

## 8bitMCU ES7P003

# data sheet

-Product introduction

- Data sheet

-Product specification

**Shanghai Neusoft Carrier Microelectronics Co., Ltd.**

**2019Year6moon13day**

## Neusoft CarrierMCUChip usage precautions

### About the power on/off of the chip

Neusoft CarrierMCUThe chip has an independent power supply pin. whenMCUWhen the chip is used in a multi-power supply system, it should beMCUPower on the chip, and then power on other parts of the system; on the contrary, when powering off, power off other parts of the system first, and then power onMCUPower off the chip. If the operation sequence is reversed, it may cause overvoltage or overcurrent of the internal components of the chip, resulting in chip failure or component degradation. For details, please refer to the data sheet of the chip.

### About chip reset

Neusoft CarrierMCUThe chip has an internal power-on reset. For different fast power-on/off or slow power-on/off systems, the internal power-on reset circuit may fail. It is recommended that users use external reset, power-off reset, watchdog reset, etc. to ensure that the reset circuit works normally. In system design, if an external reset circuit is used, it is recommended to use a triode reset circuit,RCreset circuit. If you do not use an external reset circuit, it is recommended to use the reset pin to connect a resistor to the power supply, or take necessary power supply jitter processing circuits or other protection circuits. For details, please refer to the data sheet of the chip.

### About the clock of the chip

Neusoft CarrierMCUThe chip has internal and external clock sources. The internal clock source will drift with temperature and voltage changes, which may affect the accuracy of the clock source; when the external clock source uses a ceramic or crystal oscillator circuit, it is recommended to enable the start-up delay; useRCWhen oscillating the circuit, it is necessary to consider the matching of capacitance and resistance; when using an external active crystal oscillator or clock input, it is necessary to consider the input high/low level voltage. For details, please refer to the data sheet of the chip.

### About chip initialization

Neusoft CarrierMCUThe chip has various internal and external resets. For different application systems, it is necessary to initialize chip registers, memory, functional modules, etc., especiallyI/OThe pin multiplexing function is initialized to avoid that after the chip is powered on,I/OAn indeterminate condition of the pin state occurs.

### About the pins of the chip

Neusoft CarrierMCUThe chip has a wide range of input pin levels, it is recommended that the user input high level should be in $V_{IHMIN}$ above, the low level should be at $V_{ILMAX}$ under. Avoid input voltages between $V_{IHMIN}$ and $V_{ILMAX}$ Between, to avoid fluctuation noise into the chip. For unused input/output pins, it is recommended that users set them to input state and pull them up to the power supply or pull them down to ground through a resistor, or set them as output pins, output a fixed level and float. The handling of unused pins varies with the application system, and specifically follow the relevant regulations and instructions of the application system.

### About the chipESDProtective measures

Neusoft CarrierMCUThe chip has an industrial-gradeESDStandard protection circuit. Users are advised to take appropriate electrostatic protection measures according to the chip storage/application environment. Attention should be paid to the humidity of the application environment; it is recommended to avoid the use of insulators that are prone to static electricity; storage and transportation should be in antistatic containers, antistatic shielding bags or conductive material containers; all test and measurement tools including workbenches must be grounded; The operator should wear gloves with anti-static wrist loops, and cannot directly touch chips with hands.

### About the chipEFTProtective measures

Neusoft CarrierMCUThe chip has an industrial-gradeEFTStandard protection circuit. whenMCUThe chip is used inPCBsystem, it is necessary to follow thePCBRelevant design requirements, including power and ground routing (including digital/analog power separation, single/multi-point grounding, etc.) , reset pin protection circuit, power supply and ground Coupling capacitors, high and low frequency circuits are handled separately, and single/multilayer board selection, etc.

### About the development environment of the chip

Neusoft CarrierMCUThe chip has a complete software/hardware development environment and is protected by intellectual property rights. To choose the assembler, compiler, programmer, and emulator development environment of Shanghai Neusoft Carrier Microelectronics Co., Ltd. or its designated third-party company, you must follow the regulations and instructions related to the chip.

Note: During product development, if you encounter any unclear points, please contact Shanghai Neusoft Carrier Microelectronics Co., Ltd. through sales or other means.

## Product Ordering Information

Part No.	Operating Voltage	FLASH	SRAM	Data FLASH	I/O	ADC	UART	I2CS	SPI	Timer	package type
ES7P003FGTF	2.3V~5.5V	8K word	1K Byte	256 word	18	12-bit×8+1Ch	1	1	1	8-bit×1	TSSOP20
ES7P003FGNF										16-bit×2	QFN20

Note:8K Word FLASHlast in program memory256 wordsfor fixedData Flashstorage.

land Address: Longcao Road, Shanghai, China299Tianhua Information Technology Park2Abuilding5layer

mail code:200235

E-mail:support@essemi.com

electr wds: +86-21-60910333

passTrue: +86-21-60914991

net site:<http://www.essemi.com>

all rights reserved©

**Shanghai Neusoft Carrier Microelectronics Co., Ltd.**

The content of this document is carefully compiled by Shanghai Eastsoft Carrier Microelectronics Co., Ltd. on the basis of existing data and materials, and the examples recorded in this document are based on the correct use method and standard operation. Please fully consider external conditions when waiting for examples. Shanghai Neusoft Carrier Microelectronics Co., Ltd. does not guarantee or confirm the applicability, appropriateness or completeness of these examples on the user, and Shanghai Neusoft Carrier Microelectronics Co., Ltd. Any legal responsibility for any risks or consequences that may or have been brought about by all the contents of this information. Shanghai Neusoft Carrier Microelectronics Co., Ltd. reserves the right to modify without prior notice based on reasons such as making the content of this document more complete. If the user needs to obtain the latest product information, please feel free to contact Shanghai Neusoft Carrier Microelectronics Co., Ltd. with the above contact information.

**revise history**

Version	modified date	Summary of Changes
V1.0	2018-11-6	First release
V1.1	2019-1-7	1.No.3Add the relevant content of the unique identification code of the chip in the chapter; 2.appendix3.1Add chips inESDCharacteristic parameter table; 3.IncreaseT21Description of the multi-function multiplexed output port, etc.
V1.2	2019-2-14	1.correctPBPDThe register name isPBPort weak pull-down control register; 2.Added electrical characteristicsESD CDMtest level; 3. T21modulePWMoutput0Supplementary description of the implementation method; 4.Add chip power-on and power-off working conditions table; 5. IAPsoperation and interrupt, an increase of the interrupt enable bitGIEandGIEL Supplementary instructions for operation; 6.Added a supplementary description of the package size; <b>7.changelog.</b>
V1.3	2019-4-28	1.mask configuration wordCFG_WD0<2:0>=110configuration options; 2.optimizationUARTSending and receiving sequence diagrams and related descriptions;
V1.4	2019-6-13	1.addQFN20Packaging related information; 2.exist"UARTIn the "Precautions for Use" section, addUARTDescription of baud rate redundancy.

## Table of contents

## content directory

<b>No.1chapter</b>	<b>Chip introduction.....</b>	<b>16</b>
1. 1	Overview.....	16
1. 2	Application field.....	19
1. 3	Structure diagram.....	19
1. 4	Pin Assignment Diagram.....	20
1. 4. 1	20-pin.....	20
1. 5	Pin Description.....	twenty one
<b>No.2chapter</b>	<b>Kernel Features .....</b>	<b>twenty four</b>
twenty one	CPUKernel overview .....	twenty four
twenty two	Hardware Multiplier .....	twenty four
2. 2. 1	Overview.....	twenty four
2. 2. 2	Hardware Multiplier Operation .....	twenty four
twenty three	Hardware Divider .....	25
2. 3. 1	Overview.....	25
2. 3. 2	Hardware Divider Operation .....	25
twenty four	Special Function Registers .....	25
2. 4. 1	Program Status Word Register (PSW) .....	26
2. 4. 2	Program Counter Register Low8bits (PCRL) .....	26
2. 4. 3	Program Counter Register High5bits (PCRH) .....	27
2. 4. 4	accumulatorAregister(AREG) .....	27
2. 4. 5	multiplierAregister(MULA) .....	27
2. 4. 6	multiplierBregister(MULB) .....	27
2. 4. 7	low product8bit register (MULL) .....	27
2. 4. 8	product high8bit register (MULH) .....	28
2. 4. 9	low dividend8bit register (DIVEL) .....	28
2. 4. 10	dividend high8bit register (DIVEH) .....	28
2. 4. 11	Divisor Register (DIVS) .....	28
2. 4. 12	low quotient8bit register (DIVQL) .....	28
2. 4. 13	business high8bit register (DIVQH) .....	29
2. 4. 14	remainder register (DIVR) .....	29
<b>No.3chapter</b>	<b>Storage resources .....</b>	<b>30</b>
3.1	Overview.....	30
3.2	Program/Data Addressing Space Mapping.....	30
3.3	FLASHmemory.....	32
3. 3. 1	Overview.....	32
3. 3. 2	Program Counter (PC) .....	32
3. 3. 3	Hardware stack.....	33
3. 3. 4	Chip unique identification code (UID) .....	33
3. 3. 5	FLASHmemoryIAPoperate.....	34
3. 3. 5. 1	Overview.....	.34
3. 3. 5. 2	Table lookup read operation and reference routines.....	.34

3. 3. 5. 3	memoryIAPerase.....	35
3. 3. 5. 4	Memory Page Update Flow .....	36
3. 3. 5. 5	Operation Reference Routine.....	36
3. 3. 6	Special Function Registers .....	38
3. 3. 6. 1	Look-up table address register low8bits (FRAL) .....	38
3. 3. 6. 2	Look-up table address register high8bits (FRAH) .....	38
3. 3. 6. 3	Look-up table address inversion register low8bits (FRALN) .....	39
3. 3. 6. 4	Look-up table address inversion register high8bits (FRAHN) .....	39
3. 3. 6. 5	Look-up table data register low8bits (ROMDL) .....	39
3. 3. 6. 6	Look-up table data register high8bits (ROMDH) .....	39
3. 3. 6. 7	Memory Control Register Low8bits (ROMCL) .....	39
3. 3. 6. 8	Memory Control Register High8bits (ROMCH) .....	40
3.4	online programmingISP and online debuggingICD.....	40
3.5	Data storage.....	41
3.5.1	Overview.....	41
3.5.2	General purpose data storage .....	41
3.5.3	Special Function Registers .....	41
3.5.4	Addressing Mode.....	42
3. 5. 4. 1	Direct Addressing.....	42
3. 5. 4. 2	GPRSpecial addressing.....	43
3. 5. 4. 3	Indirect addressing.....	43
3. 5. 5	Special Function Registers .....	44
3. 5. 5. 1	Indirect Addressing Data Register (IAD) .....	44
3. 5. 5. 2	Indirect Addressing Index Register Low8bits (IAAL) .....	44
3. 5. 5. 3	Indirect Addressing Index Register High8bits (IAAH) .....	45
3. 5. 5. 4	Bank Select Register (BKSR) .....	45
No.4 chapter	<b>Input/Output Ports.....</b>	<b>46</b>
4.1	Overview.....	46
4.2	Structure diagram.....	47
4.3	I/OPort function settings.....	47
4. 3. 1	I/OPort I/O Control .....	47
4.3.2	I/OPort weak pull-up/pull-down function.....	47
4. 3. 3	I/OPort analog/digital type selection function.....	48
4. 3. 4	I/OPort open-drain output .....	48
4. 3. 5	I/OPort multiplexing function.....	48
4.4	Port Interrupts.....	48
4. 4. 1	External port interrupt (PINT) .....	48
4.5	I/OPrecautions for Port Operation.....	49
4.6	Special Function Registers .....	49
4.6.1	PAPort Level Status Register (PA) .....	49
4.6.2	PAPort I/O Control Register (PAT) .....	49
4.6.3	PAPort Weak Pull-Up Control Register (PAPU) .....	50
4. 6. 4	PAPort Weak Pull-Down Control Register (PAPD) .....	50
4.6.5	PBPort Level Status Register (PB) .....	50

4. 6. 6	PBPort I/O Control Register (PBT) .....	50
4. 6. 7	PBPort Weak Pull-Up Control Register (PBPU) .....	51
4. 6. 8	PBPort Weak Pull-Down Control Register (PBPD) .....	51
4.6.9	PCPort Level Status Register (PC) .....	51
4.6.10	PCPort I/O Control Register (PCT) .....	51
4.6.11	PCPort Weak Pull-Up Control Register (PCPU) .....	52
4.6.12	Port Special Function Control Register (PORTCTR) .....	52
No.6chapter	<b>SPECIAL FUNCTIONS AND OPERATING CHARACTERISTICS.....</b>	.53
5.1	System Clocks and Oscillators .....	53
5.1.1	Overview.....	53
5.1.2	Internal structure map.....	53
5.1.3	Clock source.....	54
5.1.3.1	internal high speed16MHz RCOscillator Mode (INTHRC) .....	54
5.1.3.2	internal low speed32kHz RCOscillator Mode (INTLRC) .....	54
5.2	System Power.....	55
5.2.1	Overview.....	55
5.2.2	Internal structure map.....	55
5.3	Watchdog Timer (WDT) .....	56
5.3.1	Overview.....	56
5.3.2	Internal structure map.....	56
5.3.3	WDTtimer.....	56
5.3.4	Special Function Registers .....	57
5.3.4.1	WDTControl Register (WDTC) .....	57
5.4	Resetting the module .....	58
5.4.1	Overview.....	58
5.4.2	Power-on reset.....	58
5.4.3	Brownout reset.....	58
5.4.4	externalMRSTNPin reset.....	59
5.4.5	Watchdog Timer Overflow Reset .....	60
5.4.6	RSTCommand software reset.....	61
5.4.7	Special Function Registers .....	61
5.4.7.1	Power Control Register (PWRC) .....	61
5.4.7.2	Power Control Register (PWEN) .....	62
5.5	Low Power Operation .....	63
5.5.1	MCULow Power Mode .....	63
5.5.2	Low Power Mode Configuration .....	63
5.5.3	IDLEWake-up mode configuration.....	64
5.5.4	Wake-up Timing Diagram.....	64
5.5.5	Special Function Registers .....	65
5.5.5.1	Wake-Up Delay Control Register (WKDC) .....	65
No.6chapter	<b>Peripherals.....</b>	.66
6.1	8bit timer/counter (T8N) .....	66
6.1.1	Overview.....	66
6.1.2	Internal structure map.....	66

6.1.3	Prescaler.....	66
6.1.4	Operating mode.....	67
6.1.5	Timer Mode.....	67
6.1.6	Synchronous Counter Mode .....	68
6.1.7	Special Function Registers .....	68
6.1.7.1	T8Ncounter register (T8N) .....	69
6.1.7.2	T8NControl Register (T8NC) .....	69
6.2	16Bit Multi-function Timer (T21) .....	70
6.2.1	Overview.....	70
6.2.2	Internal structure map.....	71
6.2.3	Prescaler and Postscaler .....	71
6.2.4	Operating mode.....	71
6.2.5	Timer Mode.....	71
6.2.6	multi-precisionPWMmodel.....	72
6.2.7	Catcher Mode.....	73
6.2.8	Comparator Mode.....	74
6.2.9	Multi-function multiplexed output port.....	75
6.2.10	Special Function Registers .....	76
6.2.10.1	counter register low8bits (T21L) .....	76
6.2.10.2	counter register high8bits (T21H) .....	76
6.2.10.3	Period Register Low8bits (T21PL) .....	76
6.2.10.4	Period Register High8bits (T21PH) .....	76
6.2.10.5	multi-purpose register0Low8bits (T21R0L) .....	77
6.2.10.6	multi-purpose register0high8bits (T21R0H) .....	77
6.2.10.7	multi-purpose register1Low8bits (T21R1L) .....	77
6.2.10.8	multi-purpose register1high8bits (T21R1H) .....	77
6.2.10.9	multi-purpose register2Low8bits (T21R2L) .....	78
6.2.10.10	multi-purpose register2high8bits (T21R2H) .....	78
6.2.10.11	control register low8bits (T21CL) .....	78
6.2.10.12	control register second high8bits (T21CM) .....	79
6.2.10.13	control register high8bits (T21CH) .....	80
6.2.10.14	I/O Control Register (T21OC) .....	80
6.3	16Bit Multi-function Timer (T31) .....	81
6.3.1	Overview.....	81
6.3.2	Internal structure map.....	82
6.3.3	Clock Divider.....	82
6.3.3.1	Prescaler.....	82
6.3.3.2	Postscaler.....	83
6.3.4	Counting mode.....	84
6.3.4.1	Count Up Mode.....	84
6.3.4.2	Count Down Mode.....	85
6.3.4.3	Center Alignment Mode .....	86
6.3.5	Operating mode.....	86
6.3.6	Timer Mode.....	87

6. 3. 6. 1	Internal Clock Source Mode.....	88
6. 3. 6. 2	External Clock Source Mode1 .....	88
6. 3. 6. 3	External Clock Source Mode2 .....	89
6. 3. 7	Catcher Mode.....	89
6. 3. 8	Comparator Mode.....	91
6. 3. 9	PWMmodel.....	93
6. 3. 9. 1	PWMnormal mode .....	93
6. 3. 9. 2	PWMCenter Alignment Mode .....	94
6. 3. 9. 3	Complementary output with dead time.....	95
6. 3. 10	Single Pulse Mode .....	96
6.3.11	Shutdown Functional Mode .....	97
6.3.12	Slave mode.....	99
6.3.12.1	Encoder Mode.....	99
6. 3. 12. 2	reset mode.....	100
6. 3. 12. 3	Gated Mode.....	101
6. 3. 12. 4	Trigger Mode.....	101
6.3.13	ICDdebug in suspend modePWMoutput.....	102
6.3.14	Special Function Registers .....	102
6.3.14.1	counter register low8bits (T31CNTL) .....	102
6. 3. 14. 2	counter register high8bits (T31CNTH) .....	102
6. 3. 14. 3	Prescaler Register Low8bits (T31PRSL) .....	103
6. 3. 14. 4	Prescaler Register High8bits (T31PRSH) .....	103
6. 3. 14. 5	Counter Reload Register Low8bits (T31CNTLDL) .....	103
6. 3. 14. 6	Counter Reload Register High8bits (T31CNTLDH) .....	103
6. 3. 14. 7	postscaler register (T31POS) .....	104
6. 3. 14. 8	aisle1Capture/Compare Register Low8bits (T31CH1RL) .....	104
6. 3. 14. 9	aisle1Capture/Compare Register High8bits (T31CH1RH) .....	104
6. 3. 14. 10	aisle2Capture/Compare Register Low8bits (T31CH2RL) .....	104
6. 3. 14. 11	aisle2Capture/Compare Register High8bits (T31CH2RH) .....	104
6. 3. 14. 12	aisle3Capture/Compare Register Low8bits (T31CH3RL) .....	105
6. 3. 14. 13	aisle3Capture/Compare Register High8bits (T31CH3RH) .....	105
6. 3. 14. 14	aisle4Capture/Compare Register Low8bits (T31CH4RL) .....	105
6. 3. 14. 15	aisle4Capture/Compare Register High8bits (T31CH4RH) .....	105
6. 3. 14. 16	Dead-Band Delay Register (T31DLYT) .....	106
6. 3. 14. 17	control register0Low8bits (T31C0L) .....	106
6. 3. 14. 18	control register0high8bits (T31C0H) .....	107
6. 3. 14. 19	control register1Low8bits (T31C1L) .....	107
6. 3. 14. 20	control register1high8bits (T31C1H) .....	108
6. 3. 14. 21	control register2Low8bits (T31C2L) .....	109
6. 3. 14. 22	control register2high8bits (T31C2H) .....	110
6. 3. 14. 23	aisle1Control Register (T31CH1C) .....	111
6. 3. 14. 24	aisle2Control Register (T31CH2C) .....	112
6. 3. 14. 25	aisle3Control Register (T31CH3C) .....	114
6. 3. 14. 26	aisle4Control Register (T31CH4C) .....	116

6. 3. 14. 27	Channel Port Control Register Low8bits (T31PINCL) .....	118
6. 3. 14. 28	Channel Port Control Register High8bits (T31PINCH) .....	120
6. 3. 14. 29	Channel Output Shutdown Control Register (T31CHBK) .....	121
6. 3. 14. 30	Software Trigger Event Register (T31EVG) .....	122
6. 3. 14. 31	Interrupt Enable Control Register Low8bits (T31IEL) .....	123
6. 3. 14. 32	Interrupt Enable Control Register High8bits (T31IEH) .....	124
6. 3. 14. 33	Interrupt Disable Register Low8bits (T31IDL) .....	125
6. 3. 14. 34	Interrupt Disable Register High8bits (T31IDH) .....	125
6. 3. 14. 35	Interrupt Enable Status Register Low8bits (T31IVSL) .....	126
6. 3. 14. 36	Interrupt Enable Status Register High8bits (T31IVSH) .....	127
6. 3. 14. 37	Interrupt Flag Register Low8bits (T31IFL) .....	127
6. 3. 14. 38	Interrupt flag register high8bits (T31IFH) .....	128
6. 3. 14. 39	Interrupt Request Status Register Low8bits (T31IFML) .....	129
6. 3. 14. 40	Interrupt Request Status Register High8bits (T31IFMH) .....	130
6. 3. 14. 41	interrupt clear0register low8bits (T31ICRL) .....	130
6. 3. 14. 42	interrupt clear0register high8bits (T31ICRH) .....	131
6.4	Universal Asynchronous Receiver/Transmitter (UART) .....	132
6.4.1	Overview.....	132
6.4.2	Internal structure map.....	132
6.4.3	Baud rate configuration.....	133
6.4.4	Transmission data format .....	133
6.4.5	Asynchronous Transmitter .....	133
6.4.6	Asynchronous Receiver .....	135
6.4.7	UARTPrecautions for use.....	136
6.4.8	Special Function Registers .....	137
6.4.8.1	UARTReceive Data Register (RXB) .....	137
6.4.8.2	UARTReceive Control/Status Register (RXC) .....	137
6.4.8.3	UARTTransmit Data Register (TXB) .....	138
6.4.8.4	UARTTransmit Control/Status Register (TXC) .....	138
6.4.8.5	UARTBaud Rate Register (BRR) .....	138
6.5	I2Cbus slave (I2CS) .....	139
6.5.1	Overview.....	139
6.5.2	internal structure .....	140
6.5.3	I2CSPort Configuration.....	140
6.5.4	I2CProtocol.....	141
6.5.5	I2Coperate.....	142
6.5.6	start bitSTARTand stop bitsSTOP .....	142
6.5.7	Data Transmission and Response.....	143
6.5.8	Data transfer format reference.....	144
6.5.9	I2CCommunication Transmitter .....	144
6.5.10	I2CCommunication Receiver.....	145
6.5.11	I2CClock line automatic pull-down wait request function.....	145
6.5.12	Special Function Registers .....	146
6. 5. 12. 1	I2CSample Filter Register (I2CX16) .....	146

6.5.12.2	I2CCControl Register (I2CC) .....	146
6.5.12.3	I2CSlave Address Register (I2CSA) .....	147
6.5.12.4	I2CTransmit Data Buffer (I2CTB) .....	147
6.5.12.5	I2CReceive Data Buffer (I2CRB) .....	148
6.5.12.6	I2CInterrupt Enable Register (I2CIEC) .....	148
6.5.12.7	I2CInterrupt Flag Register (I2CIFC) .....	149
6.6	SPISynchronous serial communication module (SPI) .....	150
6.6.1	Overview.....	150
6.6.2	Structure diagram.....	150
6.6.3	SPICommunication Mode.....	150
6.6.4	SPIData Format.....	150
6.6.5	SPISynchronous Transmitter .....	152
6.6.6	SPISynchronous Receiver .....	152
6.6.7	SPICommunication Control .....	153
6.6.8	SPIDelayed reception function .....	154
6.6.9	SPIApplication note.....	154
6.6.10	Special Function Registers .....	154
6.6.10.1	SPIcontrol register0(SPICON0) .....	154
6.6.10.2	SPIcontrol register1(SPICON1) .....	155
6.6.10.3	SPISend data is written to the register (SPITBW) .....	156
6.6.10.4	SPIReceive Data Read Register (SPIRBR) .....	156
6.6.10.5	SPIInterrupt Enable Register (SPIIE) .....	156
6.6.10.6	SPIInterrupt Flag Register (SPIIF) .....	157
6.7	A/D Converter (ADC) .....	159
6.7.1	Overview.....	159
6.7.2	Internal structure map.....	160
6.7.3	ADCconfiguration.....	160
6.7.4	ADCConversion steps.....	161
6.7.5	ADSchematic diagram of timing characteristics.....	162
6.7.6	Reference routines.....	162
6.7.7	Special Function Registers .....	163
6.7.7.1	ADCConversion Value Register (ADCR) .....	163
6.7.7.2	ADCControl Register (ADCCL) .....	163
6.7.7.3	ADCControl Register (ADCCH) .....	164
6.7.7.4	Port Digital-to-Analog Control Register (ANSL) .....	165
No.7chapter	<b>Interrupt handling .....</b>	<b>166</b>
7.1	overview.....	166
7.2	Block Diagram of Interrupt Control Structure.....	166
7.3	Interrupt Mode Selection .....	167
7.3.1	Default Interrupt Mode .....	167
7.3.2	Vectored Interrupt Mode .....	168
7.3.2.1	Vector Table Configuration .....	168
7.3.2.2	Interrupt Group Configuration .....	169
7.3.2.3	Interrupt Enable Configuration .....	169

7.4	Interrupt Context Saving.....	171
7.5	Interrupt operation.....	171
7.5.1	interrupt enable bitGIEandGIELOperation of .....	171
7.5.2	External Interrupt.....	171
7.5.3	ADCInterruption.....	172
7.5.4	T8NOverflow Interrupt .....	172
7.5.5	T21Overflow Interrupt .....	172
7.5.6	T21Cycle Interrupts.....	172
7.5.7	T21Multi-function Interrupt .....	172
7.5.8	T31Interruption.....	173
7.5.9	UARTInterruption.....	173
7.5.10	I2CS INTERRUPTION.....	173
7.5.11	SPIInterruption.....	174
7.5.12	Interrupt Operation Precautions.....	174
7.6	Special Function Registers .....	174
7.6.1	Interrupt Global Register (INTG) .....	174
7.6.2	Interrupt Priority Register (INTP) .....	175
7.6.3	Interrupt Control Register0(INTC0) .....	175
7.6.4	Interrupt Flag Register0(INTF0) .....	175
7.6.5	interrupt enable register0(INTE0) .....	176
7.6.6	Interrupt Flag Register1(INTF1) .....	177
7.6.7	interrupt enable register1(INTE1) .....	177
7.6.8	Interrupt Flag Register2(INTF2) .....	178
7.6.9	interrupt enable register2(INTE2) .....	179
No.8chapter	<b>Chip Configuration Words.....</b>	<b>180</b>
No.9chapter	<b>Chip package diagram.....</b>	<b>182</b>
9.1	20-pinPackage Diagram.....	182
9.1.1	TSSOP20.....	182
9.1.2	QFN20 .....	183
appendix1	<b>Instruction Set.....</b>	<b>184</b>
appendix1.1	Overview.....	184
appendix1.2	Register Manipulation Instructions.....	184
appendix1.3	Program Control Instructions.....	184
appendix1.4	Arithmetic/Logical Operation Instructions.....	185
appendix2	<b>Summary Table of Special Function Registers .....</b>	<b>188</b>
appendix3	<b>Electrical Characteristics .....</b>	<b>194</b>
appendix3.1	Parameter characteristic table.....	194
appendix3.2	Parameter Characteristic Diagram.....	199

**Figure catalog**

picture1-1	ES7P003Structure diagram.....	19
picture1-2	ES7P003FGTFtop view.....	20
picture3-1	FLASHProgram Memory Addressing Space Map.....	31
picture3-2	FLASHData memory address space map.....	31
picture3-3	Stack Diagram .....	33
picture3-4	Page Update Reference Flowchart.....	36
picture3-5	GPRSchematic diagram of address mapping.....	41
picture3-6	Special Function Register Space .....	42
picture3-7	Direct Addressing Schematic .....	43
picture3-8	GPRSSpecial Addressing Schematic .....	43
picture3-9	Indirect Addressing Schematic .....	44
picture4-1	PA/PB/PCPort structure diagram.....	47
picture5-1	System Clock Structure Diagram.....	53
picture5-2	System Power Structure Block Diagram.....	55
picture5-3	Watchdog Timer Internal Structure Diagram.....	56
picture5-4	Schematic diagram of chip reset.....	58
picture5-5	Schematic diagram of power-on reset sequence.....	58
picture5-6	Schematic Diagram of Low Voltage Reset Timing.....	59
picture5-7	externalMRSTNPin reset.....	59
picture5-8	MRSTNReset reference circuit diagram1.....	60
picture5-9	MRSTNReset reference circuit diagram2.....	60
picture5-10	Watchdog time-out reset .....	61
picture5-11	RSTCommand software reset.....	61
picture5-12	system wake upIDLE0Timing Diagram .....	65
picture5-13	system wake upIDLE1Timing Diagram .....	65
picture5-14	system wake upIDLE2Timing Diagram .....	65
picture6-1	T8NInternal structure map.....	66
picture6-2	Timer Mode Timing Diagram.....	68
picture6-3	Counter Mode Timing Diagram (T8NEG=0,T8NCKIRising edge count) .....	68
picture6-4	T21Internal structure map.....	71
picture6-5	T21Timer Mode Timing Diagram.....	72
picture6-6	T21multi-precisionPWMPattern Diagram .....	73
picture6-7	T21Capture mode timing diagram (capturing signal on every pulse rising edge) .....	74
picture6-8	T21Comparator Mode Timing Diagram .....	75
picture6-9	T31Internal structure map.....	82
picture6-10	Count Timing Diagram for Prescaler Change (T31CNTLD=0x2D) .....	82
picture6-11	The counting timing diagram using the postscaler (T31CNTLD=0x2D) .....	83
picture6-12	Count Up Timing Diagram (Prescaler is1) .....	84
picture6-13	Count Up Timing Diagram (RLBE=1,T31CNTLDbuffered) .....	84
picture6-14	Count Up Timing Diagram (RLBE=0,T31CNTLDunbuffered) .....	85
picture6-15	Count Down Timing Diagram (RLBE=1,T31CNTLDbuffered) .....	85
picture6-16	Center Alignment Mode1Count Timing Diagram (RLBE=1) .....	86
picture6-17	Timing diagram for counting up in internal clock source mode (prescaler/postscaler is1) .....	88

picture6-18	External Clock Source Mode1The counting timing diagram (the clock source is the channel1port, count up on rising edge) .....	89
picture6-19	External Clock Source Mode2The counting timing diagram (prescaler ratio is1:2, count up on rising edge).....	89
picture6-20	Capture mode timing for measuring pulsed signal period and duty cycle (channel1, reset mode) .....	91
picture6-21	Timing diagram of comparator mode (port output flip after comparison match,CHnOBE=0, unbuffered) .....	92
picture6-22	ordinaryPWMwaveform, counting up (T31CHnR=3,T31CNTLD=0x3F,CHnP=0).....	94
picture6-23	ordinaryPWMwaveform, count down (T31CHnR=3,T31CNTLD=0x3F,CHnP=1).....	94
picture6-24	center alignmentPWMWaveform (T31CHnR=0x3D,T31CNTLD=0x3F,CHnP=0,CMC=10) ..	95
picture6-25	Complementary Outputs with Dead-Band Delay.....	95
picture6-26	Single Pulse Output Waveform (PWMMmodel1,T31CHnR=0x3D,T31CNTLD=0x3F,CHnP=1) ..96 Shutdown Timing	
picture6-27	Diagram with Complementary Outputs with Dead Time.....	98
picture6-28	Counter count timing in encoder mode (T31SM=010, channel input is not inverted) .....	100
picture6-29	Counting Timing in Reset Mode (T31CNTLD=0x3F,aisle2Triggered by the rising edge of the port signal).....	100
picture6-30	Count Timing in Gated Mode (T31CNTLD=0x3F,aisle2Port gating signal active high) ...	101
picture6-31	Count timing in trigger mode (T31CNTLD=0x3F,aisle2Triggered by the rising edge of the port signal).....	102
picture6-32	UARTTransmitter Schematic .....	132
picture6-33	UARTSchematic diagram of the receiving end.....	133
picture6-34	UARTSchematic diagram of data format.....	133
picture6-35	UARTAsynchronous Transmitter Operation Flowchart.....	134
picture6-36	UARTTransmitter Send Data Timing Diagram (9Bit Data Format, No.9The bit data is "0") .....	135
picture6-37	UARTAsynchronous Receiver Operation Flowchart.....	136
picture6-38	UARTReceiver Receive Data Timing Diagram (9bit data format) .....	136
picture6-39	I2Cinternal structure .....	140
picture6-40	Schematic diagram of open-drain output port.....	141
picture6-41	I2CSchematic diagram of bus communication protocol.....	142
picture6-42	I2CSlave waveform diagram.....	142
picture6-43	I2CStart and stop bits.....	143
picture6-44	Data Transmission and Response.....	143
picture6-45	Schematic diagram of master controller writing slave data.....	144
picture6-46	Schematic diagram of master controller reading data from slave.....	144
picture6-47	I2CSchematic diagram of sending data flow.....	144
picture6-48	I2CSchematic diagram of receiving data flow.....	145
picture6-49	I2CSchematic diagram of clock line pull-down waiting waveform.....	145
picture6-50	SPICircuit structure block diagram.....	150
picture6-51	SPISchematic diagram of clock rising edge sending and falling edge receiving waveform.....	151
picture6-52	SPISchematic diagram of clock falling edge sending and rising edge receiving waveform.....	151
picture6-53	SPISchematic diagram of clock rising edge receiving and falling edge sending waveform.....	151
picture6-54	SPISchematic diagram of clock falling edge receiving and rising edge sending waveform.....	152
picture6-55	SPISchematic diagram of sending data flow.....	152
picture6-56	SPISchematic diagram of receiving data flow.....	153
picture6-57	SPISchematic diagram of delay receiving function waveform.....	154
picture6-58	ADCInternal structure map.....	160
picture6-59	ADCTiming Characteristic Schematic (SMPS=0) .....	162
picture6-60	ADCTiming Characteristic Schematic (SMPS=1) .....	162
picture7-1	Default Interrupt Mode Interrupt Control Logic .....	166

picture7-2 Vectored Interrupt Mode Interrupt Control Logic ..... 167

table directory

surface1-1	Pin Description.....	.twenty three
surface3-1	In-circuit programming/debugging pin description.....	40
surface4-1	I/OWeak pull-up on port.....	47
surface4-2	I/OPort Weak Pulldown.....	48
surface4-3	I/OPort open-drain output .....	48
surface4-4	External Port Interrupts.....	49
surface5-1	Low Power Mode Configuration Table.....	63
surface5-2	Wake-up mode configuration table.....	64
surface5-3	Wake-up time calculation table.....	64
surface6-1	T8NPrescaler Configuration Table .....	67
surface6-2	T8NWorking mode configuration table.....	67
surface6-3	List of Channel Complementary Outputs with Shutdown .....	98
surface6-4	The relationship between the counting direction and the encoder signal (the channel input is not inverted).....	99
surface6-5	UARTBaud rate configuration table.....	133
surface6-6	UARTData format configuration table.....	133
surface6-7	I2CSPort configuration table.....	140
surface7-1	Interrupt mode selection table.....	167
surface7-2	Default interrupt mode enable configuration table.....	168
surface7-3	Vector table configuration table.....	169
surface7-4	Vector interrupt mode interrupt grouping configuration table.....	169
surface7-5	Vector interrupt mode enable configuration table.....	170

## 1. Overview

- working conditions
  - Working voltage range: 2.3V ~ 5.5V
  - range of working temperature: -40 ~ 85°C
- Design process and packaging
  - Low power consumption, high speedFLASH CMOScraft
  - 20pins, using TSSOP/QFN encapsulation
- kernel
  - ES7P RISC CPUkernel
  - 79compact instructions
  - The system clock operating frequency is up to 16MHz
  - The instruction cycle is 2 system clock cycle
  - The reset vector is located at 0000h, the default interrupt vector is located at 0004h
  - Support interrupt processing, support interrupt priority and interrupt vector table
  - Support hardware multiplier/divider
- storage resources
  - 8K Word FLASHprogram memory (where the last 1 page fixed as Data Flashdata memory)
    - common 32page, per page 256 words
    - support IAP operation, look-up table read, page erase and single address programming
    - When erasing and writing, the timer module is supported to work normally
    - When erasing and writing, interrupt processing is not supported
  - 256 Word Data FLASHdata storage
    - A total of one page
    - support IAP operation, look-up table read, page erase and single address programming
    - When erasing and writing, the timer module is supported to work normally
    - When erasing and writing, interrupt processing is not supported
  - Support chip unique identification code
  - 8level program stack
- 1K Byte SRAMdata storage
  - Program memory supports direct addressing, relative addressing and look-up table read operations
  - The data memory supports direct addressing, GPR special addressing and indirect addressing
- Programming and debugging interface
  - Support online programming (ISP) interface
  - Support online debugging (ICD) Function
  - Support programming code encryption protection

- I/Oport

- Support up to 18 individual I/Oport

- PAport(PA0~PA7)

- PBport(PB0~PB7)

- PCport(PC0~PC1)

- support 9 external port interrupt PINT(PINT0~PINT8 for input)

- Supports independent configurable internal weak pull-up/pull-down input ports

- The matching accuracy of the input port pull-up/pull-down resistors is ±3% Within (normal temperature 25°C, VDD=5V)

- support 18 independently configurable weak pull-up input ports

- support 16 independently configurable weak pull-down input ports

- support 2 independently configurable open-drain output ports

- Reset and Clock

- Embedded power-on reset circuit POR

- Embedded power-down reset circuit BOR

- BOR Reset voltage range: 2.1V, 2.5V, 3.1V

- Support external reset MRSTN

- Support independent hardware watchdog timer

- support command RSTreset

- Support internal high frequency 16MHz RCOscillation clock source

- The factory calibration accuracy is ±1% (room temperature 25°C)

- Power consumption characteristics

- IDLE0 electric current

- 6uA@5.0V, 25°C, LDOs Sleep, the system clock source stops oscillating, typical value

- IDLE1 electric current

- 70uA@5.0V, 25°C, LDOs Normal operation, the system clock source stops oscillating, typical value

- IDLE2 electric current

- 300uA@5.0V, 25°C, LDOs Normal operation, the system clock source does not stop shaking, typical value

- dynamic current

- 2.5mA@internal 16MHz, 5.0V, 25°C, typical

- peripherals

- 1 road 8bit timer T8N

- Timer mode (the count clock is the system clock 2 crossover)

- Counter mode (external irrigation clock or internal low frequency INTLRC clock)

- Supports configurable prescaler

- Support interrupt generation

- 1 road 16bit multifunction timer T21

- Support synchronous timer mode (using system clock frequency division as clock source)

- built-in 4bit prescaler and 7bit postscaler

- up to 3 independent channels for:

- input capture
- output compare
- PWMgenerate (multi-precisionPWMmodel)

- 3roadPWMOutput duty cycle can be set independently
- 3roadPWMOutput polarity can be set independently
- Support counter overflow interrupt,PWMPeriod match interrupt, capture interrupt, compare interrupt

#### -1road16bit multifunction timerT31

- 16Bit auto-reload counter, support up counting, down counting, up/down alternate counting mode
- 16Bit programmable prescaler, the counting clock prescaler range is1~65536
- 8Bit-programmable postscale with a postscale range of1~256
- 8Bit Dead-Band Delay RegisterT31DLYT
- 4individual16Bit Capture/Compare RegisterT31CH1R,T31CH2R,T31CH3RandT31CH4R
- support7Two working modes: timer mode, capture mode, compare mode,PWMmode, Single Pulse Mode, Shutdown Function Mode, Slave Mode
- Multiple slave modes supported: Encoder mode, Reset mode, Gated mode, Trigger mode
- up to4independent channels for:
  - input capture
  - output compare
  - PWMgenerate (commonPWMmode, center-aligned mode, complementary output with dead zone)
  - Single pulse mode output
- The following events will generate an interrupt request:
  - Update: counter overflow/underflow, counter initialization (by software or internal/external trigger)
  - Trigger event (counter start, stop, initialization or counting by internal/external trigger)
  - input capture
  - output compare match
  - shutdown input

#### -One high-speed asynchronous transceiverUART

- Support asynchronous full-duplex transceiver
- support8bit/9bit data format
- The agreed data is received/sent from the lowest bit
- Support interrupt generation

#### -all the wayI2Cbus

- Only supports slave mode
- support standardI2CBus protocol, maximum transfer rate400K bit/s
- support7bit addressing mode
- It is agreed that the data is received/sent from the highest bit
- Support interrupt generation

#### -One way synchronous serial communication moduleSPI

- Support master mode, slave mode
- support4data transfer format
- Support master mode communication clock rate configurable
- support4stage transmit buffer and4stage receive buffer
- Supports transmit and receive buffer empty/full interrupts

- Support receive data overflow interrupt, send data write error interrupt, send data error interrupt in slave mode
- Supports chip select change interrupt in slave mode and idle state interrupt in master mode
- Support delayed reception in master mode

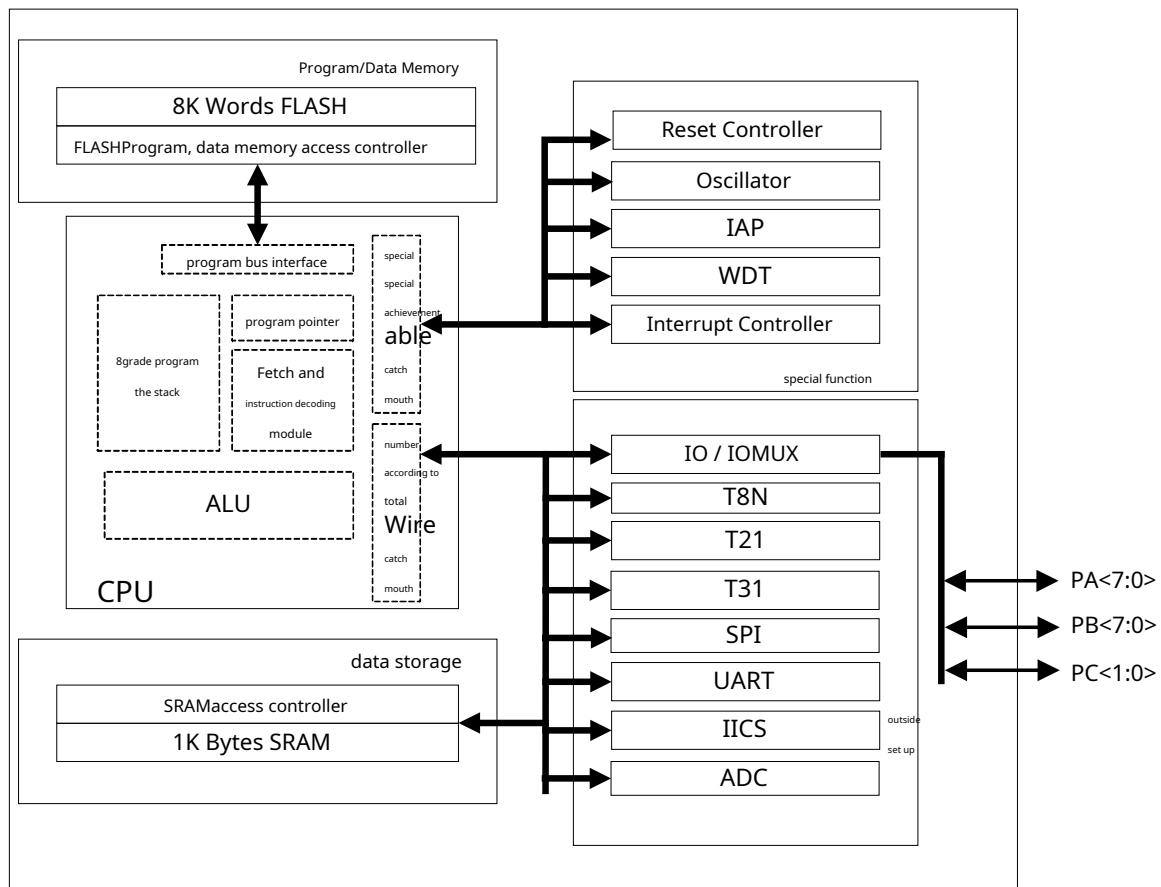
#### -Analog to Digital ConverterADC

- support12digit conversion accuracy
- support8+1channel analog input
- Support selectable reference voltage source
- Support interrupt generation

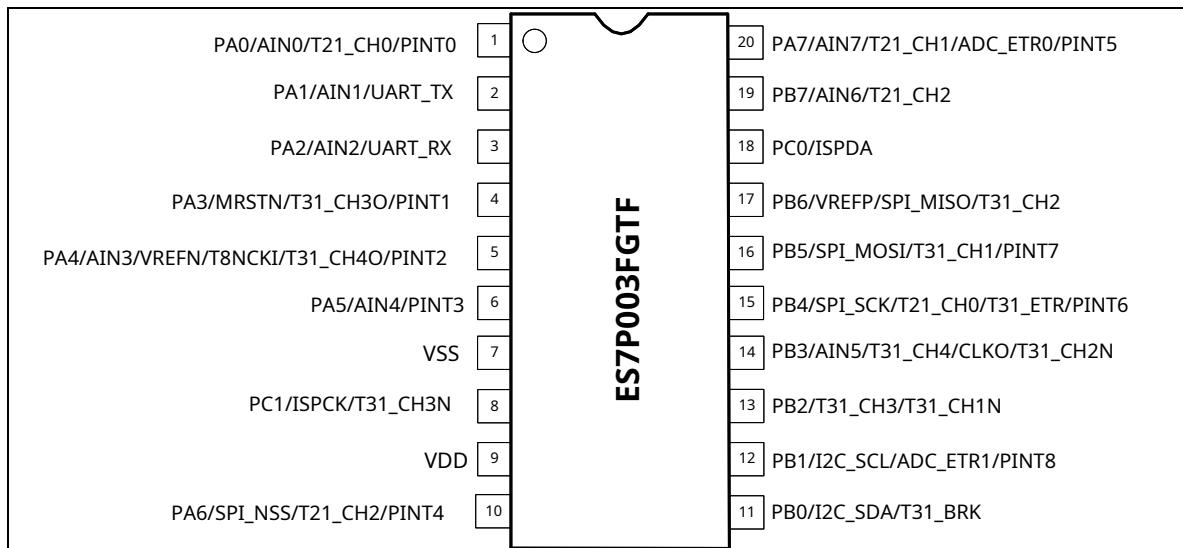
### 1. 2Application field

This chip can be used in access control systems/alarms, temperature sensing equipment, Bluetooth speakers, electric vehicle meters, digital voltmeters, gas detectors, collectors, chargers, beauty instruments, small household appliances and other fields.

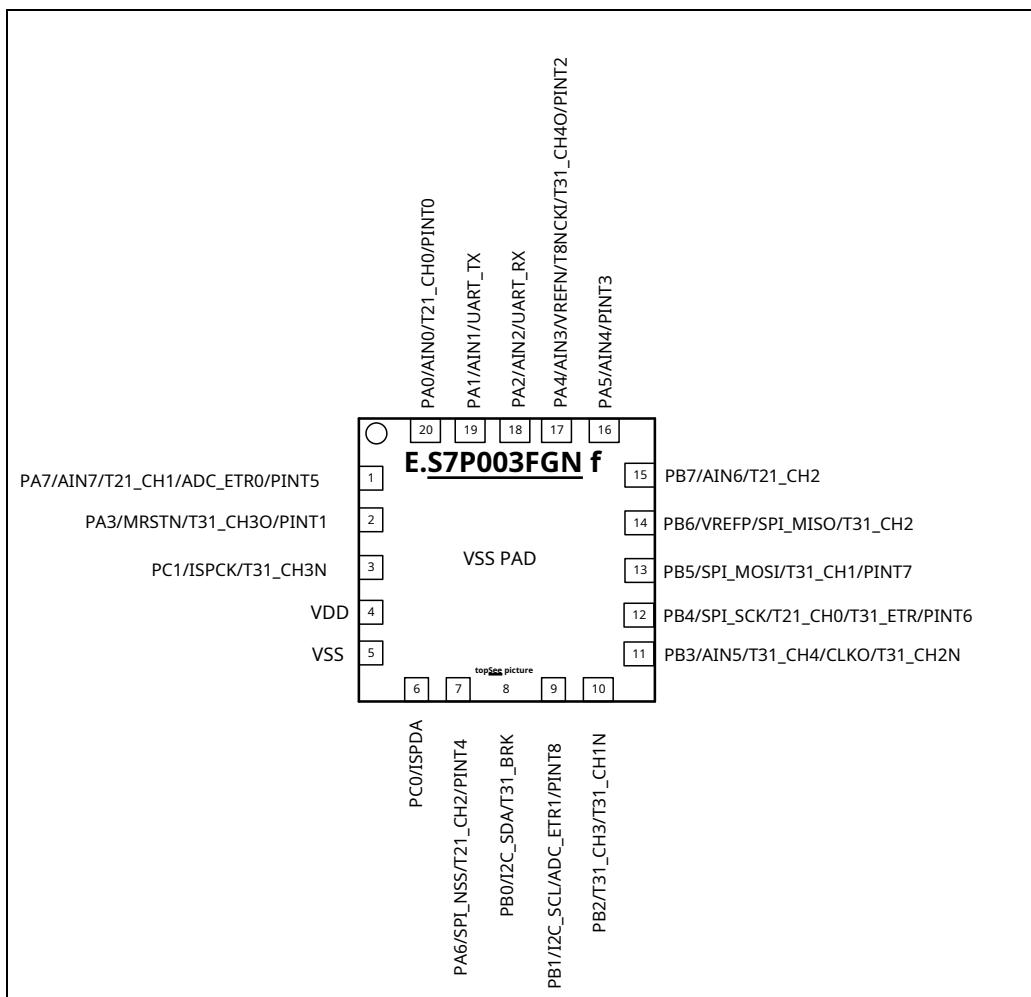
### 1. 3Structure diagram



picture1-1 ES7P003Structure diagram

**1. 4Pin Assignment Diagram**
**1. 4. 1 20-pin**


picture1-2 ES7P003FGTFtop view



picture1-3 ES7P003FGNtop view

Note1:MRSTNIndicates that the low level is active;

Note2: External reset pin can be multiplexedPA3is a digital input/output function; Note3:

PC0andPC1As a set of in-circuit programming/debugging interfaces;

Note4:chipI/OThe port input level cannot be higher than the chipVDD+0.3Vand not less thanVSS-0.3V, otherwise it may affect the normal operation of the chip; Note5:

T31aisle3(T31\_CH3) Compare output withPWMThe output can be multiplexed toPB2orPA3by registerT31CH3EN (PORTCTR<4>) choose;

Note6:31aisle4(T31\_CH4) compare the output withPWMThe output can be multiplexed toPB3orPA4, through the registerT31CH4EN(PORTCTR<5>) choose.

## 1.5Pin description

Pin name	Pin multiplexing	input type	output type	A/D	port description	Remark
PA0/AIN0/T21_CH0/ PINT0	PA0	TTL	CMOS	D.	universal/I/O	weak support up and down pull
	AIN0	—	—	A	ADCAnalog channel0enter	
	T21_CH0	TTL	CMOS	D.	T21capture input channel0 PWMonput channel0 compare output channel0	
	PINT0	TTL	—	D.	External interrupt input0	
PA1/AIN1/UART_TX	PA1	TTL	CMOS	D.	universal/I/O	weak support up and down pull
	AIN1	—	—	A	ADCAnalog channel1enter	
	UART_TX	—	CMOS	D.	UARTsend output	
PA2/AIN2/UART_RX	PA2	TTL	CMOS	D.	universal/I/O	weak support up and down pull
	AIN2	—	—	A	ADCAnalog channel2enter	
	UART_RX	TTL	—	D.	UARTreceive input	
PA3/MRSTN/ T31_CH3O/PINT1	PA3	TTL	CMOS	D.	universal/I/O	weak support up and down pull
	MRSTN	TTL	—	D.	External reset input	
	T31_CH3O	—	CMOS	D.	T31aisle3compare output PWMonput	
	PINT1	TTL	—	D.	External interrupt input1	
PA4/AIN3/VREFN/ T8NCKI/ T31_CH4O/ PINT2	PA4	TTL	CMOS	D.	universal/I/O	weak support up and down pull
	AIN3	—	—	A	ADCAnalog channel3enter	
	VREFN	—	—	A	ADCExternal reference negative input	
	T8NCKI	TTL	—	D.	T8Nexternal clock input	
	T31_CH4O	—	CMOS	D.	T31aisle4compare output PWMonput	
	PINT2	TTL	—	D.	External interrupt input2	
PA5/AIN4/PINT3	PA5	TTL	CMOS	D.	universal/I/O	weak support up and down pull
	AIN4	—	—	A	ADCAnalog channel4enter	
	PINT3	TTL	—	D.	External interrupt input3	
PA6/SPI_NSS/	PA6	TTL	CMOS	D.	universal/I/O	weak support

Pin name	Pin multiplexing	input type	output type	A/D	port description	Remark
T21_CH2/PINT4	SPI_NSS	TTL	—	D.	SPISlave Mode Chip Select Input	up and down pull
	T21_CH2	TTL	CMOS	D.	T21capture input channel2 PWMoutput channel2 compare output channel2	
	PINT4	TTL	—	D.	External interrupt input4	
PA7/AIN7/T21_CH1/ ADC_ETR0/PINT5	PA7	TTL	CMOS	D.	universalI/O	weak support up and down pull
	AIN7	—	—	A	ADCAnalog channel7enter	
	T21_CH1	TTL	CMOS	D.	T21capture input channel1 PWMoutput channel1 compare output channel1	
	ADC_ETR0	TTL	—	D.	External Interrupt5triggerADconvert signal	
	PINT5	TTL	—	D.	External interrupt input5	
PB0/I2C_SDA/ T31_BRK	PB0	TTL	CMOS	D.	universalI/O	weak support up and down pull/open Drain output
	I2C_SDA	TTL	CMOS	D.	I2Cdata input/output	
	T31_BRK	TTL	—	D.	T31Shutdown event input	
PB1/I2C_SCL/ ADC_ETR1/PINT8	PB1	TTL	CMOS	D.	universalI/O	weak support up and down pull/open Drain output
	I2C_SCL	TTL	—	D.	I2Cclock input	
	ADC_ETR1	TTL	—	D.	External Interrupt8triggerADconvert signal	
	PINT8	TTL	—	D.	External interrupt input8	
PB2/T31_CH3/ T31_CH1N	PB2	TTL	CMOS	D.	universalI/O	weak support up and down pull
	T31_CH3	TTL	CMOS	D.	T31aisle3capture input compare output,PWMoutput	
	T31_CH1N	—	CMOS	D.	T31aisle1complementary output	
PB3/AIN5/ T31_CH4/ T31_CH2N/ CLKO	PB3	TTL	CMOS	D.	universalI/O	weak support up and down pull
	AIN5	—	—	A	ADCAnalog channel5enter	
	T31_CH4	TTL	CMOS	D.	T31aisle4capture input compare output,PWMoutput	
	T31_CH2N	—	CMOS	D.	T31aisle2complementary output	
	CLKO	—	CMOS	D.	system clock16Frequency division output	
PB4/SPI_SCK/ T21_CH0/ T31_ETR/ PINT6	PB4	TTL	CMOS	D.	universalI/O	weak support up and down pull
	SPI_SCK	TTL	—	D.	SPISlave mode clock input	
	T21_CH0	TTL	CMOS	D.	T21capture input channel0 PWMoutput channel0 compare output channel0	
	T31_ETR	TTL	—	D.	T31External trigger input signal	
	PINT6	TTL	—	D.	External interrupt input6	
PB5/SPI_MOSI/ T31_CH1/PINT7	PB5	TTL	CMOS	D.	universalI/O	weak support up and down pull
	SPI_MOSI	TTL	CMOS	D.	SPIMaster output/slave input port	
	T31_CH1	TTL	CMOS	D.	T31aisle1capture input compare output	

Pin name	Pin multiplexing	input type	output type	A/D	port description	Remark
					PWMoutput	
	PINT7	TTL	—	D.	External interrupt input7	
PB6/VREFP/ SPI_MISO/ T31_CH2	PB6	TTL	CMOS	D.	universal/O	
	VREFP	—	—	A	ADCPositive input for external reference voltage	weak support
	SPI_MISO	TTL	CMOS	D.	SPIMaster input/slave output port	up and down
	T31_CH2	TTL	CMOS	D.	T31aisle2capture input compare output PWMoutput	pull
PB7/AIN6/T21_CH2	PB7	TTL	CMOS	D.	universal/O	
	AIN6	—	—	A	ADCanalog channel6enter	weak support
	T21_CH2	TTL	CMOS	D.	T21capture input channel2 PWMoutput channel2 compare output channel2	up and down pull
PC0/ISPDAT	PC0	TTL	CMOS	D.	universal/O	
	ISPDAT	TTL	CMOS	D.	ISPSerial programming/debug data input output	weak support pull up
PC1/ISPCK/ T31_CH3N	PC1	TTL	CMOS	D.	universal/O	
	ISPCK	TTL	—	D.	ISPSerial programming/debugging clock input	weak support
	T31_CH3N	—	CMOS	D.	T31aisle3complementary output	pull up
VDD	VDD	—	—	P	power supply	—
VSS	VSS	—	—	P	land,0Vreference point	—

## surface1-1Pin description

Note1:A =analog port,D =digital port,P =power/ground. Note2:

MRSTNIndicates low level reset is active.

Note3: all genericI/OPorts are TTLSchmidt input and CMOSoutput driver.

## No.2 Chapter Kernel Features

### 2.1 CPUKernel overview

-kernel features

- high performance ES7P RISC CPU kernel, 79RISC
- use 2T architecture, each machine cycle consists of two system clock cycles
- The system clock supports up to 16MHz, minimum instruction cycle 125ns
- Support interrupt priority and interrupt vector table
- Support hardware multiplier and divider

twenty two hardware multiplier

#### 2.2.1 overview

The chip instruction set does not contain multiplication instructions, and an independent hardware multiplier is integrated inside, which operates by reading and writing corresponding registers.

-main functional components

- 8bit multiplier A register (MULA, write-only)
- 8bit multiplier B register (MULB, write-only)
- 16bit product register (MULL/MULH, read only)

#### 2.2.2 Hardware Multiplier Operation

hardware multiplier complete 8bit multiplier MULA and 8bit multiplier MULB. The multiplication operation: <8bit multiplierA> x <8bit multiplierB> = 16bit product.

exist MULA and MULB after writing 1 in one machine cycle, the 16bit result high, low 8bits are stored in 2 registers MULH and MULL.

MULA and MULL share a register address, MULB and MULH share a register address. multiplier A/B After the setting is completed, the next instruction can read the result of the product.

##### Application Example: Hardware Multiplier Operation Application

```
...
MOVI    mul_operand_a
MOVA    MULA          ;write multiplierA
MOVI    mul_operand_b
MOVA    MULB          ;write multiplierB
MOV     MULL,0         ;read product low8bit
...
MOV     MULH,0         ;read product high8bit
...
```

**twenty threehardware divider****2. 3. 1** overview

The chip instruction set does not contain division instructions, and an independent hardware divider is integrated inside, which operates by reading and writing corresponding registers.

-main functional components

- 16bit dividend register (DIVEL/DIVEH, write-only)
- 8bit divisor register (DIVS, write-only)
- 16bit quotient register (DIVQL/DIVQH, read only)
- 8bit remainder register (DIVR, read only)

**2. 3. 2** Hardware Divider Operation

Hardware divider completes 16digit dividend DIVEL,DIVEHand 8bit divisor DIVS. The division operation of :<16bit dividend>  $\div$  <8 bit divisor> = 16bit quotient……8digit remainder.

