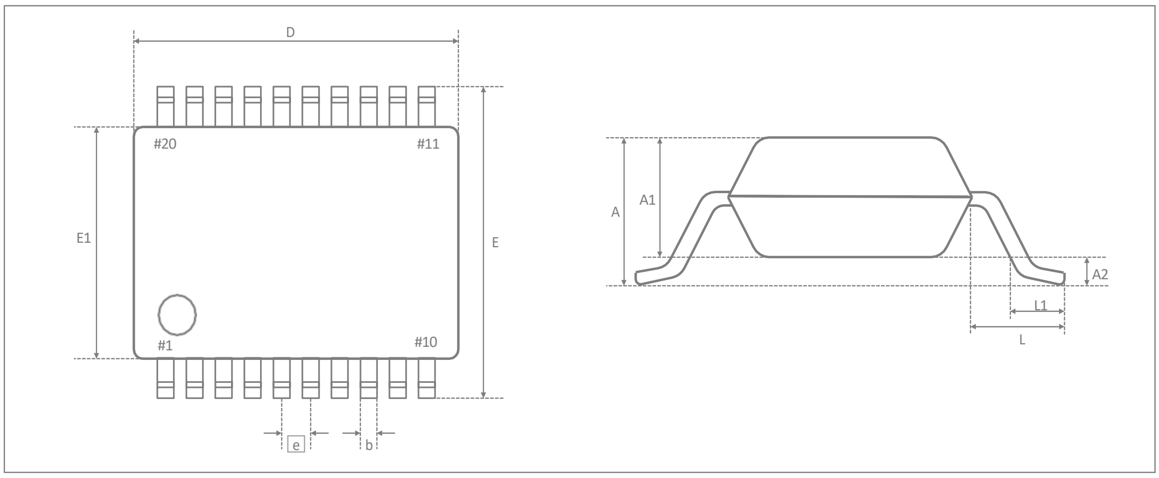
LQFP32 General Size Definition

character designator	minimum value	typical value	maximum value	unit
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 General Size Definition

character designator	minimum value	typical value	maximum value	unit
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 General Size Definition

character designator	minimum value	typical value	maximum value	unit
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

Version History

V1.0.5 2018/9/26	Removal of ADC11 function on QFP32/PB5 pins Corrects configuration in AC1 for positive end selection
V1.0.4 2017/11/15	Correction to the definition of SSOP20 PIN8/11
V1.0.3 2017/6/23	Add SSOP20 package definition Operating Instructions for Updating the TMR3 Interrupt Marker Bit
V1.0.2 2017/5/15	Update the description of automatic PWM shutdown and restart in TMR0/TRM1/TMR3 Update the description of SPI interrupt handling in the SPI chapter and update the description of the SPFR register
V1.0.1 2017/2/13	Delete the I2C1 section, this function is not available to improve the definition of some registers
V1.0.0 2016/12/29	Initial version