DIVELand DIVQLShare a register address, DIVEHand DIVQHShare a register address, DIVSand DIVRShare a register address. After the dividend and divisor are set, you need to insert 2strip NOPin instruction to read the quotient and remainder. If the divisor is "0", then the quotient is 0xFFFF, the remainder is 0xFF, indicating overflow.

**Application Example: Hardware Divider Operation Application**

```
...
MOVI      div_operand_divel
MOVA      DIVEL           ;write dividend low8bit
MOVI      div_operand_diveh
MOVA     DIVEH           ;write dividend high8bit
MOVI      mul_operand_divs
MOVA      DIVS            ;write divisor
NOP
NOP
MOV      DIVQL,0          ;low reading quotient8bit
...
MOV      DIVQH,0          ;high reading quotient8bit
...
MOV      DIVR,0            ;read remainder
...
```

**twenty fourspecial function register**

CPUThe relevant registers include 13-bit Program Counter Register PCRL/PCRH, Program Status Word Register PSW , accumulator A register AREG, the multiplier register MULA, MULB sum product register MULL, MULH, the dividend register DIVEL/DIVEH, divisor register DIVS, business register DIVQL/DIVQH and remainder register DIVR. where the program status register PSW It is used to store various status flags, including program stack overflow, negative number flag, overflow flag, zero flag, half carry/half borrow flag, and full carry or full borrow flag, etc.

## 2.4.1 Program Status Word Register (PSW)

PSW: Program status word register								
bit	7	6	5	4	3	2	1	0
name	—	UF	OF	N	OV	Z	DC	C
R/W	—	R	R	R/W	R/W	R/W	R/W	R/W
RESET	X	0	0	X	X	X	X	X
Bit 7	Unused							
Bit 6	UF: Program stack overflow flag bit							
	0: The program has not overflowed the stack							
	1: program stack overflow							
Bit 5	OF: Program push stack overflow flag							
	0: Program push stack does not overflow							
	1: Program push stack overflow							
Bit 4	N: Negative number flag							
	0: The result of a signed arithmetic or logical operation is a positive number							
	1: the result is negative							
Bit 3	OV: overflow flag							
	0: Signed arithmetic operations did not overflow							
	1: overflow occurred							
Bit 2	Z: zero flag							
	0: The result of an arithmetic or logical operation is not zero							
	1: The result of an arithmetic or logical operation is zero							
Bit 1	DC: half carry or half borrow flag bit							
	0: There is no carry in the lower four bits or there is a borrow in the lower four bits							
	1: The lower four bits have carry or the lower four bits have no borrow							
Bit 0	C: All carry or all borrow flag bit							
	0: no carry or borrow							
	1: with carry or without borrow							

Note1: Only some commands are available PSW registers for write operations, including JDEC, JINC, SWAP, BCC, BSS, BTT, MOVA and SETR.

other instruction pairs PSW. The write operation of the register only affects the corresponding status flag bit according to the operation result of the instruction.

Note2: OF and UF The bit is a read-only flag, only power-on reset, reset command and MRSTNReset will clear it, other resets will not affect these two flags bit.

## 2.4.2 Program Counter Register Low8bits (PCRL)

PCRL: Program Counter Register Low8bit								
bit	7	6	5	4	3	2	1	0
name	PCRL<7:0>							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7~0 PCRL<7:0>: Program Counter Register Low8bit

### 2.4.3 Program Counter Register High5bits (PCRH)

PCRH: Program Counter Register High5bit								
bit	7	6	5	4	3	2	1	0
name	—	—	—	PCRH<4:0>				
R/W	—	—	—	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7~5      Unused

Bit 4~0      PCRH&lt;4:0&gt;: Program Counter Register High5bit

### 2.4.4 accumulatorAregister(AREG)

AREG:accumulatorAregister								
bit	7	6	5	4	3	2	1	0
name	AREG<7:0>							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	X	X	X	X	X	X	X	X

Bit 7~0      AREG&lt;7:0&gt;: the value of the accumulator

### 2.4.5 multiplierAregister(MULA)

MULA:multiplierAregister								
bit	7	6	5	4	3	2	1	0
name	MULA<7:0>							
R/W	W	W	W	W	W	W	W	W
POR	X	X	X	X	X	X	X	X

Bit 7~0      MULA&lt;7:0&gt;:multiplierA

### 2.4.6 multiplierBregister(MULB)

MULB:multiplierBregister								
bit	7	6	5	4	3	2	1	0
name	MULB<7:0>							
R/W	W	W	W	W	W	W	W	W
POR	X	X	X	X	X	X	X	X

Bit 7~0      MULB&lt;7:0&gt;:multiplierB

### 2.4.7 low product8bit register (MULL)

MULL: product low8bit register								
bit	7	6	5	4	3	2	1	0
name	MULL<7:0>							
R/W	R	R	R	R	R	R	R	R
POR	X	X	X	X	X	X	X	X

Bit 7~0      MUL&lt;7:0&gt;: product low8bit

## 2.4.8 product high8bit register (MULH)

MULH: product height8bit register								
bit	7	6	5	4	3	2	1	0
<b>name</b>	MULH<7:0>							
<b>R/W</b>	R	R	R	R	R	R	R	R
<b>POR</b>	X	X	X	X	X	X	X	X

Bit 7~0      MUL<15:8>: product height8bit

## 2.4.9 low dividend8bit register (DIVEL)

DIVEL: low dividend8bit register								
bit	7	6	5	4	3	2	1	0
<b>name</b>	DIVEL<7:0>							
<b>R/W</b>	W	W	W	W	W	W	W	W
<b>POR</b>	X	X	X	X	X	X	X	X

Bit 7~0      DIVEL<7:0>: low dividend8bit

## 2.4.10 dividend high8bit register (DIVEH)

DIVEH: high dividend8bit register								
bit	7	6	5	4	3	2	1	0
<b>name</b>	DIVEH<7:0>							
<b>R/W</b>	W	W	W	W	W	W	W	W
<b>POR</b>	X	X	X	X	X	X	X	X

Bit 7~0      DIVEH<7:0>: high dividend8bit

## 2.4.11 Divisor Register (DIVS)

DIVS: divisor register								
bit	7	6	5	4	3	2	1	0
<b>name</b>	DIVS<7:0>							
<b>R/W</b>	W	W	W	W	W	W	W	W
<b>POR</b>	X	X	X	X	X	X	X	X

Bit 7~0      DIVS<7:0>: divisor

## 2.4.12 low quotient8bit register (DIVQL)

DIVQL: business low8bit register								
bit	7	6	5	4	3	2	1	0
<b>name</b>	DIVQL<7:0>							
<b>R/W</b>	R	R	R	R	R	R	R	R
<b>POR</b>	X	X	X	X	X	X	X	X

Bit 7~0      DIVQL<7:0>: business low8bit

### 2.4.13 business high8bit register (DIVQH)

DIVQH: Shanggao8bit register								
bit	7	6	5	4	3	2	1	0
name	DIVQH<7:0>							
R/W	R	R	R	R	R	R	R	R
POR	X	X	X	X	X	X	X	X

Bit 7~0 DIVQH<7:0>: Shanggao8bit

### 2. 4. 14 remainder register (DIVR)

DIVR: remainder register								
bit	7	6	5	4	3	2	1	0
name	DIVR<7:0>							
R/W	R	R	R	R	R	R	R	R
POR	X	X	X	X	X	X	X	X

Bit 7~0 DIVR<7:0>:remainder

### No.3 Chapter Storage Resources

#### 3.1 overview

The chip adopts the Harvard bus architecture, and the program addressing space and data addressing space are independent of each other.

On-chip memory resources include:

##### -8K word FLASHprogram memory (where the last1page fixed asData Flashdata memory)

- Divided into 31 page, per page 256 words
- storage 0000h ~ 1EFFh
- Support self-programming operation in application IAP (In-Application Programming), including program and page erase
- Support table lookup instruction to read data
- Supports page erase with an erase time of at least 2ms
- Supports single-address programming with a programming time of at least 20us
- support at least 1010,000 erasing cycles, 10 Data retention time of more than one year
- When erasing and writing, it supports the normal operation of the timer module, but does not support interrupt processing

##### -256 word Data FLASHdata store

- only one page, for 256 words
- storage 1F00h ~ 1FFFh
- Support self-programming operation in application IAP (In-Application Programming), including program and page erase
- Support table lookup instruction to read data
- Supports page erase with an erase time of at least 2ms
- Supports single-address programming with a programming time of at least 20us
- support at least 1010,000 erasing cycles, 10 Data retention time of more than one year
- When erasing and writing, it supports the normal operation of the timer module, but does not support interrupt processing

##### -128 word INFO1 information area

- storage 80h ~ FFh
- which contains 72bit chip unique identification code (UID)
- Support lookup table instruction read

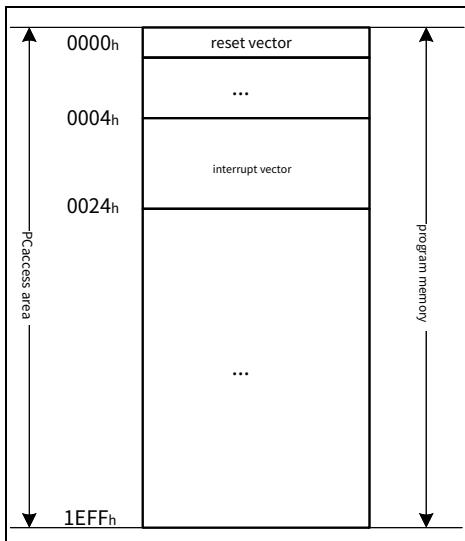
- UID is fixed at the factory and cannot be changed, and the program is read-only

##### -1K byte SRAMdata storage

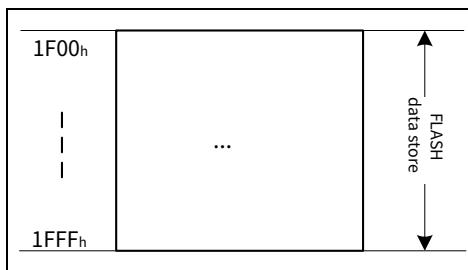
in FLASH program memory and FLASH data memory is mapped into the program address space, SRAM data memory is mapped into the data addressing space.

#### 3.2 Program/Data Addressing Space Mapping

The program/data addressing space mapping is as follows:



picture3-1 FLASHProgram memory address space map



picture3-2 FLASHData memory address space map

### 3.3 FLASHmemory

#### 3.3.1 overview

FLASHThe memory is divided into two areas, program memory and data memory.

FLASHThe program memory area is used to store user programs. Since the chip instruction bit width is16bits (2bytes), so about 8K Word FLASHProgram memory is mapped into the address space of the program addressing space0000h~1EFFh, each access address corresponds to a16bit width (2bytes) storage unit. pass13bit program counterPCPerform program addressing access.

FLASHThe data storage area is used to store key parameters in the user system, which will not be lost after the chip is powered off.

#### 3.3.2 Program Counter (PC)

Stored in the program counter is the address of the next instruction to be executed.CPURuntime,PCis automatically added after each instruction cycle1,unlessPCThe value of is overwritten by an instruction or interrupt exception.13bit program counterPC<12:0>, addressable8Kprogram memory0000h~ 1EFFh, out of address range will result inPCloop (again from0000h start access). program counterPCthe low8bitPC<7:0>accessiblePCRlread and write directly, whilePChigh5Bits cannot be read or written directly, only throughPCRHregister to indirect assignment (during executionRCALL,CALL,GOTOBefore waiting for instructions, you need toPCRHregister assignment). When reset,PCRl,PCRHandPCwill be cleared.PCHardware stack operations do not affectPCRHvalue.

when carried outFLASHWhen the data memory is erased or written, the program counterPCPause updates.

various instruction pairsPCImpact:

1. Modify directly by commandPCvalue, yesPCRLOperations for the target register can be directly modifiedPC<7:0>,Right nowPC<7:0>=PCRl<7:0>; while operatingPC<7:0>will also executePC<12:8>=PCRH<4:0>, so modifyPC, should be modified firstPCRH<4:0>, and then modify PCRl<7:0>.
2. implementRCALLcommand,PC<7:0>for the registerRthe value in ; andPC<12:8> =PCRH<4:0>.
3. implementCALL,GOTOcommand,PC<12:0>Low11bit in the instruction11bit immediate, andPC<12> =PCRH<4>. 4. implement LCALLinstruction, the instruction is shared by double-word instructions16bit immediate (operandPC<12:0>is modified to the16bit immediate value the low13bit; at the same timePCRH<4:0>is modified tol<12:8>value).
5. implementAJMPinstruction, the instruction is shared by double-word instructions16bit immediate (operandPC<12:0>is modified to the16bit immediate value Low13bit at the same timePCRH<4:0>change intol<12:8>value).
6. implementPAGEcommand,PCRH<4:3>The value of the instruction will be the immediatel<1:0>replace. 7. When executing other commands,PCThe value is automatically added1.

##### Application example: withPCRlinstruction application for the destination register

```

...
MOVI    pageaddr
MOVA    PCRH      ;Set table page address
MOVI    tableaddr ; Set the offset toRegister
CALL    TABLE     ;Call the subroutine to look up the table
...
TABLE:
ADD     PCRl, F   ;PCPlus the offset, pointing to the address of the access

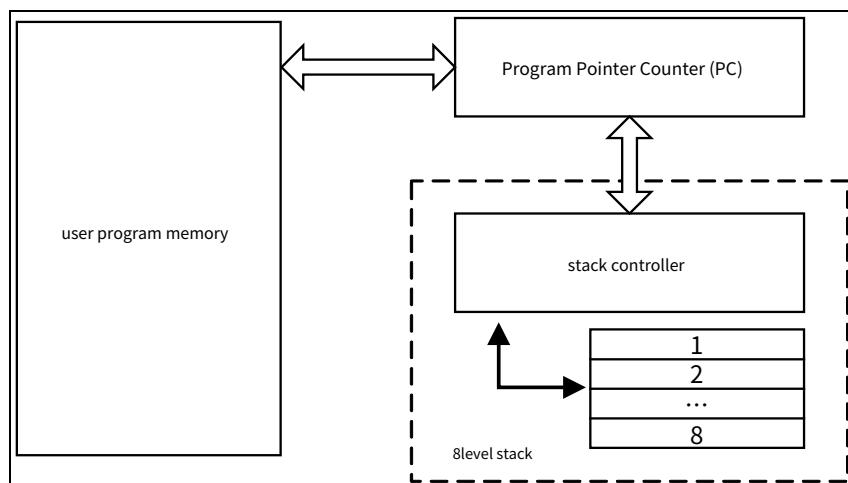
```

RETIA	0x01
RETIA	0x02
RETIA	0x03
...	

### 3.3.3 hardware stack

The chip has 8Level hardware stack, stack bit width and PC equal in bit width, for PCThe push and pop of the stack. implement CALL, LCALL and RCALLAfter the command or interrupt is serviced, the PCAutomatic stack protection; when executing RET, RETIAor RETIE instruction, the stack restores the last value pushed to the stackPC.

8Level hardware stack only supports 8Level buffer operation, that is, the hardware stack only saves the most recent 8Second push value, for consecutive over 8The second push operation, the first 9The next push data will overwrite the first 1The data pushed to the stack for the second time makes the 1The data pushed to the stack for the second time is lost. Likewise, more than 8times of continuous stack popping, the 9A stack pop operation may make the program flow uncontrollable. After a chip reset, the stack pointer will re-point to the top of the stack.



picture3-3Stack Diagram

### 3.3.4 Chip unique identification code (UID)

Each chip has a unique string of identification codes, that is, the unique identification code of the chip UID, common 9bytes 72bit, lie in INFO1 information area UID0~UID4, the user program can IAP Check the table to read. in UID4 only low8bit valid, high8Bit is FF. specific IAP For the operation of lookup table reading, see 3.3.5, 2Reference routines for chapters.

Chip unique identification code (UID0~UID4)		
address	UID0:0093h UID1:0094h UID2:0095h UID3:0096h UID4:0097h	
UID	Bit15~0	Chip unique identification code

### 3. 3. 5 FLASHmemoryIAPoperate

#### 3. 3. 5. 1

overview

FLASHmemoryIAPerase operations are performed in pages (page) as the unit, each page of program area and data area is 256 address unit, the page address corresponds to FRAH.FLASHmemory readout and IAPPProgram write operations to 1 unit of address, through FRA(FRAH,FRAL) addressing.

when FLASHmemory carry IAPerase or IAPWhen writing CPUThe kernel suspends execution, requiring software to turn off the global interrupt enable bit GIE(INTG<7>), and judge GIEWhether the register is cleared successfully, if it is not cleared, the software clearing operation needs to be performed again until the clearing is successful, the peripheral can continue to run according to the preset state, and the interrupt request of the peripheral will set the corresponding interrupt flag. when IAPerase or IAPWhen the write operation is complete, the CPUThe kernel resumes execution, and the software re-enables the global interrupt enable bit GIE, and perform corresponding interrupt processing.

FLASHBoth data memory and program memory support look-up table read and IAPerase and write operations.

#### 3. 3. 5. 2

Lookup table read operation and reference routine

of this chip FLASHThe program area, data area and information area all support table lookup and read operations. Through the look-up table read instruction will FRA (FRAH,FRAL) at the memory location pointed to by the word (word) Read ROMD(ROMDH, ROMDL) middle.

The chip lookup table write command is reserved for unused (execution only affects FRRegister) .

When performing look-up table read operations, it is necessary to set IAPSEL<2:0>(ROMCL<6:4>) to select the operation area, and the program area look-up table read also needs to enable the configuration word FREN(CFG\_WD0<13>), if FREN for 0, table look-up reading is prohibited, and all zeros are read in the program area; table look-up reading and configuration words in the data area and information area FRENWhether it is enabled or not is irrelevant.

**application routine1:FLASHProgram memory lookup table read**

```

MOVI    0x05          ; read program memory0105h
MOVA    FRAL
MOVI    0x01
MOVA    FRAH
MOVI    0x50          ;Select the operating area as the program area
MOVA    ROMCL
TBR
MOV     ROMDH, 0
...
MOV     ROMDL, 0
...
MOVI    0x00          ;quit IAPoperate
MOVA    ROMCL

```

**application routine2:FLASHData storage area lookup table read.**

```

MOVI    0x05          ; read data memory1F05h unit
MOVA    FRAL
MOVI    0X1F
MOVA    FRAH
MOVI    0x40          ; Select the operation area as the data area

```

```

MOVA    ROMCL
TBR
MOV     ROMDH, 0
...
MOV     ROMDL, 0
...
MOVI    0x00          ;quitIAPoperate
MOVA    ROMCL

```

application routine3:INFO1The information area looks up the table and reads.

```

MOVI    0x93          ; readINFO1memoryUID0(0093h)unit
MOVA    FRAL
MOVI    0X00
MOVA    FRAH
MOVI    0x60          ; Select the operating area asINFO1information area
MOVA    ROMCL
BSS    ROMCL,0         ;INFO1Zone Read Toggle Bit Enable
JBC    ROMCL,0
GOTO   $-1
BSS    ROMCL,7         ;INFO1Area lookup table instruction read enable
TBR
MOV     ROMDH, 0
...
MOV     ROMDL, 0
...
MOVI    0x00          ;quitIAPoperate
MOVA    ROMCL

```

### 3.3.5.3 memoryIAPerase

FLASHBoth program area and data area supportIAPerase and write operations.

Erase and write operations need to pass through the memory control register (ROMCH,ROMCL) together to complete while the address register FRA(FRAH,FRAL)andFRAN(FRAHN,FRALN) needs to meet the logical inversion, that is, the register FRANThe value of the registerFRAInverse of , ifFRAandFRANIf the logic inversion is not satisfied, the hardware will automatically clear the erase enable bitWREN, prohibit erasing. every timeFRAandFRANAfter the register is updated and the negation logic is satisfied, it must be reopenedFLASHErase/Program Enable BitWREN.

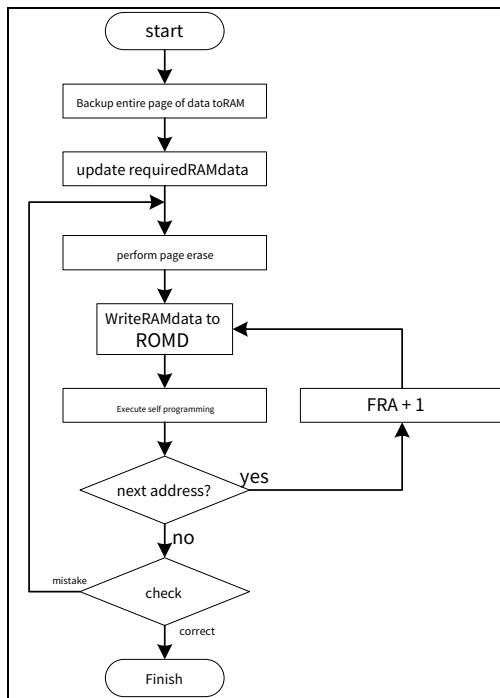
IAPerase and write includes three basic operations: data backup, page erase, and self-programming.

FLASHThe memory is erased in units of pages, and the page address register can beFRAHThe page pointed to is erased. Erase time per page is at least2ms.

FLASHThe self-programming of the memory is in units of words, which can beROMD(ROMDH,ROMDL) register in the16 bit value writeFRA(FRAH,FRAL) to the address unit pointed to. A single address programming time is at least20us.

### 3.3.5.4 Memory Page Update Flow

-programming mode



picture3-4Page Update Reference Flowchart

#### update a pageFLASHMemory steps:

1. Use the look-up table read command to back up a page to the data storage space (requires 256x2byte storage space, used to store the data volume of one page) ;
2. Modify the value to be updated in the backup data storage space;
3. By setting the registerROMCLandROMCHPerform page erase (must follow the fixed program flow) ;
4. by registerFRALandFRAHSelect the address to be updated and set the registerROMDLandROMDHdata that needs to be updated;
5. By setting the registerROMCLandROMCH, will registerROMDLandROMDHwrite content inFRAThe address in the page pointed to (must follow the fixed program flow);
6. repeat4,5Steps until full page programming is completed;
7. Use the look-up table read command to verify the write-in area.

### 3.3.5.5 Operation Reference Routine

The following routines are based on the datastore'sIAPPPage Erase and Program as an example, similarly whenIAPoperation selects program memory and also supports IAPpage erase and program operations.

#### application routine4:FLASHDatastore page erase.

The program stops, except that the timer/counter can keep running until it resumes automatically after the erase operation is complete.

MOVI	0X00	;Set the address of the erased page
MOVA	FRAL	
MOVI	0X1F	
MOVA	FRAH	
MOVI	0xFF	; Set the inverse code of the address of the page
MOVA	FRALN	
MOVI	0XE0	

MOVA	FRAHN	
MOVI	0x40	;Select the operation area as the data area
MOVA	ROMCL	
BSS	ROMCL, FPEE	;Select erase operation
BSS	ROMCL, WREN	;OpenFLASHErase/Program Enable
BCC	INTG, GIE	;Turn off the global interrupt (avoid the interruption affecting the follow-up fixed program flow);
JBC	INTG, GIE	determine whether the global interrupt is cleared
GOTO	\$-2	
MOVI	0x55	
MOVA	ROMCH	
.....		;8indivualNOPcommand, or wait8instruction cycle
MOVI	0xAA	
MOVA	ROMCH	
.....		;8indivualNOPcommand, or wait8instruction cycle
BSS	ROMCL, WR	
JBC	ROMCL,WR	;Wait for erase to end
GOTO	\$-1	
BSS	INTG, GIE	;enable global interrupt
...		
MOVI	0x00	;quitIAPoperate
MOVA	ROMCL	
CLR	FRALN	
CLR	FRAHN	

**application routine5:FLASHData storage area address unit is written.**

The program stops running, except that the timer/counter can keep running until it resumes running automatically after the write operation is completed.

MOVI	0x00	;to writeFLASHThe first part of the data store 1page1addresses
MOVA	FRAL	
MOVI	0x1F	
MOVA	FRAH	
MOVI	0xFF	;Set the inverse code of the address of the unit to be written
MOVA	FRALN	
MOVI	0XE0	
MOVA	FRAHN	
MOVI	0x12	
MOVA	ROMDH	
MOVI	0x34	
MOVA	ROMDL	;data input1234h
MOVI	0x40	;Select the operation area as the data area
MOVA	ROMCL	
BCC	ROMCL, FPEE	;Select programming operation
BSS	ROMCL, WREN	;OpenFLASHErase/program enable; disable global interrupt (to avoid
BCC	INTG, GIE	interrupt affecting subsequent fixed program flow)

JBC	INTG, GIE	;Determine whether the global interrupt is cleared
GOTO	\$-2	
MOVI	0x55	
MOVA	ROMCH	
.....		;8indivualNOPcommand, or wait8instruction cycle
MOVI	0xAA	
MOVA	ROMCH	
.....		;8indivualNOPcommand, or wait8instruction cycle
BSS	ROMCL, WR	
JBC	ROMCL,WR	;Wait for programming to end
GOTO	\$-1	
BSS	INTG, GIE	;enable global interrupt
.....		
MOVI	0x00	;quitIAPoperate
MOVA	ROMCL	
CLR	FRALN	
CLR	FRAHN	

Note: The program in the above application routine box is a fixed operation format and cannot be changed during use, otherwise it will causeIAPERase and program were unsuccessful.

### 3.3.6 special function register

#### 3.3.6.1 Look-up table address register low8bits (FRAL)

FRAL: Look-up table address register low8bit								
bit	7	6	5	4	3	2	1	0
name	FRAL<7:0>							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RESET	X	X	X	X	X	X	X	X

Bit 7~0 FRAL<7:0>: Lookup table address low8bit

#### 3.3.6.2 Look-up table address register high8bits (FRAH)

FRAH: look-up table address register high8bit								
bit	7	6	5	4	3	2	1	0
name	FRAH<7:0>							
R/W	—	—	—	R/W	R/W	R/W	R/W	R/W
RESET	X	X	X	X	X	X	X	X

Bit 7~0 FRAH<7:0>: lookup table address high8bit

**3.3.6.3** Look-up table address inversion register low8bits (FRALN)

FRALN: look-up table address inversion register low8bit								
bit	7	6	5	4	3	2	1	0
name	FRALN<7:0>							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RESET	0	0	0	0	0	0	0	0

Bit 7~0      FRALN&lt;7:0&gt;: look-up table address inversion low8bit

**3.3.6.4** Look-up table address inversion register high8bits (FRAHN)

FRAHN: look-up table address inversion register high8bit								
bit	7	6	5	4	3	2	1	0
name	FRAHN<7:0>							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RESET	0	0	0	0	0	0	0	0

Bit 7~0      FRAHN&lt;7:0&gt;: lookup table address inversion high8bit

**3.3.6.5** Look-up table data register low8bits (ROMDL)

ROMDL: Look-up table data register low8bit								
bit	7	6	5	4	3	2	1	0
name	ROMDL<7:0>							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RESET	X	X	X	X	X	X	X	X

Bit 7~0      ROMDL&lt;7:0&gt;: Lookup table data low8bit

**3.3.6.6** Look-up table data register high8bits (ROMDH)

ROMDH: look-up table data register high8bit								
bit	7	6	5	4	3	2	1	0
name	ROMDH<7:0>							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RESET	X	X	X	X	X	X	X	X

Bit 7~0      ROMDH&lt;7:0&gt;: lookup table data high8bit

**3.3.6.7** Memory Control Register Low8bits (ROMCL)

ROMCL: Memory Control Register Low8bit									
bit	7	6	5	4	3	2	1	0	
name	INFORDEN		IAPSEL<2:0>			FPEE	WREN	WR	INFORDTRG
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
RESET	0	0	0	0	0	0	0	0	

Bit7      INFORDEN:INFO1Area look-up table instruction read enable bit 0

:prohibit

1:Enable

Bit 6~4	IAPSEL<2:0>:IAP operation area selection 110 :INFO1information area 101:FLASHprogram area 100:FLASHdata area Other: ProhibitedIAP operate
Bit 3	FPEE:FLASHMemory Page Erase/Program Select Bits  0: programming 1: erase
Bit 2	WREN:FLASHMemory Page Erase/Program Enable Bit  0: forbidden ifFRAandFRANwhen the logical inversion is not satisfied, the hardware will automatically clearWREN 1 :Enable
Bit 1	WR:FLASHMemory Page Erase/Program Trigger Bit  0: The page erase/program operation is not initiated, or the operation has completed 1: Page Erase/Program operation in progress (auto-cleared by hardware)
Bit 0	INFORDTRG:INFO1Area lookup table read trigger bit  0:prohibit  1: Enable, the hardware will automatically clear after reading

### 3.3.6.8 Memory Control Register High8bits (ROMCH)

ROMCH: Memory Control Register High8bit								
bit	7	6	5	4	3	2	1	0
name	ROMCH<7:0>							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RESET	0	0	0	0	0	0	0	0

Bit 7~0 ROMCH<7:0>: Memory Page Erase/Program Control Word

Note: ROMCHThe register is a virtual register, and the read of this register is always full0.

### 3.4online programmingISP and online debuggingICD

FLASHmemory andFLASHThe data memory has the function of reprogramming, which is convenient for updating and upgrading customer codes and data. In order to enable developers to debug, update, and upgrade code more easily during the development process, this chip also supports online programmingISP and online debuggingICD, the user only needs to lead five programming and debugging interface lines on the circuit system board to realize reprogramming and debugging of the program, which is more convenient and efficient.

chip pin	Pin description
ISPCK	Program/Debug Serial Clock Port
ISPDA	Program/Debug Serial Data Port
MRSTN	reset pin
VDD	power supply
VSS	land

surface3-1In-circuit programming/debugging pin description

NOTE: For programming/debugging interfaceISPDAandISPCKpins, chip supportPC0andPC1As a set of programming/debugging interfaces.

## 3.5data storage

### 3.5.1

#### overview

-data storage by2Partial composition

- general purpose data storageGPR
- special function registerSFR

-Physical storage includes

- 1K Bytedata storage
- 128special register

-support3addressing mode

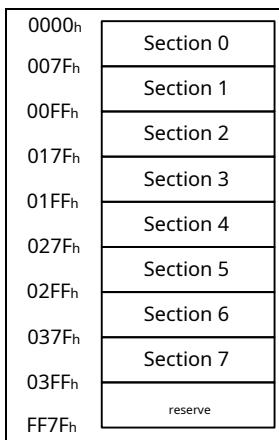
- direct addressing
- GPRspecial addressing
- indirect addressing

### 3.5.2

#### general purpose data storage

The general-purpose data memory is used to temporarily store data and control information, and can be read and written under program control. The general data memory space of this chip is1K Byte,support8bank group, the address range is0000h~03FFh. In the process of program control, when accessing these memory banks, it is necessary to pass the registerBKSRSelect the memory bank to realize the jump between different memory banks. The content of the general data memory is indeterminate after power-on reset, and the content before reset will be saved after other resets that are not powered off.

The address mapping is as follows:



picture3-5GPRSchematic diagram of address mapping

### 3.5.3

#### special function register

Special function registers are used to control and set the chip's peripheral operations. This chip supports128special registers, address rangeFF80h~FFFFh. Most of the registers can be read and written, and only a few registers are reserved for use, and user programs cannot read and write. Registers used by related functions are described in each chapter.

FF80h	IAD	FFA0h	SPICON0	FFC0h	T31C1H	FFE0h	RXB
FF81h	IAAL	FFA1h	SPICON1	FFC1h	T31C2L	FFE1h	RXC
FF82h	IAAH	FFA2h	SPIIE	FFC2h	T31C2H	FFE2h	TXB
FF83h	BKSR	FFA3h	SPIIF	FFC3h	T31IEL	FFE3h	TXC
FF84h	PSW	FFA4h	SPIRBR	FFC4h	T31IEH	FFE4h	BRR
FF85h	AREG	FFA5h	SPITBW	FFC5h	T31IDL	FFE5h	T21R1L
FF86h	PCRL	FFA6h	PWRC	FFC6h	T31IDH	FFE6h	T21R1H
FF87h	PCRH	FFA7h	WDTC	FFC7h	T31IVSL	FFE7h	T21R2L
FF88h	MULA/MULL	FFA8h	WKDC	FFC8h	T31IVSH	FFE8h	T21R2H
FF89h	MULB/MULH	FFA9h	PWEN	FFC9h	T31IFL	FFE9h	T21OC
FF8Ah	DIVEL/DIVQL	FFAAh	PA	FFCAh	T31IFH	FFEAh	T31CH2C
FF8Bh	DIVEH/DIVQH	FFABh	PAT	FFCBh	T31IFML	FFEBh	T31CH3C
FF8Ch	DIVS/DIVR	FFACh	PB	FFCCh	T31IFMH	FFECh	T31CH4C
FF8Dh	T31CH2RH	FFADh	PBT	FFCDh	T31ICRL	FFEDh	T31PINCL
FF8Eh	T31CH3RL	FFAEh	PC	FFCEh	T31ICRH	FFEEh	T31PINCH
FF8Fh	T31CH3RH	FFAFh	PCT	FFCFh	T31EVG	FFEFh	I2CX16
FF90h	FRAL	FFB0h	PAPU	FFD0h	T31CH1C	FFF0h	I2CC
FF91h	FRAH	FFB1h	PBPU	FFD1h	T21L	FFF1h	I2CSA
FF92h	ROMDL	FFB2h	PCPU	FFD2h	T21H	FFF2h	I2CTB
FF93h	ROMDH	FFB3h	T31CHBK	FFD3h	T21PL	FFF3h	I2CRB
FF94h	ROMCL	FFB4h	T31CH4RH	FFD4h	T21PH	FFF4h	I2CIEC
FF95h	ROMCH	FFB5h	PORTCTR	FFD5h	T21ROL	FFF5h	I2CIFC
FF96h	INTG	FFB6h	T31DLYT	FFD6h	T21ROH	FFF6h	T31CNTL
FF97h	INTP	FFB7h	PAPD	FFD7h	T21CL	FFF7h	T31CNTH
FF98h	INTCO	FFB8h	PBDP	FFD8h	T21CM	FFF8h	T31PRSL
FF99h	T31CH4RL	FFB9h	FRALN	FFD9h	T21CH	FFF9h	T31PRSH
FF9Ah	INTE0	FFBAh	FRAHN	FFDAh	ADCRL	FFFAh	T31CNTLDL
FF9Bh	INTFO	FFBBh	T8N	FFDBh	ADCRH	FFFBh	T31CNTLDH
FF9Ch	INTE1	FFBCh	T8NC	FFDCh	ADCCL	FFFCh	T31POS
FF9Dh	INTF1	FFBDh	T31C0L	FFDDh	ADCCH	FFFDh	T31CH1RL
FF9Eh	INTE2	FFBEh	T31C0H	FFDEh	ANSL	FFFEh	T31CH1RH
FF9Fh	INTF2	FFBFh	T31C1L	FFDFh	—	FFFFh	T31CH2RL

picture3-6special function register space

### 3.5.4 addressing mode

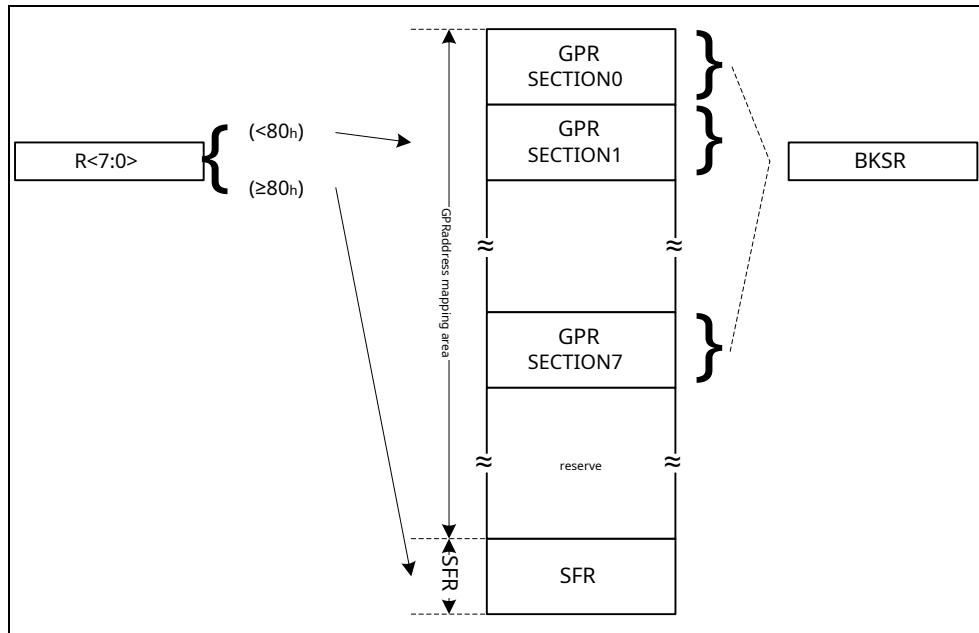
SRAMThe addressing mode of the data memory supports direct addressing,GPRSpecial addressing and indirect addressing.

#### 3. 5. 4. 1 direct addressing

The address information for direct addressing consists of two parts,BKSRand in the directive8bit address information.BKSUsed to select the bank group, the command in the8bit address information is used in theBKSAddressing within the selected bank group.

In direct addressing, when the instruction in the8bit address information greater than or equal to80hwill ignoreBKSWhile direct addressing SFRmap area. when the instruction8bit address information less than80hwhen accessingGPRaddress mapping area.

The schematic diagram is as follows:



picture3-7Direct Addressing Diagram

### 3. 5. 4. 2 GPRspecial addressing

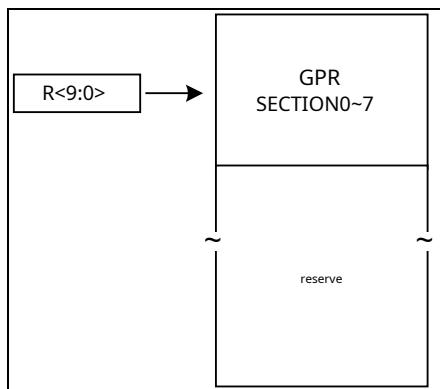
To facilitate larger data segments (such as arrays) in GPR movement in the command MOVAR and MOVRA used to pair GPR

For special addressing operations, the chip MOVAR and MOVRA command maximum support 10bit address information (

R<9:0>), directly addressable 1Kbyte address space. no need to SECTION switch between.

MOVAR and MOVRA command cannot be accessed SFR.

The schematic diagram is as follows:



picture3-8GPRSpecial Addressing Schematic

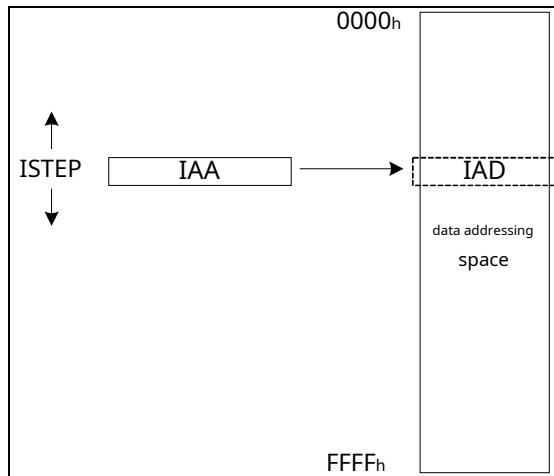
### 3. 5. 4. 3 indirect addressing

Indirect addressing is via 16bit indirect address register IAA (Depend on 2 individual 8bit register IAAH and IAAL composition) and 8Bit Dummy Data Register IAD, indirect access to storage units in the data addressing space. First store the access destination address in the indirect address register IAA, and then pass the instruction to IAD for read/write operations, the actual read/write operation object is IAA The destination address unit in the data addressing space pointed to.

IAD The registers themselves are also mapped into the data addressing space of the FF80h address, so when IAA The stored address value is FF80h

, read/write IAD Equivalent to accessing virtual registers with indirect addressing IAD itself, at which point a read operation will always read as 00h, the write operation is a no-op (may affect status bits) .

ISTEP instruction for 16bit indirect address register IAA Perform an offset operation. When executing this instruction, first put the 8bit signed immediate value is sign-extended to 16 number of digits, and then IAA The result of adding this number to the value of IAA register. ISTEP The achievable offset range is -128~127.



picture3-9 Indirect Addressing Diagram

### 3.5.5 special function register

#### 3.5.5.1 Indirect Addressing Data Register (IAD)

IAD: Indirect addressing data register								
bit	7	6	5	4	3	2	1	0
name	IAD<7:0>							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RESET	0	0	0	0	0	0	0	0

Bit 7~0      IAD<7:0>: indirect addressing data

#### 3.5.5.2 Indirect Addressing Index Register Low8bits (IAAL)

IAAL: Indirect Addressing Index Register Low8bit								
bit	7	6	5	4	3	2	1	0
name	IAAL<7:0>							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RESET	0	0	0	0	0	0	0	0

Bit 7~0      IAAL<7:0>: indirect addressing index low8bit

**3. 5. 3 Indirect Addressing Index Register High8bits (IAAH)**

IAAH: Indirect Addressing Index Register High8bit								
bit	7	6	5	4	3	2	1	0
name	IAAH<7:0>							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RESET	0	0	0	0	0	0	0	0

Bit 7~0      IAAH&lt;7:0&gt;: indirect addressing index high8bit

**3. 5. 4 Bank Select Register (BCSR)**

BCSR: Bank Select Register								
bit	7	6	5	4	3	2	1	0
name	—	—	—	—	—	DBCSR<2:0>		
R/W	—	—	—	R/W	—	R/W	R/W	R/W
RESET	0	0	0	0	0	0	0	0

Bit 7~5      Unused

Bit 4      Reserved bit, must be set by software too

Bit 3      Unused

Bit 2~0      DBCSR&lt;2:0&gt;: Data bank selection bit

000: Select memory bank0

001: Select memory bank1

010: Select memory bank2

011: Select memory bank3

100: Select memory bank4

101: Select memory bank5

110: Select memory bank6

111: Select memory bank7

Note: BCSR&lt;4&gt;bit must hold the reset value0, or the software is set to0.

## No.4 Chapter Input/Output Ports

### 4.1 overview

The input/output port is the most basic part of the chip, and the chip supports up to 18 input/output ports. all I/O ports are TTL/SMT input and CMOS output driver.

-PA Input/Output Port Functional Components

- 8bit bidirectional input/output port
- TTL/SMT input and CMOS output drive
- Port I/O Control Register (PAT)
- Port Weak Pull-Up Control Register (PAPU)
- Port Weak Pull-Down Control Register (PAPD)
- D/A Port Control Register (ANSL)

-PA0,PA3~7 Support external port interrupt function

-PB Input/Output Port Functional Components

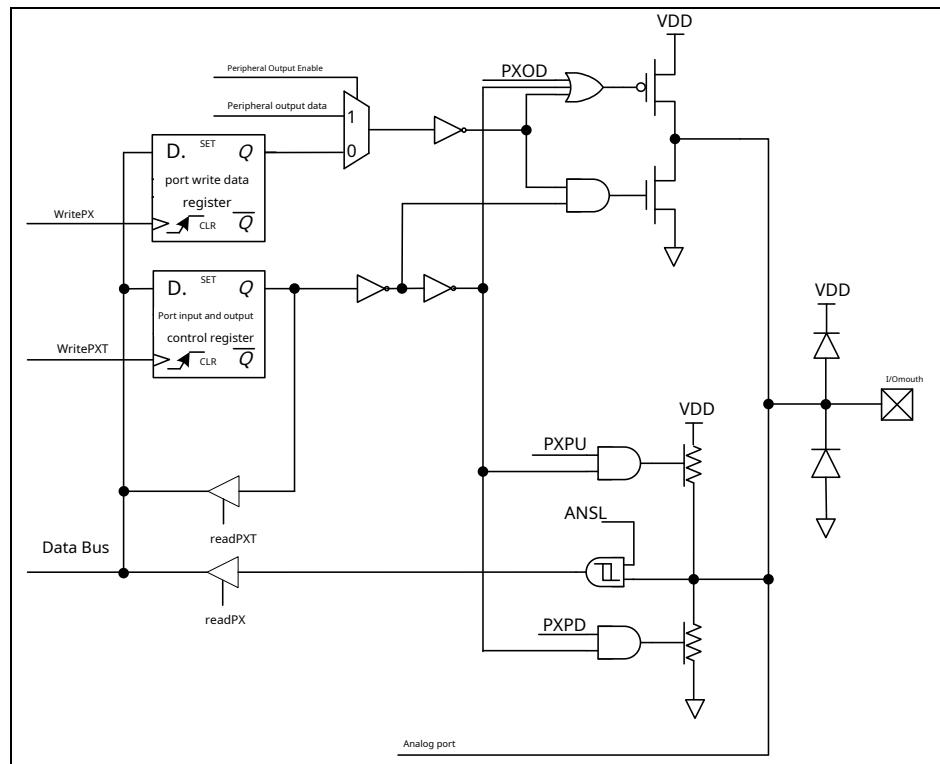
- 8bit bidirectional input/output port
- TTL/SMT input and CMOS output drive
- Port I/O Control Register (PBT)
- Port Weak Pull-Up Control Register (PBPU)
- Port Weak Pull-Down Control Register (PBPD)
- port PB0~1 With open-drain output function
- PB4~5,PB1 Support external port interrupt function

-PC Input/Output Port Functional Components

- 2bit bidirectional input/output port
- TTL/SMT input and CMOS output drive
- Port I/O Control Register (PCT)
- Port Weak Pull-Up Control Register (PCPU)

Note: When the port is set as an output or analog input port, the internal weak pull-up/pull-down is automatically disabled.

## 4.2 Structure diagram



picture4-1 PA/PB/PCPort structure diagram

Note1:PCThe port has no weak pull-down function.

Note2:removePB0/PB1Besides, other ports have no open-drain output function.

## 4.3 I/O Port function setting

### 4.3.1 I/O Port I/O Control

all in the chipI/OPorts have input/output capabilities, and the port control registerPAT/PBT/PCTInput or output function selection for the corresponding port. whenI/Owhen the port is set to digital output state,I/Oport outputPA/PB/PC register contents, that is, the correspondingI/OPort level status, readPA/PB/PCThe operation of the register is actually to read the correspondingI/O Port level status. whenI/Owhen the port is set to digital input state, readPA/PB/PCThe operation of the register is actually to read the correspondingI/OPort level status.

### 4.3.2 I/O Port weak pull-up/pull-down function

In the application of many products, the port needs to be connected with a pull-up or pull-down resistor to fix the port at a stable level state and prevent external interference and other influences. (PA3/MRSTNThe internal weak pull-up of the port is enabled by default)

pin	0	1	2	3	4	5	6	7
PA	support							
PB	support							
PC	support	support	—	—	—	—	—	—

surface4-1 I/O Port weak pull-up

pin	0	1	2	3	4	5	6	7
PA	support							
PB	support							
PC	—	—	—	—	—	—	—	—

surface4-2 I/Oport Weak Pulldown

#### 4.3.3 I/Oport analog/digital type selection function

When the digital signal and the analog signal share the pin, before using the digital signal or analog signal function of the corresponding port, the type of the port must be correctly set, otherwise the expected result may not be achieved. Part of this chip PA/PB port has independent analog/digital signal selection function by ANSLRegister control selection. When the port is configured as an analog port, read the corresponding PA/PBregister, always read "0".

#### 4.3.4 I/Oport open-drain output

port PB0~1 Support open-drain output function, which can be independently set to open-drain output, by PORTCTR register PBOD<1:0> bit control.

pin	0	1	2	3	4	5	6	7
PA	—	—	—	—	—	—	—	—
PB	support	support	—	—	—	—	—	—
PC	—	—	—	—	—	—	—	—

surface4-3 I/Oport open-drain output

#### 4.3.5 I/Oport multiplexing function

In order to optimize the rational use of resources, all the I/OAll ports have multiplexing functions. When the port is used for alternate function, the pin level is determined by the alternate function.

### 4.4 port interrupt

#### 4.4.1 External port interrupt (PINT)

This chip supports 9 external port interrupt. When PINT0~PINT8 The multiplexing port is configured as a digital input port, and when the input signal change meets the trigger condition, it will generate PINT0~PINT8 External port interrupt. It can be configured as rising edge trigger, falling edge trigger or both edge trigger. External port interrupts can be accessed by PIE0~PIE8 Enable. Interrupt generation will affect the corresponding interrupt flag PIFO~PIF8. exist IDLE mode, this interrupt can wake up CPU.

External Interrupt PINT5 and PINT8 can be triggered AD convert. When the register ADC\_ETR1EN(PORTCTR<3>) when enabled, the external interrupt PINT8 can be triggered AD conversion; when the register ADC\_ETR0EN(PORTCTR<2>) when enabled, the external interrupt PINT5 can be triggered AD conversion; ADC it must be enabled first and set to hardware sampling, ie ADCCL register ADEN and SMPSCONTROL bits need to be set to 1.

Pin name	port input	edge selection	interrupt name	interrupt enable	interrupt flag
PA0	PINT0	PEG0<1:0>	PINT0	PIE0	PIFO
PA3	PINT1	PEG0<1:0>	PINT1	PIE1	PIF1
PA4	PINT2	PEG1<1:0>	PINT2	PIE2	PIF2

Pin name	port input	edge selection	interrupt name	interrupt enable	interrupt flag
PA5	PINT3	PEG1<1:0>	PINT3	PIE3	PIF3
PA6	PINT4	PEG2<1:0>	PINT4	PIE4	PIF4
PA7	PINT5	PEG2<1:0>	PINT5	PIE5	PIF5
PB4	PINT6	PEG3<1:0>	PINT6	PIE6	PIF6
PB5	PINT7	PEG3<1:0>	PINT7	PIE7	PIF7
PB1	PINT8	PEG3<1:0>	PINT8	PIE8	PIF8

surface4-4external port interrupt

#### 4.5 I/O Port Operation Precautions

When executing an arithmetic or logical operation instruction (except bit operation instruction) targeting a port register, the chip actually performs a read-modify-write process, that is, read all I/OThe level of the port is modified and then written back to the port register. pair of bit manipulation instructionsI/O The modification operation only affects the selected bit, and the other bits in the same groupI/Odoes not affect. Therefore, it is recommended that usersI/OThe modification employs bit manipulation instructions. Also inI/OWhen the multiplexing function is enabled and disabled, the currentI/Ooutput register value of the port, and determine whether it is necessary to re-set theseI/OThe port is initialized and assigned.

#### 4.6special function register

##### 4.6.1PAPort Level Status Register (PA)

PA:PAPort Level Status Register								
bit	7	6	5	4	3	2	1	0
name	PA<7:0>							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RESET	X	X	X	X	X	X	X	X

Bit 7~0 PA&lt;7:0&gt;:PAPort level status

0: low level

1: high level

##### 4.6.2PAPort I/O Control Register (PAT)

PAT:PAPort I/O Control Register								
bit	7	6	5	4	3	2	1	0
name	PAT<7:0>							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RESET	1	1	1	1	1	1	1	1

Bit 7~0 PAT&lt;7:0&gt;:PAPort input and output status control bits

0: output state

1: input status

Note:PA3Only when configured asIOWhen it is used as a port, the input/output function is valid, and when it is used as an external reset pin, it is fixed as an input.

**4.6.3PAPort Weak Pull-Up Control Register (PAPU)**

PAPU:PAPort weak pull-up control register								
bit	7	6	5	4	3	2	1	0
name	PAPU<7:0>							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RESET	0	0	0	0	1	0	0	0

Bit 7~0 PAPU&lt;7:0&gt;:PAPort internal weak pull-up control bit

0:prohibit

1:Enable

**4.6.4PAPort Weak Pull-Down Control Register (PAPD)**

PAPD:PAPort weak pull-down control register								
bit	7	6	5	4	3	2	1	0
name	PAPD<7:0>							
R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W
RESET	0	0	0	0	0	0	0	0

Bit 7~0 PAPD&lt;7:0&gt;:PAPort internal weak pull-down control bit

0:prohibit

1:Enable

Note1:PA3Only when configured asIOWhen the port is connected, the weak pull-down function and input/output function are

valid; Note2:PA3When configured as an external reset pin, it is fixed as an input;

Note3:PA3The default weak pull-up is enabled, when configured as an external reset pin orIOAt any time, the weak pull-up enable can be disabled by software.

**4.6.5PBPort Level Status Register (PB)**

PB:PBPort Level Status Register								
bit	7	6	5	4	3	2	1	0
name	PB<7:0>							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RESET	x	x	x	x	x	x	x	x

Bit 7~0 PB&lt;7:0&gt;:PBport level status

0: low level

1: high level

**4.6.6PBPort I/O Control Register (PBT)**

PBT:PBPort I/O Control Register								
bit	7	6	5	4	3	2	1	0
name	PBT<7:0>							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RESET	1	1	1	1	1	1	1	1

Bit 7~0 PBT&lt;7:0&gt;:PBPort input and output status control bits

0: output state

1: input status

#### 4. 6. 7PBPort Weak Pull-Up Control Register (PBPU)

PBPU:PBPort weak pull-up control register								
bit	7	6	5	4	3	2	1	0
name	PBPU<7:0>							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RESET	0	0	0	0	0	0	0	0

Bit 7~0 PBPU<7:0>:PBPort internal weak pull-up control bit

0:prohibit

1:Enable

#### 4. 6. 8 PBPort Weak Pull-Down Control Register (PBPD)

PBPD:PBPort weak pull-down control register								
bit	7	6	5	4	3	2	1	0
name	PBPD<7:0>							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RESET	0	0	0	0	0	0	0	0

Bit 7~0 PBPD<7:0>:PBPort internal weak pull-down control bit

0:prohibit

1:Enable

#### 4.6.9 PCPort Level Status Register (PC)

PC:PCPort Level Status Register								
bit	7	6	5	4	3	2	1	0
name	—	—	—	—	—	—	PC<1:0>	
R/W	—	—	—	—	—	—	R/W	R/W
RESET	X	X	X	X	X	X	X	X

Bit 7~2 Unused

Bit 1~0 PC<1:0>:PC1~PC0port level status

0: low level

1: high level

#### 4.6.10 PCPort I/O Control Register (PCT)

PCT:PCPort I/O Control Register								
bit	7	6	5	4	3	2	1	0
name	—	—	—	—	—	—	PCT<1:0>	
R/W	—	—	—	—	—	—	R/W	R/W
RESET	0	0	0	0	0	0	1	1

Bit 7~2 Unused

Bit 1~0 PCT<1:0>:PC1~PC0Port input and output status control bits

0: output state

1: input status

**4.6.11 PCPort Weak Pull-Up Control Register (PCPU)**

PCPU:PCPort weak pull-up control register								
bit	7	6	5	4	3	2	1	0
name	—	—	—	—	—	—	PCPU<1:0>	
R/W	—	—	—	—	—	—	R/W	
RESET	0	0	0	0	0	0	0	0

Bit 7~2      Unused

Bit 1~0      PCPU&lt;1:0&gt;:PC1~PC0Port internal weak pull-up control bit

0:prohibit

1:Enable

**4.6.12 Port Special Function Control Register (PORTCTR)**

PORTCTR: Port Special Function Control Register								
bit	7	6	5	4	3	2	1	0
name	—	—	T31CH4EN	T31CH3EN	AD_ETR1EN	ADC_ETR0EN	PBOD<1:0>	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RESET	0	0	0	0	0	0	0	0

Bit 7~6      Reserved

bit5      T31CH4EN:T31aisle4option bit

0:PB3input capture, compare output,PWMoutput enable

1:PA4compare output,PWMoutput enable T31CH3EN:T31

bit4      aisle3option bit

0:PB2input capture, compare output,PWMoutput enable 1:PA3

compare output,PWMoutput enable ADC\_ETR1EN:External

bit3      Interrupt8triggerADCconversion enable bit 0:prohibit

1:Enable

bit2      ADC\_ETR0EN:External Interrupt5triggerADCconversion enable bit 0

:prohibit

1:Enable

Bit 1~0      PBOD&lt;1:0&gt;:PB1~PB0Port open-drain output control bit 0

:prohibit

1:Enable

Note: SpecificIOFor the drive capability of the port, please refer to the appendix3"Electrical Characteristics".

## No.5Chapter Special Functions and Operating Features

**5.1System Clocks and Oscillators****5.1.1**

## overview

The clock source required for chip operation is provided by the oscillator, and different oscillator options allow users to achieve a wider range of functions in different application requirements. The oscillators provided by this chip are2Type: Internal High SpeedRCoscillator (16MHz) and internal low speedRCoscillator (32KHz). Flexible selection of oscillators enables products to be optimized in terms of speed and power consumption. internal high speedRCIn addition to being a system clock source, the oscillator can also be used asADCConversion clock source; internal low-speed RCThe oscillator can be a watchdog timer, ADCmodule, T8NModules, etc. provide the required clock source and cannot be used as the system clock source.

**-INTHRC**

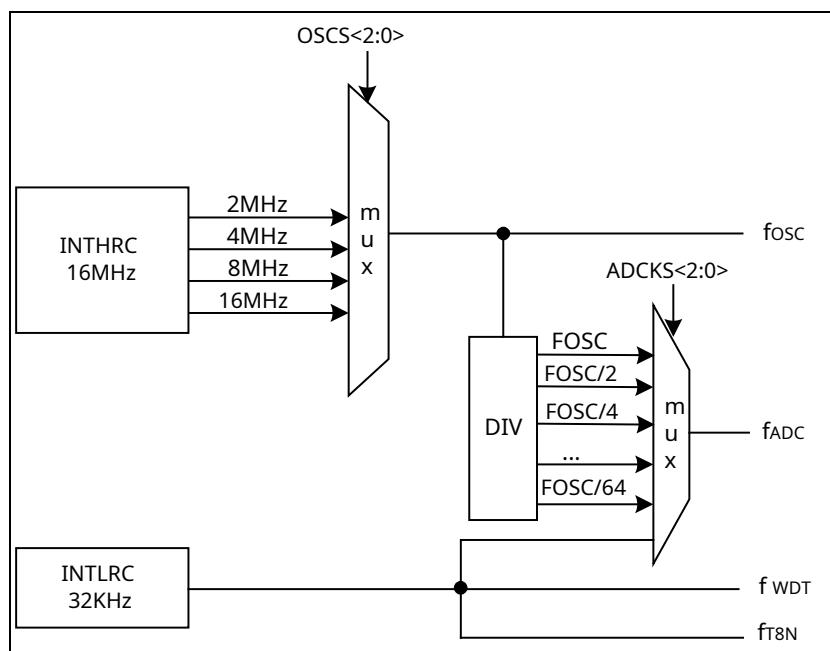
- internal16MHz RCoscillator
- Before leaving the factory, the internalRCThe oscillator frequency has been calibrated at room temperature, and the calibration accuracy is within  $\pm 1\%$ within
- Support multiple frequency division clocks, which can be selected through configuration words

**-INTLRC**

- internal32KHz RCoscillator
- WDTcounting clock
- T8Ncounting clock
- ADconversion clock
- Oscillation and Pause
  - existIDLE0mode, the main system clockINTHRCThe oscillator pauses oscillation,LDOssleep
  - existIDLE1mode, the main system clockINTHRCThe oscillator pauses oscillation,LDOsnormal work
  - existIDLE2mode, the main system clockINTHRCThe oscillator keeps vibrating,LDOsnormal work

**5.1.2**

## Internal structure map



picture5-1System Clock Structure Diagram

## 5.1.3 clock source

### 5. 1. 3. 1 internal high speed16MHz RCOscillator Mode (INTHRC)

chip built-in16MHz RCThe clock oscillator does not need to be connected to other external devices, and is used as the system clock source.

When the chip configuration wordOSCS<2:0> = 111,101,100or011when configured asINTOSCI0mode, at this time PB3Pins are multiplexed as general purposeI/Oport.

When the chip configuration wordOSCS<2:0> = 110when configured asINTOSCmode, at this timePB3Pin multiplexed as output CLKO,CLKOoutput system clock16Divided Clock (Fosc/16). Customers can choose through the programming interface.

Before leaving the factory, the chip has been calibrated at room temperature, within the working voltage range,INTHRCClock frequency calibration accuracy within ±1%within.

### 5. 1. 3. 2 internal low speed32kHz RCOscillator Mode (INTLRC)

chip built-in32KHz RCClock oscillator, no need to connect other external devices, can be used asWDT,ADCmodule, T8NThe clock source for the module.

Before leaving the factory, the chip has been calibrated at room temperature, within the working voltage range,INTLRCClock frequency calibration accuracy within ±5%within.

## 5.2 system power

### 5.2.1

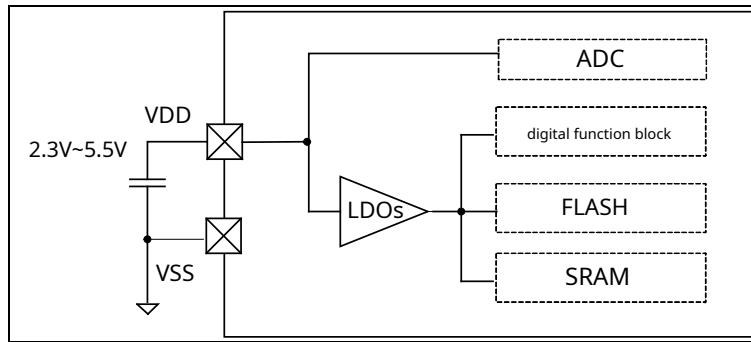
overview

The chip power supply is VDD, which corresponds to the reference ground of the chip VSS. VDD Give I/O port, ADC and inside LDOs Module powered, internal LDOs output voltage to the digital function module, FLASH memory, SRAM wait for the power supply.

LDOs The output voltage has been calibrated to 1.64V within ±20mV.

### 5.2.2

Internal structure map



picture5-2 System Power Structure Block Diagram

## 5.3 Watchdog Timer (WDT)

### 5.3.1 overview

The watchdog timer is an integral part of the chip, which can reset the chip when a software failure occurs. If the system enters a wrong working state, the watchdog can reset the chip within a reasonable time frame. When the watchdog is enabled, if the user program fails to clear the watchdog timer, the watchdog will reset the system within a predetermined time range.

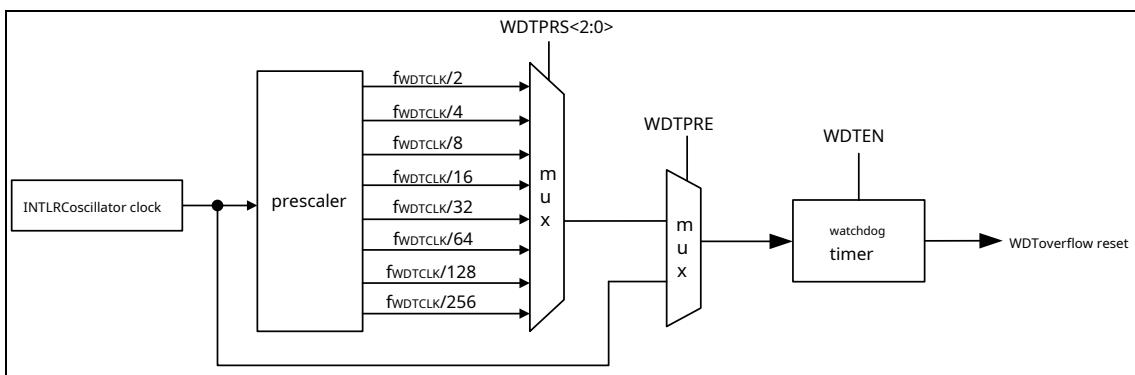
#### -WDTtimer

- 8bitWDTTiming counter (no actual physical address, not readable or writable)
- The timer clock source is the internal32KHz RCclock
- 8Bit prescaler (no actual physical address, not readable or writable)

#### -WDTControl Register (WDTC)

- wake up function
- reset function

### 5.3.2 Internal structure map



picture5-3Internal structure diagram of watchdog timer

### 5.3.3 WDTtimer

chip provided8bitWDTTiming counter, through the chip configuration wordWDTEHHardware watchdog can be enabledWDT. When the chip configuration wordWDTEHWhen enabled,WDTTimer count enable; whenWDTEHwhen off,WDTTimer counting is disabled. Customers can choose through the programmer interface.

existIDLEmode,WDTCount overflow will wake upCPU, do not reset the chip; the system is executingIDLEinstruction, the hardware will automatically clear theWDTcounter and restart counting.

In normal operating mode,WDTCount overflow will reset the chip. To avoid unnecessary resets, use theCWDIThe instruction is cleared at the right timeWDTcounter.

WDTSupports a prescaler byWDTin the registerWDTPREbit control. whenWDTPREbit is cleared to disable the prescaler at normal temperatureWDTThe count overflow time is approximately8ms.

whenWDTPRElocation1, when the prescaler is enabled, theWDTin the registerWDTPRS<2:0>bit set WDTThe prescaler ratio of the clock source, and then use the divided clock signal asWDTTimer count clock.

**5. 3. 4** special function register**5. 3. 4. 1** WDTControl Register (WDTC)

WDTC:WDTcontrol register								
bit	7	6	5	4	3	2	1	0
name	—	—	—	—	WDTPRE	WDTPRS<2:0>		
R/W	—	—	—	—	R/W	R/W	R/W	R/W
RESET	0	0	0	0	1	1	1	1

Bit 7~4      Unused

Bit 3      WDTPRE:WDTPrescaler Enable Bit

0:prohibit

1:Enable

Bit 2~0      WDTPRS&lt;2:0&gt;:WDTPrescaler division ratio selection bits

000:1:2

001:1:4

010:1:8

011:1:16

100:1:32

101:1:64

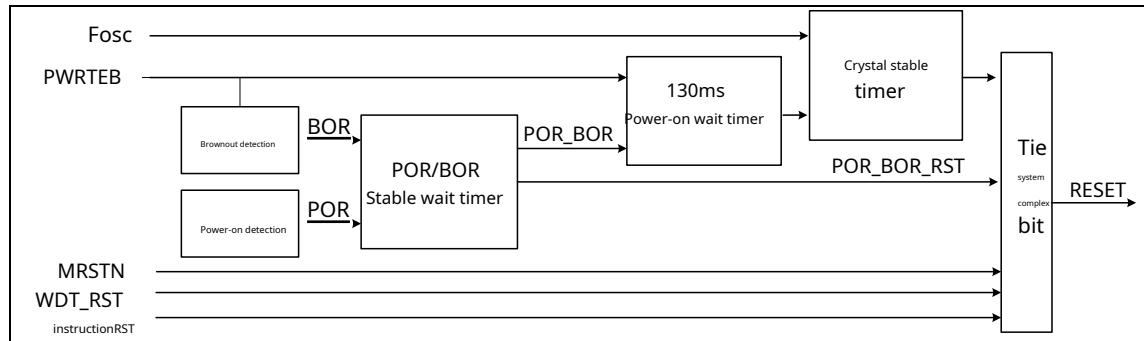
110:1:128

111:1:256

## 5.4reset module

### 5.4.1 overview

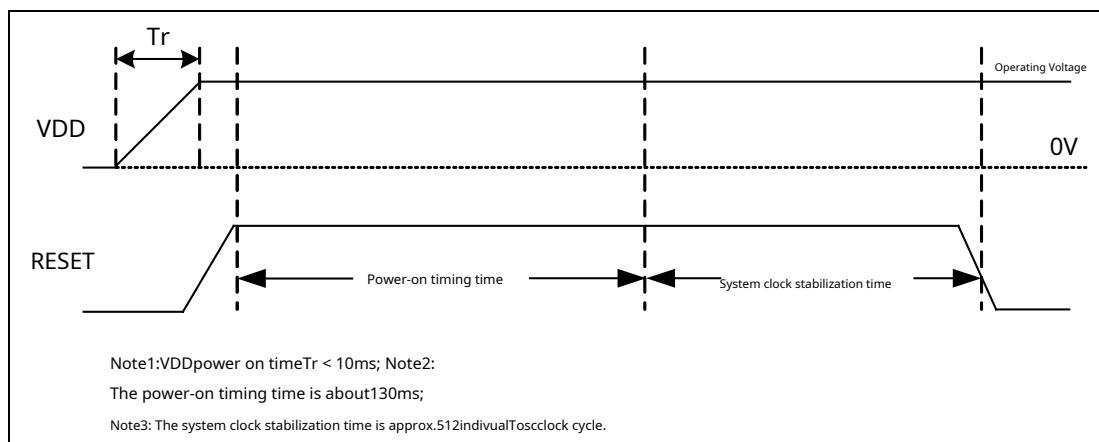
- power on resetPOR
- Brownout resetBOR, the reset voltage point is configurable
- external portMRSTNReset, active low reset
- watchdog timerWDTOverflow reset
- software execution instructionsRSTReset



picture5-4Schematic diagram of chip reset

### 5.4.2 power on reset

During chip power-up, aPORReset, and the reset signal will remain until the power supply voltage rises to the voltage where the chip can work normally. The power-on process of the system is in the form of a gradually rising curve, and it takes a certain amount of time to reach the normal level. The timing of power-on reset is as follows:

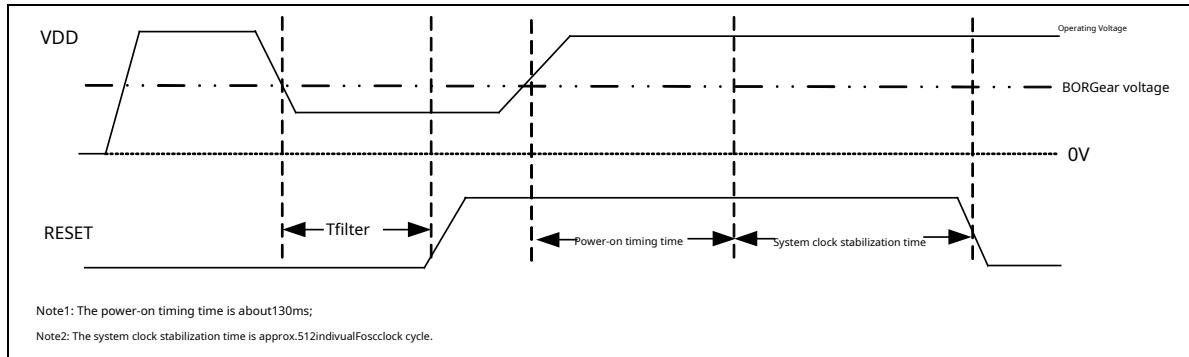


picture5-5Power-on reset timing diagram

### 5.4.3 Brownout reset

Power-down reset is aimed at system voltage drops caused by external factors (for example: battery replacement), which may cause abnormal system working status or program execution errors during power-down. The power-down reset circuit can ensure that the chip is reset during abnormal power-down status to avoid misuse. Filter time for voltage dips  $T_{filter}$ , available through the register PWEN<3:2>Make settings, according to the configured BOR. The low voltage gear and the power supply situation of the application system, choose the appropriate filter time,

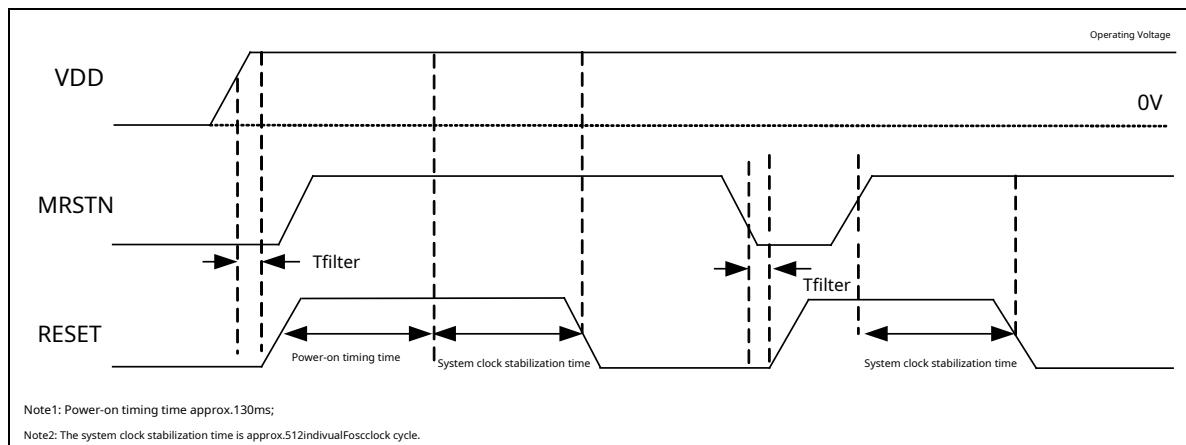
Usually left at the default.



picture5-6Low voltage reset timing diagram

#### 5. 4. 4 externalMRSTNpin reset

chip provides externalMRSTNPin for system reset. When the reset pin inputs a low level signal, the system is reset. When the reset pin is at a high level, the system operates normally. It should be noted that when the chip is configured as an external reset function, after the system is powered on, the external reset pin must input a high level, otherwise the system will remain in the reset state. It is also important to note that it is prohibited toMRSTNpins are directly connected to theVDDsuperior. External reset filter timeTfilter for200usLeft and right, can filter out the pulse width on the external reset pin is less than200usinterference pulse signal, to ensureMRSTN The external reset signal of the pin is valid, and its low-level pulse width must be greater than250us.



picture5-7externalMRSTNpin reset

Note1: When the chip configuration wordMRSTEN=1For external reset, the power-up timer can bePWRTEBshielded, and whenMRSTEN=0input for numbers

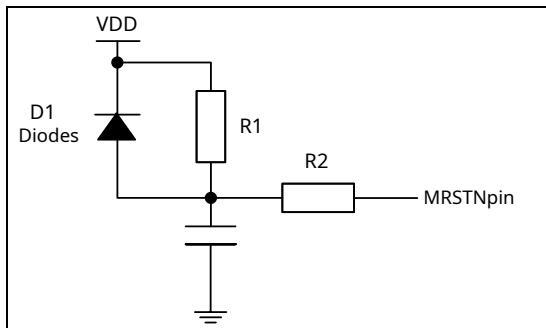
When the input and output ports are connected, the power-on timer is always enabled, and the timing time is about

130ms. Note2: When power on, the system clock stabilization time is about512individualFoscclock cycle.

externalMRSTNThere are many kinds of pin reset circuits, and the following two typical connection circuits are introduced.

##### 1. RCreset

RCThe reset circuit is externalMRSTNIt is the simplest kind of pin reset circuit, and it does not require high external environmental conditions. In some cases, this type of connection can be used.

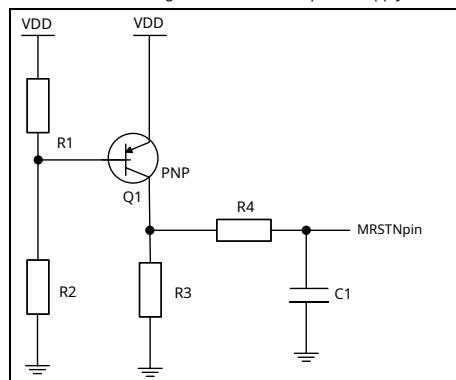


picture5-8 MRSTNReset reference circuit diagram1

Note: SamplingRCreset, where  $47\text{K}\Omega \leq R1 \leq 100\text{K}\Omega$ , capacitance  $C1(0.1\mu\text{F})$  ,  $R2$  is the current limiting resistor,  $0.1\text{K}\Omega \leq R2 \leq 1\text{K}\Omega$ .

## 2. PNPTransistor reset

PNPThe triode reset circuit is suitable for occasions with strong interference to the power supply.

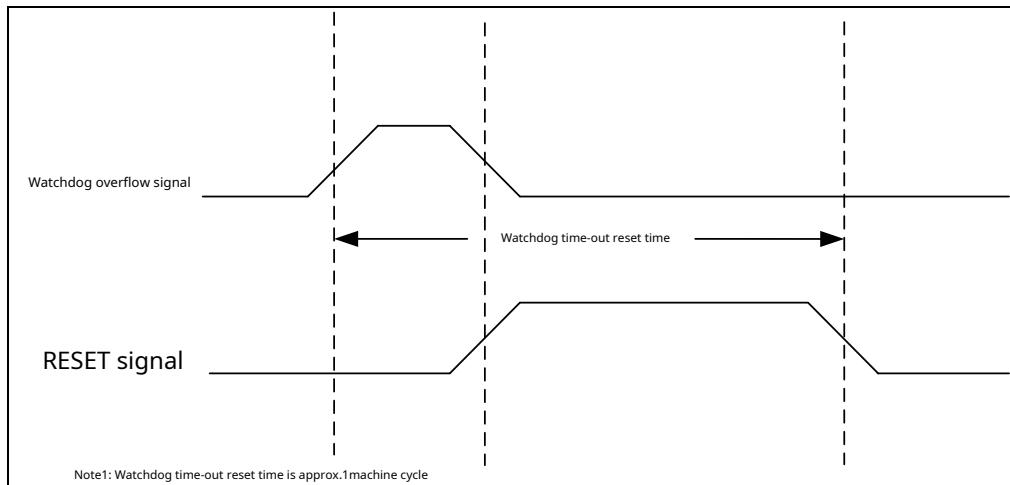


picture5-9 MRSTNReset reference circuit diagram2

Note: using PNPTransistor reset by  $R1(2\text{K}\Omega)$ and $R2(10\text{K}\Omega)$  divided voltage as the base input, the emitter connectedVDD, the collector all the way through  $R3(20\text{K}\Omega)$  to ground, the other via $R4(1\text{K}\Omega)$ and $C1(0.1\mu\text{F})$  to ground, $C1$ the other end asMRSTNenter.

## 5. 4. 5 Watchdog timer overflow reset

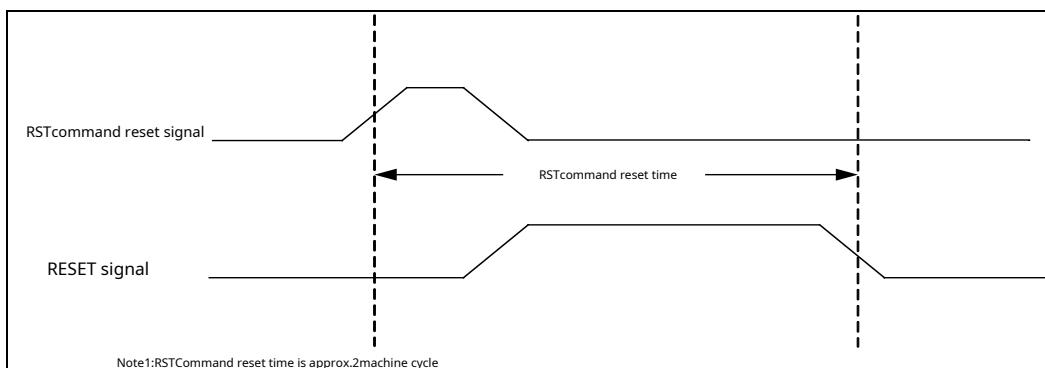
Watchdog reset is a protection setting for the system. Under normal conditions, the watchdog timer is cleared by the program. If an error occurs, the system is in an unknown state, and the program cannot clear the watchdog, which will cause the watchdog timer to overflow and cause a system reset. After the watchdog time-out reset, the system restarts into normal state.



picture5-10 Watchdog time-out reset

#### 5.4.6 RSTcommand software reset

The entire chip can be implemented by RST instruction to reset, after reset, all register status bits will be affected.



picture5-11 RSTcommand software reset

#### 5.4.7 special function register

##### 5.4.7.1 Power Control Register (PWRC)

PWRC: Power Control Register								
bit	7	6	5	4	3	2	1	0
name	LPM<1:0>		—	—	N_TO	N_PD	N_POR	N_BOR
R/W	R/W	R/W	—	R/W	R/W	R/W	R/W	R/W
RESET	0	0	0	1	1	1	0	0

Bit 7-6 LPM<1:0>: Sleep mode select bit

00:IDLE0mode, deep sleep mode 01:IDLE1mode,  
light sleep mode 10/11:IDLE2mode, fast wake-up  
sleep mode reserved unused

Bit 5-4

Bit 3 N\_TO:WDTOverflow flag

0:WDT Cleared when count overflows  
1: power-on reset or executeCWDT, IDLESet after the command1

Bit 2	N_PD: Low power consumption flag bit 0:implementIDLEcleared after command 1: power-on reset or executeCWDTpst-command1
Bit 1	N_POR: Power-on reset status bit 0: Power-on reset occurs (after power-on reset, it must be set by software) 1: No power-on reset occurs
Bit 0	N_BOR: Low voltage reset status bit 0: Low voltage reset occurs (after low voltage reset, it must be set by software) 1: No low voltage reset occurs

#### 5. 4. 7. 2 Power Control Register (PWEN)

PWEN: Power Control Register								
bit	7	6	5	4	3	2	1	0
name	ADVREFS2	—	MRSTF	PORLOST	BORFLT<1:0>	RCEN	HALT_PWM	
R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W
RESET	0	0	0	0	1	0	1	1

Bit 7 ADVREFS2:A/DReference source selection bit, this bit needs to be combined with the registerADVREFS<1:0>(ADCCH<1:0>)

At the same time to set up, in order to select the correct reference source. For details, please refer toADCCHregister.

Bit 6 Unused

Bit 5 MRSTF: External reset status bit

0: An external reset has occurred (after an external reset, it must be set by software) 1: No external reset occurred

Bit 4 PORLOST: power-on reset failure status bit

0: A power-on reset has occurred  
1: Power-on reset did not occur

Bit 3~2 BORFLT<1:0>:BORfilter time selection bit

00:about3indivialINTLRCclock cycle  
01:about5indivialINTLRCclock cycle  
10:about7indivialINTLRCclock cycle  
11:about9indivialINTLRCclock cycle

Bit 1 RCEN:IDLEstate,WDTcount enable bit (notIDLEstate is fixed to 1)

0:closureWDT  
1:EnableWDT

Bit 0 HALT\_PWM: In the paused state of online debugging,PWMoutput enable bit 0

:Enable  
1:prohibit

## 5.Low power operation

### 5.5.1 MCULow power mode

The chip supports three sleep modes:IDLE0model, IDLE1and IDLE2model.

#### -IDLE0deep sleep mode

- The clock source stops oscillating, and the main system clock is suspended
- Program pause, synchronous block pause, asynchronous block run, device power reduction
- LDOsSleep, need stabilization time after waking up
- Support low-power wake-up, wake-up time can be configured
- allI/Oport will remain in theIDLE0state before mode
- If enabledWDT, butWDTwill be cleared and keep running
- N\_PDbit is cleared,N\_TObit is set1

#### -IDLE1light sleep mode

- The clock source stops oscillating, and the main system clock is suspended
- Program pause, synchronous block pause, asynchronous block run, device power reduction
- LDOsnormal work
- Support low-power wake-up, wake-up time can be configured
- allI/Oport will remain in theIDLE1previous state
- If enabledWDT, butWDTwill be cleared and keep running
- N\_PDbit is cleared,N\_TObit is set1

#### -IDLE2Quick wake-up from sleep mode

- Clock source is active, main system clock is suspended
- Program pause, synchronous block pause, asynchronous block run, device power reduction
- LDOsnormal work
- Support fast wake-up
- allI/Oport will remain in theIDLE2previous state
- If enabledWDT, butWDTwill be cleared and keep running
- N\_PDbit is cleared,N\_TObit is set1

### 5.5.2

#### Low Power Mode Configuration

Three Low Power ModesIDLE0, IDLE1and IDLE2mode selection byPWRCin the registerLPM<1:0> bit control. whenLPM = 00when, executeIDLEinstruction, the chip entersIDLE0mode; whenLPM = 01when, executeIDLE instruction, the chip entersIDLE1mode; whenLPM = 10or11when, executeIDLEinstruction, the chip enters IDLE2model.

LPM (PWRC<7:6>)	low power mode
00	IDLE0model
01	IDLE1model
10/11	IDLE2model

surface5-1 Low Power Mode Configuration Table

To reduce power consumption, allI/OBoth pins should remain high or low. Introduced to avoid floating input pins

switching current, an external high-impedance input should be I/O. The pin is connected to a high level or a low level through a pull-up or pull-down resistor, MRSTN pin must be at a logic high level. If the product package pin count is less than the maximum pin count, the unlead and unused I/O pins need to be set to output low level.

### 5. 5. 3 IDLE Wake-up configuration

When the system enters the low power consumption mode, the program is suspended, and the system can be woken up in the following ways.

serial number	wake up method	interrupt mask	interrupt enable	interrupt mode	Remark
1	MRSTN	—	—	—	—
2	WDT	—	—	—	WDT overflow
3	PINTx	—	PIEX	default/vector	—
4	I2CINT	—	I2CIE	default/vector	only in IDLE2 wake-up mode

surface5-2Wake-up mode configuration table

Note: Low power wakeup has nothing to do with global interrupt enable. In low-power mode, if the peripheral generates an interrupt signal, even in the default interrupt mode, the global interrupt enables able GIEfor0, or in vectored interrupt mode, high priority interrupt enable GIEand low priority interrupt enable GIEL both 0, the low power consumption mode will still be woken up, but the interrupt program will not be executed after waking up.

### 5. 5. 4 Wake-up Timing Diagram

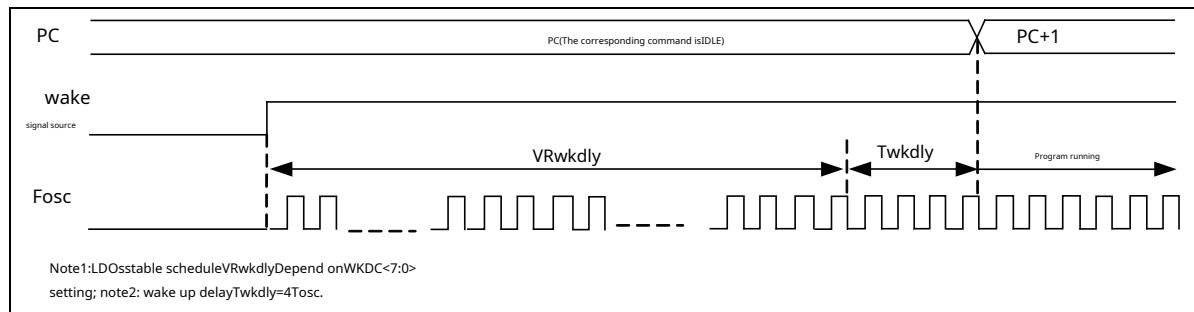
When a wake-up event occurs, the chip wake-up time is determined by LDOs stable schedule (VRwkdy) and wake-up delay (Twkdy) consists of two parts.

- existIDLE0model(LPM=00), the chip needs to wait for VRwkdy time (by WKDC<7:0> set up), this time is called LDOs Stabilization time, minimum 64 individual 16MHz clock cycle (tinthrc). Of After the master clock of the chip is stable for a period of time Twkdy execute after time IDLE the next instruction, Twkdy is called the wake-up delay, and the wake-up delay is 4 individual Tosc system clock cycle;
- existIDLE1model(LPM=01) down, no LDOs Stabilization time, only wake-up delay, wake-up delay is 16 individual Tosc system clock cycle.
- existIDLE2model(LPM=10 or 11) down, no LDOs Stabilization time, no wake-up delay, quick wake-up.

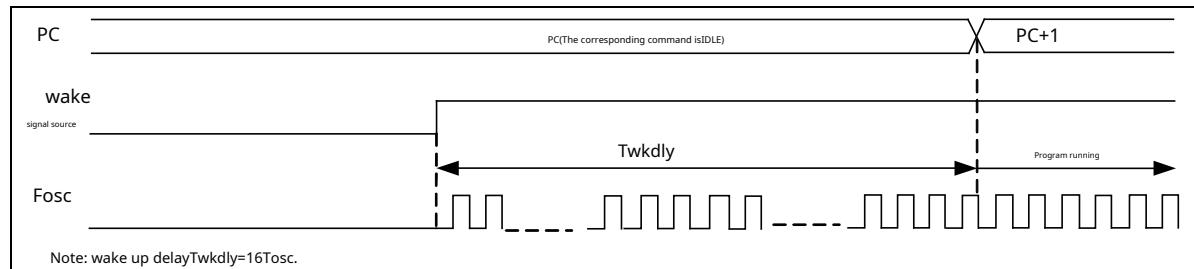
low power mode	wake up time	
IDLE0model	VRwkdy	(WKDC<7:0>) × 4 × Tinthrc
	Twkdy	4Tosc
IDLE1model	VRwkdy	0
	Twkdy	16Tosc
IDLE2model	VRwkdy	0
	Twkdy	0

surface5-3Wake Up Time Calculation Table

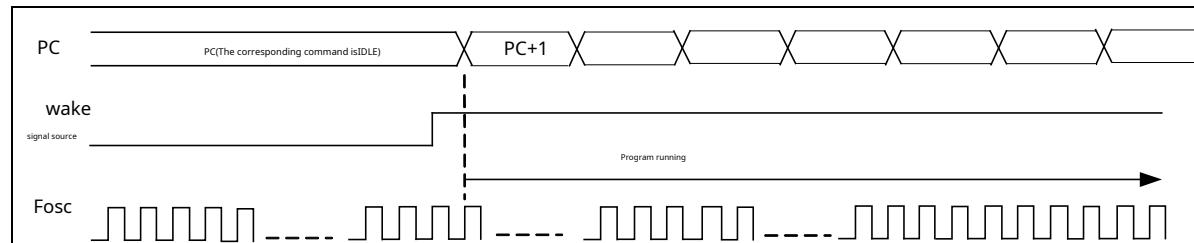
Note: The above wake-up time is a theoretical value, tinthrc for internal 16MHz clock cycle, internal 16MHz The clock frequency before the oscillator stabilizes may be lower than the 16MHz, subject to the actual chip.



picture5-12 system wake up IDLE0Timing diagram of



picture5-13 system wake up IDLE1Timing diagram of



picture5-14system wake up IDLE2Timing diagram of

## 5.5.5 special function register

### 5.5.5.1 Wake-Up Delay Control Register (WKDC)

WKDC: Wake-up Delay Control Register								
bit	7	6	5	4	3	2	1	0
name	WKDC<7:0>							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RESET	1	1	1	1	1	1	1	1

Bit 7~0 WKDC&lt;7:0&gt;: Wake-up delay time setting bit

0Fh: The shortest delay (set value less than 0Fh is invalid, still equivalent to 0Fh)

...

FFh: the longest delay

## No.6Chapter Peripherals

## 6. 1 8bit timer/counter (T8N)

### 6.1.1

overview

The 8bit timer/counter includes two working modes of timer and counter. The timer mode is timed according to the timing time set by the register, which can make the timer selectively generate interrupt requests or complete other operations. Counter mode is used for external clock signals (T8NCKI) to count.

-T8NSupport two working modes

- Timer mode (the clock source is the system clock frequency divided by two (Fosc/2)orINTLRCclock)
- Synchronous counter mode (clock source is external input clockT8NCKI)

-T8NThe following functional components are supported

- 8Bit prescaler (no actual physical address, not readable or writable)

- 8Bit Counter Register (T8N)

- 8bit control register (T8NC)

-break and pause

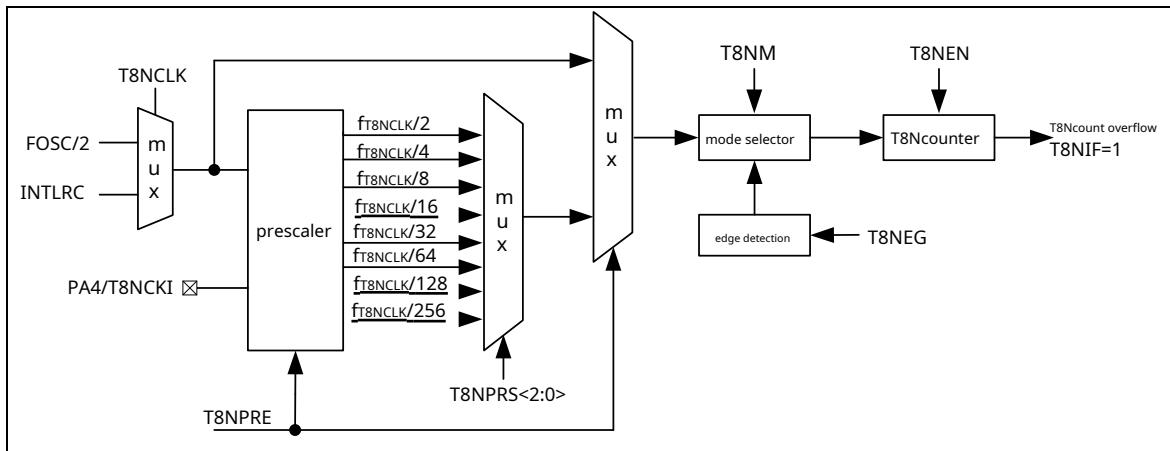
- Support overflow interrupt flag (T8NIF)

- Support interrupt handling

- existIDLEmode,T8Npause

### 6.1.2

Internal structure map



picture6-1 T8NInternal structure map

### 6.1.3

prescaler

A prescaler can provide a longer overflow period of the timer/counter. whenT8NCin the registerT8NPRE for"1"when enableT8N prescaler. any pairT8NThe write operation of the counter will clear the prescaler, and it will be updated immediately after rewriting, but it will not affect the frequency division ratio setting of the prescaler, and the count value of the prescaler cannot be read or written. The division ratio of the prescaler can be set byT8NCin the registerT8NPRS<2:0>bit to be set, the prescaler range is 1:2~1:256.

Note1: when usingINTLRCWhen it is the count clock, the prescaler control bit must be enabled.

Note2: It is not recommended to useT8NWhen counting, rewriteT8NCounter and prescaler value, otherwise it will affect the time of the first count after rewriting.

Operating mode	T8NPRE	T8NPRS<2:0>	T8Ncounting clock	
			T8NCLK=0	T8NCLK=1
timer mode	0	—	Fosc/2	—
	1	000	(Fosc/2) /2	Fintlrc /2
	1	001	(Fosc/2) /4	Fintlrc /4
	1	010	(Fosc/2) /8	Fintlrc /8
	1	011	(Fosc/2) /16	Fintlrc /16
	1	100	(Fosc/2) /32	Fintlrc /32
	1	101	(Fosc/2) /64	Fintlrc /64
	1	110	(Fosc/2) /128	Fintlrc /128
	1	111	(Fosc/2) /256	Fintlrc /256
Operating mode	T8NPRE	T8NPRS<2:0>	T8Ncounting clock	
counter mode	0	—	T8NCKI	
	1	000	T8NCKI /2	
	1	001	T8NCKI /4	
	1	010	T8NCKI /8	
	1	011	T8NCKI /16	
	1	100	T8NCKI /32	
	1	101	T8NCKI /64	
	1	110	T8NCKI /128	
	1	111	T8NCKI /256	

surface6-1 T8NPrescaler Configuration Table

### 6.1.4 Operating mode

T8NThere are two working modes, timer mode and counter mode, throughT8NMMAke a selection. Both timer and counter counting modes support prescaler. When configured in timer mode, theT8NThe clock source for the counter is available through theT8NC in the registerT8NCLKbit selection for the system clock2frequency division (Fosc/2)orINTLRC; When configured as counter mode,T8N The clock source of the counter is the system clock after frequency division by twoFosc/2Synchronized external clock input T8NCKI ,thereforeT8NCKIThe high level and low level time of the input clock signal are at least one machine cycle. passt8NCin the register T8NEGThe bit selects the counting edge of the external clock as rising or falling.T8NCKI where theIOThe port must be configured as a digital input.

T8NM	T8NCLK	Operating mode	clock source
0	0	timer mode	Fosc/2
0	1	timer mode	INTLRC
1	0	Synchronous Counter Mode	T8NCKI

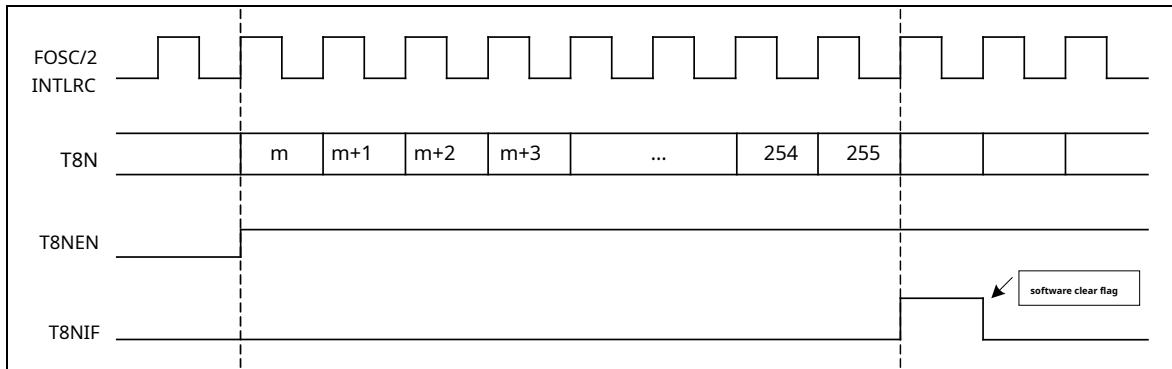
surface6-2 T8NWorking mode configuration table

### 6.1.5 timer mode

T8NThe counter counts up, and the count value is determined byFFhbecomes00hhour,T8NThe counter overflows and restarts counting.T8NWwhen the counter overflows, the interrupt flagT8NIFbit is set"1",produceT8Noverflow interrupt. existCPU

After entering sleep mode, the T8N module is inactive, so no interrupt is generated.

when T8N When configured in timer mode, if the prescaler is disabled, the T8N The counter clock can only be selected as the system clock divided by two (Fosc/2), cannot be selected as INTLRC; If the prescaler is enabled, the divider pair Fosc/2 or INTLRC For frequency division, at this time, T8N The count clock of the counter is a clock after frequency division.



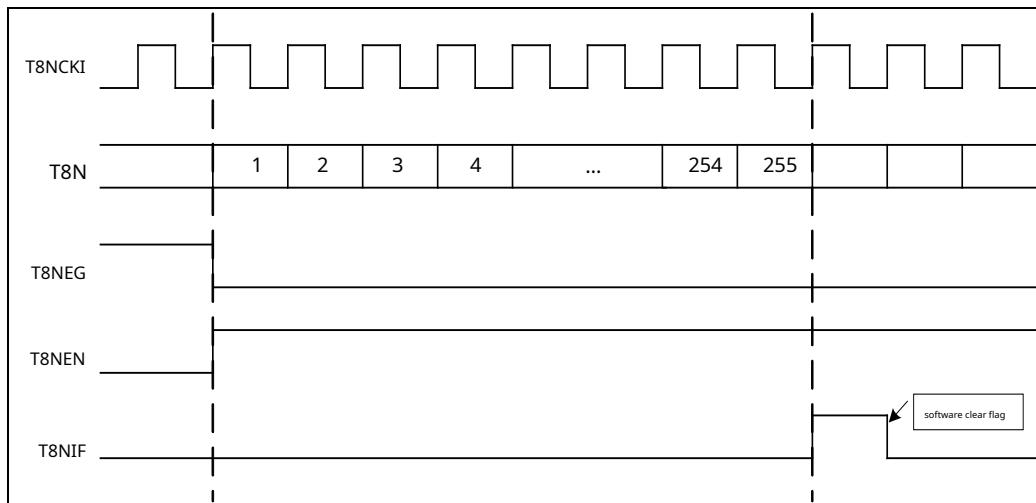
picture6-2Timer Mode Timing Diagram

### 6. 1. 6 Synchronous Counter Mode

when T8N When configured in synchronous counter mode, if the prescaler is disabled, T8N The clock of the counter is an external input clock T8NCKI, the internal phase clock p2 will clock T8NCKI to sync. so T8NCKI The time to keep the high level or low level is at least one machine cycle. by setting T8NEG (T8NC<4>) Select the counting edge of the external clock as rising or falling.

Similarly, synchronous counter mode also supports prescaler for external clock T8NCKI Carry out frequency division. and, T8NCKI Multiplexed IO The port must be configured as a digital input.

T8N The counter counts up, and the count value is determined by FFH becomes 00H hour, T8N The counter overflows and restarts counting. T8N When the counter overflows, the interrupt flag T8NIF bit is set "1", produce T8N overflow interrupt. exist CPU After entering sleep mode, the T8N module is inactive, so no interrupt is generated.



picture6-3Counter Mode Timing Diagram (T8NEG=0,T8NCKI Rising edge count)

### 6.1.7 special function register

8Bit Timer/Counter T8N Controlled by two registers, a 8bit counter register T8N and a control post

memoryT8NC.T8NThe register is used to store the count value,T8NCThe control register is used to control theT8N enabling,T8N mode selection,T8NCKICounting edge selection, prescaler enable bit, and prescaler division ratio selection.

### 6. 1. 7. 1 T8Ncounter register (T8N)

T8N:T8Ncounter register								
bit	7	6	5	4	3	2	1	0
name	T8N <7:0>							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RESET	0	0	0	0	0	0	0	0

Bit 7~0      T8N <7:0>:8bitT8Ncount value

### 6. 1. 7. 2 T8NControl Register (T8NC)

T8NC:T8Ncontrol register								
bit	7	6	5	4	3	2	1	0
name	T8NEN	T8NCLK	T8NM	T8NEG	T8NPRE	T8NPRS<2:0>		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RESET	0	0	0	0	0	0	0	0

Bit 7      T8NEN:T8Nmodule enable bit

0:closure

1:Enable

Bit 6      T8NCLK:T8NTiming clock source selection bit

0: The system clock is divided by twoFosc/2

1:INTLRCclock (must be enabled at the same timeT8NThe prescaler control bits, that isT8NPRE=1)

Bit 5      T8NM:T8Nmode selection bit

0: Timer mode

1: Synchronous Counter Mode

Bit 4      T8NEG:T8NCKISynchronous count edge select bit

0:T8NCKIRising edge count 1:

T8NCKIFalling edge count

Bit 3      T8NPRE: Prescaler enable bit

0:prohibit

1:Enable

Bit 2~0      T8NPRS<2:0>: Prescaler division ratio selection bit

000:1:2

001:1:4

010:1:8

011:1:16

100:1:32

101:1:64

110:1:128

111:1:256

## 6. 2 16Bit Multi-function Timer (T21)

### 6.2.1

overview

16bit multifunction timer T21 in total 4 Two working modes: timer mode, multi-precision PWM mode, capture mode, comparator mode.

#### -T21 support 4 working mode

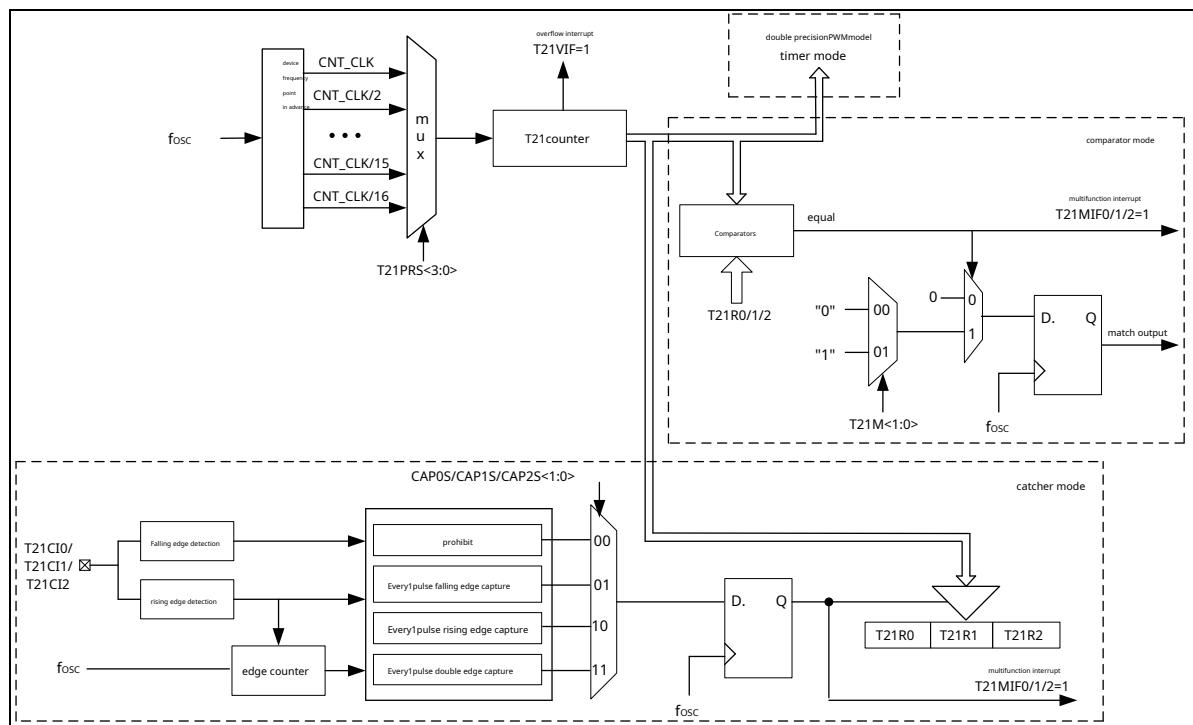
- Timer mode (clock source is Fosc)
- multi-precision PWM mode, supports 3 independent PWM output, output duty cycle, polarity can be set independently
- Catcher mode, supports 3 independent capture input, each input capture edge can be set separately
- Comparator mode, supports 3 independent compare output
- T21 The following functional components are supported
  - 4bit prescaler and 7bit postscaler (no actual physical address, not readable or writable)
  - 16bit counter T21 (the initial value of the counter can be written)
  - 16Bit capture/compare/precision registers T21R
  - 16bit period register T21P
  - Multifunction Control Register T21CL, T21CM, T21CH
  - Multi-function input and output control register T21OC
  - break and pause
  - Support overflow interrupt T21VIF, cycle matching interrupt T21PIF and multifunction interrupt T21MIF
  - exist IDLE mode, stop working

Note1: The precision register and the period register are updated, while writing the low8bit register is loaded, the value of the high register is loaded at the same time, so when the register is updated, it is necessary to write high first 8bit register, then write low 8bit register;

Note2: in the pin diagram T21\_CH0/T21\_CH1/T21\_CH2 Respectively represent the multi-precision PWM mode output channel PWM210/PWM211/PWM212, capture input mode channel T21\_CI0/T21\_CI1/T21\_CI2, Output channel of comparison output mode T21\_CO0/T21\_CO1/T21\_CO2;

Note3: T21R middle no represent 0, 1, 2.

## 6.2.2 Internal structure map



picture6-4 T21Internal structure map

## 6.2.3 Prescaler and Postscaler

The prescaler can provide a longer overflow period. T21Supports configurable prescaler. passT21CMIn the registerT21PRSBits configure the frequency division ratio of the prescaler, and the range of the prescaler ratio is1:1 ~ 1:16. Any write to the counter or control registers will clear the prescaler and postscaler, but will not change the configured divider ratio. The count value of the prescaler and postscaler cannot be read or written.

The divider ratio of the postscaler can be set byT21CHin the registerT21POS<6:0>bit to be set, the postscaler range is 1:1~1:128, by matching the counter with the period register value to perform post-division.

## 6.2.4 Operating mode

T21have4Two working modes, timer mode, multi-precisionPWMmode, capture mode, comparator mode, through T21M<3:0>Make mode selection.

T21ENplace1Before enabling, theT21MSet the working mode, select the counting clock, and configure parameters such as prescaler, postscaler, period, etc., and ensure that the counting clock is atT21ENStable when enabled.

## 6.2.5 timer mode

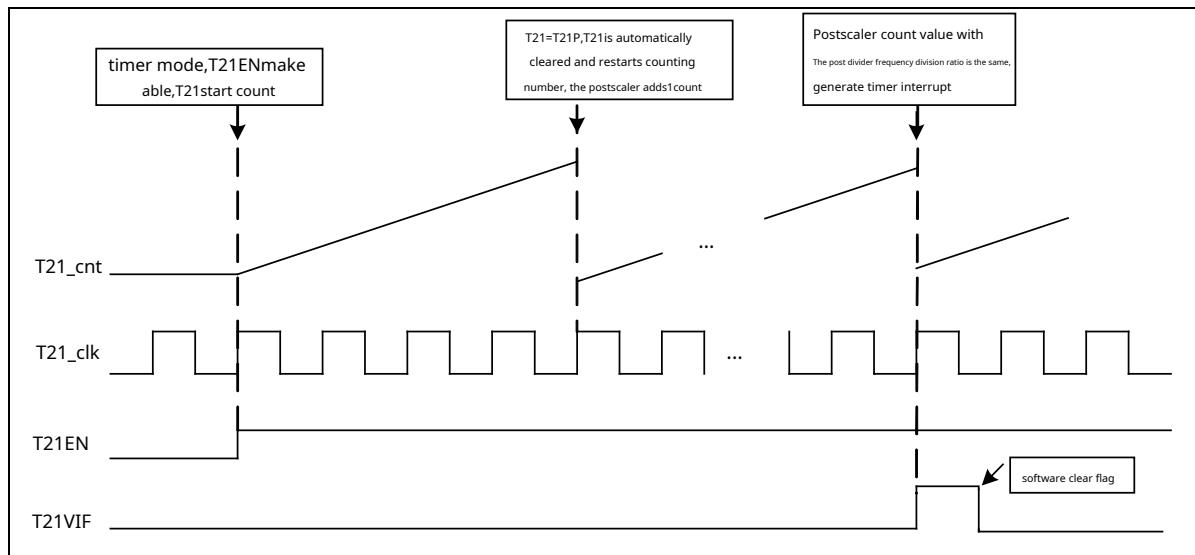
whenT21Mfor0000hour,T21Work in timer mode.

T21The clock source of the counter is the system clockFosc, and supports prescaler and postscaler.

T21The counter is a readable and writable register, which supports the setting of the initial value of the count and the change of the count value during the counting process.

T21ENWhen enabled,16bit timerT21counts up the count clock whenT21The count value and the period registerT21PWhen equal, the postscaler counter adds1,at the same timeT21The counter is automatically cleared and restarts counting.

When the count value of the post-scaler is the same as the frequency-division ratio of the post-scaler, the post-scaler is reset and the timer overflow interrupt flag T21VIFset "1", the interrupt flag needs to be cleared by software.



picture6-5 T21Timer Mode Timing Diagram

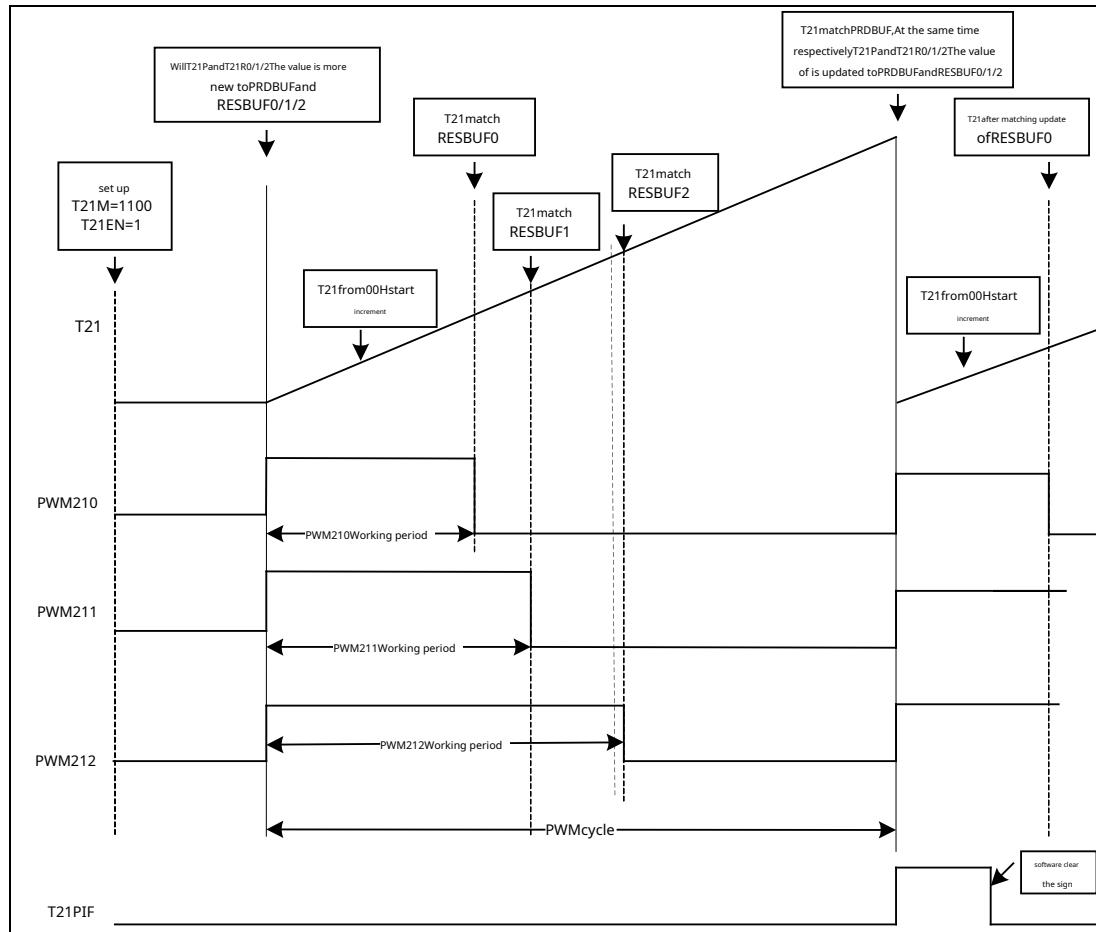
## 6. 2. 6 multi-precisionPWMmodel

T21Mfor"1100"hour,T21work in multiple precisionPWMmode, support3roadPWMoutputPWM210, PWM211andPWM212, which can be passed separatelyT21R0,T21R1andT21R2independent settingPWMduty cycle, and can be passedT21CM,T21OC The control bits of the register independently set the output polarity.

multi-precisionPWMMode count clock source is system clockFosc, and supports prescaler and postscaler. In this mode, the postscaler does not affectPWMcycle, only affects the count overflow interrupt flagT21VIF.

As shown below, whenT21ENEnable, T21TRfor0hour, PWMoutput turns off and remainsPWM210/1/2 output as0;set upT21TRfor1hour,PWMThe output waveform starts,PWM210/ PWM211/ PWM212The output starts with1, and respectively16bit period registerT21Pand16bit precision register T21R0/T21R1/T21R2register contents, updated to16bitPWMcycle bufferPRDBUFand16bit precision buffer RESBUFO/RESBUF1/RESBUF2(This buffer is not readable or writable by software) , then16bit meter CounterT21Counting up from zero, whenT21andRESBUFO/RESBUF1/RESBUF2When the value of is equal, PWM210/PWM211/ PWM212The output changes to0, and continue to increment the count. whenT21The count value of PRDBUF When equal, the postscaler counter adds1,PWM210/PWM211/PWM212The output reverts to1,at the same timePRDBUFand RESBUFO/RESBUF1/RESBUF2reload separatelyT21Pand T21R0/T21R1/T21R2register value and generate a periodic interrupt flagT21PIF, the interrupt flag needs to be cleared by software. so far a completePWMcycle completes, followed by the counterT21Count up from zero and continue to loop to generate newPWMcycle. When the count value of the postscaler is the same as the frequency division ratio of the postscaler, reset the postscaler and set the count overflow interrupt flag T21VIFset "1", the interrupt flag needs to be cleared by software.

In particular, if the precision bufferRESBUFI is not less than the value of the cycle bufferPRDBUF, when the output polarity is active high, the currentPWMwithin the cyclePWMThe output is always1; Conversely, when the output polarity is selected as active low, the currentPWMwithin the cyclePWMThe output is always0.



picture6-6 T21multi-precisionPWMPattern Diagram

PWMCalculated as follows:

$$\text{PWMcycle} = (\text{T21P} + 1) \times \text{Tosc} \times (\text{prescaler division ratio})$$

$$\text{PWMfrequency} = 1 / (\text{PWMcycle})$$

$$\text{PWMpulse width} = (\text{T21R0/1/2} + 1) \times \text{Tosc} \times (\text{prescaler division ratio})$$

$$\text{PWMDuty cycle} = (\text{PWMpulse width}) / (\text{PWMcycle})$$

given PWMfrequency, PWMThe maximum resolution of can be calculated as:

$$\text{Resolution} = \frac{\log(\frac{F_{osc}}{F_{pwm} * F_{ckps}})}{\log 2}$$

Fckpsis the division ratio of the counter's prescaler.

- Note: when T21Rn=0hour,PWMpulse width =Tosc ×(prescaler division ratio) . If needed PWMoutput as0, you can choose to set the PWM The pin corresponding to the output is low level, and set the corresponding PT2EN<1:0>,PT1ENorPT0EN<1:0>is all zeros, the hardware will automatically turn off the corresponding I/Oport PWMoutput function.

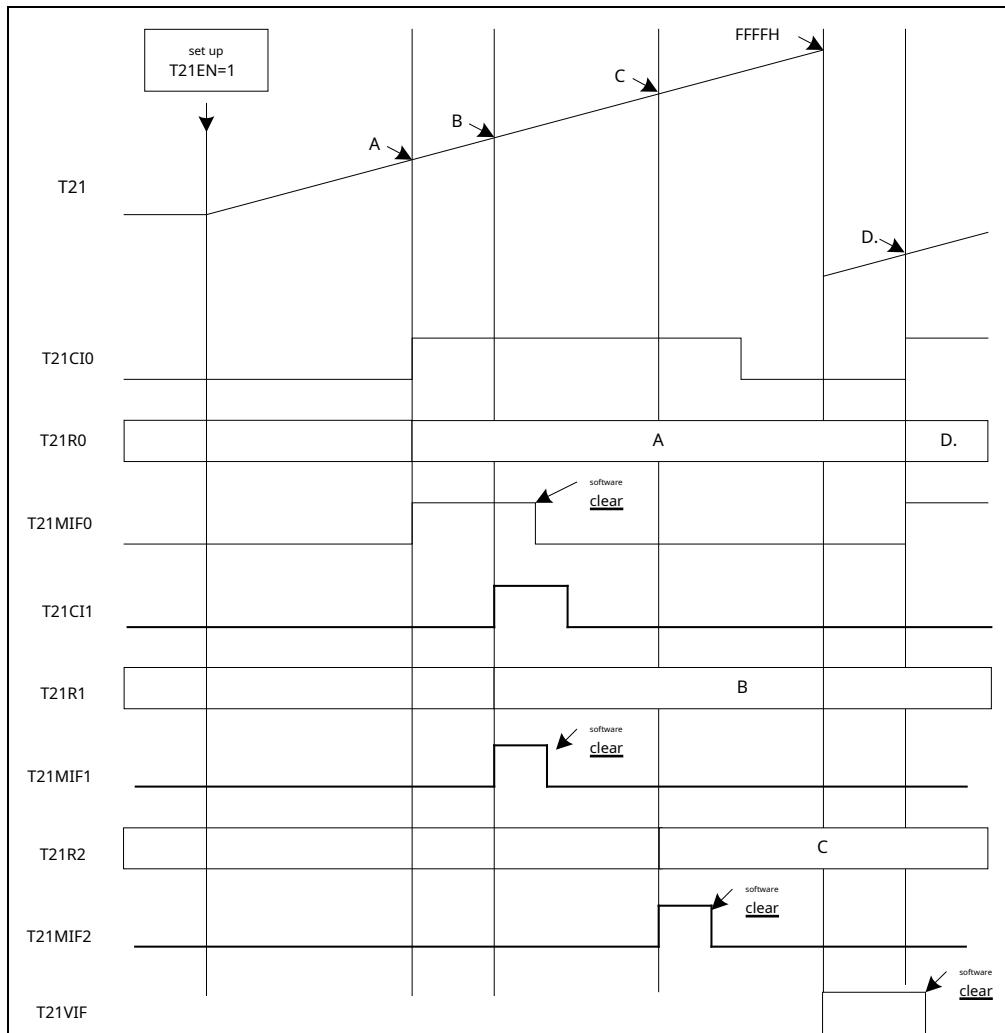
## 6. 2. 7 catcher mode

T21MSet as "01xx"hour,T21work in catcher mode,T21support3independent capture inputsT21CI0, T21CI1andT21CI2. This mode supports3A selection of capture conditions.

T21When configured in capture mode, the clock source is the system clock (Fosc) . In this mode,16bit counterT21Enter

The row count is incremented when T21CI0/T21CI1/T21CI2. When the change state of the input signal meets the capture condition, the counter T21 will be loaded into the corresponding 16bit capture register T21R0/T21R1/T21R2, and generate a multi-function interrupt T21MIF0/T21MIF1/T21MIF2, the interrupt must be cleared by software. The counter continues to count up. If the next capture event occurs, the capture register T21R0/T21R1/T21R2. The value in is not read in time and will be overwritten by the newly captured value. When the counter value overflows (i.e. from FFFF<sub>h</sub> becomes 0000<sub>h</sub>) , an overflow interrupt is generated T21VIF. Should Interrupts must be cleared by software.

T21 support 1An edge counter used to judge the capture condition, this edge counter is only valid in the capture mode. when T21 The edge counter is cleared when shutting down or switching to other modes, but in T21 of 4The edge counter will not be cleared when the two capture modes are switched between each other. Therefore, when the capture mode is switched, there may be errors in the first capture, and it may also cause wrong interrupts to be generated. To avoid false interrupts, the user should first disable the T21 Corresponding interrupt enable bit, and clear the interrupt flag.



picture6-7 T21Capture mode timing diagram (capturing signal on every pulse rising edge)

## 6. 2. 8 comparator mode

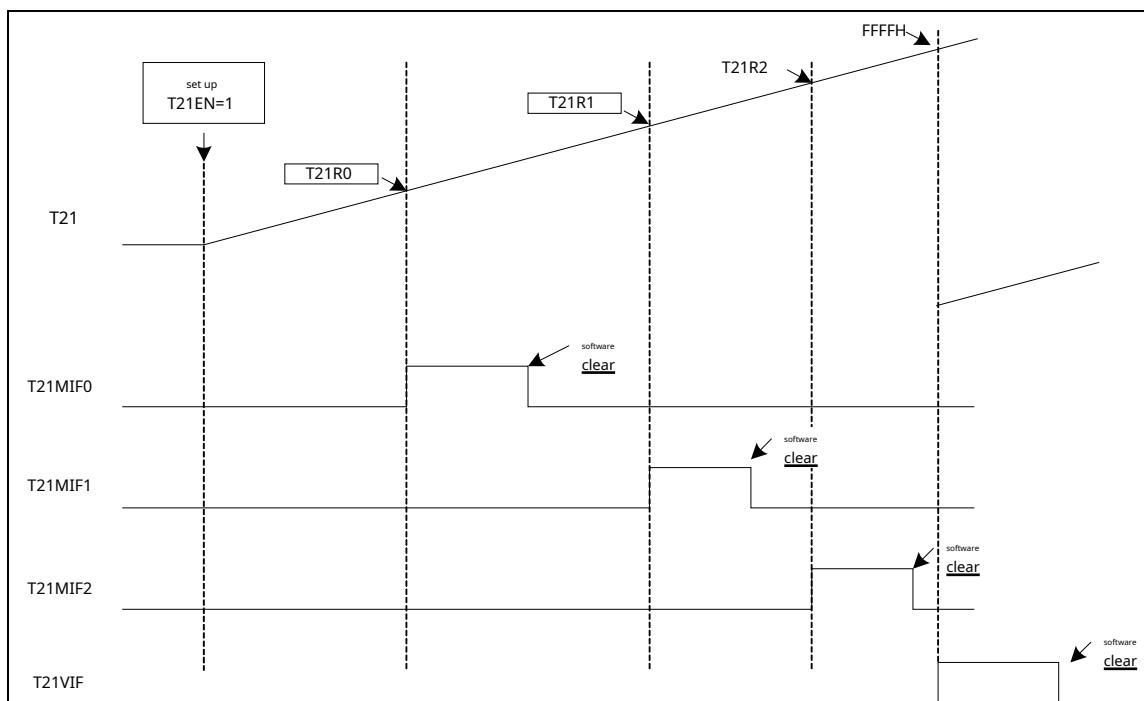
T21MSet as "10xx"hour, T21work in comparator mode, T21support3independent comparison output port T21CO0, T21CO1and T21CO2.

The comparator mode clock source is the system clock (Fosc) . In this mode, 16bit counter T21Count up.

when the counter T21The count value is compared with the compare register T21R0/T21R1/T21R2When the values in are equal, execute the corresponding comparison match event and generate a multi-function interrupt T21MIF0/T21MIF1/T21MIF2, the interrupt must be cleared by software. When the counter value overflows (i.e. from FFFFh becomes 0000h) , an overflow interrupt is generated T21VIF, the interrupt must be cleared by software zero; after the count overflows, continue from 0Start counting up.

Comparator Mode Configuration Bits T21M configured as 1000 or 1001 when T21 counter match T21R0L/T21R0H hour, T21CO0 port output 0 or 1 and keep; when T21 counter match T21R1L/T21R1H hour, T21CO1 port output 0 or 1 and keep; when T21 counter match T21R2L/T21R2H hour, T21CO2 port output 0 or 1 And kept.

Comparator Mode Configuration Bits T21M configured as 1011 when T21 counter match T21R0L/T21R0H, T21R1L/T21R1H or T21R2L/T21R2H hour, T21 is cleared and can trigger ADC convert. ADC It must be enabled first and set to hardware sampling, ie ADCCL register ADEN and SMPSC control bits need to be set to 1. Special Note: When matching T21 is cleared, so only T21R0L/T21R0H, T21R1L/T21R1H or T21R2L/T21R2H smaller values of are valid in this mode.



picture6-8 T21 Comparator Mode Timing Diagram

### 6.2.9 Multi-function output port

T21 work in multiple precision PWM and comparator mode, by setting the T21O register control bits PT0EN, PT1EN and PT2EN, optional PWM and compare output to IOPort; if the control bit PT0EN, PT1EN and PT2EN when selecting to disable the output, the IOPort will not output PWM waveform and compare the results at this time! The high/low level of the port output is determined by the IOPort register assignment decision. Note that in multiple precision and comparator modes, the corresponding IOPort the port direction register needs to be set to output.

**6.2.10special function register****6. 2. 10. 1 counter register low8bits (T21L)**

T21L:T21counter low8bits (T21L)								
bit	7	6	5	4	3	2	1	0
<b>name</b>	T21<7:0>							
<b>R/W</b>	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
<b>POR</b>	0	0	0	0	0	0	0	0

Bit 7~0 T21&lt;7:0&gt;:T21counter low8bit

**6. 2. 10. 2 counter register high8bits (T21H)**

T21H:T21counter high8bits (T21H)								
bit	7	6	5	4	3	2	1	0
<b>name</b>	T21<15:8>							
<b>R/W</b>	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
<b>POR</b>	0	0	0	0	0	0	0	0

Bit 7~0 T21&lt;15:8&gt;:T21counter high8bit

**6. 2. 10. 3 Period Register Low8bits (T21PL)**

T21PL:T21Period Register Low8bits (T21PL)								
bit	7	6	5	4	3	2	1	0
<b>name</b>	T21P<7:0>							
<b>R/W</b>	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
<b>POR</b>	1	1	1	1	1	1	1	1

Bit 7~0 T21P&lt;7:0&gt;:PWMLow cycle value8bit

**6. 2. 10. 4 Period Register High8bits (T21PH)**

T21PH:T21Period Register High8bits (T21PH)								
bit	7	6	5	4	3	2	1	0
<b>name</b>	T21P<15:8>							
<b>R/W</b>	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
<b>POR</b>	1	1	1	1	1	1	1	1

Bit 7~0 T21P&lt;15:8&gt;:PWMHigh cycle value8bit

Note: The period register is updated when writing a low8When the bit register is used, the value of the high-bit register will be loaded at the same time, so when the register is updated, the high-bit register needs to be written first8bit register, then write low8bit register.

**6. 2. 10. 5 multi-purpose register0Low8bits (T21R0L)**

T21R0L:T21precision register0Low8bits (T21R0L)								
bit	7	6	5	4	3	2	1	0
name	T21R0<7:0>							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7~0 T21R0&lt;7:0&gt;:

multi-precisionPWMmodel:PWM210Low precision value8Bit

capture mode:T21CI0low capture value8Bit compare mode:

T21CO0low comparison value8bit

**6. 2. 10. 6 multi-purpose register0high8bits (T21R0H)**

T21R0H:T21precision register0high8bits (T21R0H)								
bit	7	6	5	4	3	2	1	0
name	T21R0<15:8>							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7~0 T21R0&lt;15:8&gt;:

multi-precisionPWMmodel:PWM210High precision value8Bit

capture mode:T21CI0High capture value8Bit compare mode:

T21CO0High comparison value8bit

**6. 2. 10. 7 multi-purpose register1Low8bits (T21R1L)**

T21R1L:T21precision register1Low8bits (T21R1L)								
bit	7	6	5	4	3	2	1	0
name	T21R1<7:0>							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7~0 T21R1&lt;7:0&gt;:

multi-precisionPWMmodel:PWM211Low precision value8Bit

capture mode:T21CI1low capture value8Bit compare mode:

T21CO1low comparison value8bit

**6. 2. 10. 8 multi-purpose register1high8bits (T21R1H)**

T21R1H:T21precision register1high8bits (T21R1H)								
bit	7	6	5	4	3	2	1	0
name	T21R1<15:8>							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7~0 T21R1&lt;15:8&gt;:

multi-precisionPWMmodel:PWM211High precision value8bit

capture mode:T21CI1High capture value8Bit  
compare mode:T21CO1High comparison value8bit

### 6. 2. 10. 9 multi-purpose register2Low8bits (T21R2L)

T21R2L:T21precision register2Low8bits (T21R2L)								
bit	7	6	5	4	3	2	1	0
name	T21R2<7:0>							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7~0 T21R2<7:0>:

multi-precisionPWMmodel:PWM212Low precision value8Bit  
capture mode:T21CI2low capture value8Bit compare mode:  
T21CO2low comparison value8bit

### 6. 2. 10. 10multi-purpose register2high8bits (T21R2H)

T21R2H:T21precision register2high8bits (T21R2H)								
bit	7	6	5	4	3	2	1	0
name	T21R2<15:8>							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7~0 T21R2<15:8>:

multi-precisionPWMmodel:PWM212High precision value8Bit  
capture mode:T21CI2High capture value8Bit compare mode:  
T21CO2High comparison value8bit

NOTE: The precision register is updated on writes to low8When the bit register is used, the value of the high-bit register will be loaded at the same time, so when the register is updated, the high-bit register needs to be written first8bit sent memory, then write low8bit register;

### 6. 2. 10. 11control register low8bits (T21CL)

T21CL:T21control register low8bits (T21CL)								
bit	7	6	5	4	3	2	1	0
name	T21M<3:0>				CAP1S<1:0>		CAP0S<1:0>	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7~4 T21M<3:0>:T21working mode selection bit

- 0000: Timer mode (counter clock source isFosc)
- 0000~0011: reserved for unused
- 01xx: capture mode
  - 1000: Comparator mode, output on match1
  - 1001: Comparator mode, output on match0
  - 1010: Comparator mode, does not change the output on match, byIOoutput status decision
  - 1011: Comparator mode, reset on matchT21, and triggerADCconversion, without changing the output, byIOoutput status decision

1100: multi-precisionPWMMode

Others: reserved for unused

Bit 3~2 CAP1S<1:0>:CAPTURE1Capture input edge selection bit

00:prohibit

01: Capture every1pulse falling edge

10: Capture every1pulse rising edge 11

: Capture every1double edge

Bit 1~0 CAP0S<1:0>:CAPTURE0Capture input edge selection bit

00:prohibit

01: Capture every1pulse falling edge

10: Capture every1pulse rising edge 11

: Capture every1double edge

## 6. 2. 10. 12control register second high8bits (T21CM)

T21CM:T21in the control register8bits (T21CM)								
bit	7	6	5	4	3	2	1	0
<b>name</b>	CAP2S<1:0>	T21OM20	T21OM21	T21PRS<3:0>				
<b>R/W</b>	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
<b>POR</b>	0	0	0	0	0	0	0	0

Bit 7~6 CAP2S<1:0>:CAPTURE2Capture input edge selection bit

00:prohibit

01: Capture every1pulse falling edge

10: Capture every1pulse rising edge 11

: Capture every1double edge

Bit 5 T21OM20:PWM210Output Polarity Select Bit

0:PWM210High effective

1:PWM210effective low

Bit 4 T21OM21:PWM211Output Polarity Select Bit

0:PWM211High effective

1:PWM211effective low

Bit 3~0 T21PRS<3:0>:T21Prescaler division ratio selection bits

0000: The frequency division ratio is1:1

0001: The frequency division ratio is1:2

0010: The frequency division ratio is1:3

0011: The frequency division ratio is1:4

0100: The frequency division ratio is1:5

0101: The frequency division ratio is1:6

0110: The frequency division ratio is1:7

0111: The frequency division ratio is1:8

1000: The frequency division ratio is1:9

1001: The frequency division ratio is1:10

1010: The frequency division ratio is1:11

1011: The frequency division ratio is1:12

1100: The frequency division ratio is1:13

1101: The frequency division ratio is1:14

1110: The frequency division ratio is1:15

1111: The frequency division ratio is1:16

**6. 2. 10. 13control register high8bits (T21CH)**

T21CH:T21control register high8bits (T21CH)								
bit	7	6	5	4	3	2	1	0
name	T21EN							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7 T21EN:T21enable bit

0:closure

1:Enable

Bit 6~0 T21POS&lt;6:0&gt;:T21Postscaler frequency division value

Postscale frequency =T21POS&lt;6:0&gt; + 1

**6. 2. 10. 14I/O Control Register (T21OC)**

T21OC:T21I/O Control Register (T21OC)								
bit	7	6	5	4	3	2	1	0
name	T21TR	—	PT2EN<1:0>		T21OM22	PT1EN	PT0EN<1:0>	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7 T21TR:PWMyoutput trigger bit

0:PWMyoutput as0

1:start upPWMyoutput

Bit 6 Reserved

Bit 5~4 PT2EN&lt;1:0&gt;:T21CI2/PWM212/T21CO2Multiplexed Port Select Bits

00:closure,PA6,PA7common port 01

:choosePA6port

10:choosePB7port

11: reserved for unused

Bit 3 T21OM22:PWM212Output Polarity Select Bit

0:PWM212High effective

1:PWM212effective low

Bit 2 PT1EN:T21CI1/PWM211/T21CO1Multiplexed Port Select Bits

0:closure,PA7common port

1:choosePA7port

Bit 1~0 PT0EN&lt;1:0&gt;:T21CI0/PWM210/T21CO0Multiplexed Port Select Bits

00:closure,PA0,PB4common port 01

:choosePA0port

10:choosePB4port

11: reserved for unused

## 6.3 16Bit Multi-function Timer (T31)

### 6.3.1

overview

16bit multifunction timerT31include a16Bit auto-reload counter, support prescaler and postscaler, support multiple operating modes: timer mode, capture mode, comparator mode,PWMmode, Single Pulse Mode, Shutdown Function Mode, and Slave Mode.

-T31Support multiple working modes

- Timer mode (clock source isFosc, external clock source mode1, external clock source mode2)
- catcher mode
- comparator mode
- PWMmode (normalPWMmode, center-aligned mode, complementary output with dead zone)
- single pulse mode
- Shutdown function mode
- Slave mode (encoder mode, reset mode, gated mode, trigger mode)

-T31The following functional components are supported

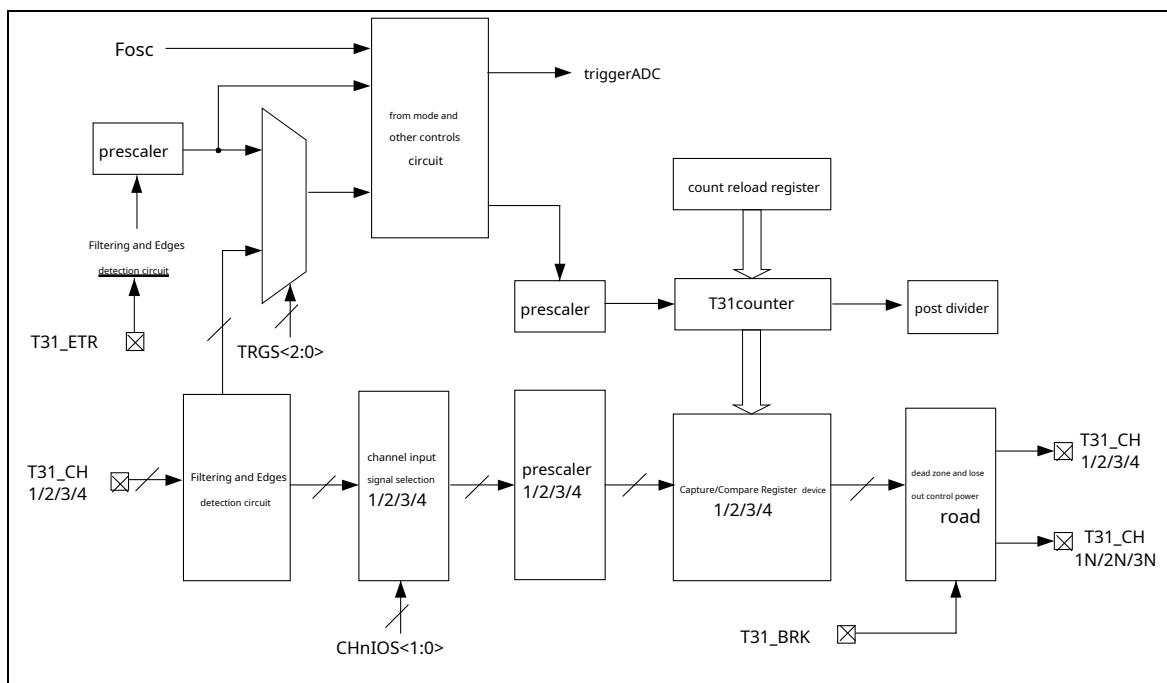
- 16Bit automatic reload counter, support up counting, down counting, up/down alternate counting
- 16Bit programmable prescaler, the counting clock prescaler range is1~65536
- 8Bit-programmable postscaler with a postscale range of1~256
- 8Bit Dead-Band Delay RegisterT31DLYT
- 4indivial16Bit Capture/Compare RegisterT31CH1R,T31CH2R,T31CH3RandT31CH4R
  - Supports four independent channels

break and pause

- Support shutdown interruptBKIF, external trigger interruptTRGIF, Complementary Channel Update InterruptCHUIF, channel multi-function interruptMIF, update interruptionUPIF, channel capture overflow interruptOVI, when an interrupt occurs, if the corresponding interrupt enable bit is1, will setT31total interrupt flagT31IF
- existIDLEmode, stop working

Note: Letters in this sectionnrepresent numbers1,2,3,4.

### 6.3.2 Internal structure map



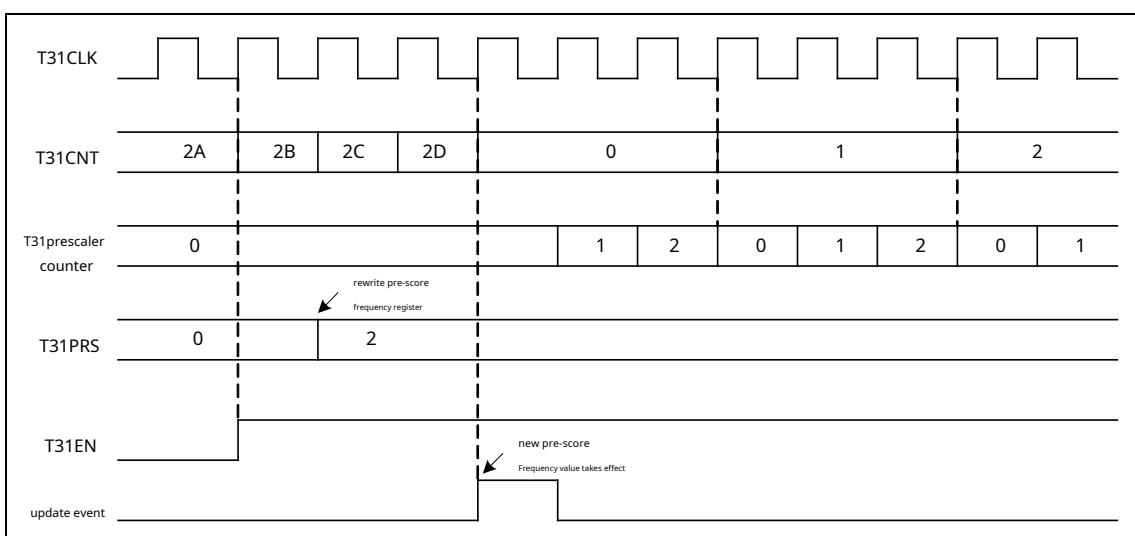
picture6-9 T31Internal structure map

### 6.3.3 clock divider

#### 6.3.3.1 prescaler

The prescaler consists of a 16Bit Prescaler Register T31PRS and a 16Bit prescaler counter with a prescaler range of 1~65536. where the prescaler register T31PRS can be read and written by software, and has a buffer function, so the register T31PRSIt can be rewritten when the prescaler counter is working, and the updated prescaler value will not take effect until the next update event occurs.

The prescaler counter is fixed to count up, from 0 start counting to T31PRSAfter the value of the register is reset, it is automatically cleared and the count is re-incremented.



picture6-10Count Timing Diagram for Prescaler Change (T31CNTLD=0x2D)

Note:T31CLKfor the system clockFosc.

### 6. 3. 3. 2 post divider

The postscaler consists of a 8bit postscaler register T31POS and a 81-bit postscaler counter with a postscaler range of 1~256. where the postscaler register T31POS can be read and written by software, and has a buffer function, so the register T31POS It can be rewritten when the post-scale frequency counter is working, and the updated post-scale frequency value will not take effect until the next update event occurs.

The post-scale frequency counter is fixed to count down, from T31POS register value begins counting down to 0, will automatically reload the T31POS register value and continue counting down.

After using the postscaler in operation, only after the postscaler counter decrements to 0 Only when the update event is generated, and the update flag is set UPIF(T31IFL<0>)

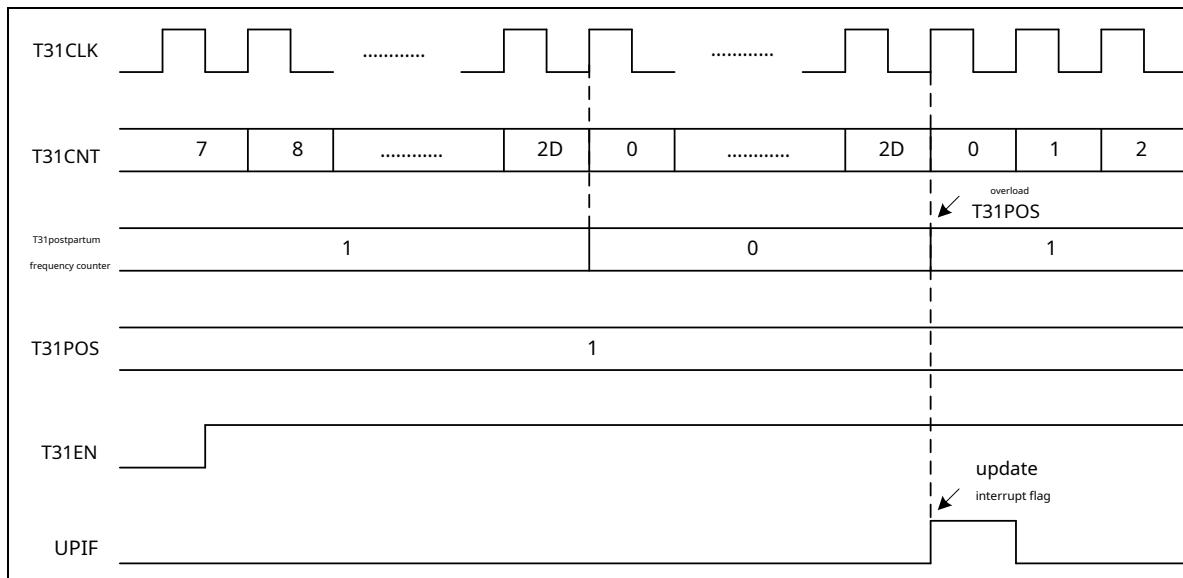
Current Postscaler Register T31POS The value is Never time N When the secondary counter overflows or underflows, an update event is generated. When the update event occurs, all count-related settings are updated:

- Postscaler counter reload T31POS register value
- counter reload T31CNTLD register value
- Prescaler counter reload T31PRS register value
- Comparator reload in comparator mode T31CHnR register value

The postscaler counter counts down and will decrement under any of the following conditions:

- Decrement every time the counter increments and counts overflow
- Decrement every time the counter counts down and underflows
- In center-aligned mode, the postscaler counter decrements each time the counter up-counts overflows and down-counts underflows

The post-scale counter is automatically loaded, when the post-scale counts down to 0 After the update event is generated, it is automatically loaded T31POS The value of the register. When the update event is set by software UPT(T31EVG<0>) bit or generated by hardware through the slave mode controller, the postscaler counter will also be loaded immediately T31POS The value of the register.



picture6-11The counting timing diagram using the postscaler (T31CNTLD=0x2D)

## 6. 3. 4 counting mode

### 6. 3. 4. 1 count up mode

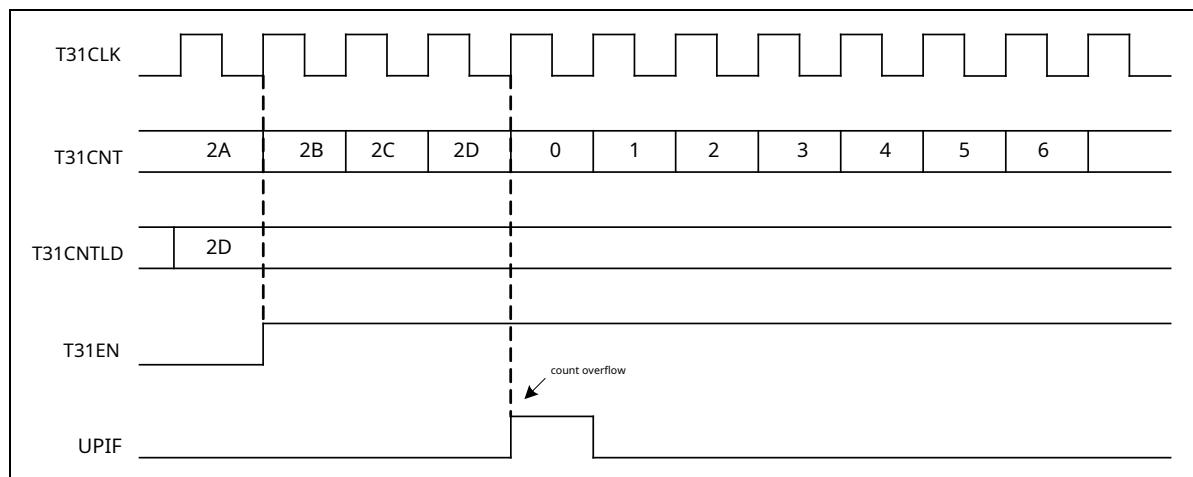
In up counting mode, the counter is controlled by 0count to count reload value (byT31CNTLDregister setting), it will restart by 0Counts and generates a counter overflow update event.

If a postscaler counter is used, it will overflow the number of times the postscaler register is counted up T31POSAn update event will be generated after the value of the counter, otherwise an update event will be generated every time the counter overflows.

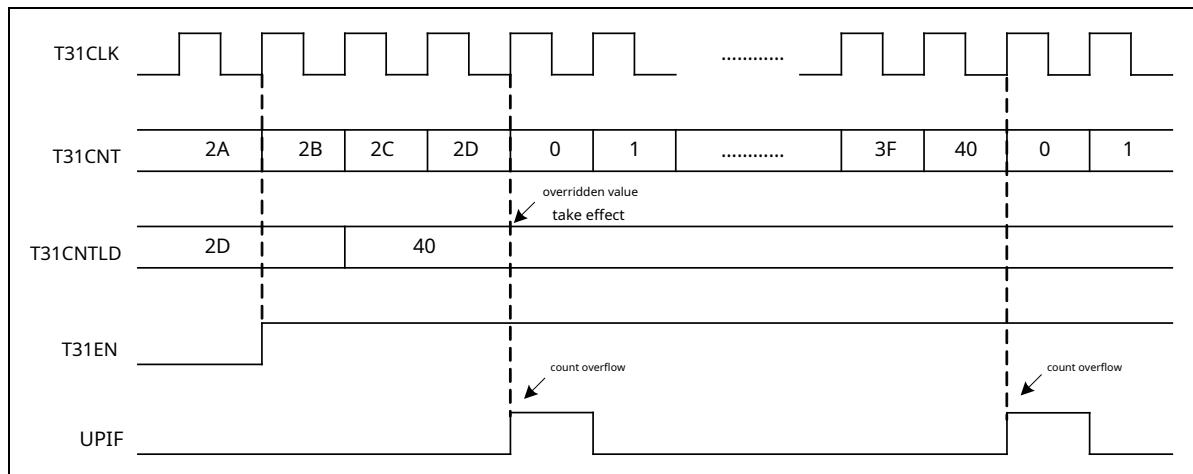
Update events can be set by setting T31COLregisterUEDbit off, this avoids reloading the count register while overwriting the T31CNTLDThe update event happens exactly at that time, making the overwritten value take effect immediately. When the update event is disabled, the counter will still be cleared and restart counting after the counter counts up and overflows, and the count of the prescaler will also be cleared and restart counting (but the prescaler ratio will not change).

When an update event occurs, all count-related settings are updated and the update flag is set UPIF(T31IFL<0>) :

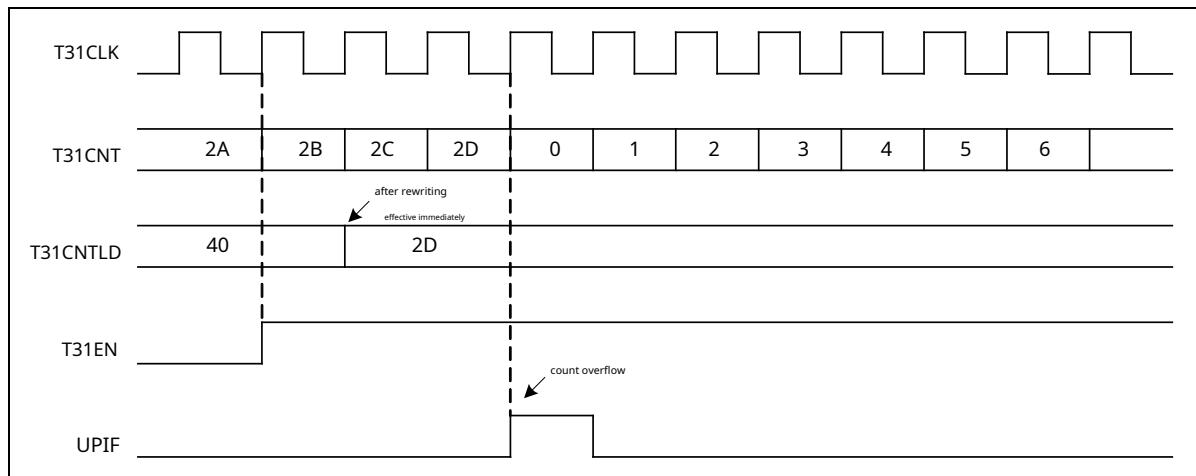
- Postscaler counter reload T31POSregister value
- counter reload T31CNTLDregister value
- Prescaler counter reload T31PRSregister value



picture6-12Count Up Timing Diagram (Prescaler is1)



picture6-13Count Up Timing Diagram (RLBE=1,T31CNTLDbuffered)



picture6-14Count Up Timing Diagram (RLBE=0,T31CNTLDunbuffered)

### 6. 3. 4. 2 Count down mode

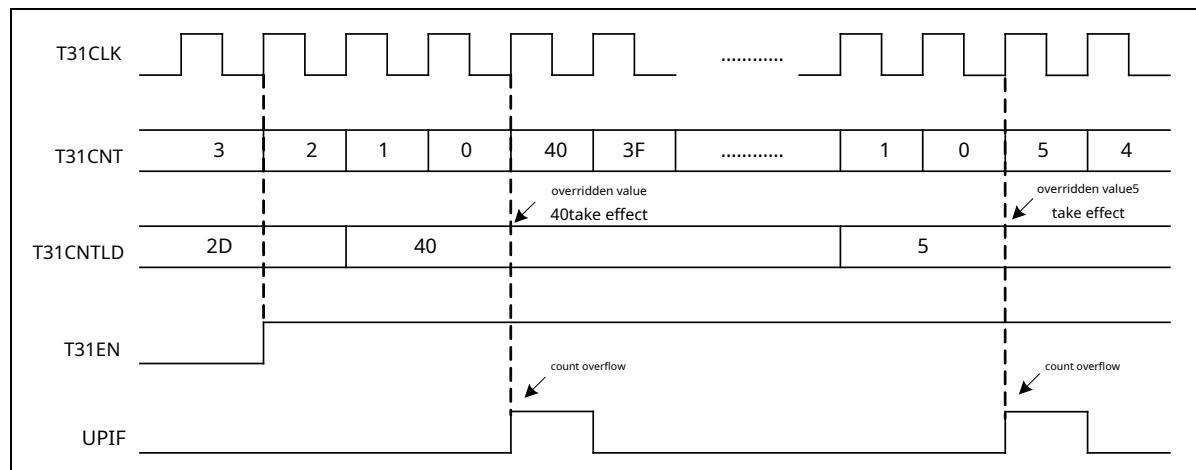
In down-counting mode, the counter is reloaded by the count value (byT31CNTLDregister setting) begins counting down to 0After that, it restarts counting by the count reload value and generates a counter underflow update event.

If a postscaler counter is used, it will underflow the number of times it reaches the postscaler register after counting downT31POSAn update event is generated after the value of the counter, otherwise an update event is generated every time the counter overflows.

Update events can be set by settingT31COLregisterUEDbit off, this avoids reloading the count register while overwriting theT31CNTLDThe update event happens exactly at that time, making the overwritten value take effect immediately. When the update event is disabled, the counter will automatically reload and restart counting after the counting down counts underflow, and the count of the prescaler will also be cleared and start counting again (but the prescaler ratio will not change).

When an update event occurs, all count related settings are updated and the update status bit is setUPIF(T31IFL<0>) :

- Postscaler counter reloadT31POSregister value
- counter reloadT31CNTLDregister value
- Prescaler counter reloadT31PRSregister value



picture6-15Count Down Timing Diagram (RLBE=1,T31CNTLDbuffered)

### 6. 3. 4. 3 Center Alignment Mode

In center-aligned mode, the counter consists of 0Start counting up to the count reload value (by T31CNTLD register setting), a counter overflow event will be generated, and then continue counting down from the count reload value to 0, and generate a counter underflow event, and then the counter is restarted by 0Start counting.

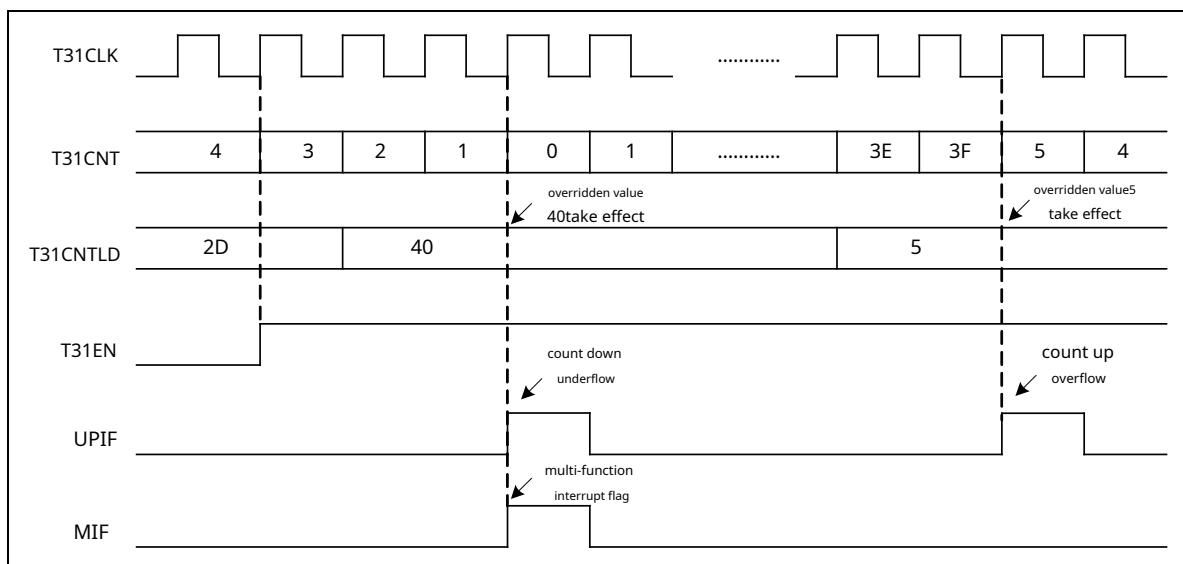
T31 support 3A center-aligned mode, which can be controlled by a register bit CMC(T31C0L<6:5>) set up. exist3 In both center-aligned modes, the counters are incremented or decremented alternately, but the multi-function interrupt flag of each output channel MIF place differently, when CMC="01" When , it is center-aligned mode1, multi-function interrupt flag only when the counter is counting down MIF will be placed1; when CMC="10" When , it is center-aligned mode2, multi-function interrupt flag only when the counter is counting up MIF will be placed1; when CMC="11" When , it is center-aligned mode3, the multi-function interrupt flag when the counter is counting up or down MIF can be set1.

In center-aligned mode, the counting direction selection bit of the counterDIRS(T31C0L<4>) is automatically controlled by the hardware circuit to realize the counter increment or decrement count.

Update events can be set by setting T31C0L register UED bit off, this avoids reloading the count register while overwriting the T31CNTLD. The update event happens exactly at that time, making the overwritten value take effect immediately. When the update event is disabled, the counter will automatically restart counting after overflow or underflow.

When an update event occurs, all count-related settings are updated and the update flag is set UPIF(T31IF<0>) :

- Postscaler counter reload T31POSregister value
- counter reload T31CNTLDregister value
- Prescaler counter reload T31PRSregister value



picture6-16Center Alignment Mode1Count Timing Diagram (RLBE=1)

### 6. 3. 5 Operating mode

T31 supports multiple working modes: timer mode, capture mode, comparator mode, PWM mode, Single Pulse Mode, Shutdown Function Mode, and Slave Mode. where the timer mode includes the internal FoscClock count, external clock source mode 1 counting, external clock source mode 2 counting; PWM mode includes PWM normal mode, PWM center-aligned mode, complementary output with dead zone; slave mode includes encoder mode, reset mode, gate control mode, trigger mode.

T31 support 4 capture/compare channels with 3. Each channel supports two complementary output ports, and the capture input and compare output functions of each channel cannot be used at the same time.

### 6.3.6 timer mode

T31 The timer includes an auto-reloadable 16bit count register T31CNT, one 16bit count reload register T31CNTLD, one 16Bit Prescaler Register T31PRS, one 8bit postscaler register T31POS, can be read and written by software, but it is recommended not to rewrite the value of the counting register when the counter is working normally, so as not to cause abnormal operation.

counter reload register T31CNTLDA write operation requires first writing a high 8bit register T31CNTLDH, then write low 8bit register T31CNTLDL. high 8The value written to the bit register does not take effect immediately, but writes low 8The value of the bit register will be updated at the same time.

count reload register T31CNTLD Support write buffer function, can pass T31COLregisterRLBBit setting whether to enable the buffer function, when the buffer is enabled, write T31CNTLD The value of the register will not take effect immediately, but will not take effect until the next update event occurs.

when T31COLRegister update event disable bit UEDfor0, an update event will be generated when any of the following events occurs:

- Counter overflow when counting up or underflow when counting down
- software settings UPTbit, generating an update event
- Updates generated from the schema controller

when T31COLRegister update event disable bit UEDfor1 When , the generation of update events is prohibited.

when T31 When working in timer mode, it is not necessary to enable the function of its capture/compare channel, but only need to set the count-related registers.

The clock source of the counter in timer mode can be the system clock Fosc or external clock, and supports two external clock source modes:

- set up T31C2Lregister T31SM=000, T31C2Hregister ECM2E=0, the configurable counter clock source is the system clock Fosc.
- set up T31C2Lregister T31SM=111, T31C2Hregister ECM2E=0, the counter can be configured as an external clock source mode1, and pass TRGSbit to select the external clock source:  
aisle1portT31\_CH1Input pulse signal edge (both edges are valid) ;  
aisle1portT31\_CH1Input signal (supports input filtering, rising or falling edge is optional);  
aisle2portT31\_CH2Input signal (supports input filtering, rising or falling edge is optional);  
External trigger input T31\_ETR (Support input filtering and independent prescaler, rising or falling edge selectable) .
- set up T31C2Hregister ECM2E=1, the counter can be configured as an external clock source mode2, using an external trigger input T31\_ETR as a clock source.

Prescaler and postscaler for count clock are supported.

set up T31COLregister CMC=00, can be selected as normal counting mode, set T31COLregister T31EN=1 After that, the counter starts counting.

Supports two counting methods of increment and decrement, which can be passed T31COLregister DIRSbit to select. Increment or pass

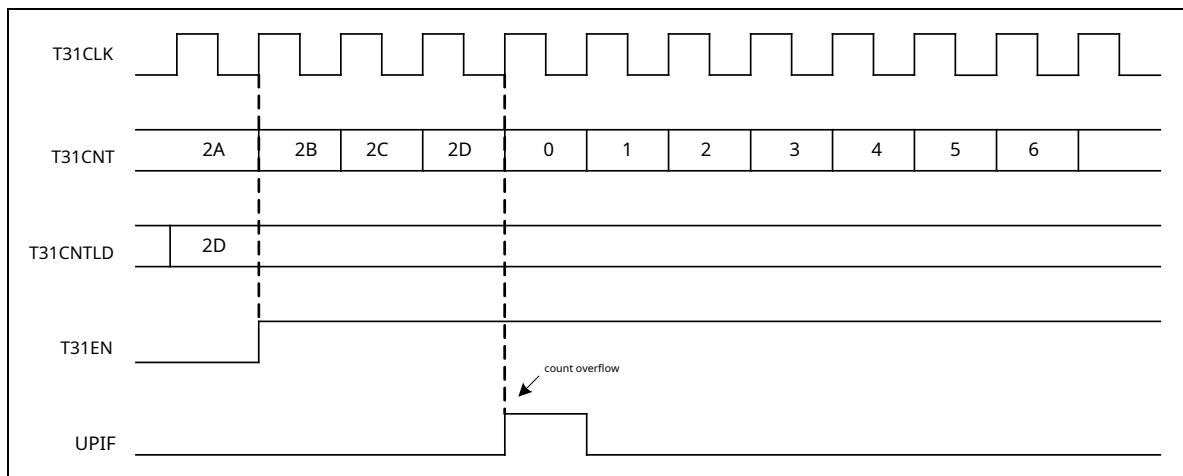
When the countdown overflows, the counter will restart counting, and a counter overflow or underflow event will be generated, and the update flag will be setUPIF(T31IFL<0>).

### 6. 3. 6. 1 Internal Clock Source Mode

set upT31C2LregisterT31SM=000,T31C2HregisterECM2E=0,ConfigurableT31In internal clock source mode, the counter clock source is the system clockFosc.

The steps to configure the counter to count for internal clock source mode are as follows:

- set upT31C2LregisterT31SM=000,T31C2HregisterECM2E=0, configured as internal clock source mode
- Set the prescaler registerT31PRS, configure the prescaler
- Set postscaler registerT31POS, after configuring the frequency divider
- set upT31C0LregisterDIRSBit, select up or down counting mode
- set upT31C0LregisterRLBBit, configure whether the counting reload register buffer is enabled
- SET COUNT RELOAD REGISTER T31CNTLD, configure the counting period
- set upT31C0LregisterT31EN=1, enable the counter



picture6-17Timing diagram for counting up in internal clock source mode (prescaler/postscaler is1)

### 6. 3. 6. 2 External Clock Source Mode1

set upT31C2LregisterT31SM=111,ConfigurableT31For external clock source mode1, in this mode, the counter counts on each rising edge or falling edge of the external input clock, and the external input clock port is optional.

Configure the counter as an external clock source mode1The counting steps are as follows:

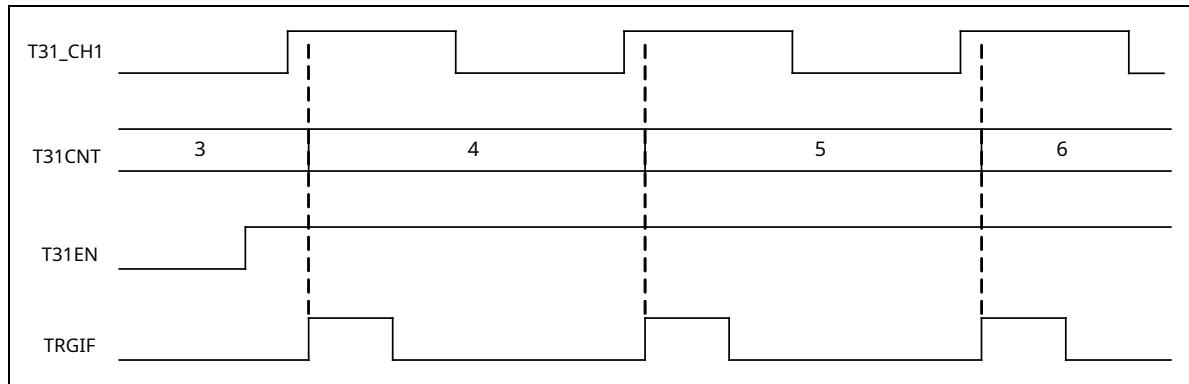
- set upT31C2LregisterT31SM=111, configured as an external clock source mode1
- set upT31C2Lin the registerTRGSbit to select an external clock source
- If the external clock source is a channel1or2port, you can setT31CHnCregisterCHnIFSbit, configure the filter time of the input signal; setT31PINregisterCHnP/CHnP, select rising edge or falling edge to be valid
- If the external clock source is an external trigger input port, you can setT31C2HregisterETFSBit, configure the filter time of the input signal, setETPRSbit, configure the prescaler of the trigger input signal, setETEG,

Select rising edge or falling edge valid

- set upT31C0LregisterT31EN=1, enable the counter

The counter works in external clock source mode1, every increment or decrement count,TRGIFFlag bits will be set1.

Because the external input clock needs to be processed by the on-chip filter sampling circuit, there is a certain delay between the actual counting time and the rising edge time of the external clock.



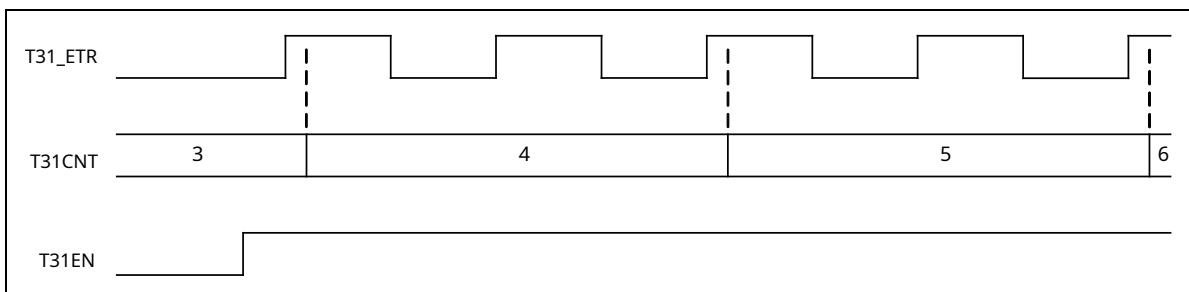
picture6-18External Clock Source Mode1The counting timing diagram (the clock source is the channel1port, count up on rising edge)

### 6. 3. 6. 3 External Clock Source Mode2

set upT31C2LregisterT31SM=000,T31C2HregisterECM2E=1,ConfigurableT31For external clock source mode2, in this mode the counter triggers the input externallyT31\_ETRThe rising or falling edge counts and can use the prescaler. External Clock Source Mode2Equivalent to the external clock source mode1when, willT31\_ETRsignal as an external clock.

Configure the counter as an external clock source mode2The counting steps are as follows:

- set upT31C2HregisterETFSbit to configure the filter time of the input signal
- set upT31C2HregisterETPRSbit to select the prescaler for the external trigger input signal
- set upT31C2HregisterETEG, choose rising edge or falling edge to be valid
- set upT31C2LregisterT31SM=000,T31C2HregisterECM2E=1, enable external clock source mode2
- set upT31C0LregisterT31EN=1, enable the counter



picture6-19External Clock Source Mode2The counting timing diagram (prescaler ratio is1:2, count up on rising edge)

### 6. 3. 7 catcher mode

T31support4capture channels, each capture channel supports two capture input ports, throughT31CHnC

registerCHnIOSbit to select. Example: for the capture channel1, available for channel1portT31\_CH1, also for channel2portT31\_CH2The input signal is captured (byCH1IOSbit to select) .

Supports filter processing and edge selection for captured input signals, which can be passedT31CHnCregister CHnIFS bit, set the filter time to capture the input signal, byT31PINRegisterCHnP/CHnPBit, set to capture the valid edge of the input signal (rising edge, falling edge or both edges)

Support capture frequency optional, can pasST31CHnCregisterQURbit, sets the frequency of capture (every 1, 2,4or8input signal edge, capture once).

set upT31PINRegisterCHnE=1, to enable the capture input of the channel.

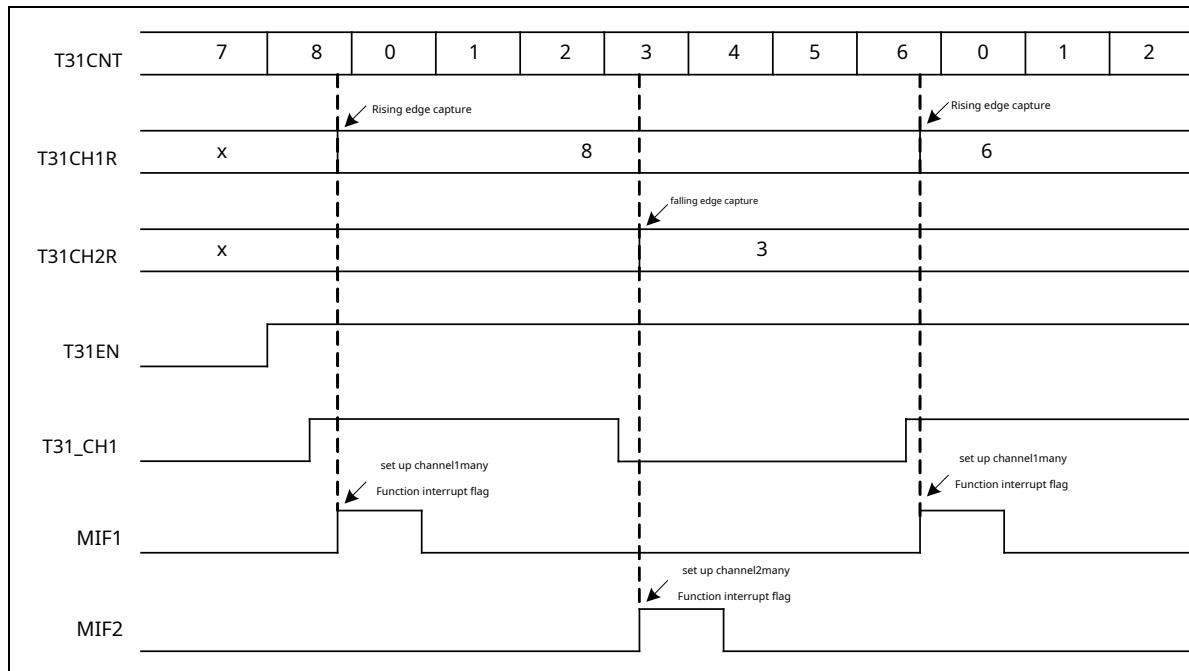
In the capture mode, when the capture input signal is detected to meet the capture condition, the current value of the counter is latched into the capture/compare registerT31CHnRmiddle.

When a capture occurs, the multi-function interrupt flag bitMIFbe placed1, if the multi-function interrupt enable bit MIEn=1, butT31 total interrupt flagT31IFalso set1, if the total interrupt enable bitT31IE=1, a capture interrupt request will be generated; if a capture occurs,MIFflag has been set, the capture overflow interrupt flagOVIwill be set 1. register byT31ICRThe corresponding interrupt is cleared0bit to write1operation, the interrupt flag bit can be clearedMIF andOVI, also by reading the capture registerT31CHnRto clear the interrupt flagMIF.

Capture condition generation can also be set by software registerT31EVGofCHnTbit to trigger whenCHnTbit write1 , a capture is triggered and the current value of the counter is latched into the capture/compare registerT31CHnR, at the same time set up the corresponding interrupt flag bitMIForOVI.

When capture overflow occurs, it is recommended to read the data first, and then process the capture overflow flag to avoid repeated capture overflow.

The capture mode can be used to measure the period of the input pulse signal. If two capture channels are used to capture the rising edge and falling edge of the input signal on one port at the same time, the period and duty cycle of the pulse signal can be measured. In the application of measuring pulse signal, it can be setT31C2LregisterT31SM=100, configure the slave mode controller in reset mode, reinitialize the counter with the rising edge of the trigger input signal, and setT31C2Lregister TRGSBit, select the capture input of the channel port as the trigger input signal.



picture6-20Capture mode timing for measuring pulsed signal period and duty cycle (channel1, reset mode)

### 6. 3. 8 comparator mode

set upT31ChnCregisterQUR, configurable comparator mode.T31support4comparison output channels, output portT31\_CHn(n=1,2,3,4) multiplexed with the capture channel, setT31ChnCregister CHnIOS=00, to configure the channel port as a compare output.

Each compare channel supports a16Bit Capture/Compare RegisterT31ChnR, can perform software read and write operations, when writing operations, you need to write high8bit registerT31ChnRH, then write low8bit registerT31ChnRL,high8The value written to the bit register does not take effect immediately, but writes low8The value of the bit register will be updated at the same time. pass T31ChnC registerOBEbit, configurableT31ChnRWhether the register takes effect immediately or when the next update event occurs. Update events have no effect on comparator outputs and channel port outputs.

In comparator mode, the counter can use internal or external clock for counting, and supports counting clock prescaler.

In comparator mode, the count value of the counter is compared with the compare register, and when a compare match occurs, it can be passed T31ChnCregisterQURbit andT31PINOutput Polarity Control Bits in RegisterChnP, to control the channel portT31\_CHnoutput level. When a compare match occurs:

- ifCHnOM=000, the output level of the channel port remains unchanged;
- ifCHnOM=001, the comparator output1, the channel port level depends onCHnPbit;
- ifCHnOM=010, the comparator output0, the channel port level depends onCHnPbit;
- ifCHnOM=011, the channel port level flips. When a comparison match occurs, the multi-function interrupt flag bit

corresponding to the comparison channel will be setMIF, if the corresponding multi-function interrupt enable bitMIEn=1, butT31total interrupt flagT31IFalso set1, if the total interrupt enable bitT31IE=1, a compare interrupt request will be generated.

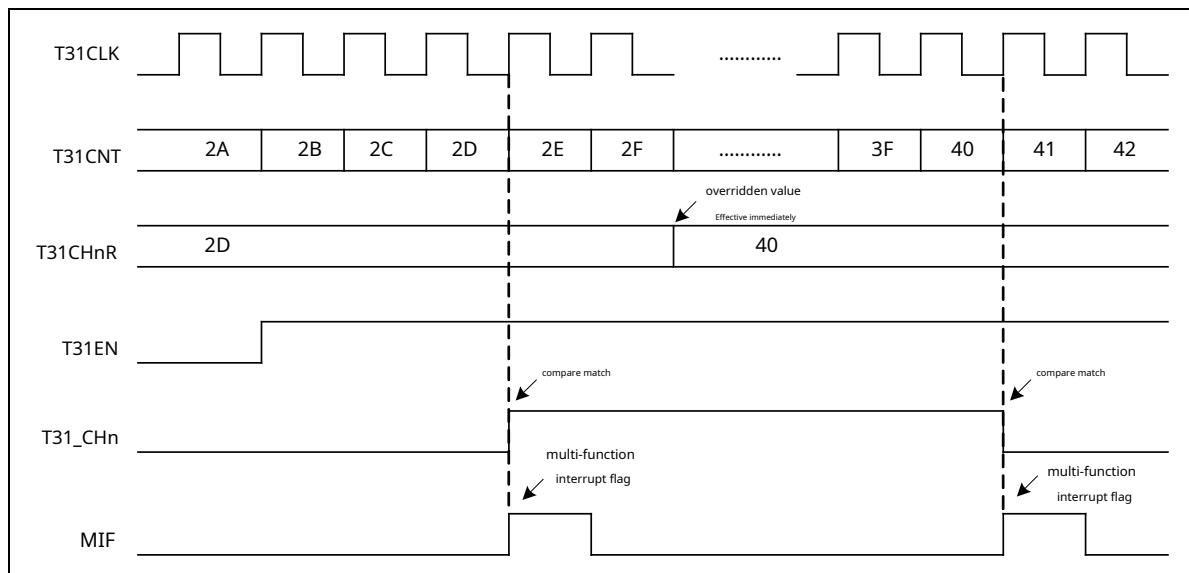
After a comparison match occurs, the counter continues to increase or decrease until the count overflows, and the update interrupt flag will be setUPIF, if a postscaler is used, until the postscaler decrements to0After that, when the count overflow occurs again, the

Set the update interrupt flag bitUPIF. For example, when counting up, it will continue to count to the reload registerT31CNTLDAn overflow occurs after the value of .

In Comparator mode, software is also supported to force the comparator output to a fixed level. set upT31CHnRegister CHnOM=100, which forces the comparator output to0;CHnOM=101, which forces the comparator output to1. After the comparator output is forced to a fixed level, the channel port level still depends onCHnPbit, the hardware comparison circuit will still continue to compare with the count value, the corresponding flag will be set up, and a corresponding interrupt will be generated.

The configuration steps for comparator mode are as follows:

- Selection of counter clock (internal, external) and prescaler
- SET COUNT RELOAD REGISTERT31CNTLDand compare registerT31CHnR
- passQURSet the port status when comparing and matching, throughCHnPSet port level polarity
- passOBEset compare registerT31CHnRIs it effective immediately or with a buffer
- set upT31IELregisterMIEn=1,JNTE0registerT31IE=1, can generate compare interrupt request
- set upT31PINregisterCHnE=1, to enable compare channel port output
- set upT31C0LregisterT31EN=1, enable the counter



picture6-21Timing diagram of comparator mode (port output flip after comparison match,CHnOBE=0, unbuffered)

Comparator output for each channel, supports external trigger input signalT31\_ETRclear0. set upT31CHnC register CHnCOCE=1,T31C2LregisterCOCE=1, the comparator output of the corresponding channel can be cleared by an external input signal, and the cleared comparator output remains low until the next update event occurs. Note that this feature can only be used in comparator mode andPWMmode, but not for comparator forced output mode (CHnOM=100or101) .

When using an external trigger input signal to clear the comparator output, it is necessary to setT31C2HregisterECM2E=0, disable external clock source mode2, cannot put the external input signalT31\_ETRas a counter clock.

accessibleT31C2HregisterETEGandETFSBit, configure the valid edge and filter time of the external trigger input signal.

### 6.3.9 PWMmodel

set up T31CHnCregisterQUR, Configurable PWM model. T31 support 4 individual PWM output channels, each channel's PWM Modes can be set independently, based on the comparison output of the comparator to generate PWM output waveform. Including 3 individual PWM. Each channel supports two complementary output ports T31\_CHn and T31\_CHnN (n=1, 2, 3), which can generate complementary output signals with dead-band delays, pp. 4 individual PWM Channel supports only one output port T31\_CH4.

Support two PWM mode, the output waveform is complementary, through T31CHnCregisterQUR Can be set PWM mode when CHnOM=110 when PWM model 1, when CHnOM=111 when PWM model 2.

exist PWM mode, through T31CNTLD register to set the pulse period by T31CHnR. The register sets the pulse width, which determines the duty cycle. must be set T31CHnCregisterCHnOBE=1, Enable T31CHnR register buffer function, set the T31C0L register RLBE=1, Enable T31CNTLD. The buffering function of the register ensures that when the period or duty cycle is continuously changed, each PWM The pulse waveforms are all complete. Before the counter starts counting, it must be set T31EVG register UPT bit, initialize the buffer that updates all registers.

able to pass T31PINC in the register CHnPbit set port T31\_CHn. The output polarity of the port, you can choose whether the port level is inverted relative to the output of the comparator. pass T31PINC Register channel output enable bit CH, complementary output enable bit CHN and T31CHBK. The channel output status total enable bit of the register CHOE, the port idle state select bit in idle mode NOFFS and port invalid state selection bits in run mode ROFFS, to control the port T31\_CHn output status.

exist PWM mode, the count value T31CNT and compare register value T31CHnR is always comparing, and based on QUR bit set PWM mode, the comparator outputs the corresponding comparison result.

set up T31C0L register CMC bit, optional PWM The signal waveform is normal mode or center-aligned mode.

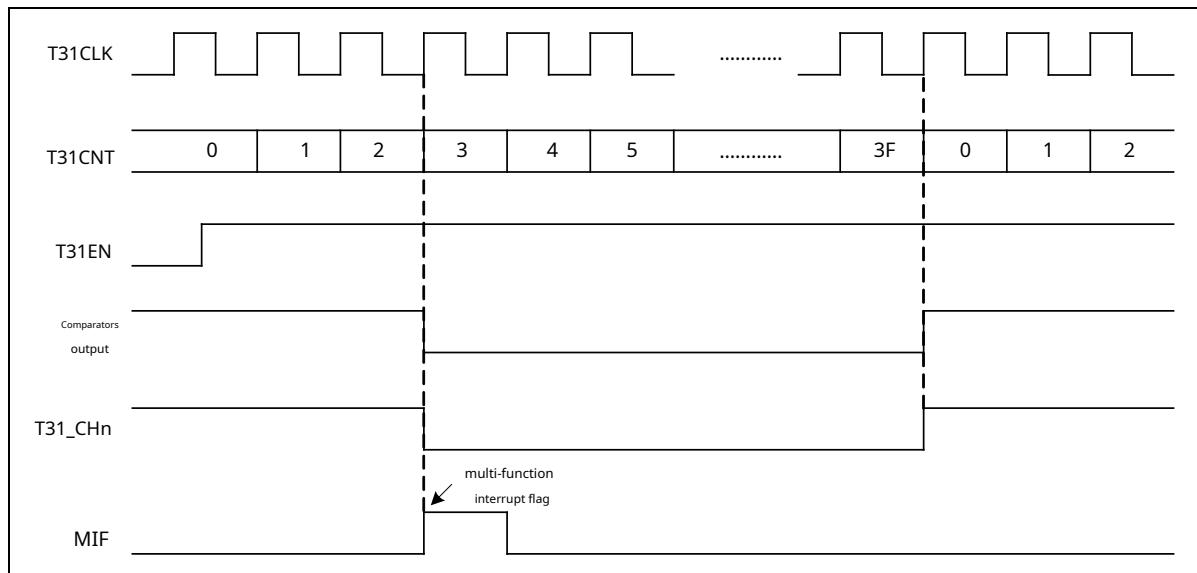
Note: in ICD In debug mode, software fixed settings are required T31C0H register HTOEOF=1, otherwise PWM The output may be abnormal.

#### 6.3.9.1 PWMnormal mode

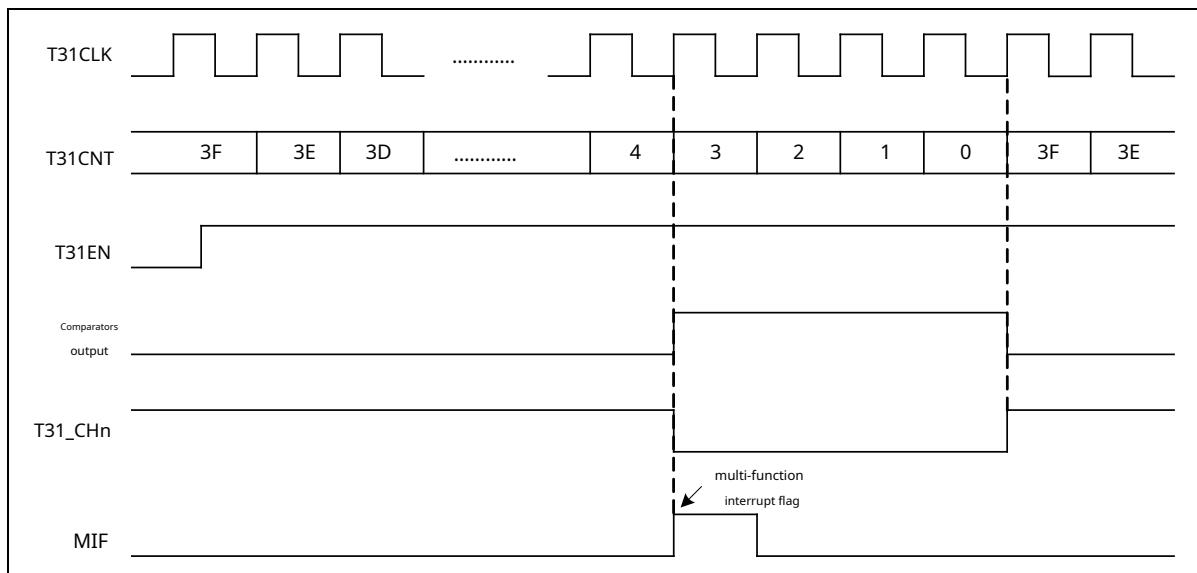
set up T31C0L register CMC=00, Configurable PWM for normal mode, in PWM In normal mode, through T31C0L register DIRS Bit, which can control the counter to count up or down.

by PWM model 1 For example, when counting up only T31CNT < T31CHnR, the output of the comparator is high, otherwise the output of the comparator remains low, and the channel port T31\_CHn The output level depends on the CHnPbit polarity control. if T31CHnR The comparison value in is greater than T31CNTLD The count reload value in , the output of the comparator remains high; if T31CHnR The comparison value in 0, the output of the comparator remains low.

by PWM model 1 For example, when counting down as long as T31CNT > T31CHnR, the output of the comparator is low, otherwise the output of the comparator remains high, and the channel port T31\_CHn The output level depends on the CHnPbit polarity control. if T31CHnR The comparison value in is greater than T31CNTLD In the counting reload value, the output of the comparator is always kept high; when counting down, the duty cycle cannot be generated. 0 of PWM waveform (even if T31CHnR=0, when counting down to T31CNT=0, the output of the comparator will still transition to a high level) .



picture6-22ordinaryPWMwaveform, counting up (T31CHnR=3,T31CNTLD=0x3F,CHnP=0)

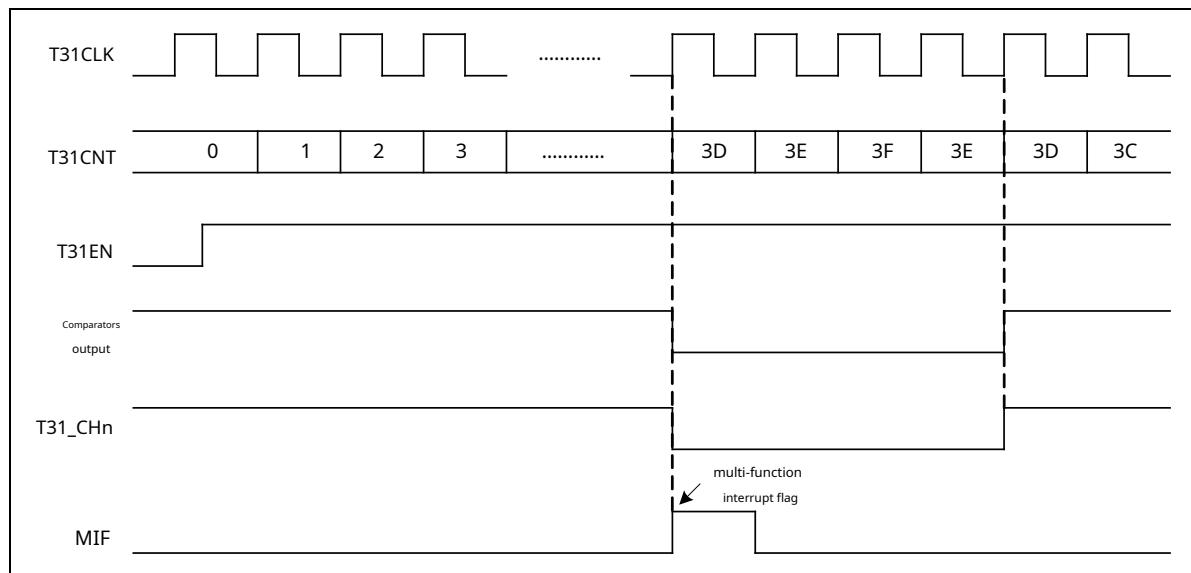


picture6-23ordinaryPWMwaveform, count down (T31CHnR=3,T31CNTLD=0x3F,CHnP=1)

### 6. 3. 9. 2 PWMCenter Alignment Mode

set up T31C0LregisterCMCbit is not 00configurable PWMFor the center-aligned mode, a total of support3A center-aligned mode for multi-function interrupt flags for each channelMIFSet on down count match1, Increment count match when set1, decrement/increment count match set1.

In the center-aligned mode, before the counter starts to work, it needs to be set by softwareT31C0Lin the registerDIRS bits, and other control registers, then set theT31EVGregisterUPTbit to generate a software update event. After the counter is enabled based onDIRSThe initial value of the bit is used to determine whether to count up or down, and the counting process is automatically controlled by the hardwareDIRSbit, to achieve counting in an alternate manner of increment and decrement, and software modification is invalid. During software initialization, the DIRSandCMCDo not modify the bits at the same time, and do not modify the value of the counter during the counting process, so as to avoid abnormalities.



picture6-24center alignmentPWMWaveform (T31CHnR=0x3D,T31CNTLD=0x3F,CHnP=0,CMC=10)

### 6. 3. 9. 3

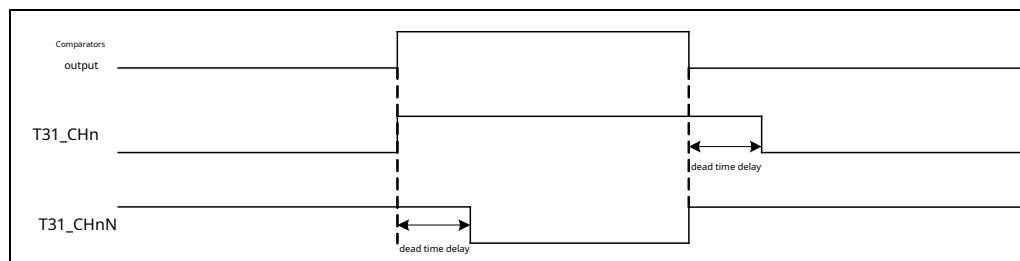
Complementary output with dead zone

T31 have 3 individual PWMs. Each channel supports two complementary output ports T31\_CHn and T31\_CHnN (n=1,2,3), can generate a complementary output signal with a dead-time delay, and the dead-time is configurable.

pass T31PINRegister CHnP and CHnNP bit, the polarity of the output signal of each port can be independently set.

Complementary output signals can be set via several control bits: T31PINRegister channel output and complementary output enable bits CHn and CHnN, T31CHBKThe channel output status total enable bit of the register CHOE, port idle and inactive state selection bits for idle and run modes NOFFS and ROFFS, T31C1H Register channel output and complementary output idle state selection bits ONS and ONSN.

pass T31DLYTRegister, can set the dead time of the complementary output signal, all channels share the dead time, according to the output signal of the comparator, generate 2 complementary output T31\_CHn and T31\_CHnN. If the dead time is greater than the effective pulse width of the output signal, no pulse output will be generated.



picture6-25Complementary Outputs with Dead-Band Delay

when PWMWhen the channel is configured as complementary output, the following register control bits will be buffered: QUR, CHn and CHnN. When a complementary channel update event occurs, these register bits will actually take effect, so that the next configuration can be pre-set and the configuration of all complementary channels will be updated at the same time.

Complementary channel update events can be set by setting T31EVG register CHUT=1 generated, or by a trigger signal (by T31C2L register TRGS bit selection trigger signal).

When a complementary channel update event occurs, the complementary channel update interrupt flag will be setCHUIF(T31IF<5>), if the complementary channel update interrupt enable bitCHUIE=1, and the total interrupt enable bitT31IE=1, a complementary channel update interrupt request is generated.

### 6. 3. 10single pulse mode

Single-pulse mode is actually based on comparator mode orPWMmode to generate waveforms, which can be set byT31C0LregisterSPME=1, to achieve a single pulse output. In one-shot mode, the counter will automatically stop counting when the next update event occurs.

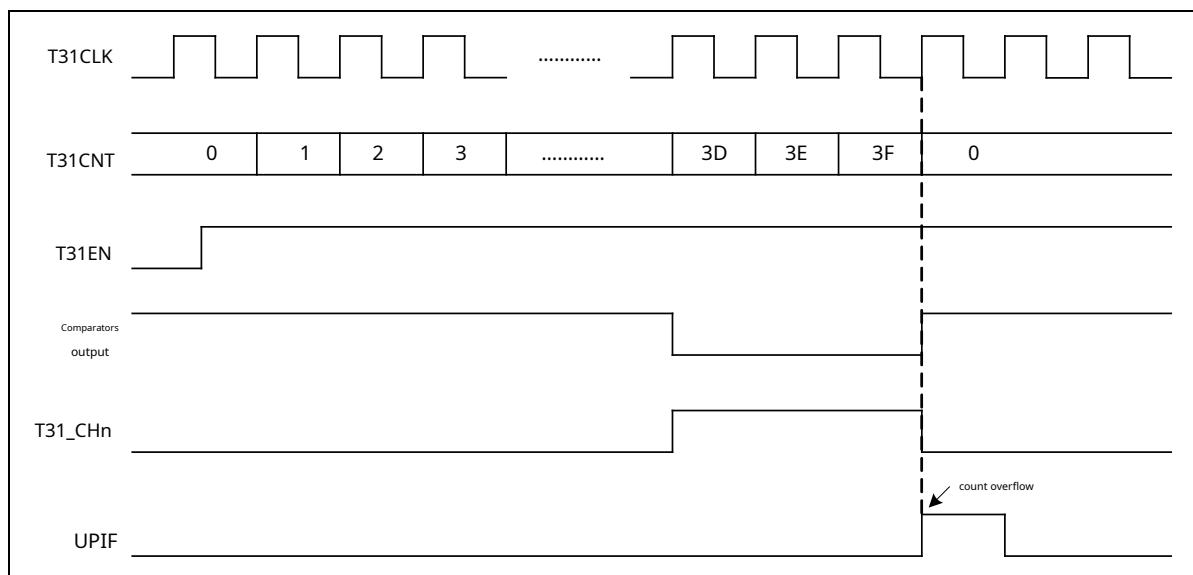
In order to ensure that the single pulse can be generated correctly, before the counter starts counting, the count value of the counter, the comparison value and the value of the count reload register must meet the following conditions:

when counting upT31CNT<T31CHnR≤T31CNTLD; while counting downT31CNT>T31CHnR.

By compare registerT31CHnRand count reload registerT31CNTLD, you can set the delay and pulse width time of single pulse output.

based on PWMThe steps of mode setting single pulse output waveform are as follows:

- set upT31CHnCregisterQURbit, choosePWMmodel1or2;
- set upT31PINregisterCHnPbit, select the channel portT31\_CHnoutput polarity;
- set upT31C0LregisterDIRS,CMC,SPMEbits, configured to count up or down,PWM Ordinary waveform mode, single pulse mode enabled;
- set upT31CHnCregisterCHnOBE=1,T31C0LregisterRLBE=1, enable the buffering function of the comparison register and the counting reload register (buffering can also be disabled according to the actual situation) ;
- set upT31CHnRegister andT31CNTLDRegister, configure single pulse output delay and pulse width time;
- set upPTbit to generate an update event;
- set upT31C0LregisterT31EN=1To start the counter, or in the trigger mode, the hardware can be automatically set by an external trigger input signalT31EN=1.



picture6-26Single Pulse Output Waveform (PWMmodel1,T31CHnR=0x3D,T31CNTLD=0x3F,CHnP=1)

In single pulse mode, if you want to quickly trigger a single pulse output through an external trigger input signal, you can set T31CHnCregisterCHnOFE=1, enable the comparator output acceleration function, so that the comparator output and the channel portT31\_CHnImmediately output the pulse waveform when the external trigger, without waiting for the count comparison match before outputting. have to be aware of is,CHnOFEbit only if the channel is configured asPWMmodel1andPWMmodel2only valid.

### 6.3.11 Shutdown function mode

The shutdown function mode can be set by the following control bits:T31CHBKregisterCHOE,NOFFSand ROFFSbit,T31C1HregisterONSandONSbit. When a shutdown occurs, the channel outputs an idle state level.

Two shutdown methods are supported: one is throughT31\_BRKThe input pin is powered down, and the other is setT31EVGregister BKT=1, to generate a shutdown event.

set upT31CHBKin the registerBKEbit can enable the shutdown function, and the polarity of the shutdown input signal can be set by BKPSbit selection,BKEandBKPScan be modified at the same time.

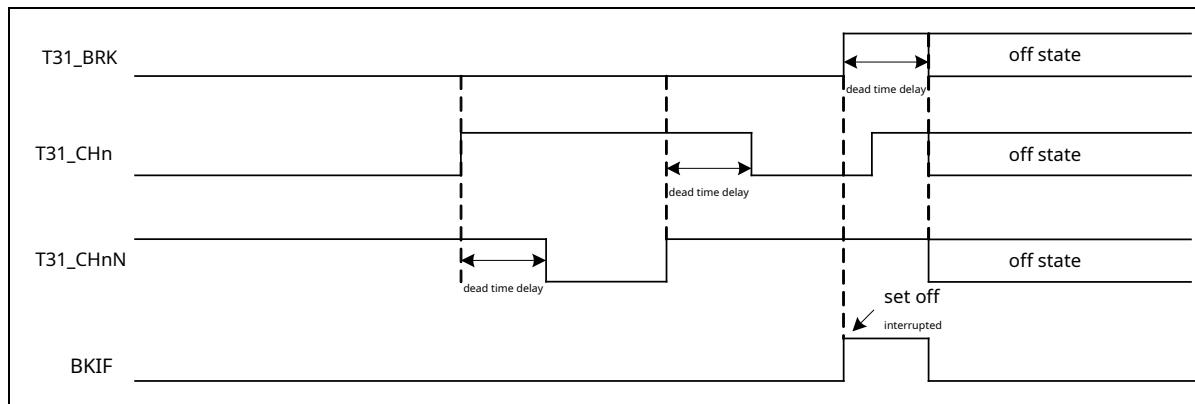
Shutdown circuit control register has write protection function, supports3protection level, which can be passedT31CHBKregister PROTBits to select the implementation of the dead time, the channel portT31\_CHn/T31\_CHnNoutput polarity and idle state,QR Write protection for configuration, shutdown enable and polarity, etc.

When shutdown occurs, do the following:

- CHOEBit is cleared immediately0, disables or forces the channel output to an idle state (byNOFFSbit selection), this operation is asynchronous, so it is still valid even when the chip system clock is turned off.
- In idle state, the output level of each channel can be controlled byT31C1HregisterONSandONSbit setting. When the channel output is disabled, theT31Port output enable is no longer controlled.
- When using complementary outputs: the channel output is first put in an inactive state, which is an asynchronous operation, and the operation is still active even if the timer does not have a clock; if the timer's clock is present, the dead time is still in effect and will restart dead time timing, after the dead time according toONSandONSWhen the bit is set, the port outputs the idle state level. Note that since the resyncCHOE, the dead time will be slightly longer than usual (approx.1~2clock cycles)
- 
- if enabledT31IELShutdown interrupt enable bit in registerBKIE, then when the interrupt flag is turned offBKIF (T31IFL<7>)for1, an interrupt request is generated.
- if enabledT31CHBKin the registerAROEbit, then when the shutdown event is canceled, at the next update eventCHOEbit is automatically set, otherwiseCHOEbit remains low until set again by software1.

Note1: Because the shutdown input signal is level active, it cannot be set simultaneously (automatically or by software) when the shutdown input is activeCHOE, and interrupt flagBKIFcannot be cleared;

Note2: After chip reset,T31CHBKin the registerPROTSBits can only be written once.



picture6-27 Shutdown Timing Diagram with Complementary Outputs with Dead Time

The channel port output supports the following four states:

- The output is disabled, the input and output status of the port is determined by the IOPort I/O Control Register PxTcontrol, the output data is controlled by the Port Level Status Register Pxcontrol.
- Output idle state, need to set CHOE=0, the port level is determined by T31C1HregisterONSn/ONSnNbit setting, all channel ports can be implemented T31\_CHn/T31\_CHnN simultaneously outputs the idle state.
- Output invalid state, need to set CHOE=1, the port level is determined by T31PINRegisterCHnP/CHnNP Bit setting, one of the two complementary output ports can output an invalid state, and the other can output a valid state.
- output valid state, Need to set CHOE=1, The port outputs the comparison result of the comparator, The output polarity is determined by T31PIN register CHnP/CHnNP whether the bit setting is inverted, two complementary outputs can be realized.

CHOE	NOFFS	ROFFS	CH	CHN	T31_CHn output state	T31_CHnN output state
1	X	0	0	0	output disable	output disable
		0	0	1	output disable	Active state, polarity configurable
		0	1	0	Active state, polarity configurable	output disable
		0	1	1	Both complementary outputs are active with configurable polarity and dead time	
		1	0	0	output disable	output disable
		1	0	1	Inactive state, level configurable	Active state, polarity configurable
		1	1	0	Active state, polarity configurable	Inactive state, level configurable
		1	1	1	Both complementary outputs are active with configurable polarity and dead time	
0	X	0	0	0	output disable	output disable
		0	0	1	output disable	output disable
		0	1	0		
		0	1	1		
		1	0	0	output disable	output disable
		1	0	1	output idle state, the port level is determined by the ONSn/ONSnN setting, and if a dead time is set, the CHOE=0. After that, restart the dead time counting, after the set	After the set dead time, the output idle state
		1	1	0		
		1	1	1		

surface6-3 Channel Complementary Output List with Shutdown

Note: When the channel output is disabled, the corresponding channel port can be used as a normal IOPort usage.

### 6.3.12 slave mode

T31The following extended slave modes are supported: Encoder mode, Reset mode, Gated mode, Trigger mode.

#### 6.3.12.1 encoder mode

T31channel1and channel2support3Encoder modes, available throughT31C2LregisterT31SMbit configuration: whenT31SM=001, the counter is only on the channel2input signalT31\_CH2edge counting, the counting direction is determined by the channel1Signal T31\_CH1level control; whenT31SM=010, the counter is only on the channel1input signal T31\_CH1edge counting, the counting direction is determined by the channel2SignalT31\_CH2level control; whenT31SM=011 , the counter is on the channel2input signal T31\_CH2and channel1input signalT31\_CH1The edge of each channel is counted. When counting, the channel signal with edge change is used as the clock, and the counting direction is controlled by the signal level of the other channel.

by settingT31PINCLregisterCH1NP/CH1PandCH2NP/CH2Pbit, you can chooseT31\_CH1 andT31\_CH2Whether the input signal of the port is inverted or not, and the filter time of the input signal can also be set according to the actual situation.

In encoder mode, the counter uses theT31\_CH1andT31\_CH2Signal edge count for both channel inputs.

After the counter starts (T31C0LregisterT31EN=1) , using one of the channel input signals as the counting clock, Both the rising and falling edges of the clock signal are valid, and the hardware automatically controls the counting direction selection bit according to the level change of another input signalDIRS(T31C0L<4>), to count up or down.

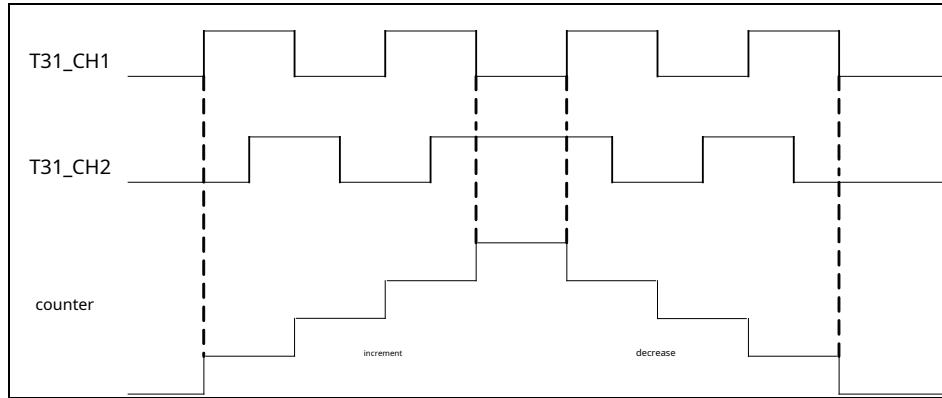
In encoder mode, the counter is equivalent to using an external clock with direction selection, and the counter is only in0arrive T31CNTLDThe register's count is continuously incremented or decremented between reload values. In encoder mode, it is forbidden to enable the external clock source mode2, both cannot work at the same time.

In encoder mode, the counter automatically adjusts counting according to the speed and direction of the external incremental encoder. The value of the counter reflects the position of the encoder, and the counting direction corresponds to the direction of rotation of the externally connected sensor. The table below shows how the counter counts for different input combinations of signals:

counting clock	The level of the other signal	T31_CH1		T31_CH2	
		rising edge	falling edge	rising edge	falling edge
T31_CH1	T31_CH2=1	decrease	increment	not count	not count
	T31_CH2=0	increment	decrease	not count	not count
T31_CH2	T31_CH1=1	not count	not count	increment	decrease
	T31_CH1=0	not count	not count	decrease	increment
T31_CH1and T31_CH2	1	decrease	increment	increment	decrease
	0	increment	decrease	decrease	increment

surface6-4Relationship between counting direction and encoder signal (channel input is not inverted)

An external incremental encoder can interface directly to the chip without external interface logic, but a comparator is usually used to convert the encoder's differential output to a digital signal, which enhances noise immunity. The third output terminal of the external incremental encoder is used to indicate the mechanical zero point, which can be connected to an external interrupt input pin of the chip to trigger a counter reset.



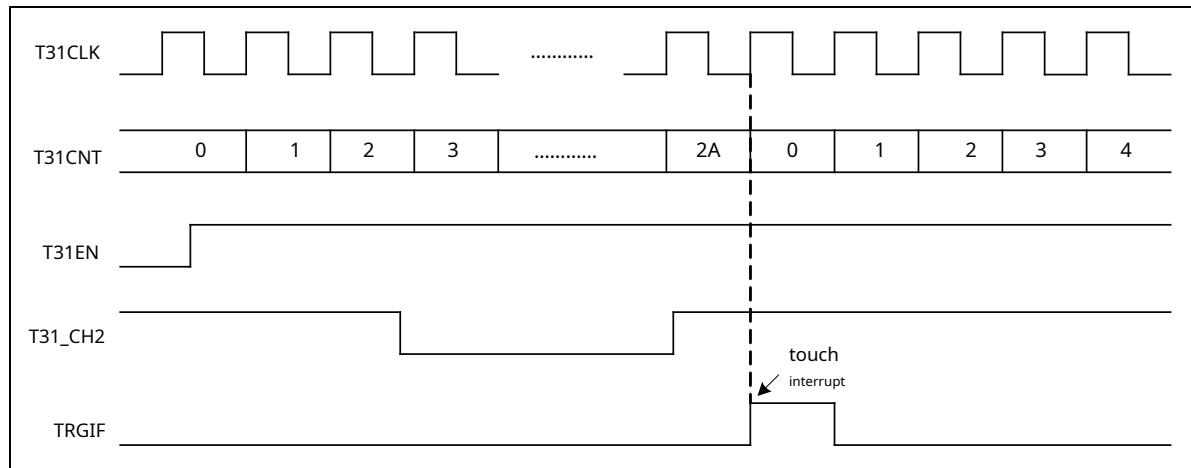
picture6-28Counter count timing in encoder mode (T31SM=010, channel input is not inverting)

### 6. 3. 12. 2 reset mode

set up T31C2L register T31SM=100, can be configured for reset mode. In reset mode, the counter uses the internal FoscClock count or external clock source mode2Counting, when a trigger input event occurs, the counter and prescaler can be reinitialized, and the prescaler counter will be cleared (the prescaler ratio is not affected) , the counter is incremented when counting It will also be cleared. When counting down, the counter will reload the value of the counting register; at the same time, if T31COL register UES=0, also generates an update event that makes all reloadable registers (T31CNTLD, T31CHnR) are overloaded and initialized.

In reset mode, the trigger input signal can be T31C2L register TRGS, select the trigger signal source; through T31PIN register CHnNP/CHnPbit or T31C2H register ETEGBit, select the effective edge of the trigger signal; you can also configure the filter time of the input signal according to actual needs.

set up T31COL register T31EN=1, to start counting, when a trigger input is generated, the counter is cleared from 0Restart counting and set the trigger interrupt flag at the same time TRGIF(T31IF<6>), if the interrupt enable bit at this time TRGIE=1 , an interrupt request will be generated.



picture6-29Counting Timing in Reset Mode (T31CNTLD=0x3F, aisle2Port signal rising edge trigger)

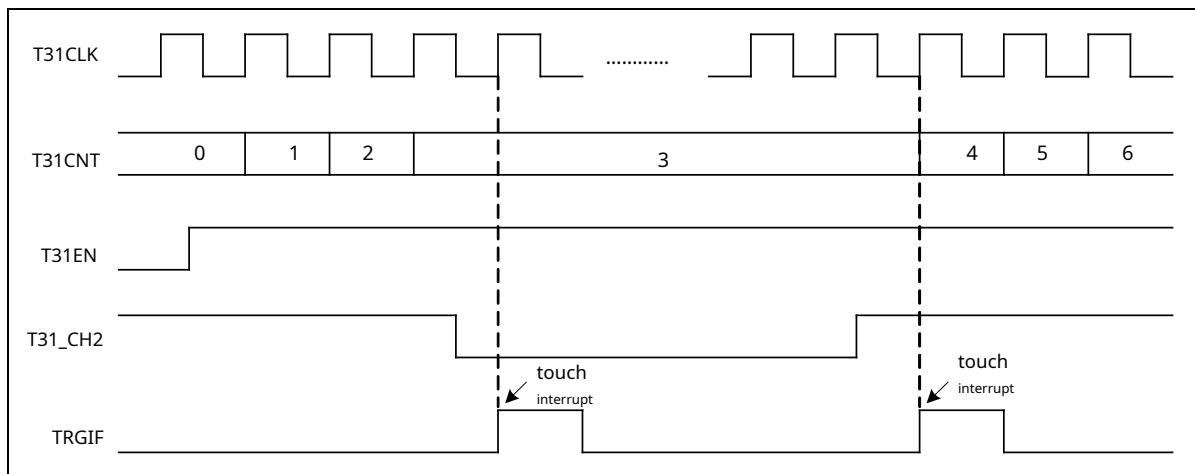
Note: In external clock source mode2operation, the reset mode can be enabled at the same time, at this time T31\_ETR signal as an external clock input, you need to select another input as a trigger signal for reset mode, prohibiting the T31\_ETR as a trigger signal (i.e. TRGScannot be set to 11) .

### 6. 3. 12. 3 gating mode

set up T31C2LregisterT31SM=101, can be configured as a gated mode. In gated mode, the counter uses an internalFosc Clock count or external clock source mode2Counting, the counting of the counter can be enabled or stopped by the trigger input signal as a gating signal (counter enable bitT31ENstill need to remain as1).

In gated mode, the trigger input signal can be passed through T31C2LregisterTRGS, select the trigger signal source; through T31PINregisterCHnNP/CHnPbit or T31C2HregisterETEGBit, select the active level of the trigger signal. When the trigger signal level is valid, the counter keeps counting, otherwise it stops counting. The filter time of the input signal can also be configured according to actual needs.

set up T31C0LregisterT31EN=1, start counting, as long as the gate signal level is valid, the counter will count normally, once the gate signal level is invalid, it will stop counting. When the counter stops counting or restarts counting, the trigger interrupt flag will be set TRGIF(T31IF<6>) , if the interrupt enable bit at this time TRGIE=1, will generate a interrupt request.



picture6-30Count Timing in Gated Mode (T31CNTLD=0x3F, aisle2Port gating signal active high)

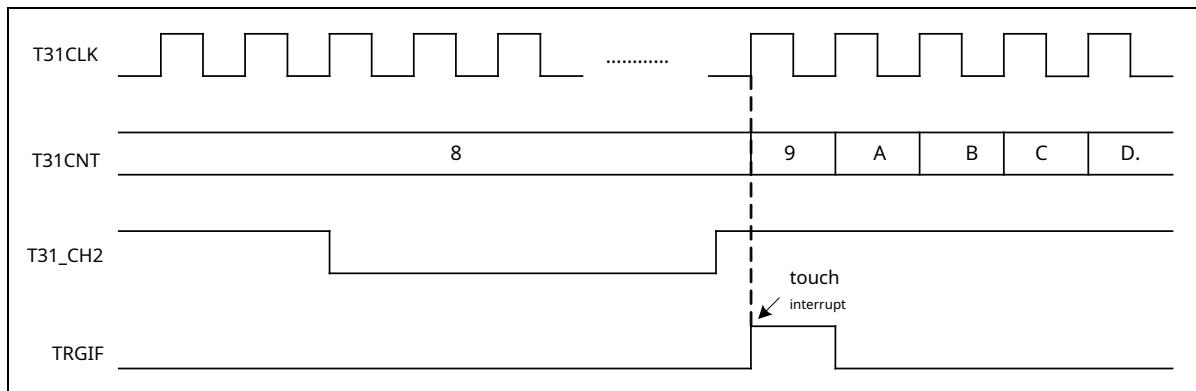
Note: In external clock source mode2When working, the gated mode can be enabled at the same time, at this time T31\_ETRsignal as an external clock input, you need to select another input as a gating signal for gating mode, prohibiting the T31\_ETRAs a gating signal (i.e. TRGScannot be set to111) .

### 6. 3. 12. 4 trigger mode

set up T31C2LregisterT31SM=110, can be configured as trigger mode, in trigger mode, the counter uses the internalFoscClock count or external clock source mode2Counting, the start of the counter can be enabled by the valid edge of the trigger input signal, and the hardware is automatically set T31EN=1, no software setup required.

In trigger mode, the trigger input signal can be T31C2LregisterTRGS, select the trigger signal source; through T31PINregisterCHnNP/CHnPbit or T31C2HregisterETEGBit, select the effective edge of the trigger signal; you can also configure the filter time of the input signal according to actual needs.

When a trigger input is generated, the hardware automatically sets the T31EN=1, start the counter to count, and set the trigger interrupt flag at the same time TRGIF(T31IF<6>), if the interrupt enable bit at this time TRGIE=1, an interrupt request will be generated.



picture6-31Count timing in trigger mode (T31CNTLD=0x3F, aisle2Port signal rising edge trigger)

Note: In external clock source mode2When working, the trigger mode can be enabled at the same time, at this timeT31\_ETRsignal as an external clock input, you need to select another input as a trigger signal in trigger mode, prohibiting theT31\_ETRas a trigger signal (i.e.TRGScannot be set to111) .

### 6.3.13 ICDsdebug in suspend modePWMoutput

existICDIn debug mode, software fixed settings are requiredHTOE<sub>OFF</sub>(T31C0H<6>)=1, otherwisePWMThe output may be abnormal. existICDWhen debugging is paused,PWMoutput byHALT\_PWM(PWEN<0>) bit decision.

when settingHALT\_PWM=1, when debugging is suspended, the counter stops counting and shuts down thePWMoutput, at this time PWMThe input/output status of the port is determined by the corresponding port input and output control registerPxTThe corresponding control bit determines ifPxTIf the port is configured as an input, the corresponding port is in a high-impedance state. IfPxTConfigure the port as an output, then the corresponding port level is determined by the port level status registerPxDecide.

when settingHALT\_PWM=0, when debugging is suspended, the counter continues to count, keepingPWMoutput.

### 6.3.14 special function register

#### 6.3.14.1 counter register low8bits (T31CNTL)

bit	7	6	5	4	3	2	1	0
name	CNT<7:0>							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7~0      CNT<7:0>: The count value is low8bit

#### 6.3.14.2 counter register high8bits (T31CNTH)

bit	7	6	5	4	3	2	1	0
name	CNT<15:8>							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7~0      CNT<15:8>: The count value is high8bit

**6. 3. 14. 3 Prescaler Register Low8bits (T31PRSL)**

T31PRSL: Prescaler Register Low8bits (T31PRSL)								
bit	7	6	5	4	3	2	1	0
<b>name</b>	PRS<7:0>							
<b>R/W</b>	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
<b>POR</b>	0	0	0	0	0	0	0	0

Bit 7~0 PRS&lt;7:0&gt;: Low prescaler value8bit

**6. 3. 14. 4 Prescaler Register High8bits (T31PRSH)**

T31PRSH: Prescaler Register High8bits (T31PRSH)								
bit	7	6	5	4	3	2	1	0
<b>name</b>	PRS<15:8>							
<b>R/W</b>	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
<b>POR</b>	0	0	0	0	0	0	0	0

Bit 7~0 PRS&lt;15:8&gt;: High prescaler value8bit

Note: The counter uses the prescaled clock to count, and the prescaler ratio isPRS&lt;15:0&gt; + 1.

**6. 3. 14. 5 Counter Reload Register Low8bits (T31CNTLDL)**

T31CNTLDL: Counter Reload Register Low8bits (T31CNTLDL)								
bit	7	6	5	4	3	2	1	0
<b>name</b>	CNTLD<7:0>							
<b>R/W</b>	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
<b>POR</b>	1	1	1	1	1	1	1	1

Bit 7~0 CNTLD&lt;7:0&gt;: counter count reload value low8bit

**6. 3. 14. 6 Counter Reload Register High8bits (T31CNTLDH)**

T31CNTLDH: Counter Reload Register High8bits (T31CNTLDH)								
bit	7	6	5	4	3	2	1	0
<b>name</b>	CNTLD<15:8>							
<b>R/W</b>	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
<b>POR</b>	1	1	1	1	1	1	1	1

Bit 7~0 CNTLD&lt;15:8&gt;: Counter count reload value high8bit

Note1: When an update event occurs, T31CNTLDThe value of will be automatically reloaded, re-initializing the counter, when T31CNTLD&lt;15:0&gt;=0time, count

After re-initialization, it will not continue counting;

Note2: pair register T31CNTLDA write operation requires first writing a high8bit register T31CNTLDH, then write low8bit register T31CNTLDL.

**6. 3. 14. 7 postscaler register (T31POS)**

T31POS: Postscaler Register (T31POS)								
bit	7	6	5	4	3	2	1	0
<b>name</b>	POS<7:0>							
<b>R/W</b>	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
<b>POR</b>	0	0	0	0	0	0	0	0

Bit 7~0      POS&lt;7:0&gt;: Postscale value

**6. 3. 14. 8 aisle1Capture/Compare Register Low8bits (T31CH1RL)**

T31CH1RL:aisle1Capture/Compare Register1Low8bits (T31CH1RL)								
bit	7	6	5	4	3	2	1	0
<b>name</b>	CH1R<7:0>							
<b>R/W</b>	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
<b>POR</b>	0	0	0	0	0	0	0	0

Bit 7~0      CH1R&lt;7:0&gt;:aisle1Capture/Compare Value Low8bit

**6. 3. 14. 9 aisle1Capture/Compare Register High8bits (T31CH1RH)**

T31CH1RH:aisle1Capture/Compare Register High8bits (T31CH1RH)								
bit	7	6	5	4	3	2	1	0
<b>name</b>	CH1R<15:8>							
<b>R/W</b>	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
<b>POR</b>	0	0	0	0	0	0	0	0

Bit 7~0      CH1R&lt;15:8&gt;:aisle1Capture/compare value high8bit

NOTE: For register T31CH1RA write operation requires first writing a high8bit register T31CH1RH, then write low8bit register T31CH1RL.

**6. 3. 14. 10aisle2Capture/Compare Register Low8bits (T31CH2RL)**

T31CH2RL:aisle2Capture/Compare Register Low8bits (T31CH2RL)								
bit	7	6	5	4	3	2	1	0
<b>name</b>	CH2R<7:0>							
<b>R/W</b>	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
<b>POR</b>	0	0	0	0	0	0	0	0

Bit 7~0      CH2R&lt;7:0&gt;:aisle2Capture/Compare Value Low8bit

**6. 3. 14. 11aisle2Capture/Compare Register High8bits (T31CH2RH)**

T31CH2RH:aisle2Capture/Compare Register High8bits (T31CH2RH)								
bit	7	6	5	4	3	2	1	0
<b>name</b>	CH2R<15:8>							
<b>R/W</b>	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
<b>POR</b>	0	0	0	0	0	0	0	0

Bit 7~0      CH2R&lt;15:8&gt;:aisle2Capture/compare value high8bit

NOTE: For registerT31CH2RA write operation requires first writing a high8bit registerT31CH2RH, then write low8bit registerT31CH2RL.

#### 6. 3. 14. 12aisle3Capture/Compare Register Low8bits (T31CH3RL)

T31CH3RL:aisle3Capture/Compare Register Low8bits (T31CH3RL)								
bit	7	6	5	4	3	2	1	0
<b>name</b>	CH3R<7:0>							
<b>R/W</b>	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
<b>POR</b>	0	0	0	0	0	0	0	0

Bit 7~0 CH3R<7:0>:aisle3Capture/Compare Value Low8bit

#### 6. 3. 14. 13aisle3Capture/Compare Register High8bits (T31CH3RH)

T31CH3RH:aisle3Capture/Compare Register High8bits (T31CH3RH)								
bit	7	6	5	4	3	2	1	0
<b>name</b>	CH3R<15:8>							
<b>R/W</b>	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
<b>POR</b>	0	0	0	0	0	0	0	0

Bit 7~0 CH3R<15:8>:aisle3Capture/compare value high8bit

NOTE: For registerT31CH3RA write operation requires first writing a high8bit registerT31CH3RH, then write low8bit registerT31CH3RL.

#### 6. 3. 14. 14aisle4Capture/Compare Register Low8bits (T31CH4RL)

T31CH4RL:aisle4Capture/Compare Register Low8bits (T31CH4RL)								
bit	7	6	5	4	3	2	1	0
<b>name</b>	CH4R<7:0>							
<b>R/W</b>	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
<b>POR</b>	0	0	0	0	0	0	0	0

Bit 7~0 CH4R<7:0>:aisle4Capture/Compare Value Low8bit

#### 6. 3. 14. 15aisle4Capture/Compare Register High8bits (T31CH4RH)

T31CH4RH:aisle4Capture/Compare Register High8bits (T31CH4RH)								
bit	7	6	5	4	3	2	1	0
<b>name</b>	CH4R<15:8>							
<b>R/W</b>	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
<b>POR</b>	0	0	0	0	0	0	0	0

Bit 7~0 CH4R<15:8>:aisle4Capture/compare value high8bit

NOTE: For registerT31CH4RA write operation requires first writing a high8bit registerT31CH4RH, then write low8bit registerT31CH4RL.

**6.3. 14. 16Dead-Band Delay Register (T31DLYT)**

T31DLYT: Dead-Band Delay Register (T31DLYT)								
bit	7	6	5	4	3	2	1	0
name	DLYT<7:0>							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7~0

DLYT&lt;7:0&gt;: Dead zone delay setting value

DLYT&lt;7:5&gt;=0xx: The dead time is DLYT&lt;7:0&gt; x Tdfck

DLYT&lt;7:5&gt;=10x: The dead time is (128+2 x DLYT&lt;5:0&gt;)x Tdfck

DLYT&lt;7:5&gt;=110: The dead time is (256+8 x DLYT&lt;4:0&gt;)x Tdfck

DLYT&lt;7:5&gt;=111: The dead time is (512+16 x DLYT&lt;4:0&gt;)x

Note1:when T31CHBKregisterPROTS=01,10or11(set as protection level1,2or3) ,T31DLYTThe register will not be rewritten;  
 Note 2:Tdfckis the digital filter operating clock period, byDFCKS(T31C0H<1:0>) bit setting.

**6.3. 14. 17control register0Low8bits (T31C0L)**

T31C0L: Control register0Low8bits (T31C0L)								
bit	7	6	5	4	3	2	1	0
name	RLB	CMC<1:0>		DIRS	SPME	UES	UED	T31EN
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7

RLB: Count reload register buffer enable bit

0:prohibit,T31CNTLDThe register is not buffered and takes effect immediately after writing 1:Enable,

T31CNTLDRegisters are buffered and take effect on the next update event CMC&lt;1:0&gt;: Center-aligned

Bit 6~5

mode control bit (recommended only inT31EN=0overwrite this bit)

00: Normal mode, the counter is based on the direction bitDIRSCount up or down, controlled by software.

01: center alignment mode1, the counter counts up or down in an alternating fashion. Only when the counter is counting down, the multi-function interrupt flag bitMIFwill be placed1(The corresponding channel needs to be configured as an output) .

10: center alignment mode2, the counter counts up or down in an alternating fashion. Only when the counter is counting up, the multi-function interrupt flag bitMIFwill be placed1.

11: center alignment mode3, the counter counts up or down in an alternating fashion. When the counter counts up or down, the multi-function interrupt flagMIFwill be set1.

Bit 4

DIRS: Counter counting direction selection bit (in center-aligned and encoder mode, this bit is read-only)

0: count up

1: count down

Bit 3

SPME: Single pulse mode enable bit

0: disabled, the counter does not stop counting when an update event occurs

1: Enabled, when the next update event occurs,T31ENbit cleared0, the counter stops counting UES:

Bit 2

update event source selection bit

0: Counter overflow/underflow, software settingUPT=1, an update generated from the pattern controller, can generate an update event

1: The counter overflows/underflows, which can generate update events

Bit 1      UED: update event disable bit

0: Do not suppress generation of update events

1: Prohibit generating update events

Bit 0      T31EN:T31counter enable bit

0:prohibit

1:Enable

Note1: When the counter is enabled (T31EN=1) to prohibit the transition from normal mode to center-aligned mode;

Note2: When the counter is configured in center-aligned mode or encoder mode, DIRSThe bit can only be read and cannot be rewritten by software;

Note3:rightT31ENbit, automatically set by hardware in trigger mode1, without software setting1.

### 6. 3. 14. 18control register0high8bits (T31C0H)

T31C0H: Control register0high8bits (T31C0H)								
bit	7	6	5	4	3	2	1	0
name	—	HTOE OFF	—	—	—	—	DFCKS<1:0>	
R/W	—	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7      Not used, the software needs to be fixed as0

Bit 6      HTOE OFF:existICDWhen debugging,PWMstate control (notICDIn debug mode, this bit has no effect) 0:

Reserved, forbidden to use

1:existICDIn debug mode, this bit needs to be fixed by software as1, otherwise PWMThe output may be abnormally unused, and

Bit 5~2      the software needs to be fixed as0

Bit 1~0      DFCKS<1:0>: Dead zone generator and digital filter operating clock frequencyFdckoption bit

00:Fosc

01:Fosc/2

10:Fosc/4

11: Reserved, forbidden to use

Note:PWMDead zone generator for complementary output signals, external trigger inputT31\_ETRand the digital filter of the input signal of each channel are based on the internal

Department clockFoscOr its frequency-divided clock works.

### 6. 3. 14. 19control register1Low8bits (T31C1L)

T31C1L: Control register1Low8bits (T31C1L)								
bit	7	6	5	4	3	2	1	0
name	—	ADTRGS<2:0>			—	CHCUS	—	CHCBE
R/W	R/W	R/W	R/W	R/W	—	R/W	—	R/W
POR	0	0	0	0	0	0	0	0

Bit 7      Not used, the software needs to be fixed as0

Bit 6~4      ADTRGS<2:0>:ADCTrigger source selection bit, start after triggerADCconvert

000: counter is reinitialized: setT31EVGregisterUPT=1An update event is generated; or a trigger input event occurs in reset mode

001: The counter is enabled to count: setT31C0LregisterT31EN=1Enable counting; or in trigger mode

In the mode, the trigger input signal enables counting; or in the gated mode, the counting is enabled

010: generation of update event

011:aisle1The multi-function interrupt flagMIF1 100:aisle1A

compare match has occurred 101:aisle2A compare match

has occurred 110:aisle3A compare match has occurred 111

:aisle4A comparison match has occurred and is not used,

and the software needs to be fixed as0

**Bit 3**

**Bit 2**

CHCUS: Complementary channel control bit update selection bit (only in complementary output mode, andCHCBE=1valid)

0: only inCHUT=1These control bits are only updated when a complementary channel update event is triggered

1:existCHUT=1Or when the trigger signal has a rising edge, these control bits will be updated and unused, and

**Bit 1**

the software needs to be fixed as0

**Bit 0**

CHCBE: Channel control bit buffer enable bit (valid only in complementary output mode)

0:CH,CHNandQURNo buffering, effective immediately after rewriting

1:CH,CHNandQURWith buffering, it does not take effect immediately after rewriting, depending onCHCUSsetting

### 6. 3. 14. 20control register1high8bits (T31C1H)

T31C1H: Control register1high8bits (T31C1H)								
bit	7	6	5	4	3	2	1	0
name	—	ONS4	ONS3N	ONS3	ONS2N	ONS2	ONS1N	ONS1
R/W	—	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0
Bit 7	Not used, the software needs to be fixed as0							
Bit 6	ONS4:aisle4Idle state select bit for output 0: output when idle0 1: output when idle1							
Bit 5	ONS3N:aisle3Idle state selection bits for complementary outputs 0: output when idle0 1: output when idle1							
Bit 4	ONS3:aisle3Output idle state3option bit 0: output when idle0 1: output when idle1							
Bit 3	ONS2N:aisle2Idle state selection bits for complementary outputs 0: output when idle0 1: output when idle1							
Bit 2	ONS2:aisle2Idle state select bit for output 0: output when idle0 1: output when idle1							
Bit 1	ONS1N:aisle1Idle state selection bits for complementary outputs 0: output when idle0 1: output when idle1							
Bit 0	ONS1:aisle1Idle state select bit for output							

Bit 7 Not used, the software needs to be fixed as0

Bit 6 ONS4:aisle4Idle state select bit for output

0: output when idle0

1: output when idle1

Bit 5 ONS3N:aisle3Idle state selection bits for complementary outputs

0: output when idle0

1: output when idle1

Bit 4 ONS3:aisle3Output idle state3option bit

0: output when idle0

1: output when idle1

Bit 3 ONS2N:aisle2Idle state selection bits for complementary outputs

0: output when idle0

1: output when idle1

Bit 2 ONS2:aisle2Idle state select bit for output

0: output when idle0

1: output when idle1

Bit 1 ONS1N:aisle1Idle state selection bits for complementary outputs

0: output when idle0

1: output when idle1

Bit 0 ONS1:aisle1Idle state select bit for output

0: output when idle0

1: output when idle1

Note1:set upT31CHBRegisterCHOE=0,NOFFS=1After that, the channel outputs the idle state;

Note2:set upT31CHBRegisterPROTSAfter the bit is write-protected, the T31C1HThe bits in the register can no longer be rewritten.

### 6. 3. 14. 21control register2Low8bits (T31C2L)

T31C2L: Control register2Low8bits (T31C2L)								
bit	7	6	5	4	3	2	1	0
name	—	TRGS<2:0>			COCE	T31SM<2:0>		
R/W	R/W	R/W	R/W	R/W	—	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7 Not used, the software needs to be fixed as0

Bit 6~4 TRGS<2:0>: Trigger source selection bit, the selected trigger source is used as a trigger signal to synchronize the counter

000~011:reserve

100: aisle1Input pulse signal edge (both rising and falling edges are valid, can passCH1IOSbit selection channel input port)

101: aisle1Input signal (supports input filtering, can pass throughCH1IOSbit select channel input port) 110: aisle2 Input signal (supports input filtering, can pass throughCH2IOSbit select channel input port) 111: External trigger inputT31\_ETR(Supports input filtering and prescaler) COCE: All channel comparator outputs are cleared0total

Bit 3 enable bit

0:prohibit

1: Enable, external trigger input signalT31\_ETRis high, if theCHnCOCE=1, then the corresponding channel's comparator output can be cleared0

Bit 2~0

T31SM<2:0>:T31Slave mode select bit

000: Slave mode is disabled, the counter uses the system clockFoscThe internal clock source counts (ECM2E=0) or outside External clock source mode2count(ECM2E=1)

001: encoder mode1, the count clock for the channel2portT31\_CH2The rising and falling edges of the signal, the meter channel1portT31\_CH1level control

010: encoder mode2, the count clock for the channel1portT31\_CH1The rising and falling edges of the signal, the meter channel2portT31\_CH2level control

011: encoder mode3, the count clock for the channel1portT31\_CH1and channel2portT31\_CH2letter The rising edge and falling edge of the signal, when any one of the input signals has a valid edge, it will count, and the counting direction is controlled by the level of the other input signal

100: Reset mode, re-initialize the counter on the rising edge of the trigger signal, and update the register 101: Gate control mode, when the trigger signal is high level, the counter counts, when it is low level, the counter stop counting and keep

110: Trigger mode, the counter is enabled by the rising edge of the trigger signal and starts counting 111:

External clock source mode1, the counter clock is the trigger signal, counting on the rising edge

Note1: In order to avoid false edge detection, it is recommended that when the trigger signal does not work (such asT31SM=000) to select the trigger

source; Note2: When using the gated mode, the channel cannot be1port input pulse edge as a trigger signal (TRGS=100) , because the trigger signal at this time is only a

Pulse signal, not the level signal required by the gating mode;

Note 3: When selecting an external input signal, the effective edge of the trigger signal is related to the polarity setting of the external input signal;

Note 4: In encoder mode, the input signal of the channel port used for counting clock and counting direction control can be passed through the corresponding CHnNP/CHnPbit set

Input polarity.

### 6. 3. 14. 22control register2high8bits (T31C2H)

T31C2H: Control register2high8bits (T31C2H)								
bit	7	6	5	4	3	2	1	0
name	ETEG	ECM2E	ETPRS<1:0>			ETFS<3:0>		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7                    ETEG: External trigger signal edge selection bit

    0: rising edge

    1: falling edge

Bit 6                    ECM2E: External clock source mode2enable bit

    0:prohibit

    1: Enable, the counter clock is an external input T31\_ETRThe rising or falling edge of the

Bit 5~4                signal ETPRS<1:0>: External input clock T31\_ETRPrescaler selection bits

    00:1:1

    01:1:2

    10:1:4

    11:1:8

Bit 3~0                ETFS<3:0>: External input clock T31\_ETRThe filter time selection bit

    0000: No filtering

    0001:2\*Tosc(Toscfor system clock cycles)

    0010:4\*Tosc

    0011:8\*Tosc

    0100:12\*Tdfck(Tdfckis the digital filter operating clock period, byDFCKSbit set)

    0101:16\*Tdfck

    0110:24\*Tdfck

    0111:32\*Tdfck

    1000:48\*Tdfck

    1001:64\*Tdfck

    1010:80\*Tdfck

    1011:96\*Tdfck

    1100:128\*Tdfck

    1101:160\*Tdfck

    1110:192\*Tdfck

    1111:256\*Tdfck

Note1: External clock source mode2, equivalent to the external clock source mode1when, will T31\_ETRsignal as a trigger signal (TRGS=111) ;

Note2: If the external clock source mode1and external clock source mode2enable at the same time, External Clock Source Mode2high priority, Then the external clock input is T31\_ETR

Signal;

Note3: External input clock T31\_ETRThe frequency cannot exceed the counter clock frequency of the 1/4, can use prescaler to reduce T31\_ETRFrequency of.

### 6. 3. 14. 23aisle1Control Register (T31CH1C)

comparator mode

bit	7	6	5	4	3	2	1	0
name	CH1COCE	CH1OM<2:0>			CH1OBE	CH1OFE	CH1IOS<1:0>	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7 CH1COCE:aisle1The comparator output is cleared0Enable bit (only when the enable bit is always cleared to 0COCE=1valid)  
0:prohibit  
1: Enable, when the external trigger input signalT31\_ETRis high, the comparator output will be cleared

Bit 6~4 0CH1OM<2:0>:aisle1comparator andPWMworking mode selection bit  
000: Comparator mode, do not change port output when matching  
001: Comparator mode, comparator output on match1, the channel port output level depends onCH1Pbit 010: Comparator mode, comparator output on match0, the channel port output level depends onCH1Pbit 011: Comparator mode, the output of the comparator is inverted when it matches, and the output level of the channel port is inverted 100: The comparator output is forced to0, the channel port output level depends onCH1Pbit 101: The comparator output is forced to1, the channel port output level depends onCH1Pbit  
110:PWMmodel1, during the increment counting process, whenT31CNT<T31CH1R, the comparator output1, otherwise output0;During counting down, whenT31CNT>T31CH1R, the comparator output0, otherwise output1. The channel port output level depends on theCH1Pbit  
111:PWMmodel2, during the increment counting process, whenT31CNT<T31CH1R, the comparator output0, otherwise output1;During counting down, whenT31CNT>T31CH1R, the comparator output1, otherwise output0. The channel port output level depends on theCH1Pbit CH1OBE:aisle1Compare register buffer enable bit

Bit 3 0:prohibit,T31CH1RThe value after rewriting the register will take effect immediately  
1:Enable,T31CH1RThe register is buffered, and the rewritten value will not take effect immediately until the next update event occurs.PWMmode is recommended to enable this bit to ensure PWMComplete waveform

Bit 2 CH1OFE:aisle1Compare output acceleration enable bit (only inPWM1orPWM2mode)  
0: Disabled, after detecting the external trigger input signal, the channel output port still needs to wait until the count value of the counter matches the comparison register before generating the required output level according to the specific setting  
1: Enable, after detecting the external trigger input signal, the channel output port will immediately generate the required output level according to the specific settings after the comparison match occurs, without waiting for the count value of the counter to actually match the comparison register, so that Minimizes delay time between channel output and trigger input

Bit 1~0 signal CH1IOS<1:0>:aisle1I/O select bits (only inCH1E=0,aisle1writable when closed)  
00:aisle1is an output, the channel1After enabling, the channel1portT31\_CH1is set to output 01:aisle1For input, use the channel1portT31\_CH1as input 10:aisle1For input, use the channel2portT31\_CH2as input 11:aisle1input, fixed input0

Note1:whenT31CHBRegisterPROTS=11(set as protection level3) ,andCH1IOS=00(channel is in output mode),CH1OM andCH1OBEbits will not be rewritten;

Note2: in the comparator andPWMmode, channel1needs to be set as an output,CH1IOS=00.

## catcher mode

bit	7	6	5	4	3	2	1	0
name	CH1IFS<3:0>				CH1IM<1:0>		CH1IOS<1:0>	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7~4 CH1IFS<3:0>:aisle1Capture input filter time selection bits

- 0000: No filtering
- 0001:2\*Tosc(Toscfor system clock cycles)
- 0010:4\*Tosc
- 0011:8\*Tosc
- 0100:12\*Tdfck(Tdfckclock period for digital filter operation)
- 0101:16\*Tdfck
- 0110:24\*Tdfck
- 0111:32\*Tdfck
- 1000:48\*Tdfck
- 1001:64\*Tdfck
- 1010:80\*Tdfck
- 1011:96\*Tdfck
- 1100:128\*Tdfck
- 1101:160\*Tdfck
- 1110:192\*Tdfck
- 1111:256\*Tdfck

Bit 3~2 CH1IM<1:0>:aisle1Catcher operating mode selection bit

- 00:Every1input signal edge, capture once
- 01:Every2input signal edge, capture once
- 10:Every4input signal edge, capture once
- 11:Every8input signal edge, capture once

Bit 1~0 CH1IOS<1:0>:aisle1I/O select bits (only inCH1E=0,aisle1writable when closed)  
 00:aisle1is an output, the channel1After enabling, the channel1portT31\_CH1is set to  
 output 01:aisle1For input, use the channel1portT31\_CH1as input 10:aisle1For input,  
 use the channel2portT31\_CH2as input 11:aisle1input, fixed input0

Note: In capture mode, channel1needs to be set as input,CH1IOS=01or10.

## 6. 3. 14. 24aisle2Control Register (T31CH2C)

## comparator mode

bit	7	6	5	4	3	2	1	0
name	CH2COCE	CH2OM<2:0>			CH2OBE	CH2OFE	CH2IOS<1:0>	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7	CH2COCE:aisle2compare output clear0Enable bit (only when the total clear0enable bitCOCE=1valid) 0:prohibit 1: Enable, when the external trigger input signalT31_ETRis high, the comparator output will be cleared
Bit 6~4	0CH2OM<2:0>:aisle2comparator andPWMworking mode selection bit  000: Comparator mode, do not change port output when matching 001: Comparator mode, comparator output on match1, the channel port output level depends onCH2Pbit 010: Comparator mode, comparator output on match0, the channel port output level depends onCH2Pbit 011: Comparator mode, the output of the comparator is inverted when it matches, and the output level of the channel port is inverted 100: The comparator output is forced to0, the channel port output level depends onCH2Pbit 101: The comparator output is forced to1, the channel port output level depends onCH2Pbit 110:PWMmodel1, during the increment counting process, whenT31CNT<T31CH2R, the comparator output1, otherwise output0;During counting down, whenT31CNT>T31CH2R, the comparator output0, otherwise output1. The channel port output level depends on theCH2Pbit 111:PWMmodel2, during the increment counting process, whenT31CNT<T31CH2R, the comparator output0, otherwise output1;During counting down, whenT31CNT>T31CH2R, the comparator output1, otherwise output0. The channel port output level depends on theCH2Pbit CH2OBE:aisle2Compare register buffer enable bit
Bit 3	0:prohibit,T31CH2RThe value after rewriting the register will take effect immediately 1:Enable,T31CH2RThe register is buffered, and the rewritten value will not take effect immediately until the next update event occurs.PWMmode is recommended to enable this bit to ensure PWMComplete waveform CH2OFE
Bit 2	:aisle2Compare compare output acceleration enable bit (only inPWM1orPWM2mode)  0: Disabled, after detecting the external trigger input signal, the channel output port still needs to wait until the count value of the counter matches the comparison register before generating the required output level according to the specific setting 1: Enable, after detecting the external trigger input signal, the channel output port will immediately generate the required output level according to the specific settings after the comparison match occurs, without waiting for the count value of the counter to actually match the comparison register, so that Minimizes delay time between channel output and trigger input
Bit 1~0	signal CH2IOS<1:0>:aisle2I/O select bits (only inCH2E=0,aisle2writable when closed) 00:aisle2is an output, the channel2After enabling, the channel2portT31_CH2is set to output 01:aisle2For input, use the channel2portT31_CH2as input 10:aisle2For input, use the channel1portT31_CH1as input 11:aisle2input, fixed input0

Note1:whenT31CHBRegisterPROTS=11(set as protection level3) ,andCH2IOS=00(channel is in output mode),CH2OM andCH2OBEbits will not be rewritten;

Note2: in the comparator andPWMmode, channel2needs to be set as an output,CH2IOS=00.

#### catcher mode

T31CH2C:aisle2Control Register (T31CH2C)							
bit	7	6	5	4	3	2	1 0
name	CH2IFS<3:0>				CH2IM<1:0>		CH2IOS<1:0>
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0
Bit 7~4	CH2IFS<3:0>:aisle2Capture input filter time selection bits						

	0000: No filtering
	0001:2*Tosc(Toscfor system clock cycles)
	0010:4*Tosc
	0011:8*Tosc
	0100:12*Tdfck(Tdfckclock period for digital filter operation)
	0101:16*Tdfck
	0110:24*Tdfck
	0111:32*Tdfck
	1000:48*Tdfck
	1001:64*Tdfck
	1010:80*Tdfck
	1011:96*Tdfck
	1100:128*Tdfck
	1101:160*Tdfck
	1110:192*Tdfck
	1111:256*Tdfck
Bit 3~2	CH2IM<1:0>:aisle2Catcher operating mode selection bit 00:Every1input signal edge, capture once 01:Every2input signal edge, capture once 10:Every4input signal edge, capture once 11:Every8input signal edge, capture once
Bit 1~0	CH2IOS<1:0>:aisle2I/O select bits (only inCH2E=0,aisle2writable when closed) 00:aisle2is an output, the channel2After enabling, the channel2portT31_CH2is set to output 01:aisle2For input, use the channel2portT31_CH2as input 10:aisle2For input, use the channel1portT31_CH1as input 11:aisle2input, fixed input0

Note: In capture mode, channel2needs to be set as input,CH2IOS=01or10.

### 6. 3. 14. 25aisle3Control Register (T31CH3C)

comparator mode

T31CH3C:aisle3Control Register (T31CH3C)								
bit	7	6	5	4	3	2	1	0
name	CH3COCE	CH3OM<2:0>			CH3OBE	CH3OFE	CH3IOS<1:0>	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7 CH3COCE:aisle3The comparator output is cleared0Enable bit (only when the enable bit is always cleared to 0COCE=1valid)

0:prohibit

1: Enable, when the external trigger input signalT31\_ETRis high, the comparator output will be cleared

Bit 6~4 0CH3OM<2:0>:aisle3comparator andPWMworking mode selection bit

000: Comparator mode, do not change port output when matching

001: Comparator mode, comparator output on match1, the channel port output level depends onCH3Pbit

010: Comparator mode, comparator output on match0, the channel port output level depends onCH3Pbit

011: Comparator mode, the output of the comparator is inverted when it matches, and the output level of the channel port is inverted  
 100: The comparator output is forced to 0, the channel port output level depends on CH3Pbit 101: The comparator output is forced to 1, the channel port output level depends on CH3Pbit  
 110: PWMmodel1, during the increment counting process, when T31CNT < T31CH3R, the comparator output 1, otherwise output 0; During counting down, when T31CNT > T31CH3R, the comparator output 0, otherwise output 1. The channel port output level depends on the CH3Pbit  
 111: PWMmodel2, during the increment counting process, when T31CNT < T31CH3R, the comparator output 0, otherwise output 1; During counting down, when T31CNT > T31CH3R, the comparator output 1, otherwise output 0. The channel port output level depends on the CH3Pbit CH3OBE: aisle3Compare register buffer enable bit

**Bit 3**

0: prohibit, T31CH3R The value after rewriting the register will take effect immediately  
 1: Enable, T31CH3R The register is buffered, and the rewritten value will not take effect immediately until the next update event occurs. PWMmode is recommended to enable this bit to ensure PWMThe waveform is complete.

**Bit 2**

CH3OFE: aisle3Compare output acceleration enable bit (only PWM1 or PWM2 mode)  
 0: Disabled, after detecting the external trigger input signal, the channel output port still needs to wait until the count value of the counter matches the comparison register before generating the required output level according to the specific setting  
 1: Enable, after detecting the external trigger input signal, the channel output port will immediately generate the required output level according to the specific settings after the comparison match occurs, without waiting for the count value of the counter to actually match the comparison register, so that Minimizes delay time between channel output and trigger input signal  
 CH3IOS<1:0>: aisle3I/O select bits (only in CH3E=0, aisle3writable when closed)  
 00: aisle3 is an output, the channel3After enabling, the channel3portT31\_CH3is set to output 01: aisle3For input, use the channel3portT31\_CH3as input 10: aisle3For input, use the channel4portT31\_CH4as input 11: aisle3input, fixed input 0

**Bit 1~0**

Note1: when T31CHBRegisterPROTS=11(set as protection level3) , and CH3IOS=00(channel is in output mode), CH3OM and CH3OBEbits will not be rewritten;

Note2: in the comparator and PWM mode, channel3needs to be set as an output, CH3IOS=00.

**catcher mode****T31CH3C: aisle3Control Register (T31CH3C)**

<b>bit</b>	<b>7</b>	<b>6</b>	<b>5</b>	<b>4</b>	<b>3</b>	<b>2</b>	<b>1</b>	<b>0</b>
<b>name</b>	CH3IFS<3:0>				CH3IM<1:0>		CH3IOS<1:0>	
<b>R/W</b>	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
<b>POR</b>	0	0	0	0	0	0	0	0

Bit 7~4 CH3IFS<3:0>: aisle3Capture input filter time selection bits

- 0000: No filtering
- 0001: 2\*Tosc(Tosc for system clock cycles)
- 0010: 4\*Tosc
- 0011: 8\*Tosc
- 0100: 12\*Tdfck(Tdfckclock period for digital filter operation)
- 0101: 16\*Tdfck
- 0110: 24\*Tdfck
- 0111: 32\*Tdfck

	1000:48*Tdfck
	1001:64*Tdfck
	1010:80*Tdfck
	1011:96*Tdfck
	1100:128*Tdfck
	1101:160*Tdfck
	1110:192*Tdfck
	1111:256*Tdfck
Bit 3~2	CH3IM<1:0>:aisle3Catcher operating mode selection bit 00:Every1input signal edge, capture once 01:Every2input signal edge, capture once 10:Every4input signal edge, capture once 11:Every8input signal edge, capture once
Bit 1~0	CH3IOS<1:0>aisle3I/O select bits (only inCH3E=0,aisle3writable when closed) 00:aisle3is an output, the channel3After enabling, the channel3portT31_CH3is set to output 01:aisle3For input, use the channel3portT31_CH3as input 10:aisle3For input, use the channel4portT31_CH4as input 11:aisle3input, fixed input0

Note: In capture mode, channel3needs to be set as input,CH3IOS=01or10.

### 6. 3. 14. 26aisle4Control Register (T31CH4C)

comparator mode

T31CH4C:aisle4Control Register (T31CH4C)								
bit	7	6	5	4	3	2	1	0
name	CH4COCE	CH4OM<2:0>			CH4OBE	CH4OFE	CH4IOS<1:0>	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7	CH4COCE:aisle4The comparator output is cleared0Enable bit (only when the total clear0enable bitCOCE=1valid) 0:prohibit 1: Enable, when the external trigger input signalT31_ETRis high, the comparator output will be cleared
Bit 6~4	0CH4OM<2:0>:aisle4comparator andPWMworking mode selection bit 000: Comparator mode, do not change port output when matching 001: Comparator mode, comparator output on match1, the channel port output level depends onCH4Pbit 010: Comparator mode, comparator output on match0, the channel port output level depends onCH4Pbit 011: Comparator mode, the output of the comparator is inverted when it matches, and the output level of the channel port is inverted 100: The comparator output is forced to0, the channel port output level depends onCH4Pbit 101: The comparator output is forced to1, the channel port output level depends onCH4Pbit 110:PWMmodel1, during the increment counting process, whenT31CNT<T31CH4R, the comparator output1, otherwise output0;During counting down, whenT31CNT>T31CH4R, the comparator output0, otherwise output1. The channel port output level depends on theCH4Pbit 111:PWMmodel2, during the increment counting process, whenT31CNT<T31CH4R, the comparator output0,

otherwise output1; During counting down, when T31CNT>T31CH4R, the comparator output1, otherwise output0.

The channel port output level depends on the CH4Pbit CH4OBE: aisle4Compare register buffer enable bit

### Bit 3

0: prohibit, T31CH4R The value after rewriting the register will take effect immediately

1: Enable, T31CH4R The register is buffered, and the rewritten value will not take effect immediately until the next update event occurs. PWM mode is recommended to enable this bit to ensure PWM complete waveform CH4OF

### Bit 2

: aisle4Compare output acceleration enable bit (only PWM1 or PWM2 mode)

0: Disabled, after detecting the external trigger input signal, the channel output port still needs to wait until the count value of the counter matches the comparison register before generating the required output level according to the specific setting

1: Enable, after detecting the external trigger input signal, the channel output port will immediately generate the required output level according to the specific settings after the comparison match occurs, without waiting for the count value of the counter to actually match the comparison register, so that minimizes delay time between channel output and trigger input

### Bit 1~0

: aisle4I/O select bits (only in CH4E=0, aisle4 writable when closed)

00: aisle4 is an output, the channel4 After enabling, the channel4 port T31\_CH4 is set to output  
 01: aisle4 For input, use the channel4 port T31\_CH4 as input  
 10: aisle4 For input, use the channel3 port T31\_CH3 as input  
 11: aisle4 input, fixed input 0

Note1: when T31CHB register PROTS=11 (set as protection level3), and CH4IOS=00 (channel is in output mode), CH4OM and CH4OBE bits will not be rewritten;

Note2: in the comparator and PWM mode, channel4 needs to be set as an output, CH4IOS=00.

### catcher mode

T31CH4C: aisle4 Control Register (T31CH4C)								
bit	7	6	5	4	3	2	1	0
name	CH4IFS<3:0>				CH4IM<1:0>		CH4IOS<1:0>	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

### Bit 7~4

CH4IFS<3:0>: aisle4 Capture input filter time selection bits

0000: No filtering

0001: 2\*Tosc (Tosc for system clock cycles)

0010: 4\*Tosc

0011: 8\*Tosc

0100: 12\*Tdfck (Tdfck clock period for digital filter operation)

0101: 16\*Tdfck

0110: 24\*Tdfck

0111: 32\*Tdfck

1000: 48\*Tdfck

1001: 64\*Tdfck

1010: 80\*Tdfck

1011: 96\*Tdfck

1100: 128\*Tdfck

1101: 160\*Tdfck

	1110:192*Tdfck
	1111:256*Tdfck
Bit 3~2	CH4IM<1:0>:aisle4Catcher operating mode selection bit 00:Every1input signal edge, capture once 01:Every2input signal edge, capture once 10:Every4input signal edge, capture once 11:Every8input signal edge, capture once
Bit 1~0	CH4IOS<1:0>:aisle4I/O select bits (only inCH4E=0,aisle4writable when closed) 00:aisle4is an output, the channel4After enabling, the channel4portT31_CH4is set to output 01:aisle4For input, use the channel4portT31_CH4as input 10:aisle4For input, use the channel3portT31_CH3as input 11:aisle4input, fixed input0

Note: In capture mode, channel4needs to be set as input,CH4IOS=01or10.

### 6. 3. 14. 27Channel Port Control Register Low8bits (T31PINCL)

T31PINCL: Channel Port Control Register Low8bits (T31PINCL)								
bit	7	6	5	4	3	2	1	0
name	CH2NP	CH2NE	CH2P	CH2E	CH1NP	CH1NE	CH1P	CH1E
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0
<b>aisle2or1When configured as output:</b>								
Bit 7	CH2NP:aisle2Complementary Output Polarity Control Bits 0:T31_CH2NWhen the port outputs a valid state, it does not invert, and when it outputs an invalid state, the port is 0 1:T31_CH2NWhen the port outputs a valid state, it inverts, and when it outputs an invalid state, the port is1							
Bit 6	CH2NE:aisle2Complementary output enable bit 0:prohibit,T31_CH2Nport output closed 1 :Enable,T31_CH2Nport output open CH2P							
Bit 5	:aisle2Output Polarity Control Bits 0:T31_CH2When the port outputs a valid state, it does not invert, and when it outputs an invalid state, the port is0 1:T31_CH2When the port outputs a valid state, it inverts, and when it outputs an invalid state, the port is1							
Bit 4	CH2E:aisle2output enable bit 0:prohibit,T31_CH2port output closed 1:Enable, T31_CH2port output open CH1NP:aisle1							
Bit 3	Complementary Output Polarity Control Bits 0:T31_CH1NWhen the port outputs a valid state, it does not invert, and when it outputs an invalid state, the port is 0 1:T31_CH1NWhen the port outputs a valid state, it inverts, and when it outputs an invalid state, the port is1							
Bit 2	CH1NE:aisle1Complementary output enable bit 0:prohibit,T31_CH1Nport output closed 1 :Enable,T31_CH1Nport output open CH1P							
Bit 1	:aisle1Output Polarity Control Bits 0:T31_CH1When the port outputs a valid state, it does not invert, and when it outputs an invalid state, the port is0 1: T31_CH1When the port outputs a valid state, it inverts, and when it outputs an invalid state, the port is1							

Bit 0 CH1E:aisle1output enable bit  
 0:prohibit,T31\_CH1port output closed  
 1:Enable,T31\_CH1port output open

**aisle2or1When configured as input:**

Bit 7 CH2NP:aisle2Input Polarity Control Bits  
 This bit needs to beCH2Pused together to control the channel2polarity of the input signal, seeCH2Pbit description. CH2NE  
 Bit 6 :aisle2Complementary output enable bit This bit is invalid in channel input state

Bit 5 CH2P:aisle2Input Polarity Control Bits  
 Depend onCH2NP/CH2PCombined Control Input Signal Polarity:  
 00:aisle2The input is not inverted, and the rising edge or high level of the input signal is active  
 01:aisle2Input inversion, the falling edge or low level of the input signal is active 10: reserved  
 for unused  
 11: Daodao2The input is not inverted, and both edges (rising/falling edges) or high levels of the input signal are active (this configuration is prohibited in encoder mode)

Bit 4 CH2E:aisle2input enable bit  
 0: Forbidden, cannot capture  
 1: enable, can capture

Bit 3 CH1NP:aisle1Input Polarity Control Bits  
 This bit needs to beCH1Pused together to control the channel1polarity of the input signal, seeCH1Pbit description. CH1NE  
 Bit 2 :aisle1Complementary output enable bit This bit is invalid in channel input state

Bit 1 CH1P:aisle1Input Polarity Control Bits  
 Depend onCH1NP/CH1PCombined Control Input Signal Polarity:  
 00:aisle1The input is not inverted, and the rising edge or high level of the input signal is active  
 01:aisle1Input inversion, the falling edge or low level of the input signal is active 10: reserved  
 for unused  
 11: Daodao1The input is not inverted, and both edges (rising/falling edges) or high levels of the input signal are active (this configuration is prohibited in encoder mode)

Bit 0 CH1E:aisle1input enable bit  
 0: Forbidden, cannot capture  
 1: enable, can capture

Note1:whenT31CHBRegisterPROTS=10or11(set as protection level2or3), and the channel2and1In output mode,CH2NP,  
 CH2P,CH1NP,CH1Pbits will not be rewritten;

Note2:aisle2input depends onT31CH2CregisterCH2IOSbit setting, whenCH2IOS=01when the channel2port input, when  
 CH2IOS=10when the channel1port input;

Note3:aisle1input depends onT31CH1CregisterCH1IOSbit setting, whenCH1IOS=01when the channel1port input, when  
 CH1IOS=10when the channel2port input;

Note4:aisle2output portT31\_CH2When closed and opened, its port level depends onCHOE,NOFFS,ROFFS,ONS2,CH2Eand  
 CH2NEcontrol;

Note5:aisle2Complementary output portT31\_CH2NWhen closed and opened, its port level depends onCHOE,NOFFS,ROFFS,ONS2N,  
 CH2EandCH2NEcontrol;

Note6:aisle1output portT31\_CH1When closed and opened, its port level depends onCHOE,NOFFS,ROFFS,ONS1,CH1Eand CH1NEcontrol;

Note7:aisle1Complementary output portT31\_CH1NWhen closed and opened, its port level depends onCHOE,NOFFS,ROFFS,ONS1N, CH1EandCH1NEcontrol;

Note 8:CHnPandCHnPControl of channel port output polarity is only available on theCHOE=1valid.

### 6. 3. 14. 28Channel Port Control Register High8bits (T31PINCH)

T31PINCH: Channel Port Control Register High8bits (T31PINCH)								
bit	7	6	5	4	3	2	1	0
name	CH4NP	—	CH4P	CH4E	CH3NP	CH3NE	CH3P	CH3E
R/W	R/W	—	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

#### aisle4or3When configured as output:

- Bit 7      CH4NP:aisle4Input polarity control bit This bit  
is invalid in the channel output state
- Bit 6      Not used, the software needs to be fixed as0
- Bit 5      CH4P:aisle4Output Polarity Control Bits  
0:T31\_CH4When the port outputs a valid state, it does not invert, and when it outputs an invalid state, the port  
is0 1:T31\_CH4When the port outputs a valid state, it inverts, and when it outputs an invalid state, the port is1
- Bit 4      CH4E:aisle4output enable bit  
0:prohibit,T31\_CH4port output closed 1:Enable,  
T31\_CH4port output open CH3NP:aisle3
- Bit 3      Complementary Output Polarity Control Bits  
0:T31\_CH3NWhen the port outputs a valid state, it does not invert, and when it outputs an invalid state, the port is  
0 1:T31\_CH3NWhen the port outputs a valid state, it inverts, and when it outputs an invalid state, the port is1
- Bit 2      CH3NE:aisle3Complementary output enable bit  
0:prohibit,T31\_CH3Nport output closed 1  
:Enable,T31\_CH3Nport output open CH3P
- Bit 1      :aisle3Output Polarity Control Bits  
0:T31\_CH3When the port outputs a valid state, it does not invert, and when it outputs an invalid state, the port  
is0 1:T31\_CH3When the port outputs a valid state, it inverts, and when it outputs an invalid state, the port is1
- Bit 0      CH3E:aisle3output enable bit  
0:prohibit,T31\_CH3port output closed  
1:Enable,T31\_CH3port output open

#### aisle4or3When configured as input:

- Bit 7      CH4NP:aisle4Input Polarity Control Bits  
This bit needs to beCH4Pused together to control the channel4polarity of the input signal, seeCH4Pbit description. Not  
used, the software needs to be fixed as0
- Bit 6      CH4P:aisle4Input Polarity Control Bits  
Depend onCH4NP/CH4PCombined Control Input Signal Polarity:  
00:aisle4The input is not inverted, and the rising edge or high level of the input signal is active  
01:aisle4Input inversion, the falling edge or low level of the input signal is active

	10: reserved for unused
	11: Daodao4The input is not inverted, and both edges (rising/falling edges) or high levels of the input signal are active (this configuration is prohibited in encoder mode)
Bit 4	CH4E:aisle4input enable bit 0: Forbidden, cannot capture 1: enable, can capture
Bit 3	CH3NP:aisle3Input Polarity Control Bits This bit needs to beCH3Pused together to control the channel3polarity of the input signal, seeCH3Pbit description. CH3NE
Bit 2	:aisle3Complementary output enable bit This bit is invalid in channel input state
Bit 1	CH3P:aisle3Input Polarity Control Bits Depend onCH3NP/CH3PCombined Control Input Signal Polarity: 00:aisle3The input is not inverted, and the rising edge or high level of the input signal is active 01:aisle3Input inversion, the falling edge or low level of the input signal is active 10: reserved for unused 11: Daodao3The input is not inverted, and both edges (rising/falling edges) or high levels of the input signal are active (this configuration is prohibited in encoder mode)
Bit 0	CH3E:aisle3input enable bit 0: Forbidden, cannot capture 1: enable, can capture

Note1:whenT31CHBKregisterPROTS=10or11(set as protection level2or3), and the channel4and3In output mode,CH4P, CH3NP,CH3Pbits will not be rewritten;

Note2:aisle4input depends onT31CH4CregisterCH4IOSbit setting, whenCH4IOS=01when the channel4port input, when CH4IOS=10when the channel3port input;

Note3:aisle3input depends onT31CH3CregisterCH3IOSbit setting, whenCH3IOS=01when the channel3port input, when CH3IOS=10when the channel4port input;

Note4:aisle4output portT31\_CH4When closed and opened, its port level depends onCHOE,NOFFS,ROFFS,ONS4andCH4E control;

Note5:aisle3output portT31\_CH3When closed and opened, its port level depends onCHOE,NOFFS,ROFFS,ONS3,CH3Eand CH3NEcontrol;

Note6:aisle3Complementary output portT31\_CH3NWhen closed and opened, its port level depends onCHOE,NOFFS,ROFFS,ONS3N, CH3EandCH3NEcontrol;

Note 7:CHnPandCHnPControl of channel port output polarity is only available on theCHOE=1valid.

### 6. 3. 14. 29Channel Output Shutdown Control Register (T31CHBK)

T31CHBK: Channel Output Shutdown Control Register (T31CHBK)								
bit	7	6	5	4	3	2	1	0
name	CHOE	AROE	BKPS	BKE	ROFFS	NOFFS	PROTS<1:0>	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7 CHOE: channel output status enable general control bit

0: The output is off, and the channel port is in idle mode. whenNOFFS=1When, each channel output and complementary output are

It is forced to idle state, if the dead zone delay is set, it will enter the idle state after the dead zone delay; when NOFFS=0, each channel output and complementary output are disabled.

1: Output is enabled, the channel port is in the running mode, and the output of each channel and the complementary output are valid or invalid, and the specific status depends on CH<sub>n</sub> and ROFFS bit. This bit is cleared asynchronously by hardware as soon as the shutdown input is asserted. This bit can be set by software or set automatically after the shutdown event is removed, Depend on AROE bit to control. AROE: Channel auto recovery enable bit

#### Bit 6

0: Prohibited, software reset is required CHOEBit

1: Enabled, after the shutdown event is cancelled, when the next update event occurs, CHOEBit auto-recovery 1 BKPS

#### Bit 5

: Shutdown input polarity selection bit

0: Low effective

1: High effective

#### Bit 4

BKE: shutdown enable bit

0: prohibit

1: Enable

#### Bit 3

ROFFS: Inactive state select bit in run mode (only in CHOE=1 valid)

0: channel output enable bit CH<sub>n</sub> or CH<sub>n</sub> for 0, the corresponding channel port output is disabled

1: channel output enable bit CH<sub>n</sub> and CH<sub>n</sub> only one for 0, the enable bit is 0. The corresponding channel port outputs the invalid state level, when both bits are 0, the channel port output is prohibited

#### Bit 2

NOFFS: Idle state select bit in idle mode (only in CHOE=0 valid)

0: all channel ports T31\_CHn/T31\_CHnN output disable

1: channel output enable bit CH<sub>n</sub> and CH<sub>n</sub> only one for 0, then the channel port T31\_CHn and T31\_CHnN both output the idle state level, when both bits are 0, the channel port output is disabled PROT<sub>S</sub><1:0>: Write

#### Bit 1~0

protection level selection bit

00: Close the write protection, the software can normally write the register bit

01: Write protection level 1: T31\_DLYT register, T31\_C1H register ONS and ONS and

T31\_CHBK register BKE, BKPS and AROEn longer writable

10: Write protection level 2: Except for the write-protected level 1, in addition to the protected register bits, the following register bits are no longer writable: T31\_PIN register CH<sub>n</sub>P and CH<sub>n</sub>N P bit (only the polarity control bit of the output channel is write-protected), as well as ROFFS and NOFFS bit

11: Write protection level 3: Except for the write-protected level 2, in addition to the protected register bits, the following register bits are no longer writable: T31\_CHn register QUR and OBE bit (write protection is only provided for the operating mode of the output channel and the compare register buffer control bit)

Note: After chip reset, T31\_CHBK register PROT<sub>S</sub> bit can only be written once by software, so the write-protected register bit must be set before writing PROT<sub>S</sub> bit, set the corresponding write protection level.

#### 6. 3. 14. 30 Software Trigger Event Register (T31EVG)

T31EVG: Software trigger event register (T31EVG)								
bit	7	6	5	4	3	2	1	0
name	BKT	TRGT	CHUT	wxya	CH3T	CH2T	CH1T	UPT
R/W	W	W	W	W	W	W	W	W
POR	0	0	0	0	0	0	0	0

Bit 7	BKT: Shutdown event trigger bit 0:invalid 1: The software triggers a shutdown event and generates a shutdown interruptBKIF, this bit is automatically cleared by hardware0
Bit 6	hardware0 TRGT: Trigger event generation bit 0:invalid 1: The software generates a trigger event and generates a trigger interruptTRGIF, this bit is automatically cleared by hardware0
Bit 5	CHUT: Complementary channel update event trigger bit (only valid for complementary output channels) 0:invalid 1: The software triggers a complementary channel update event, generating a complementary channel update interruptCHUIF, this bit is automatically cleared by hardware0
Bit 4	wxya:aisle4Capture/Compare Match Event Trigger Bit 0:invalid 1: The software triggers a channel4A capture/compare match event that generates a multi-function interruptMIF4 or capture overflow interruptOVIF4, this bit is automatically cleared by hardware0 CH3T:aisle3Capture/Compare Match Event Trigger Bit
Bit 3	0:invalid 1: The software triggers a channel3A capture/compare match event that generates a multi-function interruptMIF3 or capture overflow interruptOVIF3, this bit is automatically cleared by hardware0 CH2T:aisle2Capture/Compare Match Event Trigger Bit
Bit 2	0:invalid 1: The software triggers a channel2A capture/compare match event that generates a multi-function interruptMIF2 or capture overflow interruptOVIF2, this bit is automatically cleared by hardware0 CH1T:aisle1Capture/Compare Match Event Trigger Bit
Bit 1	0:invalid 1: The software triggers a channel1A capture/compare match event that generates a multi-function interruptMIF1 or capture overflow interruptOVIF1, this bit is automatically cleared by hardware0 UPT: update event trigger bit
Bit 0	0:invalid 1: The software triggers a re-event, generating an update interruptUPIF, this bit is automatically cleared by hardware0. After the update event is triggered, the effect is the same as the trigger event in reset mode, the counter and prescaler are re-initialized, and the prescaler counter will be cleared (the prescaler ratio is not affected), the counter will also be cleared when counting up. When counting down, the counter reloads the value of the count register. At the same time if T31C0Lregister UES=0, also makes all reloadable registers (T31CNTLD, T31CHnR) are overloaded and initialized.

**6. 3. 14. 31Interrupt Enable Control Register Low8bits (T31IEL)**

T31IEL: Interrupt Enable Control Register Low8bits (T31IEL)								
bit	7	6	5	4	3	2	1	0
name	BKIE	TRGIE	CHUIE	MIE4	MIE3	MIE2	MIE1	UPIE
R/W	W	W	W	W	W	W	W	W
POR	0	0	0	0	0	0	0	0

Bit 7 BKIE: Shutdown interrupt enable bit

0:prohibit

1:Enable

Bit 6	TRGIE: trigger interrupt enable bit 0:prohibit 1:Enable
Bit 5	CHUIE: Complementary channel update interrupt enable bit 0:prohibit 1:Enable
Bit 4	MIE4:aisle4Multi-function interrupt enable bit 0:prohibit 1:Enable
Bit 3	MIE3:aisle3Multi-function interrupt enable bit 0:prohibit 1:Enable
Bit 2	MIE2:aisle2Multi-function interrupt enable bit 0:prohibit 1:Enable
Bit 1	MIE1:aisle1Multi-function interrupt enable bit 0:prohibit 1:Enable
Bit 0	UPIE: update interrupt enable bit 0:prohibit 1:Enable

Note: channelnoWhen a capture or compare match occurs, the multi-function interrupt flag of the channel will be setMIF, if the channel's multi-function interrupt enable bit MIEn=1, the corresponding multi-function interrupt request will be generated.

#### 6. 3. 14. 32Interrupt Enable Control Register High8bits (T31IEH)

T31IEH: Interrupt Enable Control Register High8bits (T31IEH)								
bit	7	6	5	4	3	2	1	0
name	—	—	—	OVIE4	OVIE3	OVIE2	OVIE1	—
R/W	—	—	—	W	W	W	W	—
POR	0	0	0	0	0	0	0	0

Bit 7~5	Not used, the software needs to be fixed as0
Bit 4	OVIE4:aisle4Capture overflow interrupt enable bit 0:prohibit 1:Enable
Bit 3	OVIE3:aisle3Capture overflow interrupt enable bit 0:prohibit 1:Enable
Bit 2	OVIE2:aisle2Capture overflow interrupt enable bit 0:prohibit 1:Enable
Bit 1	OVIE1:aisle1Capture overflow interrupt enable bit 0:prohibit 1:Enable

Bit 0 Not used, the software needs to be fixed as0

Note: When both the interrupt enable bit and the interrupt disable bit of an interrupt are written 1, the bit written later is valid, for example, the bit written firstOVID1=1, then writeOVID1=1hour, actually forbidden channel1Capture overflow interrupt, otherwise enable the channel1Capture overflow interrupt.

### 6. 3. 14. 33Interrupt Disable Register Low8bits (T31IDL)

T31IDL: Interrupt Disable Register Low8bits (T31IDL)								
bit	7	6	5	4	3	2	1	0
name	BKID	TRGID	CHUID	MID4	MID3	MID2	MID1	UPID
R/W	W	W	W	W	W	W	W	W
POR	0	0	0	0	0	0	0	0

Bit 7 BKID: shutdown interrupt disable bit

0:invalid

1:prohibit

Bit 6 TRGID: trigger interrupt disable bit

0:invalid

1:prohibit

Bit 5 CHUID: Complementary channel update interrupt disable bit

0:invalid

1:prohibit

Bit 4 MID4:aisle4Multi-function interrupt disable bit

0:invalid

1:prohibit

Bit 3 MID3:aisle3Multi-function interrupt disable bit

0:invalid

1:prohibit

Bit 2 MID2:aisle2Multi-function interrupt disable bit

0:invalid

1:prohibit

Bit 1 MID1:aisle1Multi-function interrupt disable bit

0:invalid

1:prohibit

Bit 0 UPID: update interrupt disable bit

0:invalid

1:prohibit

### 6. 3. 14. 34Interrupt Disable Register High8bits (T31IDH)

T31IDH: Interrupt Disable Register High8bits (T31IDH)								
bit	7	6	5	4	3	2	1	0
name	—	—	—	OVID4	OVID3	OVID2	OVID1	—
R/W	—	—	—	W	W	W	W	—
POR	0	0	0	0	0	0	0	0

Bit 7~5	Not used, the software needs to be fixed as0
Bit 4	OVID4:aisle4Capture overflow interrupt disable bit 0:invalid 1:prohibit
Bit 3	OVID3:aisle3Capture overflow interrupt disable bit 0:invalid 1:prohibit
Bit 2	OVID2:aisle2Capture overflow interrupt disable bit 0:invalid 1:prohibit
Bit 1	OVID1:aisle1Capture overflow interrupt disable bit 0:invalid 1:prohibit
Bit 0	Not used, the software needs to be fixed as0

Note: When both the interrupt enable bit and the interrupt disable bit of an interrupt are written 1, the bit written later is valid, for example, the bit written firstOVID1=1, then writeOVID1=1hour, actually forbidden channel1Capture overflow interrupt, otherwise enable the channel1Capture overflow interrupt.

### 6. 3. 14. 35Interrupt Enable Status Register Low8bits (T31IVSL)

T31IVSL: Interrupt Enable Status Register Low8bits (T31IVSL)								
bit	7	6	5	4	3	2	1	0
name	BKIS	TRGIS	CHUIS	MIS4	MIS3	MIS2	MIS1	UPIS
R/W	R	R	R	R	R	R	R	R
POR	0	0	0	0	0	0	0	0

Bit 7	BKIS: Shutdown interrupt enable status bit 0:prohibit 1:Enable
Bit 6	TRGIS: Trigger interrupt enable status bit 0:prohibit 1:Enable
Bit 5	CHUIS: Complementary channel update interrupt enable status bit 0:prohibit 1:Enable
Bit 4	MIS4:aisle4Capture interrupt/compare match interrupt enable status bit 0:prohibit 1:Enable
Bit 3	MIS3:aisle3Capture interrupt/compare match interrupt enable status bit 0:prohibit 1:Enable
Bit 2	MIS2:aisle2Capture interrupt/compare match interrupt enable status bit 0:prohibit 1:Enable
Bit 1	MIS1:aisle1Capture interrupt/compare match interrupt enable status bit

	0:prohibit
	1:Enable
Bit 0	UPIS: Update interrupt enable status bit
	0:prohibit
	1:Enable

### 6. 3. 14. 36Interrupt Enable Status Register High8bits (T31IVSH)

T31IVSH: Interrupt Enable Status Register High8bits (T31IVSH)								
bit	7	6	5	4	3	2	1	0
name	—	—	—	OVIS4	OVIS3	OVIS2	OVIS1	—
R/W	—	—	—	R	R	R	R	—
POR	0	0	0	0	0	0	0	0

Bit 7~5	Not used, the software needs to be fixed as0
Bit 4	OVIS4:aisle4Capture overflow interrupt enable status bit
	0:prohibit
	1:Enable
Bit 3	OVIS3:aisle3Capture overflow interrupt enable status bit
	0:prohibit
	1:Enable
Bit 2	OVIS2:aisle2Capture overflow interrupt enable status bit
	0:prohibit
	1:Enable
Bit 1	OVIS1:aisle1Capture overflow interrupt enable status bit
	0:prohibit
	1:Enable
Bit 0	Not used, the software needs to be fixed as0

NOTE: By setting the registerT31IEandT31ID, to configure the registerT31IVSThe interrupt active status bit. When the registerT31IEandT31IDpair in  
When the corresponding bits are all 1, the bit written later is valid, for example, the bit written firstOVIEn=1, then writeOVID1=1, the channel is actually prohibited1capture overflow  
interrupt, the correspondingOVISe=0, and vice versaOVISe=1.

### 6. 3. 14. 37Interrupt Flag Register Low8bits (T31IFL)

T31IFL: Interrupt flag register low8bits (T31IFL)								
bit	7	6	5	4	3	2	1	0
Name	BKIF	TRGIF	CHUIF	MIF4	MIF3	MIF2	MIF1	UPIF
R/W	R	R	R	R	R	R	R	R
POR	0	0	0	0	0	0	0	0

Bit 7	BKIF: shutdown interrupt flag bit
	0: No shutdown event occurred
	1: A shutdown event occurs, after the shutdown event is canceled, writeT31ICRLThe corresponding interrupt bit of the register is cleared0
Bit 6	TRGIF: External trigger interrupt flag bit

	0: No trigger event occurred
	1: A trigger event occurs, writeT31ICRLThe corresponding interrupt bit of the register is cleared0 CHUIF:
Bit 5	Complementary channel update interrupt flag bit 0: Complementary channel update event did not occur 1: Complementary channel update event occurs, writeT31ICRLThe corresponding interrupt bit of the register is cleared0 MIF4:aisle4multi-function interrupt flag
Bit 4	0: No capture event/compare match event 1: A capture event/compare match event occurs, writeT31ICRLThe corresponding interrupt bit of the register is cleared0, when a capture occurs, it can also be read by readingT31CH4Rregister to clear0 MIF3:aisle3multi-function interrupt flag
Bit 3	0: No capture event/compare match event 1: A capture event/compare match event occurs, writeT31ICRLThe corresponding interrupt bit of the register is cleared0, when a capture occurs, it can also be read by readingT31CH3Rregister to clear0 MIF2:aisle2multi-function interrupt flag
Bit 2	0: No capture event/compare match event 1: A capture event/compare match event occurs, writeT31ICRLThe corresponding interrupt bit of the register is cleared0, when a capture occurs, it can also be read by readingT31CH2Rregister to clear0 MIF1:aisle1multi-function interrupt flag
Bit 1	0: No capture event/compare match event 1: A capture event/compare match event occurs, writeT31ICRLThe corresponding interrupt bit of the register is cleared0, when a capture occurs, it can also be read by readingT31CH1Rregister to clear0 UPIF: update interrupt flag bit
Bit 0	0: no update event occurred 1: update event occurs, writeT31ICRLThe corresponding interrupt bit of the register is cleared0

Note: channelnoWhen a capture or compare match occurs, the multi-function interrupt flag of the channel will be setMIF, if the channel's multi-function interrupt enable bit

MIEn=1, the corresponding multi-function interrupt request will be generated.

### 6. 3. 14. 38Interrupt flag register high8bits (T31IFH)

T31IFH: Interrupt Flag Register High8bits (T31IFH)								
bit	7	6	5	4	3	2	1	0
name	—	—	—	OVIF4	OVIF3	OVIF2	OVIF1	—
R/W	—	—	—	R	R	R	R	—
POR	0	0	0	0	0	0	0	0

Bit 7~5	Not used, the software needs to be fixed as0
Bit 4	OVIF4:aisle4capture overflow interrupt flag 0: no capture overflow occurs 1: Capture overflow occurs, writeT31ICRLThe corresponding interrupt bit of the register is cleared0 OVIF3:aisle3capture overflow interrupt flag
Bit 3	0: no capture overflow occurs 1: Capture overflow occurs, writeT31ICRLThe corresponding interrupt bit of the register is cleared0 OVIF2:aisle2capture overflow interrupt flag
Bit 2	cleared0 OVIF1:aisle1capture overflow interrupt flag

	0: no capture overflow occurs
	1: Capture overflow occurs, writeT31ICRLThe corresponding interrupt bit of the register is cleared0 OVIF1:aisle1capture overflow interrupt flag
Bit 1	0: no capture overflow occurs
	1: Capture overflow occurs, writeT31ICRLThe corresponding interrupt bit of the register is cleared0 Not
Bit 0	used, the software needs to be fixed as0

#### 6. 3. 14. 39Interrupt Request Status Register Low8bits (T31IFML)

T31IFML: Interrupt Request Status Register Low8bits (T31IFML)								
bit	7	6	5	4	3	2	1	0
name	BKIM	TRGIM	CHUIM	MIM4	MIM3	MIM2	MIM1	UPIM
R/W	R	R	R	R	R	R	R	R
POR	0	0	0	0	0	0	0	0

Bit 7	BKIM: Shutdown interrupt request status bit 0: No interrupt request 1: There is an interrupt request,
Bit 6	TRGIM: Trigger interrupt request status bit 0: No interrupt request 1: There is an interrupt request
Bit 5	CHUIM: Complementary channel update interrupt request status bit 0: No interrupt request 1: There is an interrupt request
Bit 4	MIM4:aisle4Multi-function interrupt request status bit 0: No interrupt request 1: There is an interrupt request
Bit 3	MIM3:aisle3Multi-function interrupt request status bit 0: No interrupt request 1: There is an interrupt request
Bit 2	MIM2:aisle2Multi-function interrupt request status bit 0: No interrupt request 1: There is an interrupt request
Bit 1	MIM1:aisle1Multi-function interrupt request status bit 0: No interrupt request 1: There is an interrupt request
Bit 0	UPIM: Update interrupt request status bit 0: No interrupt request 1: There is an interrupt request

**6. 3. 14. 40Interrupt Request Status Register High8bits (T31IFMH)**

T31IFMH: Interrupt Request Status Register High8bits (T31IFMH)								
bit	7	6	5	4	3	2	1	0
name	—	—	—	OVIM4	OVIM3	OVIM2	OVIM1	—
R/W	—	—	—	R	R	R	R	—
POR	0	0	0	0	0	0	0	0

Bit 7~5 Not used, the software needs to be fixed as0

Bit 4 OVIM4:aisle4Capture overflow interrupt request status bit

0: No interrupt request

1: There is an interrupt request

Bit 3 OVIM3:aisle3Capture overflow interrupt request status bit

0: No interrupt request

1: There is an interrupt request

Bit 2 OVIM2:aisle2Capture overflow interrupt request status bit

0: No interrupt request

1: There is an interrupt request

Bit 1 OVIM1:aisle1Capture overflow interrupt request status bit

0: No interrupt request

1: There is an interrupt request

Bit 0 Not used, the software needs to be fixed as0

Note: RegisterT31IFMThe status bits in the0Indicates that there is no interrupt request. At this time, at least one of the corresponding interrupt flag bit and interrupt enable bit is0;

for1Indicates that there is an interrupt request, and the corresponding interrupt flag bit and interrupt enable bit are both1, and will simultaneously setT31total interrupt flagT31IF.

**6. 3. 14. 41interrupt clear0register low8bits (T31ICRL)**

T31ICRL: interrupt clear0register low8bits (T31ICRL)								
bit	7	6	5	4	3	2	1	0
name	BKIC	TRGIC	CHUIC	MIC4	MIC3	MIC2	MIC1	UPIC
R/W	W	W	W	W	W	W	W	W
POR	0	0	0	0	0	0	0	0

Bit 7 BKIC: shutdown interrupt clear0bit

0:invalid

1: Interrupt flag clear0

Bit 6 TRGIC: trigger interrupt clear0bit

0:invalid

1: Interrupt flag clear0

Bit 5 CHUIC: Complementary channel update interrupt clear0bit

0:invalid

1: Interrupt flag clear0

Bit 4 MIC4:aisle4Multifunction Interrupt Clear0bit

0:invalid

1: Interrupt flag clear0

Bit 3	MIC3:aisle3Multifunction Interrupt Clear0bit 0:invalid 1: Interrupt flag clear0
Bit 2	MIC2:aisle2Multifunction Interrupt Clear0bit 0:invalid 1: Interrupt flag clear0
Bit 1	MIC1:aisle1Multifunction Interrupt Clear0bit 0:invalid 1: Interrupt flag clear0
Bit 0	UPIC: update interrupt clear0bit 0:invalid 1: Interrupt flag clear0

**6. 3. 14. 42interrupt clear0register high8bits (T31ICRH)**

T31ICRH: interrupt clear0register high8bits (T31ICRH)								
bit	7	6	5	4	3	2	1	0
name	—	—	—	OVIC4	OVIC3	OVIC2	OVIC1	—
R/W	—	—	—	W	W	W	W	—
POR	0	0	0	0	0	0	0	0

Bit 7~5	Not used, the software needs to be fixed as0
Bit 4	OVIC4:aisle4Capture Overflow Interrupt Clear0bit 0:invalid 1: Interrupt flag clear0
Bit 3	OVIC3:aisle3Capture Overflow Interrupt Clear0bit 0:invalid 1: Interrupt flag clear0
Bit 2	OVIC2:aisle2Capture Overflow Interrupt Clear0bit 0:invalid 1: Interrupt flag clear0
Bit 1	OVIC1:aisle1Capture Overflow Interrupt Clear0bit 0:invalid 1: Interrupt flag clear0
Bit 0	Not used, the software needs to be fixed as0

## 6.4Universal Asynchronous Receiver/Transmitter (UART)

### 6.4.1 overview

This chip supports a set of full-duplex universal asynchronous receiver transmitterUART, it uses serial transceiver to transmit data with external equipment, and can communicate with other external equipment with asynchronous receiver and transmitter.

#### -Support two working modes

- asynchronous receiver
  - asynchronous transmitter
- Transmission baud rate configuration
- high speed mode
  - low speed mode
  - support8/9Bit transmission data format, agreeing that data is received/sent from the lowest bit
  - Support full duplex mode
  - UARTThe following functional components are supported

- Receive Data Register (RXB)
- Receive Control Register (RXC)
- Transmit Data Register (TXB)
- Transmit Control Register (TXC)
- Transmit Shift Register (TXR) (No actual physical address, not readable or writable)
- Baud Rate Register (BRR)

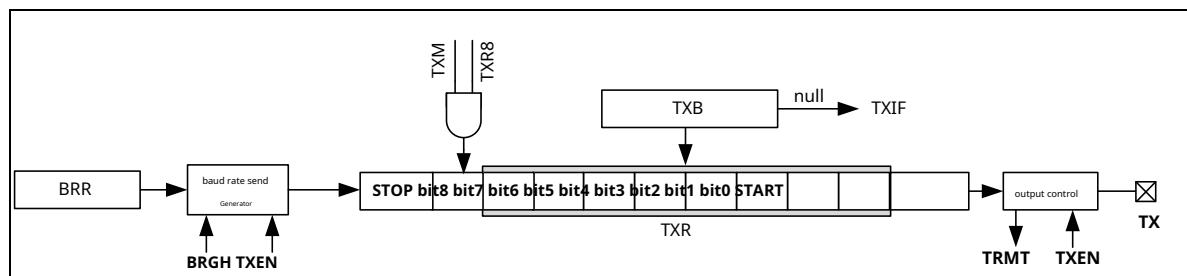
#### -break and pause

- Supports receive interrupt flag (RXIF, read only)
- Support for sending interrupt flags (TXIF, read only)
- Support interrupt handling
- existIDLEmode, suspend receiving and sending

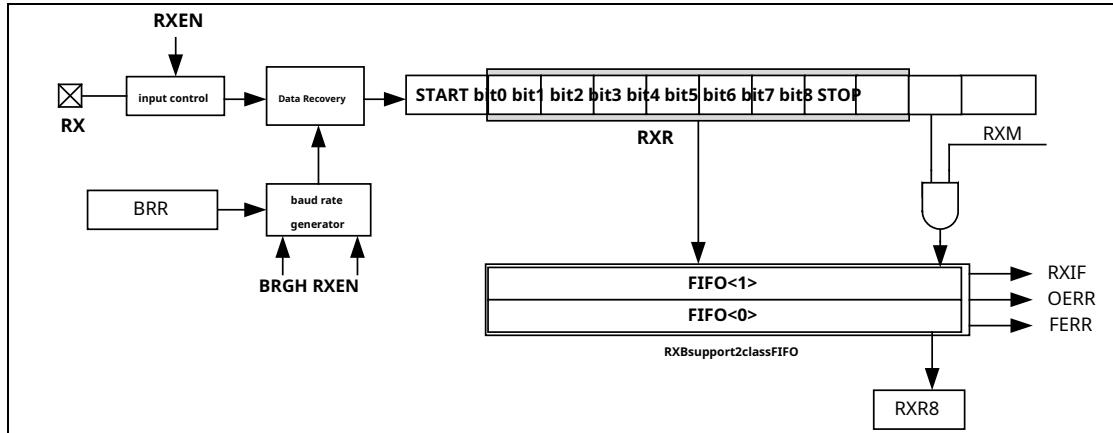
#### -compatibleRS-232/RS-442/RS-485communication interface

Note: In this sectionTX/RXThe port represents the chip pin diagramUART\_TX/UART\_RXport.

### 6.4.2 Internal structure map



picture6-32 UARTsSchematic diagram of the sender



picture6-33 UARTs Receiver Schematic

### 6.4.3

#### Baud rate configuration

UARTIt has a baud rate generator by itself, through which the data transmission rate can be set. The baud rate is determined by an independent internal8bit counter generated by theBRRRegister andTXCregisterBRGHto control.BRGHWhether to determine whether the baud rate generator is in high-speed mode or low-speed mode determines the selection of the calculation formula.

BRGH	baud rate	Calculation formula
0	low speed mode	$Fosc/(64x(BRR<7:0>+1))$
1	high speed mode	$Fosc/(16x(BRR<7:0>+1))$

surface6-5 UARTs Baud Rate Configuration Table

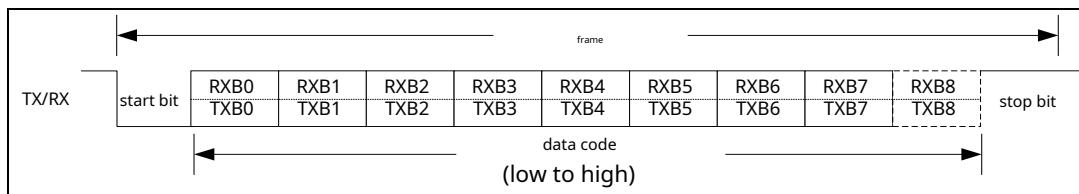
### 6.4.4

#### Transmission data format

UARTThere are two options for the transfer data format,8bit or9bit. take over9bit data when readingRXCin the registerRXR8bits can be received by the first9bit data. send9bit data, available throughTXCin the registerTXR8 bit sets the number of bits to be sent9bit data.

RXM	TXM	Transmission data format
0	0	8bit
1	1	9bit

surface6-6 UARTs Data format configuration table



picture6-34 UARTs Schematic diagram of data format

### 6.4.5

#### asynchronous transmitter

When an asynchronous transmitter sends data, the start bit (START) and end bits (STOP) is generated inside the chip, the user only needs to enable the asynchronous transmitter and write the data to be sent intoTXBandTXR8Within, asynchronous sending can be realized,

The asynchronous transmitter can also realize the continuous transmission of data.

When sending data, the asynchronous transmitter must be enabled first, and then write the send data registerTXB, otherwise the sent data written is invalid; if it is9bit data format, you need to write the first bit after enabling the asynchronous transmitter9bit dataTXR8, then writeTXB, otherwise the first9Bit data may be sent incorrectly.

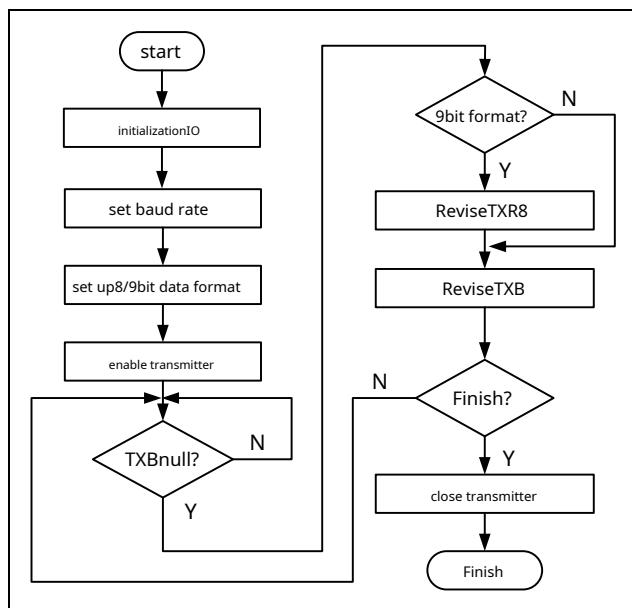
support1Stage Transmit Shift RegisterTXR(This register is not user accessible) , when the data is sent, the hardware circuit will send send data registerTXBandTXR8The data in is first transferred to the transmit shift register, and then through the transmit portTXSend data. transmit shift registerTXRWhen empty, the empty flag will be setTRMT, when writing the sending data again, it will be clearedTRMT. Disable asynchronous send (TXEN=0), the empty flag will also be setTRMT.

After the current data is sent, the interrupt flag bitTXIFwas placed "1" . If the interrupt enable bit is sentTXIEand globally enable bitGIEboth for "1", then toCPUissueUARTSend an interrupt request. interrupt flagTXIFIt is read-only and cannot be cleared by software.TRMT=0When writing to the transmit data registerTXB, or disable asynchronous send (TXEN=0), can be clearedTXIF. Enable asynchronous transmission (TXEN=1), before the first data is sent, the send interrupt flag will also be set TXIF, the first write to the transmit data registerTXBAfter, the data is automatically transferred to the shift registerTXR, and clearTRMT, but not clearedTXIF, the second writeTXBwill be cleared whenTXIF.

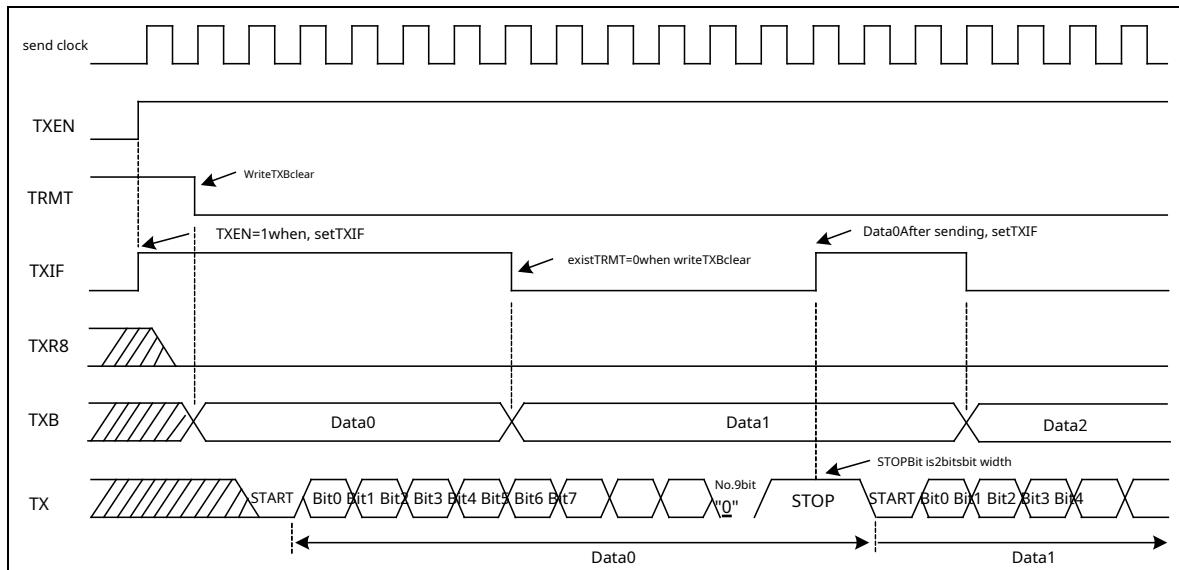
When sending data, the user can query the empty flag bit of the sending shift registerTRMTor send interrupt flag bit TXIF, to determine whether the data has been sent, whenTRMT=1orTXIF=1, you can write the next data to be sent; you can also write the next data to be sent in the sending interrupt service program to realize the continuous sending of data.

becauseUARTTransmitter send portTx and I/O Port multiplexing, in useUARTBefore sending the port, you need to set the multiplexedI/OThe port is in the output state and outputs a high level.

The operation flow chart is as follows:



picture6-35 UARTsAsynchronous Transmitter Operation Flowchart



picture6-36 UARTsTransmitter Send Data Timing Diagram (9Bit Data Format, No.9The bit data is "0")

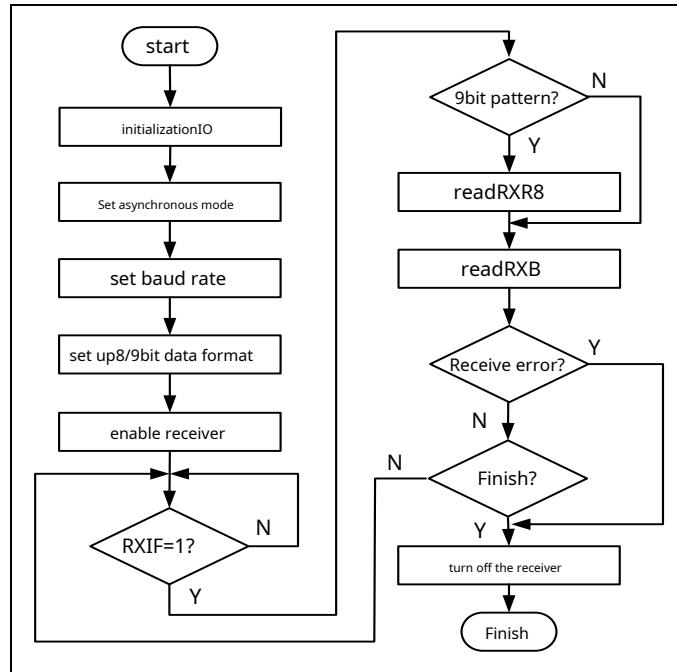
#### 6. 4. 6 asynchronous receiver

When an asynchronous receiver receives data, the user can queryRXIFInterrupt flag bit to determine whether a complete frame of data has been received, and by readingRXBandRXR8Obtain data, or read data in the receive interrupt service routine. provided inside the chip2class9bitFIFOAs a receiving data buffer (the data buffer cannot be directly accessed by the user, it needs to be readRXBto get the data in the buffer) , if the user is at3Before the data is received, it is not readRXB, then overflow flagOERRwill place1. At the end of a frame of data, if the asynchronous receiver does not receive the end bitSTOP, the frame format error flagFERRnwill place1.

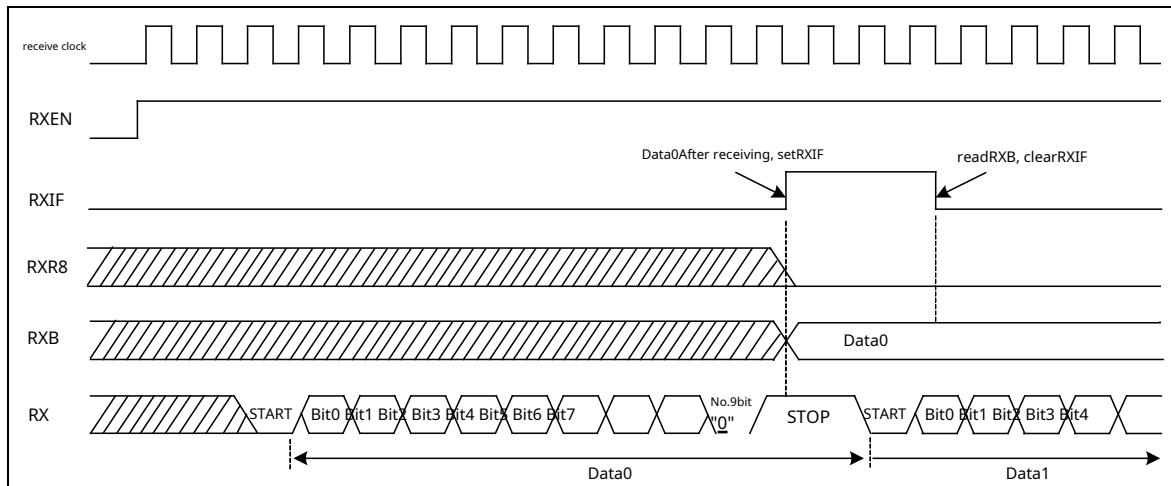
After the current data is received, the interrupt flag bitRXIFwas placed "1" . If the receive interrupt enable bitRXIEand globally enable bitGIEboth for "1", then toCPUissueUARTReceive an interrupt request. interrupt flagRXIFfor read-only, Cannot be cleared by software, read the receive data registerRXB, or disable asynchronous receive (RXEN=0), can be clearedRXIF.

becauseUARTReceiver receiving portRXandI/OPort multiplexing, in useUARTBefore receiving the port, you need to set the multiplexedI/OThe port is in input state.

The operation flow chart is as follows:



picture6-37 UARTsAsynchronous Receiver Operation Flowchart



picture6-38 UARTsReceiver Receive Data Timing Diagram (9bit data format)

#### 6.4.7 UARTsPrecautions for use

existUARTBefore enabling the module, first set the multiplexedI/Oport is a digital port and willTXThe pin is set as an output port,RXThe pin is set as an input port to ensure that theUARTAfter the module is enabled, the data is sent or received successfully. In addition, users are not recommended to switch frequently in the programTXandRXThe input/output type of the pin.

UARTThe transmission baud rate is less than38400bps, the baud rate redundancy is about $\pm 3\%$ , if the external device'sUARTThe deviation of the transmission baud rate relative to the baud rate set by this chip is $\pm 3\%$ Within, the chip can work normally UARTRData transmission; under high and low temperature conditions due to internalINHRCclock frequency deviation,UARTThe baud rate redundancy is approx.  $\pm 2\%$ .

**6. 4. 8 special function register**

UARTModules are related to 5 registers, including a receive data register RXB, a receive control/status register RXC, a transmit data register TXB, a transmit control/status register TXC, and a baud rate register BRR. RXB and TXB The registers are used to store the received data and the data to be sent respectively, RXC The register is used for the enable control of the receiver, the selection of the received data format, and the receive overflow flag, the frame format error flag and the storage of the first 9 bit to receive data, etc. TXC The register is used for the enable control of the transmitter, the selection of the transmission data format, the selection of the baud rate mode, and the storage of the first 9 bit transmit data as well as the transmit shift register (TXR) display of empty flags, etc. BRR Registers are used for UART baud rate setting, the setting range is 00h~FFh.

**6. 4. 8. 1 UARTReceive Data Register (RXB)**

RXB:UARTReceive Data Register								
bit	7	6	5	4	3	2	1	0
name	RXB<7:0>							
R/W	R	R	R	R	R	R	R	R
POR	0	0	0	0	0	0	0	0

Bit 7~0      RXB<7:0>:UARTreceived data

**6. 4. 8. 2 UARTReceive Control/Status Register (RXC)**

RXC:UARTReceive Control/Status Register								
bit	7	6	5	4	3	2	1	0
name	RXEN	RXM	—	—	—	OERR	FERR	RXR8
R/W	R/W	R/W	R/W	R/W	R/W	R	R	R
POR	0	0	0	0	0	0	0	x

Bit 7      RXEN: Receiver enable bit

0:prohibit

1:Enable

Bit 6      RXM: Receiver data format selection bit

0:8Bit data reception format

1:9Bit data reception format

Bit 5~3      Unused

Bit 2      OERR: receive overflow flag

0: no overflow error

1: There is an overflow error (clear RXEN bit to clear this bit) FERR

Bit 1      : frame format error flag bit

0: No frame format error

1: wrong frame format (read RXB, the bit is

refreshed) RXR8: No.9bit receive data bit

0: No.9Bit data is 0

1: No.9Bit data is 1

**6. 4. 8. 3      UARTTransmit Data Register (TXB)**

TXB:UARTsend data register								
bit	7	6	5	4	3	2	1	0
name	TXB<7:0>							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7~0      TXB&lt;7:0&gt;:UARTsent data

**6. 4. 8. 4      UARTTransmit Control/Status Register (TxC)**

TxC:UARTTransmit Control/Status Register								
bit	7	6	5	4	3	2	1	0
name	TXEN	TXM	BRGH	—	—	—	TRMT	TXR8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W
POR	0	0	0	0	0	0	1	0

Bit 7      TXEN: Transmitter enable bit

0:prohibit

1:Enable

Bit 6      TXM: Transmitter data format selection bit

0:8bit data transmission format

1:9bit data transmission format

Bit 5      BRGH: Baud rate mode selection bit

0: Low speed mode

1: High speed mode

Bit 4~2      Unused

Bit 1      TRMT: Transmit Shift Register (TXR) Empty flag

0:TXRnot empty

1:TXRnull

Bit 0      TXR8: No.9bit send data set bit

0: No.9Bit data is0

1: No.9Bit data is1

**6. 4. 8. 5      UARTBaud Rate Register (BRR)**

BRR:UARTbaud rate register								
bit	7	6	5	4	3	2	1	0
name	BRR<7:0>							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7~0      BRR&lt;7:0&gt;:UARTbaud rate setting,00h~FFh

## 6.5 I2Cbus slave (I2CS)

### 6.5.1

overview

This series of chips supports all the wayI2CSlave, slave modules support7bit slave address match, byI2CThe host controls sending or receiving data.

- Only supports slave mode
- support7bit slave address
- support standardI2CBus protocol, maximum transfer rate400Kbit/s
- supportI2CSportSCL/SDAOpen-drain or push-pull output
- support2level send/receive buffer
- Support automatic clock pull-down wait function
- Support automatic sending "unanswered" function
- It is agreed that the data is received/sent from the highest bit

-I2CSThe following functional components are supported

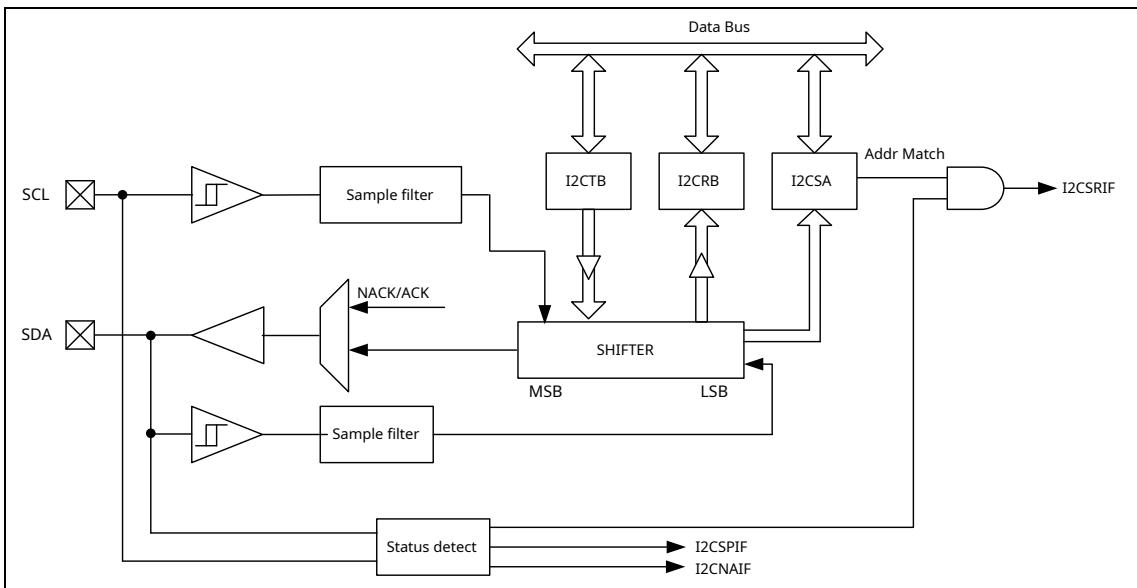
- 5bitI2CSample Filter Register (I2CX16)
- I2CControl Register (I2CC)
- Slave Address Register (I2CSA)
- Transmit Data Buffer (I2CTB)
- Receive Data Buffer (I2CRB)
- Interrupt Enable Register (I2CIEC)
- Interrupt Flag Register (I2CIFC)

-break and pause

- Support receiving "start bit + slave address match + send acknowledge bit" interrupt flag (I2CSRIF)
- Support receive end bit interrupt flag (I2CSPIF)
- Supports transmit buffer empty interrupt flag (I2CTBIF, read only)
- Support receive buffer not empty interrupt flag (I2CRBIF, read only)
- Support for sending error flags (I2CTEIF)
- Support receive overflow interrupt flag (I2CROIF)
- Supports receive unacknowledged flag (I2CNAIF)
- existIDLE0/1mode, suspend receiving and sending
- existIDLE2In this mode, if the sampling filter is disabled, the "start bit + slave address match + send acknowledge bit" can wake upCPU, that is, the slave can be woken up after receiving the matching slave address

Note: In this sectionSDA/SCLThe port represents the chip pin diagramI2C\_SDA/I2C\_SCLport.

## 6.5.2 internal structure



picture6-39 I2Cinternal structure

## 6.5.3 I2CSport configuration

I2CA bus slave consists of a serial data line SDA and a serial clock line SCL. I2C port and normal I/O port multiplexing, available through I2CC in the register I2CTE. Bit selects the function of the multiplexed port. When I2CTE=0 When the multiplexed port is used as a normal I/O; when I2CTE=1, the multiplexed port is used as I2C communication port used.

I2CTE	I2CSClock port configuration	I2CSData port configuration
1	SCL	SDA
0	PB1	PB0

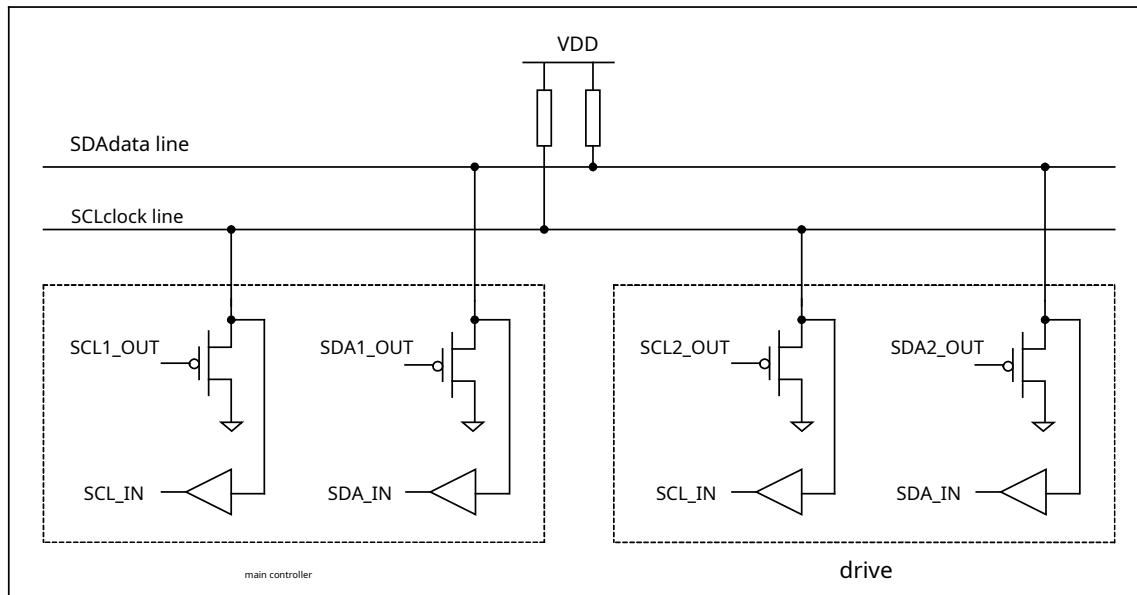
surface6-7 I2CSPort Configuration Table

I2C communication port SCL and SDA, both support push-pull output and open-drain output modes, configuration I2CC register I2COD, which can be selected separately.

The push-pull output is I/O port stdio, output data 0 and 1 hour, I/O The port levels are also respectively 0 and 1.

For push-pull output mode, there is a risk of port level conflict. For example: when the master controller outputs 0, while the slave output 1 When , port signal level conflicts will occur, resulting in uncertain port status.

The open-drain output is I2C The standard mode in the bus protocol can avoid the problem of port level conflict. The schematic diagram of the open-drain output port is shown below:



picture6-40Schematic diagram of an open-drain output port

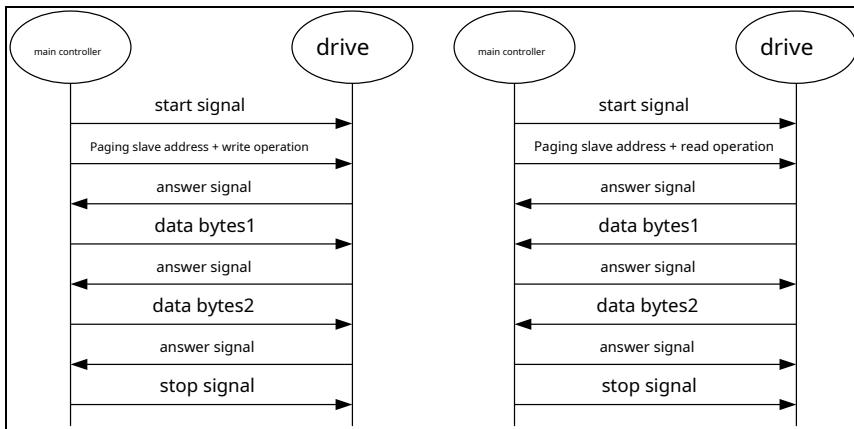
The high level of the open-drain output port is controlled by the I2C bus pull-up resistors are provided. The low level is determined jointly by the master controller and the slave.

Either side can pull down the bus level to 0, but only when both parties release the bus, the bus level can be pulled up to 1.

#### 6.5.4 I2CProtocol

I2C in communication, the following protocol must be followed:

- The communication is initiated by the master controller, and the start signal is sent (start) control bus, send stop signal (end) release the bus;
- There can be multiple masters on the bus at the same time (provided that each master supports the multi-master arbitration mechanism), but until one less slave is needed, and each slave must have an independent and unique paging address;
- After the master controller sends the start signal, it immediately pages the slave address and sends the read and write mode bits;
- Read and write control bits R/#W (called the direction bit) is used to inform the slave of the direction of data transfer, "0" indicates that this communication is "written" by the master controller to the slave device; "1" indicates that this communication is "read" from the master to the slave data;
- I2C The communication protocol supports the response mechanism, that is, every time the sender transmits a byte of data (including the paging address), the receiver must reply with a response signal (ACK or NACK), and the sender proceeds to the next step according to the response signal;
- If the master and slave clock lines (SCL) all use open-drain design, and the master supports the clock line waiting request operation, then the slave can pull down the clock line when the clock line is low, so that the master waits for the slave until the slave releases the clock line;
- Each data byte is transmitted MSB first.



picture6-41 I2CSchematic diagram of bus communication protocol

### 6.5.5 I2Coperate

I2CThere can be multiple masters on the bus at the same time (provided each master supports the multi-master arbitration mechanism), but at least one slave is required. Each device on the bus does not have a selection line, but each corresponds to a unique address forI2C communication.

The slave module includes two operation modes: the master sends data to the slave and the master reads data from the slave.

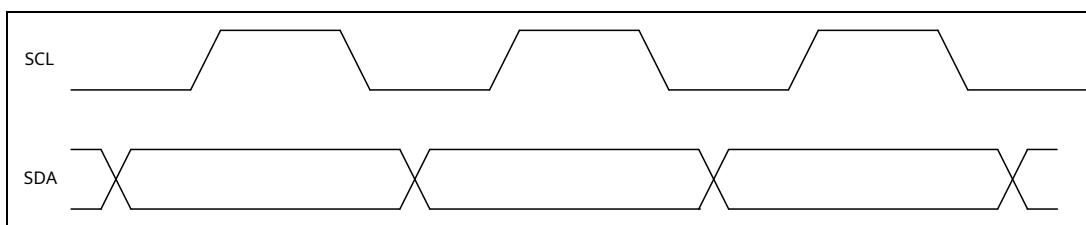
When the master sends data to the slave, the slave usually judges that the receive buffer is not empty and the interrupt flag bitI2CRBIF, if the receive buffer is not empty, that is, the host data is received, read the data in the receive buffer.

When the master reads data from the slave, the slave usually judges the transmit buffer empty interrupt flagI2CTBIF, if the send buffer is empty, write the data to be sent in sequence.

To avoid sending data by mistake, it is recommended that each complete communication be completed (e.g. receivedSTOP flag), use a software reset I2Cmodule bitI2CRSTSet and reset onceI2Cmodule to clear the receive and transmit data buffers while reinitializingI2CCandI2CIECregister, for the nextI2CNewsletter ready. (I2Chost toI2CSlave send command read command, after sending the slave address, you need to wait for about30usleft and right to read the data)

existIDLE0/1mode,I2CSModule communication suspended.

existIDLE2mode, if the sampling filter is disabled,I2CWKUPENThe register is set to1,I2CAddress match wake-up enable, then "start bit + slave address match + send acknowledge bit" can wake upCPU.

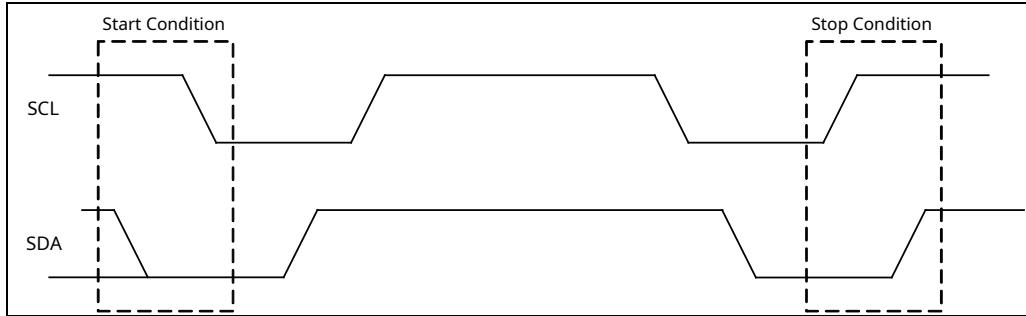


picture6-42 I2CSlave Waveform

### 6. 5. 6 start bitSTARTand stop bitsSTOP

according toI2CProtocol: During data transfer, whenSCLWhen high, theSDAMust maintain a fixed level, the waveform is shown in the figure; during no data transmission period,SCLandSDAshould remain high. whenSCLline is high when the

SDAline toggles from high to low to indicate a start condition (S) . whenSCLis high when theSDAline by low power A level toggle high indicates a STOP condition (P) The start and stop conditions are generally generated by the host, as shown in the figure.



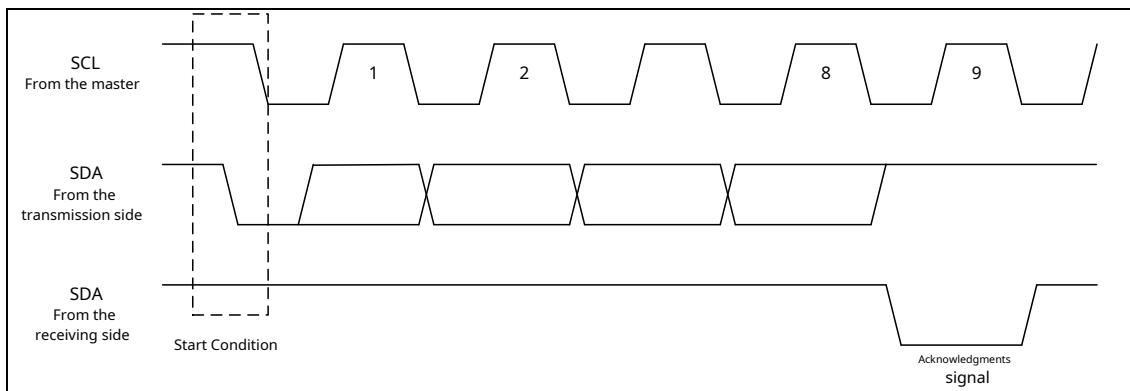
picture6-43 I2Cstart and stop bits

### 6. 5. 7 Data transmission and reply

into the start condition (S) After that, the data is transferred serially as one byte (8bit). Receiver finishes receiving each time8After the bit data, an acknowledgment signal needs to be sent to the sender. When data is transferred to the8individualSCLWhen the falling edge occurs, the receiver immediately sends a response signal, and the sender releases theSDAcontrol, while the receiver willSDAgoes low. When a byte of data is sent immediately following the reception of a previous byte (or when the receiver switches to the sender and begins data transmission), the receiver9 individualSCLFalling edge releaseSDAControl.

When the host is the sender, if the automatic unansweredI2CNAEbit is not enabled, after the paging address matches, the slave will send ACKacknowledge signal, andI2CTASSelect bit don't care; if auto unacknowledgedI2CNAEbit enable, after a page address match, if the2stage receive buffer is full, then sendNACKsignal, sent if not fullACKsignal, withI2CTAS The selection bits are irrelevant. When the slave receives data, the if auto unansweredI2CNAEbit enable, clock pull down waitI2CCSE bit is disabled, and2If the stage receive buffer is full, after receiving one byte of data, it will sendNACKsignal; otherwise sendACKorNACKsignal byI2CTASSelect bit control.

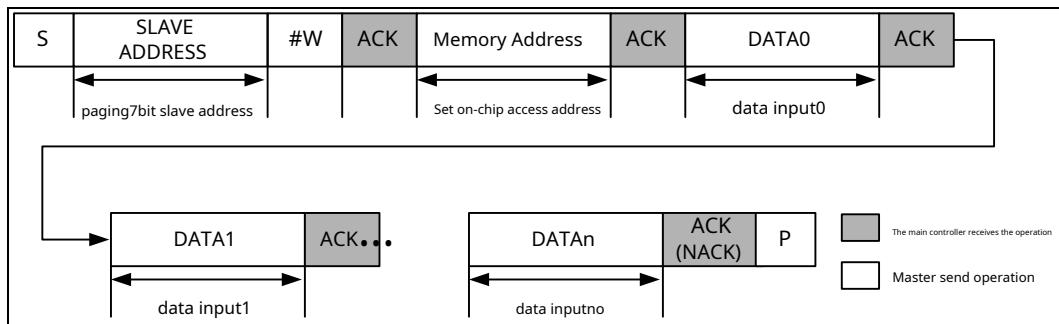
When the host is the receiver, if the automatic unansweredI2CNAEbit is not enabled, after the paging address matches, the slave will send ACKacknowledge signal, andI2CTASSelect bit don't care; if auto unacknowledgedI2CNAEbit enable, after a page address match, if the2stage send buffer is empty, then sendNACKsignal, sent if not nullACKsignal, withI2CTAS The selection bits are irrelevant. Each time the host receives a byte of data, the host sendsACKAcknowledgment signal, when the last byte of the slave is received, no answer signal is generated to inform the sending device that the data transmission is complete. on the9individualSCLFalling edge, the slave (sender) continues to releaseSDAcontrol so that the host can generate a STOP condition (P) .



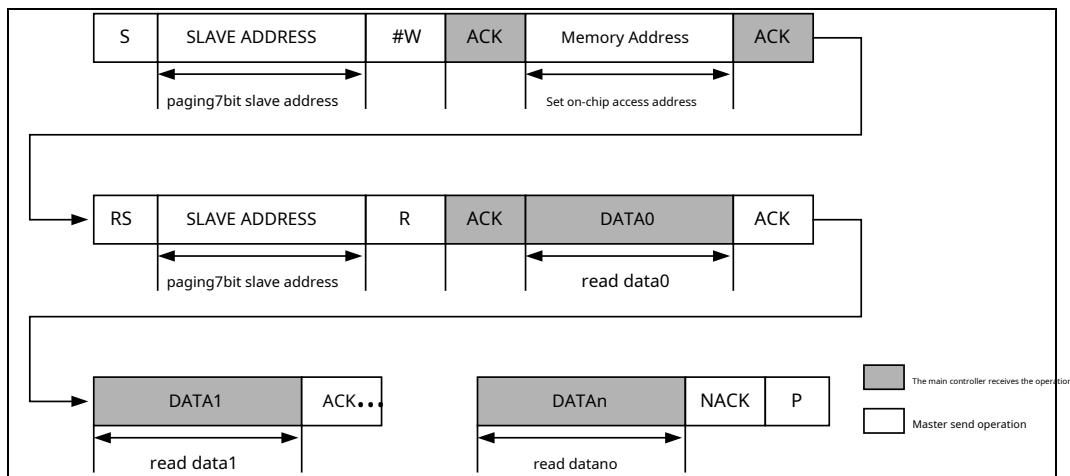
picture6-44Data transmission and reply

## 6.5.8 Data Transfer Format Reference

I2CThe data transmission reference format of the slave is as follows:



picture6-45Schematic diagram of master controller writing slave data



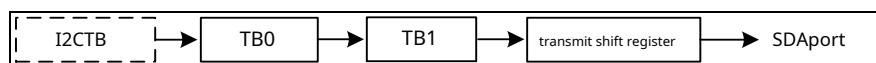
picture6-46Schematic diagram of master controller reading slave data

## 6.5.9 I2Ccommunication transmitter

support2Stage Transmit BufferTB0,TB1and1Stage send shift register, data can be continuously sent until the send buffer and shift register are completely empty, and can be continuously written and sent at most3frame data. send bufferTB0~TB1not accessible, only by sending the data registerI2CTBwrite.

WriteI2CTBWhen the register address unit is used, the sending data is actually written to the sending bufferTB0~TB1, then transferred to the transmit shift register, through the data portSDASend data.

The schematic diagram of the data flow of sending data from writing to sending to the port is as follows:



picture6-47 I2CSchematic diagram of sending data flow

Support transmit buffer empty interrupt, when transmit bufferTB0andTB1When both are empty, the transmit buffer empty interrupt flag will be setI2CTBIF;

Support transmit error interrupt, when transmit bufferTB0andTB1When it is all empty, when the communication clock provided by the host is received, the sending error interrupt flag will be setI2CTEIF.

### 6.5.10 I2C communication receiver

support 2stage receive buffer RB0, RB1 and 1stage receive shift register, which can continuously receive data until the receive buffer and shift register are full, and can receive continuously at most 3Frame data, and then execute the data read operation. Read Receive Data Register I2CRB, the received data can be obtained.

read I2CRB, actually read the receive buffer RB0The data.

The schematic diagram of the data flow of received data from the data port to the buffers at all levels is as follows:



picture6-48 I2CSchematic diagram of receiving data flow

Support receive overflow interrupt when 2stage receive buffer and 1When the stage receiving shift registers are full, the received data overflow interrupt flag will be set I2CROIF, and no new data will be received.

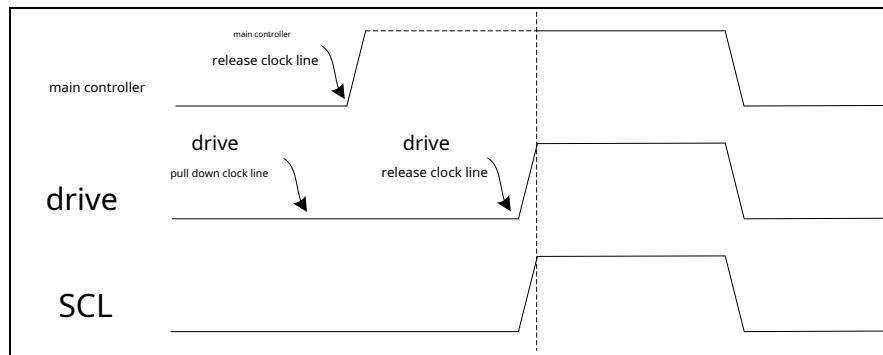
Support receive buffer buffer is not empty interrupt, when the receive buffer RB0 and RB1 When data is received, the receive buffer is not empty interrupt flag will be set I2CRBIF.

### 6.5.11 I2C Clock line automatic pull-down waiting request function

I2C The slave supports the clock line automatic pull-down waiting request function, configuration I2CCregister I2CCSE, to enable the function.

to achieve I2C The pull-down wait request function of the clock line needs to be configured I2CCregister I2COD, the communication port SCL Select the open-drain output mode, provide a high level through the pull-up resistor, so that the slave can pull down the clock line to make the master wait.

Under normal circumstances, the slave is in the state of releasing the clock line, and the clock line SCL Fully controlled by the main controller. However, when the slave has abnormal conditions and cannot continue data transmission in a short period of time, the slave can SCL output when low 0 (It is not possible to output at high level 0, otherwise it will destroy the data transfer process) , forcibly SCL Keep Low level, the master enters the communication waiting state until the slave releases the clock line. The schematic diagram of the clock line pull-down waiting request waveform is as follows:



picture6-49 I2CSchematic diagram of clock line pull-down waiting waveform

I2C After the slave clock is automatically pulled down and the waiting request function is enabled, when receiving the paging address and read operation bit of the chip, if I2C The transmit buffer and transmit shift register are all empty, and the automatic transmission not acknowledged enable bit I2CANAE=0, then after sending the response signal, it will automatically pull down the clock line and wait for the sending buffer to write data; when receiving the paging address and write operation bit of the chip, if I2C The receive buffer and receive shift register are full, and the automatic

Transmit Not Acknowledged Enable Bit I2CNAE=0, then after the response signal is sent, the clock line will be pulled down automatically, waiting for the receive buffer to be read.

### 6.5.12 special function register

#### 6.5.12.1 I2CSample Filter Register (I2CX16)

I2CX16:I2CSampling Filter Register								
bit	7	6	5	4	3	2	1	0
name	—	—	—	I2CX16<4:0>				
R/W	—	—	—	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7~5      Unused

Bit 4~0      I2CX16<4:0>: sampling filter control bit

  00h: disable sampling filter

  01h~1Fh: Communication clock and data sampling filter time is  $T_{osc}(I2CX16+1) \times 3$

#### 6.5.12.2 I2CCControl Register (I2CC)

I2CC:I2CCcontrol register								
bit	7	6	5	4	3	2	1	0
name	I2CTE	I2CPU	I2COD	I2CTAS	I2CNAE	I2CCSE	I2CRST	I2CEN
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7      I2CTE:I2CCommunication port enable bit

  0:prohibit

  1:Enable

Bit 6      I2CPU:I2CInternal weak pull-up enabled

  0:prohibit

  1:Enable

when I2CTE=1hour,I2CPUcontrolSCL/SDAWeak pull-up function of the port.

Bit 5      I2COD:I2COpen-drain output enable bit

  0: push-pull output

  1: Open-drain output

Bit 4      I2CTAS:I2Csend acknowledgment set bit

  0:sendACK

  1:sendNACK

Bit 3      I2CNAE:I2CAuto No Ack Enable Bit

  0:prohibit

  1:Enable

Bit 2      I2CCSE:I2CClock pull-down wait enable bit

  0:prohibit

  1:Enable

Bit 1      I2CRST: Software resetI2Cmodule bit

  0:prohibit

1: Enable (after reset, the hardware will automatically clear)

Bit 0 I2CEN:I2Cmodule enable bit

0:prohibit

1:Enable

Note1: After enabling the clock automatic pull-down function (and prohibiting the automatic unacknowledged (I2CNAE=0)) , when the slave receives a matching paging address:

1.1) If it is a host read operation, when the transmit buffer and the shift register are all empty, send theACK, and then pull down the clock, after writing the sent data, release the clock line, and send the data bit directly;

1.2) If it is a host write operation, when the receive buffer and shift register are full, sendACK, then pull down the clock, read the received data, release the clock line, and receive new data bits.

Note2:I2CAfter auto no answer is enabled (I2CNAE=1) ,

When off-chip main controller paging local address+Rwhen, if2If the sending data buffer of the stage is completely empty, the response bit after the local address is "NACK" ;

When off-chip main controller paging local address +#Wwhen, if before the data is received,2When the stage receiving data buffer is full, the response bit after the local address is "

NACK" ; if after receiving data, andI2CCSE=0,2When the receiving data buffer of the stage is full, the response bit after receiving the data is "NACK" .

Note3:whenI2CTE=1hour,I2CPUcontrolSCL/SDAweak pull-up function of the port; otherwise, byPBPU<1:0>controlPB1/0Port's weak pull-up Function.

### 6. 5. 12. 3 I2CSlave Address Register (I2CSA)

I2CSA:I2CSlave Address Register								
bit	7	6	5	4	3	2	1	0
name	I2CSADDR<6:0>							I2CRW
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R
POR	0	0	0	0	0	0	0	0

Bit 7~1 I2CSADDR<6:0>: slave address

Used for match comparison after receiving "start/restart"

Bit 0 I2CRW: After the slave address is matched, the read/write bit is automatically updated

0:Write

1:read

### 6. 5. 12. 4 I2CTransmit Data Buffer (I2CTB)

I2CTB:I2Csend data buffer								
bit	7	6	5	4	3	2	1	0
name	I2CTB<7:0>							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7~0 I2CTB<7:0>: send data buffer

Note: The first data to be sent should be written into the send data buffer before the send is enabled.

**6. 5. 12. 5 I2CReceive Data Buffer (I2CRB)**

I2CRB:I2CReceive data buffer								
bit	7	6	5	4	3	2	1	0
<b>name</b>	I2CRB<7:0>							
<b>R/W</b>	R	R	R	R	R	R	R	R
<b>POR</b>	0	0	0	0	0	0	0	0

Bit 7~0      I2CRB&lt;7:0&gt;: receive data buffer

**6. 5. 12. 6 I2CInterrupt Enable Register (I2CIEC)**

I2CIEC:I2Cinterrupt enable register								
bit	7	6	5	4	3	2	1	0
<b>name</b>	I2CWKUPEN	I2CNAIE	I2CROIE	I2CTEIE	I2CRBIE	I2CTBIE	I2CSPIE	I2CSRIE
<b>R/W</b>	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
<b>POR</b>	0	0	0	0	0	0	0	0

- Bit 7      I2CWKUPEN:I2CAddress match wake-up enable bit 0  
               0:prohibit  
               1:Enable
- Bit 6      I2CNAIE:I2CReceive Not Acknowledged Interrupt Enable Bit  
               0:prohibit  
               1:Enable
- Bit 5      I2CROIE:I2CReceive overflow interrupt enable bit  
               0:prohibit  
               1:Enable
- Bit 4      I2CTEIE:I2Ctransmit error interrupt enable bit  
               0:prohibit  
               1:Enable
- Bit 3      I2CRBIE:I2CReceive buffer not empty interrupt enable bit  
               0:prohibit  
               1:Enable
- Bit 2      I2CTBIE:I2CTransmit buffer empty interrupt enable bit  
               0:prohibit  
               1:Enable
- Bit 1      I2CSPIE:I2CReceive end bit interrupt enable bit  
               0:prohibit  
               1:Enable
- Bit 0      I2CSRIE:I2CReceive "start bit + slave address match + send acknowledge bit" interrupt enable bit  
               0:prohibit  
               1:Enable

### 6. 5. 12. 7 I2CInterrupt Flag Register (I2CIFC)

I2CIFC:I2CInterrupt Flag Register								
bit	7	6	5	4	3	2	1	0
name	—	I2CNAIF	I2CROIF	I2CTEIF	I2CRBIF	I2CTBIF	I2CSPIF	I2CSRIF
R/W	—	R/W	R/W	R/W	R	R	R/W	R/W
POR	1	0	0	0	0	1	0	0
Bit 7	Unused							
Bit 6	I2CNAIF:I2CReceive unacknowledged interrupt flag 0: not received or not sentNACK 1: receive or sendNACK, an interrupt flag is generated (cleared by software)							
Bit 5	I2CROIF:I2CReceive overflow interrupt flag 0:2stage receive data buffer andI2CThe shift register did not overflow 1:2stage receive data buffer andI2CShift register overflow, generate interrupt flag (cleared by software)							
Bit 4	I2CTEIF:I2Csend error interrupt flag 0: No send error occurred  1: Send error occurred:2When the sending data buffer of the stage is completely empty, the communication clock provided by the host is received, and an interrupt flag is generated (cleared by software)							
Bit 3	I2CRBIF:I2CReceive buffer not empty interrupt flag bit 0:2Stage Receive Data Buffer Empty 1:2When the stage receive data buffer is not completely empty, an interrupt flag is generated (readI2CRBregister clearable interrupt flag bit)							
Bit 2	I2CTBIF:I2CTransmit buffer empty interrupt flag 0:2Stage send data buffer is not empty 1:2When the stage transmit data buffer is completely empty, an interrupt flag is generated (writeI2CTBregister clearable interrupt flag bit)							
Bit 1	I2CSPIF:I2CReceive end bit interrupt flag bit 0: End bit not received 1: Receive end bit, generate interrupt flag (cleared by software)							
Bit 0	I2CSRIF:I2CReceive "start bit + slave address match + send acknowledge bit" interrupt flag bit 0: "Start bit + address bit and address match + send acknowledge bit" not received 1: After receiving "start bit + address bit and address match + send acknowledge bit", an interrupt flag will be generated (cleared by software)							

Note1: Clear the total interrupt flag bitI2CIFCbefore, clearI2CIFCRegister related interrupt flag bits. Note2:

Continuously receive data exceeding3When, receive overflow occurs, and the first4Received data will be lost.

Note3:I2CAfter the module sends each frame of data and receives the end bit, the hardware automatically clears the send buffer register.

## 6.6 SPISynchronous serial communication module (SPI)

### 6. 6. 1

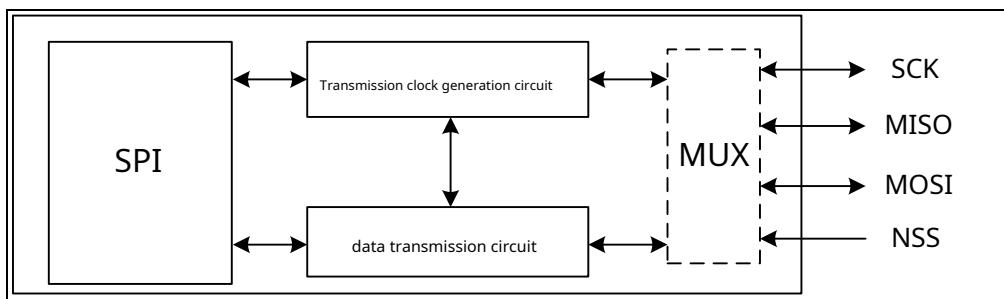
#### overview

- Support master mode, slave mode
- support4data transfer format
- Support master mode communication clock rate configurable
- support4stage transmit buffer and4stage receive buffer
- Supports transmit and receive buffer empty/full interrupts
- Support receive data overflow interrupt, send data write error interrupt, send data error interrupt in slave mode
- Supports chip select change interrupt in slave mode and idle state interrupt in master mode
- Support delayed reception in master mode

Note: In this sectionSCK/NSS/MISO/MOSIThe port represents the chip pin diagramSPI\_SCK/SPI\_NSS/SPI\_MISO/SPI\_MOSIport.

### 6.6.2

#### Structure diagram



picture6-50 SPICircuit structure block diagram

### 6.6.3 SPIcommunication mode

SPISupport master control and slave two communication modes, configurationSPICON1registerMSbit to select the communication mode.

The communication clock port isSCK, the chip select signal port in slave mode isNSS, the data output portMISO, the data input portMOSI, the commonI/OThe port is used as the chip selection signal port of the off-chip slave device, and the data output portMOSI, the data input portMISO. See the table below for details:

SPIcommunication port	SPImaster mode	SPIslave mode
SCK	support	support
MOSI	support	support
MISO	support	support
NSS	—	support

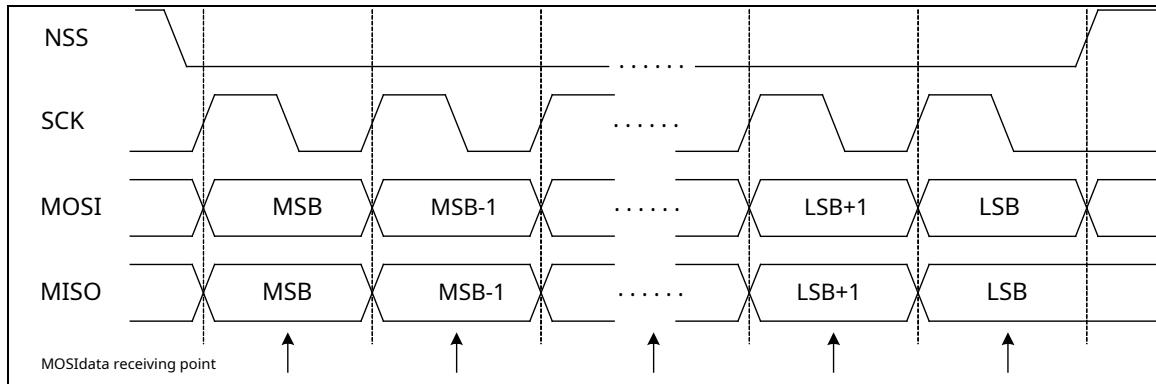
### 6. 6. 4 SPIData Format

configurationSPICON1registerDFS<1:0>, optionalSPICommunication data format, when data is sent and received, the high bit comes first and the low bit comes after. If the sending data comes first and the receiving data comes later, the output port MOSI(orMISO) will be in the firstSCKclock edge, the output data'sMSBbit; otherwise the output portMOSI(orMISO) at

oneSCKBefore the clock edge, the output data'sMSBbit.

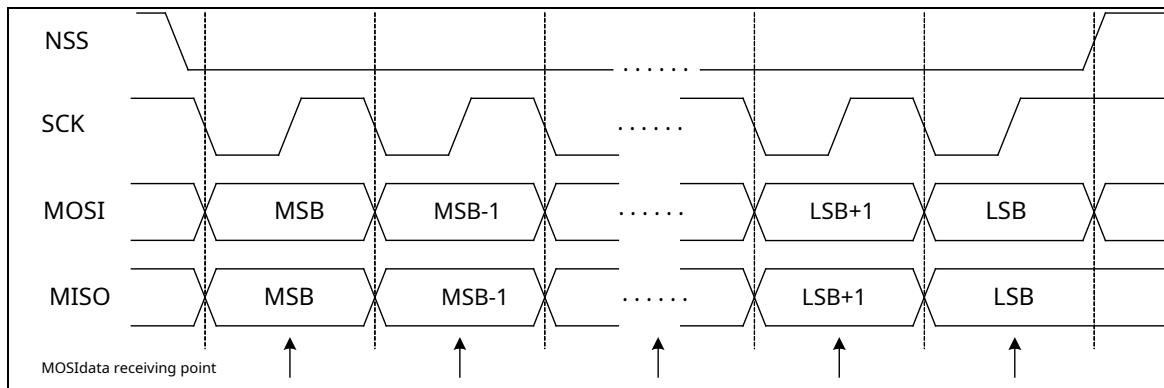
the following toSPITaking the slave communication mode as an example, the data communication sequence will be described.

1) rising edge to send (first), falling edge to receive (after) :



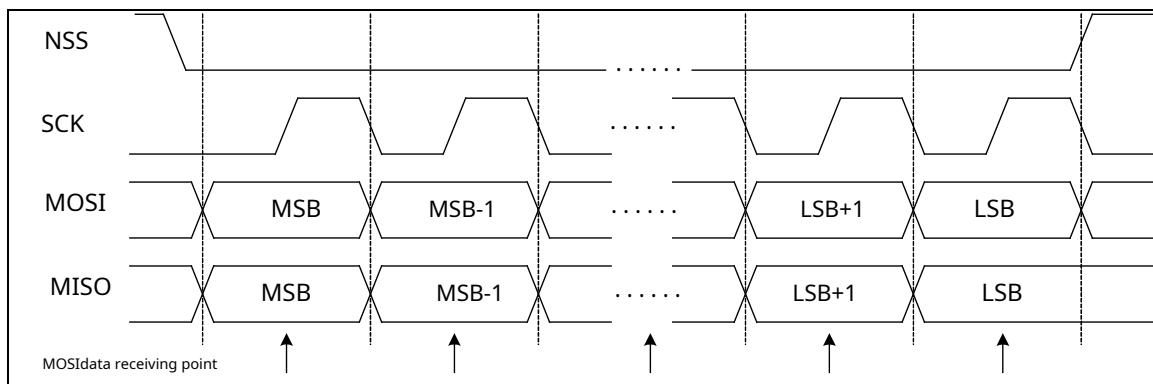
picture6-51 SPISchematic diagram of clock rising edge sending and falling edge receiving waveform

2) falling edge transmits (first), rising edge receives (after) :



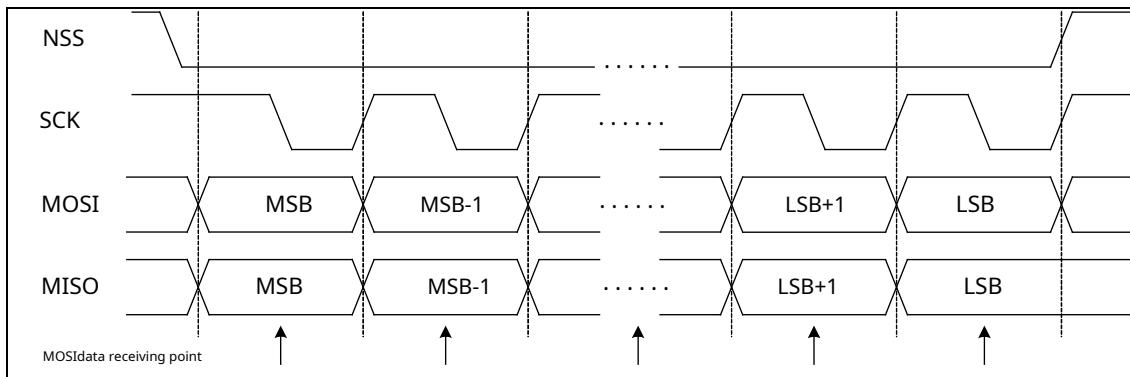
picture6-52 SPISchematic diagram of clock falling edge sending and rising edge receiving waveform

3) rising edge to receive (first), falling edge to send (after) :



picture6-53 SPISchematic diagram of clock rising edge receiving and falling edge sending waveform

4) falling edge receive (first), rising edge transmit (after) :



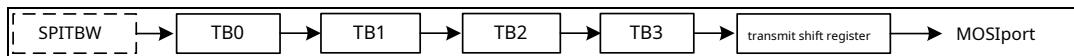
picture6-54 SPI schematic diagram of clock falling edge receiving and rising edge sending waveform

### 6.6.5 SPIsync transmitter

support4Stage Transmit BufferTB0,TB1,TB2,TB3and1Stage send shift register, data can be continuously sent until the send buffer and shift register are completely empty, and can be continuously written and sent at most5frame data. send buffer TB0~TB3and the transmit shift register are virtual registers, not accessible only through the transmit data registerSPITBW write.

send data registerSPITBW, when writing the register address unit, actually write the send data into the send bufferTB0, and then transferred to the transmit shift register step by step, through the transmit data portMOSI(orMISO) to send data.

The schematic diagram of the data flow of sending data from writing to sending to the port is shown below (take the master control mode as an example)



picture6-55 SPI schematic diagram of sending data flow

Support transmit buffer empty interrupt, configurationSPICON0registerTBIM, the interrupt mode can be selected.

TBIM<1:0>=00,forTB0An interrupt is generated when the transmit buffer is emptyTB0when empty,SPIIFregister will set the interrupt flagTBIF;

TBIM<1:0>=01,forTB0andTB1An interrupt is generated when the transmit buffer is emptyTB0andTB1are empty,SPIIF register will set the interrupt flagTBIF;

TBIM<1:0>=10,forTB0~TB3All empty generates an interrupt, i.e. transmit bufferTB0~TB3are empty,SPIIF register will set the interrupt flagTBIF.

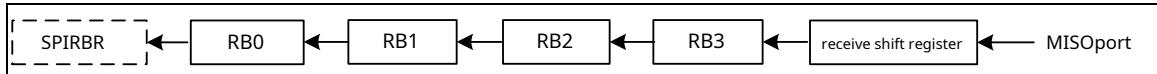
Support transmit data registerSPITBWMiswrite interrupt, when the rightSPITBWWwhen writing to the transmit bufferTB0~TB3 at full capacity,SPIIFregister will set the miswrite interrupt flagTBWEIF.

### 6. 6. 6 SPIsync receiver

support4stage receive bufferRB0,RB1,RB2,RB3and1stage receive shift register, which can continuously receive data until the receive buffer and shift register are full, and can receive continuously at most5Frame data, and then execute the data read operation.RB0,RB1,RB2,RB3and receive shift registers are virtual registers, inaccessible only by reading the receive data registerSPIRBRGet the received data.

Receive Data RegisterSPIRBR, when reading the register address unit, actually read the receive bufferRB0data in .

The schematic diagram of the data flow of received data from the receiving port to the buffers at all levels is as follows (take the master control mode as an example):



picture6-56 SPISchematic diagram of receiving data flow

The receive sequence for a synchronous receiver is as follows:

When the synchronous receiver is completely empty, the data from the receive shift register is automatically shifted into the RBO;

onlyRB1~RB3When empty, the data of the receiving shift register is automatically shifted intoRB1;

onlyRB2~RB3When empty, the data of the receiving shift register is automatically shifted intoRB2;

onlyRB3When empty, the data of the receiving shift register is automatically shifted intoRB3.

when4stage receive buffer and1When the stage receive shift registers are full, if data bits are received again,SPIIFThe receive data overflow interrupt flag will be set in the registerROIF, while no new data will be received, and the buffer data will remain.

Support receive buffer full interrupt, configurationSPICON0registerRBIM, the interrupt mode can be selected.

RBIM<1:0>=00,forRBOInterrupt generated when the receive buffer is fullRB0when full,SPIIFThe interrupt flag will be set in the registerRBIF;

RBIM<1:0>=01,forRBOandRB1Interrupt generated when the receive buffer is fullRB0andRB1are full,SPIIF The interrupt flag will be set in the registerRBIF;

RBIM<1:0>=10,forRBO~RB3Interrupt generated when full, i.e. receive bufferRB0~RB3are full,SPIIF The interrupt flag will be set in the registerRBIF.

#### 6. 6. 7 SPIcommunication control

configurationSPICommunication mode, data format; for the main control mode, it needs to be configuredSPICON0ofCKSRegister, set the transmission clock rate, for the slave mode, the transmission clock is provided by the host side;SPICON1Configure in registerSPIEN andREN, enable data transmission and reception; write the data to be transmitted into the transmission data registerSPITBW, you can start sending data, read the receive data registerSPIRBR, the received data can be obtained.

SPIIn the master control mode, after the data in the transmit buffer and the transmit shift register are sent, it enters the idle state. SPIIFidle interrupt flag generated in registerIDIF.

SPIIn slave mode, if the transmit bufferTB0~TB3When both the transmission shift register and the transmission shift register are empty, and the communication clock provided by the host is received, thenSPIIFThe transmit error interrupt flag will be set in the registerTEIF.

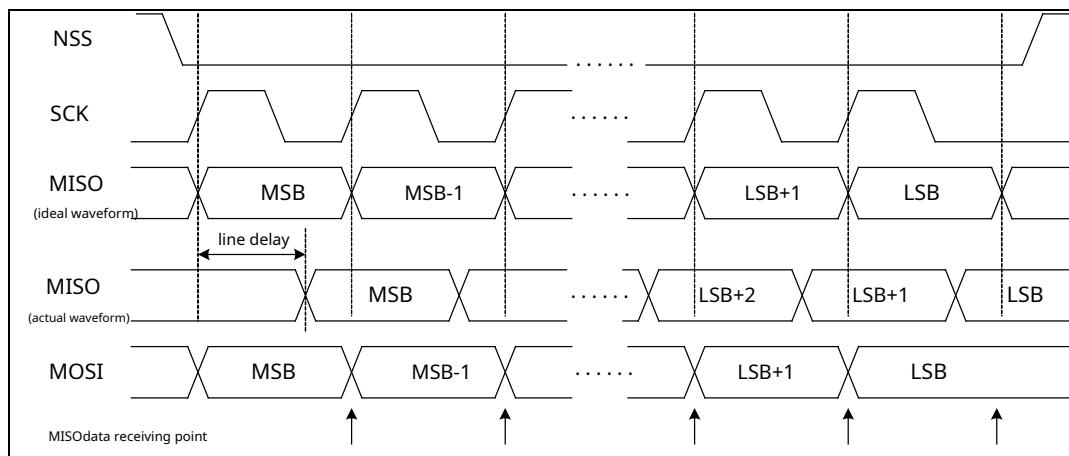
SPISlave mode, support chip select signal change interrupt, configurationSPIIERegisterNSSIE, to enable the interrupt. configurationSPICON1registerSPIRSTbit, can beSPICommunication module software reset, after reset: prohibit data communicationSPIEN=0;SPIIERelated interrupts are disabled in the registerTBIE=0, TBWEIE=0,RBIE=0,TEIE=0, ROIE=0,IDIE=0,NSSIE=0;SPIIFReset the related interrupt flag in the register to the default valueTBIF=1, TBWEIF=0,RBIF=0,TEIF=0,ROIF=0,IDIF=0,NSSIF=0.

### 6. 6. 8 SPIDelayed reception function

SPIDuring communication, the rising/falling edge of the clock is used to synchronize the sending and receiving of data respectively. During normal communication, for the data received by the master, the data sent by the slave should arrive at the receiving port of the master within half a clock cycle, otherwise the data received by the master will be lost.

SPIMaster control mode supports delayed reception function, configurationSPICON1registerDRE, this function can be enabled, the host will delay another half clock cycle, and receive and collect data at the edge of the next sending clock. Therefore, after the delay receiving function is enabled, the line delay between the slave sending port and the master receiving port can be as close as possible to the maximum1communication clock cycle.

for exampleSPIDelayed reception function:SPICON1in registerDFS<1:0>=00, the rising edge is sent (first), the falling edge is received (after) .



picture6-57 SPISchematic diagram of delay receiving function waveform

Note: When the communication rate is selectedFosc/2, it is recommended to enableSPIdelay receiving function to avoid communication abnormality caused by line delay.

### 6. 6. 9 SPIapplication note

to guaranteeSPInormal communication,SPIThe configuration must comply with the following requirements:

1) Since different communication data formats have different requirements for the initial level of the port (see the above communication waveform diagrams), if it is not possible to determine the enableSPIpreviousSPIThe initial value of the port must first configure the communication data format control bitDFS<1:0>,rightSPIThe initial level of the port is automatically set; and then through theSPICON1register SPIENandRENplace1to enableSPIsend and receive, that is toSPICON1Registers are written in two steps.

2) The master device and the slave device need to be configured in the same communication data format.

### 6. 6. 10special function register

#### 6. 6. 10. 1 SPIcontrol register0(SPICON0)

SPICON0:SPIcontrol register0								
bit	7	6	5	4	3	2	1	0
name	RXCLR	TXCLR	CKS<1:0>		RBIM<1:0>		TBIM<1:0>	
R/W	W	W	W	W	R/W	R/W	R/W	R/W
POR	0	0	1	1	0	0	0	0

Bit 7 RXCLR:SPIReceive buffer clear control bit

0:invalid

	1: Clear the receive buffer
Bit 6	TXCLR:SPI Transmit buffer clear control bit 0:invalid 1: Clear the send buffer
Bit 5~4	CKS<1:0>:SPI Communication baud rate selection bit (only supported by the master control mode) (Note: this bit can only be written, not read, read out0) 00:Fosc/2 01:Fosc/4 10:Fosc/8 11:Fosc/16
Bit 3~2	RBIM<1:0>:SPI Receive buffer full interrupt mode select bit 00:RB0full interrupt 01:RB0andRB1full interrupt 10: RB0~RB3full full interrupt 11 :reserve
Bit 1~0	TBIM<1:0>:SPI Transmit buffer empty interrupt mode select bit 00:TB0empty interrupt 01:TB0andTB1empty interrupt 10 :TB0~TB3all empty interrupt 11 :reserve

## 6. 6. 10. 2 SPIcontrol register1(SPICON1)

SPICON1:SPIcontrol register1								
bit	7	6	5	4	3	2	1	0
name	DFS<1:0>		DRE	—	REN	MS	SPIRST	SPIEN
R/W	R/W	R/W	R/W	—	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7~6	DFS<1:0>:SPI Communication data format 00: Rising edge sending (first), falling edge receiving (later) 01: Falling edge sending (first), rising edge receiving (later) 10: Rising edge receiving (first), falling edge sending (later) 11: Falling edge reception (first)the rising edge is sent (after)
Bit 5	DRE:SPI Delayed receive enable bit (only supported by master mode) 0:prohibit 1:Enable
Bit 4	Reserved
Bit 3	REN:SPI receive enable bit 0:prohibit 1: enable (requires SPIEN enable at the same time)
Bit 2	MS:SPI Communication mode selection bit 0: master mode 1: slave mode

Bit 1	SPIRST:SPIsoftware reset 0: always read as0 1: Software resetSPImodule, auto-clear SPIEN:
Bit 0	SPIcommunication enable bit 0:prohibit 1:Enable(SPIcommunication enabled, but only data transmission is enabled)

Note1: Due to different communication data formats, the initial level requirements of the ports are different (see the above communication waveform diagrams) , so if it is not possible to determine the ableSPIpreviousSPIThe initial value of the port must first configure the communication data format control bitDFS<1:0>,rightSPIThe initial level of the port is automatically set; and then through theSPICON1registerSPIENandRENplace1to enableSPIsend and receive. Right nowSPICON1The register needs to be written twice, otherwise communication errors will easily occur.

Note2: The master device and the slave device need to be configured in the same communication data format.

#### 6. 6. 10. 3 SPISend data is written to the register (SPITBW)

SPITBW:SPISend data write register								
bit	7	6	5	4	3	2	1	0
name	TBW<7:0>							
R/W	W	W	W	W	W	W	W	W
POR	0	0	0	0	0	0	0	0

Bit 7~0 TBW<7:0>: Write send data

#### 6. 6. 10. 4 SPIReceive Data Read Register (SPIRBR)

SPIRBR:SPIReceive data read register								
bit	7	6	5	4	3	2	1	0
name	RBR<7:0>							
R/W	R	R	R	R	R	R	R	R
POR	0	0	0	0	0	0	0	0

Bit 7~0 RBR<7:0>: Read received data

#### 6. 6. 10. 5 SPIInterrupt Enable Register (SPIIE)

SPIIE:SPIinterrupt enable register								
bit	7	6	5	4	3	2	1	0
name	—	TBWEIE	NSSIE	IDIE	ROIE	TEIE	RBIE	TBIE
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7 Unused

Bit 6 TBWEIE:SPISend data write error interrupt enable bit

0:prohibit

1:Enable

Bit 5 NSSIE:SPIChip select change interrupt enable bit (only supported in slave mode)

0:prohibit

	1:Enable
Bit 4	IDIE:SPIidle state interrupt enable bit (supported in master mode only)
	0:prohibit
	1:Enable
Bit 3	ROIE:SPIREceive data overflow interrupt enable bit
	0:prohibit
	1:Enable
Bit 2	TEIE:SPISend data error interrupt enable bit (only supported by slave mode)
	0:prohibit
	1:Enable
Bit 1	RBIE:SPIREceive buffer full interrupt enable bit
	0:prohibit
	1:Enable
Bit 0	TBIE:SPITransmit buffer empty interrupt enable bit
	0:prohibit
	1:Enable

#### 6. 6. 10. 6 SPIInterrupt Flag Register (SPIIF)

SPIIF:SPIInterrupt Flag Register								
<b>bit</b>	<b>7</b>	<b>6</b>	<b>5</b>	<b>4</b>	<b>3</b>	<b>2</b>	<b>1</b>	<b>0</b>
<b>name</b>	—	TBWEIF	NSSIF	IDIF	ROIF	TEIF	RBIF	TBIF
<b>R/W</b>	—	R/W	R/W	R/W	R/W	R/W	R	R
<b>POR</b>	0	0	0	0	0	0	0	1

Bit 7	Unused
Bit 6	TBWEIF:SPISend data write error interrupt flag bit
	0: No write error occurred
	1: write error occurred : yesSPITBWwhen writing,TB0~TB3In a full state the software writes1clear the flag bit, write0invalid
Bit 5	NSSIF:SPIChip select change interrupt flag (only supported in slave mode)
	0: The chip select signal has not changed
	1: The chip select signal changes
	software write1clear the flag bit, write0invalid
Bit 4	IDIF:SPIidle interrupt flag bit (only supported in master mode)
	0: not in idle state
	1: enter idle state
	software write1clear the flag bit, write0Invalid; or software write registerSPITBWclear flag
Bit 3	ROIF:SPIREceive data overflow interrupt flag
	0: not overflowed
	1:overflow
	software write1clear the flag bit, write0invalid
Bit 2	TEIF:SPITransmit error interrupt flag (only supported by slave mode)
	0: No send error occurred

1: A send error occurs: When the send buffer and send shift register are all empty, the communication clock provided by the master is received

software write1clear the flag bit, write0invalid

Bit 1 RBIF:SPIReceive buffer full interrupt flag

0: Receive buffer is not full

1: receive buffer full

readSPIRBRRegister Clearable Interrupt Flag Bits

Bit 0 TBIF:SPITransmit buffer empty interrupt flag

0: Send buffer is not empty

1: send buffer empty

WriteSPITBWRegister Clearable Interrupt Flag Bits

Note1:SPIWhen the interrupt is disabled, if the condition is met, the corresponding interrupt flag bit will still be set, but no interrupt request will be generated.

Note2:rightSPIIEach interrupt flag bit in the register, write0void, write1To clear the flag bit; when reading, the read value is1Indicates that there is an interruption

occur.

## 6.7A/D Converter (ADC)

### 6.7.1

#### overview

An analog-to-digital converter is used to convert an analog signal into a digital signal composed of a set of binary codes. The analog signal is input via a multiplexed input pin and connected to the input of the converter through a sample-and-hold circuit.

chip support12-bit,8+1of channelsA/Dconverter, throughA/Dconverted by the converter12-bitBinary data is stored inADCdata registerADCRH,ADCRLmiddle.

#### -A/D Converter Characteristics

- 12bitA/DSampling accuracy
- 8+1optional analog input channels,8external channels and1internal channel
- 12Bit conversion result, supports high-order aligned placement or low-order aligned placement
- ConfigurableA/Dsampling time
- Multiple Conversion Clock Frequencies Available
- A variety of reference sources can be configured. When using an external reference voltage, the reference voltage cannot be lower than3.0V@VDD =5.0V,  
**2.5V@ VDD=3.3V**

#### -main functional components

- ADCConversion Value Register (ADCRL,ADCRH)
- ADCControl Register (ADCCL,ADCCH)
- Digital-to-Analog Port Control Register (ANSL)

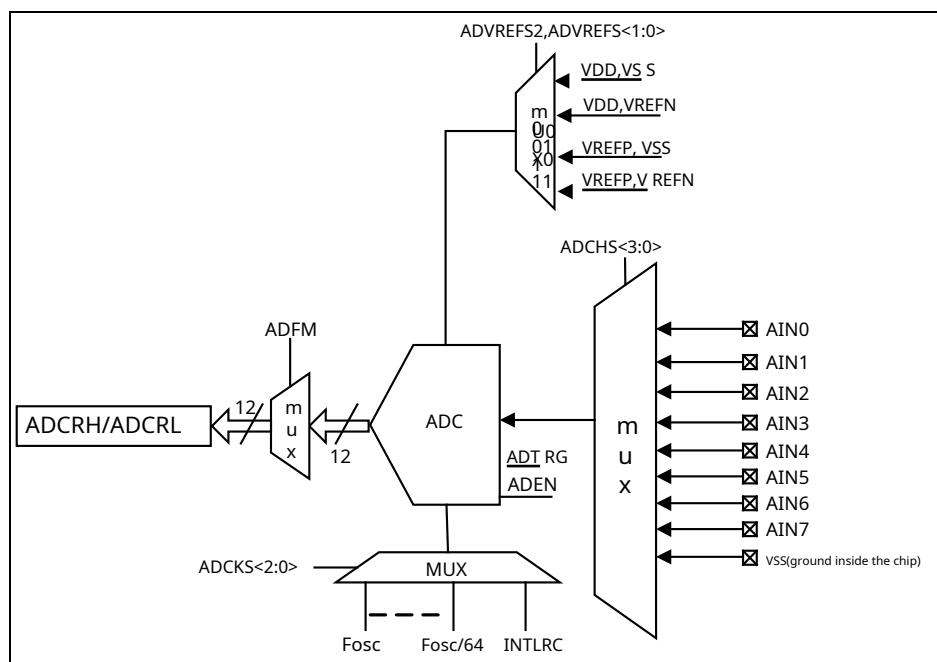
#### -ADCConversion supports the following launch methods

- set registerADCCLofADTRG=1(SMPS=1) to initiate conversion
- set registerADCCLofSMPON=1(SMPS=0) to initiate conversion
- External InterruptPINT5(ADC\_ETR0/PA7) to start conversion, need to setADC\_ETR0EN=1
- External InterruptPINT8(ADC\_ETR1/PB1) to start conversion, need to setADC\_ETR1EN=1
- T21The comparison match of the module triggers the conversion, and the register needs to be setT21CLOfT21M=1011
- T31The module's update event or compare match triggers the conversion, which can be set by the registerT31C1LofADTRGSbit  
set

#### -break and pause

- supportADconversion interrupt (ADIE/ADIF)
- existIDLEmode,A/Dconversion pause

## 6.7.2 Internal structure map



picture6-58 ADCsInternal structure map

## 6.7.3 ADCs configuration

ADC Before the circuit is used, the following aspects should be correctly configured as required to obtain correct conversion results.

### clock selection

ADC The conversion clock of the circuit has 8 group optional, Fosc~Fosc/64 or INTLRC, accessible ADCCH register ADCKS<2:0> bits to select the desired clock.

### Reference voltage selection

ADC The circuit can choose to use external reference voltage input, which are reference voltage positive polarity input and reference voltage negative polarity input respectively, and the corresponding external reference voltage input pins are respectively VREFP and VREFN, accessible ADCCH in the register ADVREFS <1:0> bit and PWEN in the register ADVREFS2 common choice.

### Sampling time selection

ADC The sampling time of the circuit can be set by ADCCH in the register ADST<1:0> bit selection, the sampling time has 2 individual Tadclk, 4 individual Tadclk, 8 individual Tadclk as well as 16 individual Tadclk four options. If the conversion signal jumps relatively large, it is recommended to set the sampling time to a longer gear, such as 8 individual Tadclk or 16 individual Tadclk.

### Sampling mode selection and control

This chip ADC two modes of software sampling and hardware sampling can be selected, through ADCCL in the register SMPSS bit selection. When software sampling is selected, it can be ADCCL in the register SMPON Bits control the start and stop of sampling.

### Multiplexing port type selection

In the chip ADC all analog input channels of the circuit AINx, reference voltage external input pin and PA/PB Port multiplexing, in use ADC Before converting the circuit, the used pins must be passed through ANSL The register is set to analog type.

### Analog signal input channel selection

ADC Before the circuit is enabled, it is necessary to select A/D Analog channel. This chip ADC circuit support 8 external channels

AIN0~AIN7,1 internal channelVSS.A/DAnalog channels are available throughADCCLin the registerADCHS<3:0> bit selection. select internalVSSAs an input channel, detectableADCsmall signaloffset.

#### Alignment selection

This chipADCThe result of circuit conversion supports two alignment methods, low-order alignment and high-order alignment, which can be passedADCCH in the registerADFMbit to select.

### 6. 7. 4 ADCconversion steps

The following outlines the implementationADCthe various steps in the conversion processIn practical applications, it is also necessary to considerADCown work establishment (see see registerADCCLThe following remarks describe the content) .

Step1:chooseADCConvert the clock byADCCHin the registerADCKS<2:0>chooseADCConvert clock. ADCThe conversion clock frequency is recommended to be set at512KHz~2MHzbetween.

Step 2:chooseADCreference voltage source, throughADCCHin the registerADVREFS <1:0>bit andPWEN in the registerADVREFS2selected together.

Step 3:chooseADCsampling time, throughADCCHin the registerA/DSampling Time Selection BitsADST<1:0> set up. It is generally recommended to set8indivualTadclk.

Step 4:chooseADCsampling mode, throughADCCLin the registerA/DSampling Mode Select BitsSMPSSelect Software Sampling or Hardware Sampling.

Step 5: Set the multiplexing port as analog type, that is, which pins are selected asADCconversion input pin, controlled by the port digital-to-analog registerANSLControl selection.

Step 6: Select the analog signal input channelAINx,passADCCLin the registerADCHS<3:0>chooseADC Analog channel.

Step7: Set the alignment of the conversion result, throughADCCHin the registerADFMBit, select high-order alignment or low-order alignment.

Step 8: If interrupts are to be used, the interrupt control registers need to be set correctly to ensureA/DThe interrupt function is correctly activated. In the default interrupt mode, it is necessary to set the global interrupt enable bit/high priority interrupt enable bitGIEset "1",WillADC interrupt enable location"1";In vectored interrupt mode, the global interrupt enable bit/high priority interrupt enable bitGIEplace "1",according to A/DThe priority of the interrupt group determines whether to enable the low priority interrupt enable bitGIEL,WillADC interrupt enable location"1".

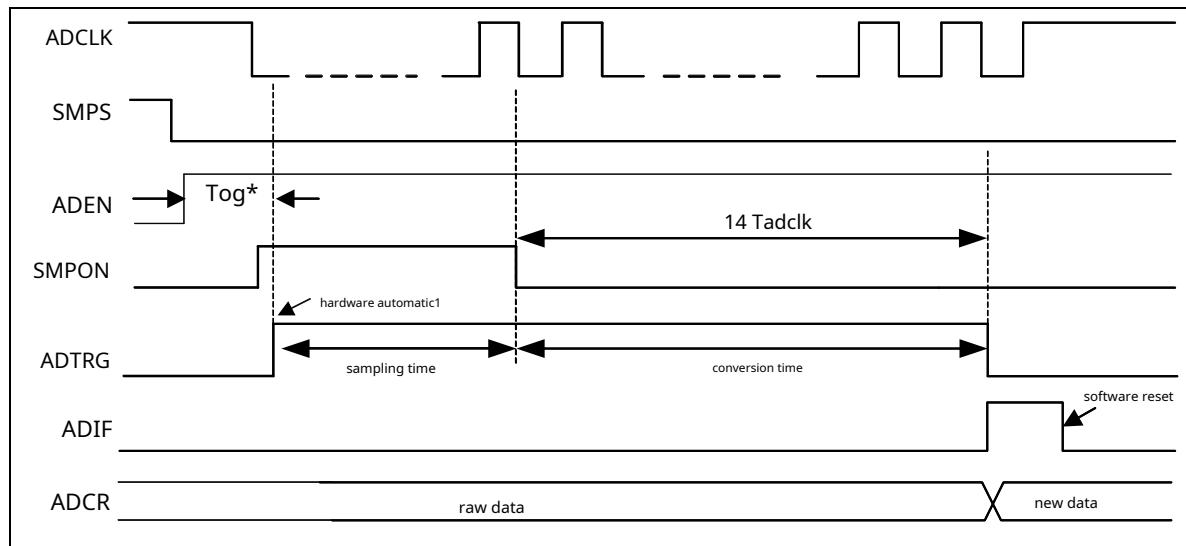
Step 9:EnableADCcircuit, willADCCLin the registerADCenable bitADENSet as"1".

Step 11:whenADCCLin the registerSMP=0, select software sampling mode, setADCCLin the registerSMPON=1 start sampling,ADCCLin the registerADTRGbit hardware auto-set1;whenSMP=1When the hardware sampling mode is selected, theADCconversion enable bitADTRGbit is set to "1",startADCconvert.

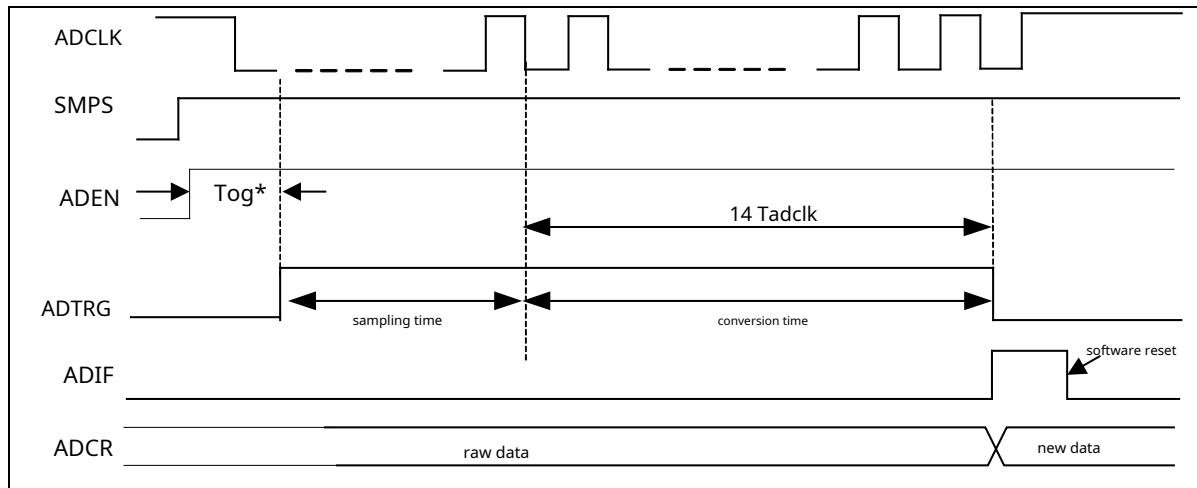
Step 10:pollingADCCLConversion Status Bits in RegisterADTRGbit, confirm this timeA/DWhether the conversion is complete.

Step 11: readADCRHandADCRLConversion result in register.

## 6. 7. 5 ADSchematic diagram of timing characteristics



picture6-59 ADCsTiming Characteristic Schematic (SMPS=0)



picture6-60 ADCsTiming Characteristic Schematic (SMPS=1)

Note1:Tog > 80us;

Note2:ADConversion Clock PeriodTadclk,accessibleADCKS<2:0>registers to configure different frequencies.

## 6. 7. 6 Reference routine

application routine1: For analog input channels0 (AIN0)Perform analog-to-digital conversion

```

...
BCC      ADCCH, ADFM          ;The conversion result is placed high-order aligned
MOVI     0x05                  ;hardware controlADCsampling mode
MOVA     ADCCL                ;EnableADCConverter, selected channel0;
...
BSS      ADCCL, ADTRG         ;delay waiting80us
AD_WAIT:

```

```

JBC      ADCCL, ADTRG           ;waitADCconversion complete
GOTO    AD_WAIT
MOV      ADCRH, 0              ;read high8bit conversion result
...
MOV      ADCRL, 0              ;read low4bit conversion result
...

```

### 6.7.7 special function register

ADCThe function is controlled by four control registers and two data registers. inADCRLandADCRHRegisters are used to storeADC Transformed data result, the result alignment is given byADCCHin the registerADFMbit control selection;ADCCLRegisters are used for ADCModule enable control,ADCsampling mode selection,ADCconversion launch control andADCAnalog channel selection, etc.;ADCCH Registers are used forADCSampling time selection, positive and negative reference voltage selection,ADCClock selection and result alignment selection, etc.;ANSLThe register is used to control the digital-to-analog type of the multiplexed port.

#### 6.7.7.1 ADCConversion Value Register (ADCR)

ADFM	ADCRH:ADCConversion Value Register High8bit								ADCRL:ADCConversion Value Register Low8bit							
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
1	—	—	—	—	ADCR<11:8>				ADCR<7:0>							
0	ADCR<11:4>								ADCR<3:0>			—	—	—	—	—

ADCR<11:0>:A/Dconversion result

#### 6.7.7.2 ADCControl Register (ADCCL)

ADCCL:ADCControl register								
bit	7	6	5	4	3	2	1	0
name	ADCHS<3:0>				SMPON	SMPS	ADTRG	ADEN
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	1	1	1	1	0	1	0	0

Bit 7~4      ADCHS<3:0>:A/DAnalog Channel Select Bits  
0000:aisle0(AIN0) 0001  
:aisle1(AIN1) 0010:aisle  
2(AIN2) 0011:aisle3(  
AIN3) 0100:aisle4(AIN4)  
0101:aisle5(AIN5) 0110  
:aisle6(AIN6) 0111:aisle  
7(AIN7) 1000: reserved  
for unused

1001~1110:VSS(Ground inside the chip) 1111:  
Shield channel selection

Bit 3      SMPON:A/DSampling Software Control Bits  
0: end sampling  
1: start sampling

Bit 2      SMPS:A/DSampling Mode Select Bits  
0: Enable software sampling, disable hardware sampling

1: Disable software sampling, enable hardware sampling

Bit 1 ADTRG:A/DSample Conversion Status Bits

0:A/DNo conversion is done, orA/DSample conversion completed 1:

A/DSample conversion in progress

whenSMPS=1, this bit is set by the software1start upA/Dsample conversion; whenSMPS=0hour,SMPON=1Start sampling, this bit is automatically set by hardware1.

Bit 0 ADEN:A/Dconversion enable bit

0:closureA/Dconverter

1:runA/Dconverter

Note1:existADENAAfter enabling,ADCYou need to complete the establishment of your own work before you can get the correct conversion result. ADCThe circuit operation is established as,

ADENAAfter enabling, delay80usAbove, start the first timeADCconvert(ADTRG=1) , after the conversion, and then delay30usabove,ADC

The work is established, and the follow-up startADCConvert, you can get the correct conversion result. For applications, theADENAAfter enabling, the first time ADCBefore and after conversion, add at least80usand30usdelay, whileADCThe conversion result obtained during the establishment process deviates greatly from the theoretical value and is unpredictable, so it needs to be discarded in the application programADENThe first conversion result after enabling.

Note2: because every timeADENAAfter re-enabling, you need to execute the aboveADCThe work establishment process, so in the application, it is not recommended to close the chip when the chip is running normally

closeADC,KeepADEN=1, only when enteringIDLEBefore sleep mode, you can turn off theADC, reducing sleep power consumption.

### 6. 7. 7. 3 ADCControl Register (ADCCH)

ADCCH:ADCcontrol register								
bit	7	6	5	4	3	2	1	0
name	ADFM	ADCKS<2:0>			ADST<1:0>		ADVREFS<1:0>	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	1	0	0	1	0	0	0

Bit 7 ADFM:A/DConversion data placement format selection bits

0: high-aligned (ADCRH<7:0>, ADCRL<7:4>) 1: little

aligned (ADCRH<3:0>, ADCRL<7:0>) ADCKS<2:0>:A/D

Bit 6~4 conversion clock frequency (Tadclk) option bit

000:Fosc

001:Fosc/2

010:Fosc/4

011:Fosc/8

100:Fosc/16

101:Fosc/32

110:Fosc/64

111:INTLRC(32KHz WDT RCclock) ADST<1:0>:A/

Bit 3~2 DHardware sample time selection bits

00:About2indivualTadclk 01

:About4indivualTadclk 10

:About8indivualTadclk 11

:About16indivualTadclk

Bit 1~0 ADVREFS<1:0>:A/DReference source selection bit, this bit needs to be combined with the registerADVREFS2(PWEN<7>) at the same time to select the correct reference source

ADVREFS2	ADVREFS<1:0>	Reference source selection
0	00	The positive terminal of the reference voltage is VDD, the negative terminal is VSS
1	01	The positive terminal of the reference voltage is VDD, the negative terminal is VREFN
0	10	The positive terminal of the reference voltage is an external VREFP, the negative terminal is VSS
1	11	The positive terminal of the reference voltage is an external VREFP, the negative terminal is an external VREFN

Note1: if in A/D During the conversion process, the conversion clock is switched, and the first time after the switch A/D There may be errors in the conversion results;

Note2: A/D The conversion clock frequency is recommended not to be greater than 2MHz;

Note3: When using an external reference voltage, the reference voltage cannot be lower than 3.0V@VDD = 5.0V, 2.5V@ VDD = 3.3V, otherwise it would result in ADC different job often.

#### 6. 7. 7. 4 Port Digital-to-Analog Control Register (ANSL)

ANSL: Port digital-analog control register								
bit	7	6	5	4	3	2	1	0
name	ANSL7	ANSL6	ANSL5	ANSL4	ANSL3	ANSL2	ANSL1	ANSL0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

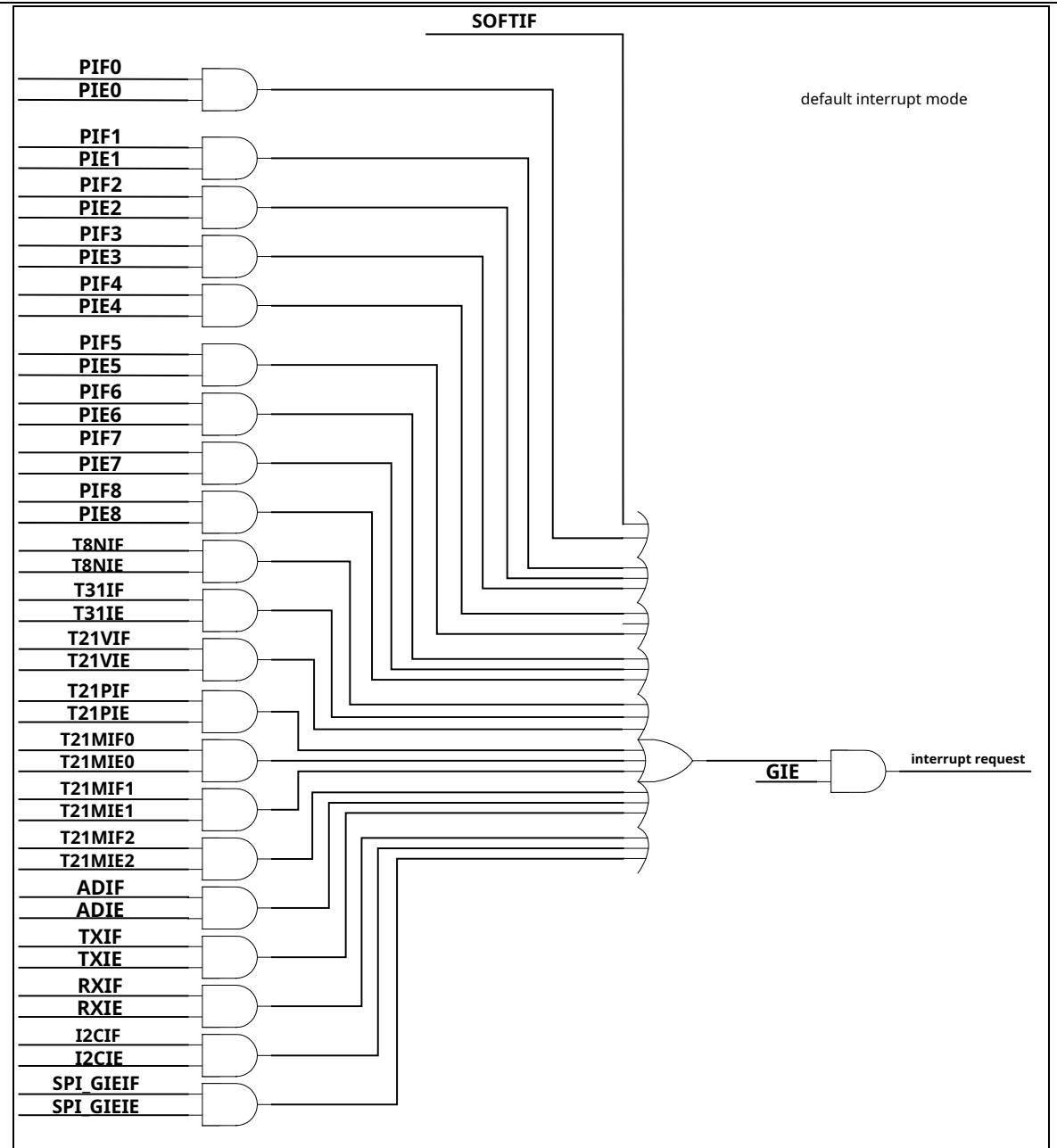
- Bit 7 ANSL7:PA7/AIN7Port D/A selection bit
  - 0: Analog input port
  - 1: Digital input/output port
- Bit 6 ANSL6:PB7/AIN6Port D/A selection bit
  - 0: Analog input port
  - 1: Digital input/output port
- Bit 5 ANSL5:PB3/AIN5Port D/A selection bit
  - 0: Analog input port
  - 1: Digital input/output port
- Bit 4 ANSL4:PA5/AIN4Port D/A selection bit
  - 0: Analog input port
  - 1: Digital input/output port
- Bit 3 ANSL3:PA4/AIN3Port D/A selection bit
  - 0: Analog input port
  - 1: Digital input port
- Bit 2 ANSL2:PA2/AIN2Port D/A selection bit
  - 0: Analog input port
  - 1: Digital input/output port
- Bit 1 ANSL1:PA1/AIN1Port D/A selection bit
  - 0: Analog input port
  - 1: Digital input/output port
- Bit 0 ANSL0:PA0/AIN0Port D/A selection bit
  - 0: Analog input port
  - 1: Digital input/output port

## No.7 Chapter Interrupt Handling

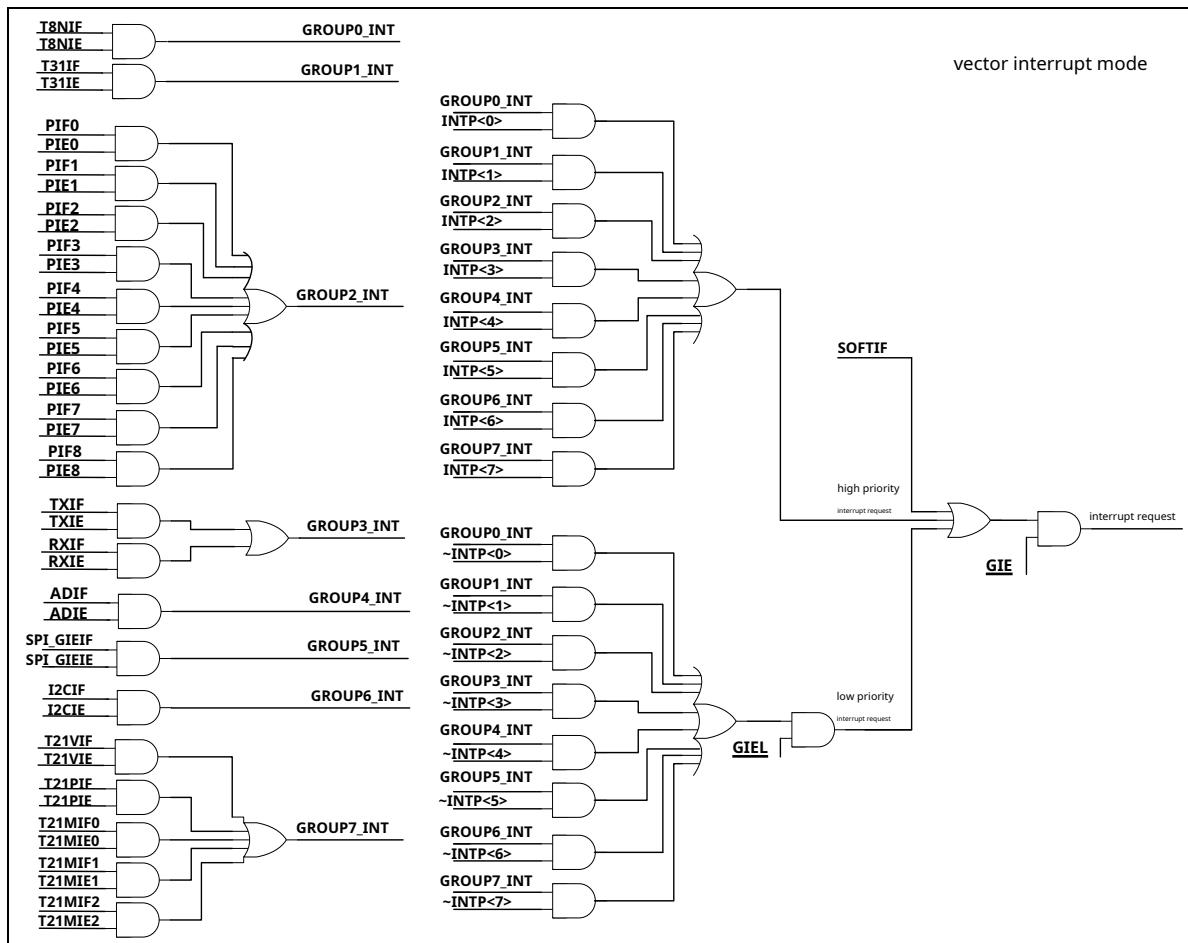
## 7.1 overview

Interrupt is an important function of the chip. It can wake up the chip from sleep mode, and can also make the system respond to unexpected events during normal operation, suspend and save the information of the current running program, and jump to the entrance of the request interrupt service program. address, execute the corresponding interrupt service routine, and handle emergencies. There are two interrupt modes supported by this chip: default interrupt mode and vector interrupt mode, which can support up to twenty two interrupt sources: one software interrupt and twenty one hardware interrupt.

## 7.2 Block Diagram of Interrupt Control Structure



picture7-1Default Interrupt Mode Interrupt Control Logic



picture7-2Vectored Interrupt Mode Interrupt Control Logic

### 7.3 Interrupt mode selection

The chip supports two interrupt modes, default interrupt mode and vector interrupt mode, which can be INTG in the register INTVEN0 bits and Chip Configuration Words in the INTVEN1 bit selection. have to be aware of is, INTVEN0 and INTVEN1 Only at the same time "1", the vectored interrupt mode is valid.

INTVEN0 (INTG<2>)	INTVEN1 (CFG_WD0<11>)	interrupt mode
0	0	default interrupt mode
0	1	
1	0	
1	1	vector interrupt mode

surface7-1Interrupt Mode Selection Table

The default interrupt mode only supports 1 interrupt entry, namely 0004h Entry address, does not support interrupt priority and interrupt nesting. Vectored interrupt mode supports multiple interrupt entries, and supports interrupt priority and interrupt nesting.

#### 7.3.1 default interrupt mode

When configured as the default interrupt mode, the entry addresses of all interrupt vectors are located at 0004h. The user needs to judge each interrupt flag and interrupt enable bit through the interrupt service program, confirm the interrupt source that causes the interrupt operation, and then execute the corresponding

Interrupt service routine. This mode does not support interrupt priority configuration.

serial number	interrupt source	interrupt name	interrupt flag	interrupt enable	global enable	Remark
1	soft interrupt	soft interrupt	SOFTIF	—	GIE	software settings1
2	External Interrupt	PINT0	PIF0	PIE0	GIE	—
3		PINT1	PIF1	PIE1	GIE	—
4		PINT2	PIF2	PIE2	GIE	—
5		PINT3	PIF3	PIE3	GIE	—
6		PINT4	PIF4	PIE4	GIE	—
7		PINT5	PIF5	PIE5	GIE	—
8		PINT6	PIF6	PIE6	GIE	—
9		PINT7	PIF7	PIE7	GIE	—
10		PINT8	PIF8	PIE8	GIE	—
11	T8NTimer/Counter Overflow Out of interrupt	T8NINT	T8NIF	T8NIE	GIE	—
12	T31total interruption	T31INT	T31IF	T31IE	GIE	—
13	T21Timer overflow interrupt	T21VINT	T21VIF	T21VIE	GIE	—
14	T21Cycle Match Interrupt	T21PINT	T21PIF	T21PIE	GIE	—
15	T21capture/compare interrupt0	T21MINT0	T21MIFO	T21MIE0	GIE	—
16	T21capture/compare interrupt1	T21MINT1	T21MIF1	T21MIE1	GIE	—
17	T21capture/compare interrupt2	T21MINT2	T21MIF2	T21MIE2	GIE	—
18	ADCto interrupt	ADINT	ADIF	ADIE	GIE	—
19	UART TXto interrupt	TXINT	TXIF	TXIE	GIE	—
20	UART RXto interrupt	RXINT	RXIF	RXIE	GIE	—
twenty one	I2Ctotal communication interruption	I2CINT	I2CIF	I2CIE	GIE	—
twenty two	SPItotal communication interruption	SPIINT	SPI_GIE IF	SPI_GIEI E.	GIE	—

surface7-2Default interrupt mode enable configuration table

### 7.3.2 vector interrupt mode

#### 7.3.2.1 Vector table configuration

When configured as vector interrupt mode, each interrupt source is divided into groups, and each group of interrupts corresponds to an interrupt vector entry address. The soft interrupt entry address is 0004h, the highest priority; other hardware interrupts are grouped (IG0-IG7), through the interrupt global register INTGmiddleINTV<1:0>. The configuration supports different vector table prioritization, and corresponds to an interrupt entry address. Each group of hardware interrupts can be set with high and low priority respectively to respond to interrupt nesting. Through the interrupt priority register INTPmiddleIGPxConfiguration, divide all hardware interrupt sources into high and low priority arbitration areas. According to INTV<1:0> The setting of the hardware interrupt group in the arbitration area is prioritized, and the highest priority is responded to. The high and low priority arbitration areas are respectively controlled by the high priority interrupt enable bit GIE and low priority interrupt enable bits GIEL to enable. When executing low-priority interrupt service routines, high-priority interrupt groups can be nested to respond.

priority		0(high )	1	2	3	4	5	6	7	8(Low )
entry address		0004h	0008h	000Ch	0010h	0014h	0018h	001Ch	0020h	0024h
INTV	00	soft interrupt	IG0	IG1	IG2	IG3	IG4	IG5	IG6	IG7
	01		IG0	IG1	IG6	IG7	IG4	IG5	IG2	IG3
	10		IG4	IG5	IG2	IG3	IG0	IG1	IG6	IG7
	11		IG7	IG6	IG5	IG4	IG3	IG2	IG1	IG0

surface7-3Vector table configuration table

### 7.3.2.2

Interrupt Group Configuration

serial number	interrupt group number	high and low priority selection	interrupt name	Remark
1	IG0	IGP0	T8NINT	—
2	IG1	IGP1	T31INT	—
3	IG2	IGP2	PINT0	—
4			PINT1	—
5			PINT2	—
6			PINT3	—
7			PINT4	—
8			PINT5	—
9			PINT6	—
10			PINT7	—
11			PINT8	—
12	IG3	IGP3	TXINT	—
13			RXINT	—
14	IG4	IGP4	ADINT	—
15	IG5	IGP5	SPIINT	—
16	IG6	IGP6	I2CINT	—
17	IG7	IGP7	T21VINT	—
18			T21PINT	—
19			T21MINT0	—
20			T21MINT1	—
twenty one			T21MINT2	—

surface7-4Vector interrupt mode interrupt grouping configuration table

### 7.3.2.3

Interrupt Enable Configuration

serial number	interrupt source	interrupt name	interrupt flag	interrupt enable	IGPx	low priority medium enable bit	global interrupt enable bit	Remark	
1	soft interrupt	soft interrupt	SOFTIF	—	—	—	GIE	software settings1	
2	External Interrupt	PINT0	PIFO	PIE0	0	GIEL	GIE	—	
3					1	—	GIE	—	
3		PINT1	PIF1	PIE1	0	GIEL	GIE	—	
					1	—	GIE	—	

serial number	interrupt source	interrupt name	interrupt flag	interrupt enable	<b>IGPx</b>	low priority medium enable bit	global interrupt enable bit	Remark
4		PINT2	PIF2	PIE2	0	GIEL	GIE	—
5		PINT3	PIF3		1	—	GIE	—
6		PINT4	PIF4	PIE4	0	GIEL	GIE	—
7		PINT5	PIF5		1	—	GIE	—
8		PINT6	PIF6	PIE6	0	GIEL	GIE	—
9		PINT7	PIF7		1	—	GIE	—
10		PINT8	PIF8	PIE8	0	GIEL	GIE	—
11	T8Ntimer/meter counter overflow interrupt	T8NINT	T8NIF		1	—	GIE	—
12	T31total interruption	T31INT	T31IF	T31IE	0	GIEL	GIE	—
13	T21timer overflow Out of interrupt	T21VINT	T21VIF		1	—	GIE	—
14	T21cycle matching to interrupt	T21PINT	T21PIF	T21PIE	0	GIEL	GIE	—
15	T21capture/compare to interrupt0	T21MINT0	T21MIFO		1	—	GIE	—
16	T21capture/compare to interrupt1		T21MIE0	0	GIEL	GIE	—	
17	T21capture/compare to interrupt2	T21MINT2		T21MIF2		1	—	GIE
18	ADCto interrupt	ADINT	ADIF	ADIE	0	GIEL	GIE	—
19	UART TXto interrupt	TXINT	TXIF		1	—	GIE	—
20	UART RXmiddle broken	RXINT	RXIF	RXIE	0	GIEL	GIE	—
twenty one	I2CSCommunication Center broken	I2CINT	I2CIF		1	—	GIE	—
twenty two	SPICommunication Center broken	SPIINT	SPI_GIEIf	SPI_GIEIE.	0	GIEL	GIE	—
					1	—	GIE	—

surface7-5Vector interrupt mode enable configuration table

## 7.4 Interrupt Context Saving

Interrupt context saving is a very important part of the interrupt service routine.

In the command system there are PUSH(push) and POP(Pop stack) instruction, which can conveniently save and restore the current working state.A,PSW,PCRHandBKSRegisters, each with its own two-stage mirror registerAS1,PSWS1,PCRHS1,BKSRS1and AS0,PSWS0,PCRHS0,BKSRS0, used to save and restore the corresponding register. Mirror registers have no physical address, they can only be accessed by PUSH and POP. The instruction automatically completes the corresponding save and restore actions, and the two-level mirror register adopts the stack operation mode.

## 7.5 Interrupt operation

### 7.5.1 interrupt enable bitGIEandGIELoperation

If an interrupt event condition occurs, the related interrupt flag will be set to "1". After the interrupt flag is generated, the program will jump to the corresponding service program address for execution, and the following conditions must be met:

- 1) When the corresponding interrupt enable bit is "1", continue to judge whether the second condition is met; when the corresponding interrupt enable bit is "0", even if the interrupt flag is "1", the interrupt will not occur, and the program will not jump to the interrupt service routine address for execution.
- 2) In the default interrupt mode, when the global interrupt enable bitGIEfor"0", all interrupt requests are masked. When the global interrupt enable bitGIEfor"1", the program will jump to the interrupt service routine address for execution. In vectored interrupt mode, when the global interrupt enable bitGIEfor"0", all interrupt requests are masked. When the global interrupt enable bitGIEfor"1", if the group corresponding to the interrupt is of high priority, the program will jump to the address of the interrupt service routine for execution; if the group of the corresponding interrupt is of low priority, when the low priority interrupt enable bitGIELfor"1" When there is no high-priority interrupt request, the program will jump to the interrupt service address for execution. When the low-priority interrupt enable bitGIELfor"0", all low-priority interrupt requests are masked.

To ensure that the registerGIEandGIELIf the software write operation is successful, follow the steps below:

- 1) In default interrupt mode or vectored interrupt mode, the GIEbit software clear0operation, you need to turn off all peripheral interrupt enable first, and then set GIEbit clear0; or at GIEbit clear0After operation, query GIEIs the bit0, not for 0 then proceed to clear0operate until successful;  
right GIEbit software setting1operation, no special requirements, it is recommended to enable the required peripheral interrupt first, and then enable GIE Location1.
- 2) In vectored interrupt mode, the GIELbit software clear0operation, you need to turn off all peripheral interrupt enable first, and then set GIEL bit clear0; or at GIELbit clear0After operation, query GIELIs the bit0, not for 0 then proceed to clear0operate until successful;  
right GIELbit software setting1operation, with GIEPosition at the same time1, or first GIELLocation1, and then GIELLocation 1.

### 7.5.2 External Interrupt

when PINTxThe multiplexing port is configured as a digital input port, and when the input signal change meets the trigger condition, it will generate PINTxExternal port interrupt, the corresponding interrupt flag PIFx was placed "1". When the global interrupt control bitGIEand external port interrupt control bitsPIEx are set to "1" when, then to CPU issue PINTxExternal port interrupt request. When the interrupt condition is allowed, the system will enter the corresponding interrupt service program entry address to process the interrupt program.

It should be noted that the corresponding interrupt flag bit PIFx and interrupt enable bits PIEx need to be cleared by software, INTC0The register is used to configure the trigger conditions, which can be configured as rising edge trigger, falling edge trigger or double edge trigger respectively.

### 7.5.3 ADCsto interrupt

ADCinterrupted byADCSwitch motion controls whenADCWhen the conversion is complete, theADCinterruption,ADCinterrupt flagADIFwas placed "1". whenADCinterrupt control bitADIEset to "1", and the global interrupt control bitGIEand low priority interrupt enable bitsGIELWhen properly enabled according to the interrupt mode, theCPUissueADCinterrupt request.CPURespond to the current interrupt request according to the priority of the interrupt, whenADCWhen the interrupt condition is allowed, the system will enter the entry address of the corresponding interrupt service program to process the interrupt program. have to be aware of is,ADCinterrupt flagADIFand interrupt enable bitsADIEBoth need to be cleared by software.

### 7.5.4 T8Noverflow interrupt

8Bit Timer/CounterT8NIn Timer mode or Counter mode, whenT8NThe counter is incremented by theFFhbecomes00h hour, T8NThe counter overflows, will interrupt flagT8NIFLocation"1". whenT8NOverflow interrupt enable bitT8NIE set to "1", and the global interrupt control bitGIEand low priority interrupt enable bitsGIELWhen properly enabled according to the interrupt mode, theCPUissue T8NOverflow interrupt request.CPURespond to the current interrupt request according to the priority of the interrupt, whenT8N When the overflow interrupt condition is allowed, the system will enter the corresponding interrupt service program entry address to process the interrupt program. have to be aware of is, T8NOverflow interrupt flagT8NIFand interrupt enable bitsT8NIEBoth need to be cleared by software.

### 7.5.5 T21overflow interrupt

16bit timerT21The overflow interrupt can be generated in various working modes:

#### Timer Mode/Multiple PrecisionPWMmodel

16bit timerT21In Timer Mode/Multiple PrecisionPWMIn the mode, count up the count clock, whenT21 When the count value of the postscaler is the same as the division ratio of the postscaler, an overflow interrupt will be generated.

#### Capture Mode/Comparator Mode

16bit timerT21In capture mode/comparator mode, count up the count clock whenT21When the count value overflows (i.e. fromFFFFhbecomes0000h) , an overflow interrupt is generated.

T21When an overflow interrupt occurs, set the interrupt flagT21VIFLocation"1". whenT21Overflow interrupt enable bitT21VIESet as "1", and the global interrupt control bitGIEand low priority interrupt enable bitsGIELWhen properly enabled according to the interrupt mode, theCPUissueT21 Overflow interrupt request.CPURespond to the current interrupt request according to the priority of the interrupt, whenT21When the overflow interrupt condition is allowed, the system will enter the corresponding interrupt service program entry address to process the interrupt program. have to be aware of is, T21overflow interrupt flagT21VIFand interrupt enable bitsT21VIEBoth need to be cleared by software.

### 7.5.6 T21cycle interruption

16bit timerT21in multiple precisionPWMmode,T21Counting up from zero, whenT21andT21PWWhen the values of the registers are equal, theT21 cycle interrupt, interrupt flagT21PIFwas placed "1". If the interrupt enable bitT21PIE set to "1", and the global interrupt control bitGIEand low priority interrupt enable bitsGIELWhen properly enabled according to the interrupt mode, theCPUissueT21Periodic interrupt request.CPURespond to the current interrupt request according to the priority of the interrupt, whenT21 When the periodic interrupt condition is allowed, the system will enter the corresponding interrupt service program entry address to process the interrupt program. have to be aware of is, T21Periodic Interrupt FlagT21PIFand interrupt enable bitsT21PIEBoth need to be cleared by software.

### 7.5.7 T21multifunction interrupt

16bit timerT21Multi-function interrupt can be generated when in capture mode/comparator mode.

#### catcher mode

16bit timerT21When in catcher mode,T21count up, whenT21CI0/T21CI1/T21CI2lose

When the change state of the input signal meets the capture condition, the counter T21 will be loaded into the corresponding 16bit capture register T21R0/T21R1/T21R2, and generate the corresponding multi-function interrupt T21MIF0/T21MIF1/T21MIF2.

#### comparator mode

16bit timer T21 When in comparator mode, the T21Count up. when the counter T21The count value is compared with the compare register T21R0/T21R1/T21R2 When the comparison values in are equal, execute the corresponding comparison match event and generate the corresponding multi-function interrupt T21MIF0/T21MIF1/T21MIF2.

T21 When a multi-function interrupt occurs, the corresponding interrupt flag T21MIF0/T21MIF1/T21MIF2 Location "1". when the corresponding T21 Multi-function interrupt enable bit T21MIE0/T21MIE1/T21MIE2 set to "1", and the global interrupt enable bit GIE and low priority interrupt enable bits GIEL When properly enabled according to the interrupt mode, the CPU issue T21 multifunction interrupt 0/1/2 ask. CPU Respond to the current interrupt request according to the priority of the interrupt, when T21 multifunction interrupt 0/1/2 When conditions permit, the system will enter the corresponding interrupt service program entry address to process the interrupt program. have to be aware of is, T21 multi-function interrupt flag T21MIF0/T21MIF1/T21MIF2 and multi-function interrupt enable bit T21MIE0/T21MIE1/T21MIE2 Both need to be cleared by software.

### 7.5.8 T31 to interrupt

when BKIF, TRGIF, CHUIF, MIF4, MIF3, MIF2, MIF1, UPIF, OVIF4, OVIF3, OVIF2, OVIF1 Any interrupt flag position in 1, and its corresponding interrupt enable bit is also 1 hour, T31 total interrupt flag T31IF will set 1. if T31 Total interrupt enable bit T31IE place 1, and the global interrupt control bit GIE and low priority interrupt enable bits GIEL When properly enabled according to the interrupt mode, the CPU issue T31 interrupt request. CPU Respond to the current interrupt request according to the priority of the interrupt, when T31 When the interrupt condition is allowed, the system will enter the entry address of the corresponding interrupt service program to process the interrupt program. It should be noted that software clearing is required T31IF, but in cleared T31IF Before the total interrupt flag bit, it needs to be cleared BKIF, TRGIF, CHUIF, MIF4, MIF3, MIF2, MIF1, UPIF, OVIF4, OVIF3, OVIF2, OVIF1 and other related interrupt flags.

### 7.5.9 UART to interrupt

UART There are two types of interrupts: sending interrupts and receiving interrupts.

when UART Transmit Data Register for Asynchronous Transmitter TXB is empty, or when the asynchronous receiver completes a data reception, produces UART Send/receive interrupt, send/receive interrupt flag bit TXIF/RXIF is set to "1". If the transmit/receive interrupt enable bit TXIE/RXIE set to "1", and the global interrupt control bit GIE and low priority interrupt enable bits GIEL When properly enabled according to the interrupt mode, the CPU issue UART Send/receive interrupt request. CPU Respond to the current interrupt request according to the priority of the interrupt, when UART When the sending/receiving interrupt condition is allowed, the system will enter the corresponding interrupt service program entry address to process the interrupt program. It should be noted that the transmit/receive interrupt flag bit TXIF/RXIF It is read-only, cannot be cleared by software, read the receive data register RXB, can be cleared RXIF, write transmit data register TXB, can be cleared TXIF; Transmit/receive interrupt enable bit TXIE/RXIE Need to be cleared by software.

### 7.5.10 I2C to interrupt

when I2CSRIF, I2CSPIF, I2CTBIF, I2CRBIF, I2CTEIF, I2CROIF and I2CNAIF Any interrupt flag position in 1 hour, I2C total interrupt flag I2CIF will set 1. if I2C interrupt enable bit I2CIE set to "1", and the global interrupt control bit GIE and low priority interrupt enable bits GIEL When properly enabled according to the interrupt mode, the CPU issue I2C interrupt request. CPU Respond to the current interrupt request according to the priority of the interrupt, when I2C When the interrupt condition is allowed, the system will enter the entry address of the corresponding interrupt service program to process the interrupt program. It should be noted that it needs to be cleared by software I2CIF, but in cleared I2CIF Before the total interrupt flag bit, it needs to be cleared I2CSRIF, I2CSPIF, I2CTEIF, I2CROIF and I2CNAIF and other related interrupt flags.

### 7.5.11 SPIto interrupt

when TBWEIF, NSSIF, IDIF, ROIF, TEIF, RBIF and TBIF Any interrupt flag position in 1 hour, I2C total interrupt flag SPI\_GIEIF will set 1. if SPI interrupt enable bit SPI\_GIEIE set to "1", and the global interrupt control bit GIE and low priority interrupt enable bits GIEL When properly enabled according to the interrupt mode, the CPU issues SPI interrupt request. CPU responds to the current interrupt request according to the priority of the interrupt, when SPI When the interrupt condition is allowed, the system will enter the entry address of the corresponding interrupt service program to process the interrupt program. It should be noted that software clearing is required SPI\_GIEIF, but in cleared SPI\_GIEIF Before the total interrupt flag bit, it needs to be cleared TBWEIF, NSSIF, IDIF, ROIF, TEIF, RBIF and TBIF and other related interrupt flags.

### 7.5.12 Precautions for interrupt operation

The user needs to clear the corresponding interrupt flag before enabling the interrupt to avoid false triggering of the interrupt.

Except for read-only interrupt flags (cleared by hardware), the remaining interrupt flags must be cleared by software.

In order to avoid the conflict between the occurrence of an interrupt and the clearing operation of the interrupt flag, the clearing of the interrupt flag is unsuccessful. It is recommended that the user perform a software judgment on whether the interrupt flag is cleared successfully after performing the interrupt flag clearing operation. If the operation is unsuccessful, the interrupt flag clearing operation is performed again until the interrupt flag is cleared successfully. The user can also perform two consecutive interrupt flag clearing operations to achieve the same purpose.

## 7.6 special function register

The interrupt function is jointly controlled by a series of control registers and chip configuration words.

### 7.6.1 Interrupt Global Register (INTG)

INTG: Interrupt global register								
bit	7	6	5	4	3	2	1	0
name	GIE	GIEL	—	—	SOFTIF	INTVENO	INTV<1:0>	
R/W	R/W	R/W	—	—	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7            GIE: Global interrupt enable bit, or high priority interrupt enable bit

    0: Disable all interrupts, or disable high priority interrupts

    1: Enable all unmasked interrupts, or enable high priority interrupts GIEL:

Bit 6            Low priority interrupt enable bit (vectored interrupt mode)

    0: Disable low priority interrupt

    1: Enable low priority interrupt

Bit 5~4          Unused

Bit 3            SOFTIF: soft interrupt flag bit

    0: no soft interrupt

    1: There is a soft interrupt

Bit 2            INTVENO: Interrupt mode selection bit

    0: default interrupt mode

    1: Vectored Interrupt Mode (Chip Configuration Word INTVEN1(CFG\_WD0<11>) must be 1) INTV<1:0>: Interrupt

Bit 1~0          vector table selection bit, refer to vector table configuration

Note: Cleared by software GIE or GIEL position, need to judge GIE or GIEL whether the clearing is successful, if not cleared, you need to perform the software clearing operation again,

until clearing is successful. software set GIE and GIEL, it needs to be set first GIEL, then reset GIE, or both GIE and GIEL.

### 7.6.2 Interrupt Priority Register (INTP)

INTP: Interrupt Priority Register								
bit	7	6	5	4	3	2	1	0
name	IGP<7:0>							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7~0      IGP<7:0>:IG7-IG0Interrupt priority setting

0: low priority

1:high priority

### 7.6.3 Interrupt Control Register0(INTC0)

INTC0: Interrupt Control Register0								
bit	7	6	5	4	3	2	1	0
name	PEG3<1:0>		PEG2<1:0>		PEG1<1:0>		PEG0<1:0>	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7~6      PEG3<1:0>:PINT8~PINT6Trigger edge selection bit

00:PINT8~PINT6Falling edge trigger 01:PINT8~PINT6

Rising edge trigger 1x:PINT8~PINT6Double edge

trigger PEG2<1:0>:PINT5~PINT4Trigger edge

Bit 5~4      selection bit

00:PINT5~PINT4Falling edge trigger 01:PINT5~PINT4

Rising edge trigger 1x:PINT5~PINT4Double edge

trigger PEG1<1:0>:PINT3~PINT2Trigger edge

Bit 3~2      selection bit

00:PINT3~PINT2Falling edge trigger 01:PINT3~PINT2

Rising edge trigger 1x:PINT3~PINT2Double edge

trigger PEG0<1:0>:PINT1~PINT0Trigger edge

Bit 1~0      selection bit

00:PINT1~PINT0Falling edge trigger

01:PINT1~PINT0Rising edge trigger 1x

:PINT1~PINT0Double edge trigger

### 7.6.4 Interrupt Flag Register0(INTF0)

INTF0: Interrupt Flag Register0								
bit	7	6	5	4	3	2	1	0
name	SPI_GIEIF	—	—	—	PIF8	T31IF	T8NIF	ADIF
R/W	R/W	—	—	—	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7      SPI\_GIEIF:SPIModule total interrupt flag bit

0:Yet to happenSPIto interrupt

1:occurSPIInterrupt (must be cleared by software)

Bit 6~4	Reserved
Bit 3	PIF8: External port interrupt8flag bit 0: external portPINT8No interrupt signal on 1: external portPINT8There is an interrupt signal (must be cleared by software) T31IF:T31total interrupt flag
Bit 2	0:T31No interruption occurred 1:T31An interrupt occurred (must be cleared by software) T8NIF:T8Noverflow interrupt flag
Bit 1	0:T8Ncount did not overflow 1:T8NCount overflow (must be cleared by software) ADIF:ADCinterrupt flag
Bit 0	0: in progressADconvert 1:ADconversion has completed (must be cleared in software)

### 7.6.5 interrupt enable register0(INTEO)

INTE0: Interrupt Enable Register0								
bit	7	6	5	4	3	2	1	0
name	SPI_GIEIE	—	—	—	PIE8	T31IE	T8NIE	ADIE
R/W	R/W	—	—	—	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7	SPI_GIEIE:SPIModule total interrupt enable bit 0:prohibit 1:Enable
Bit 6~4	Reserved
Bit 3	PIE8: External port interrupt8enable bit 0:prohibit 1:Enable
Bit 2	T31IE:T31Total interrupt enable bit 0:prohibit 1:Enable
Bit 1	T8NIE:T8NOverflow interrupt enable bit 0:prohibit 1:Enable
Bit 0	ADIE:ADCinterrupt enable bit 0:prohibit 1:Enable

### 7. 6. 6 Interrupt Flag Register1(INTF1)

INTF1: Interrupt Flag Register1								
bit	7	6	5	4	3	2	1	0
name	PIF7	PIF6	PIF5	PIF4	PIF3	PIF2	PIF1	PIFO
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0
Bit 7	PIF7: External port interrupt7flag bit 0: external portPINT7No interrupt signal on 1: external portPINT7There is an interrupt signal (must be cleared by software)							
Bit 6	PIF6: External port interrupt6flag bit 0: external portPINT6No interrupt signal on 1: external portPINT6There is an interrupt signal (must be cleared by software)							
Bit 5	PIF5: External port interrupt5flag bit 0: external portPINT5No interrupt signal on 1: external portPINT5There is an interrupt signal (must be cleared by software)							
Bit 4	PIF4: External port interrupt4flag bit 0: external portPINT4No interrupt signal on 1: external portPINT4There is an interrupt signal (must be cleared by software)							
Bit 3	PIF3: External port interrupt3flag bit 0: external portPINT3No interrupt signal on 1: external portPINT3There is an interrupt signal (must be cleared by software)							
Bit 2	PIF2: External port interrupt2flag bit 0: external portPINT2No interrupt signal on 1: external portPINT2There is an interrupt signal (must be cleared by software)							
Bit 1	PIF1: External port interrupt1flag bit 0: external portPINT1No interrupt signal on 1: external portPINT1There is an interrupt signal (must be cleared by software)							
Bit 0	PIFO: External port interrupt0flag bit 0: external portPINT0No interrupt signal on 1: external portPINT0There is an interrupt signal (must be cleared by software)							

### 7. 6. 7 interrupt enable register1(INTE1)

INTE1: Interrupt Enable Register1								
bit	7	6	5	4	3	2	1	0
name	PIE7	PIE6	PIE5	PIE4	PIE3	PIE2	PIE1	PIE0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0
Bit 7	PIE7: External port interrupt7enable bit 0:prohibit 1:Enable							
Bit 6	PIE6: External port interrupt6enable bit 0:prohibit							

	1:Enable
Bit 5	PIE5: External port interrupt5enable bit 0:prohibit 1:Enable
Bit 4	PIE4: External port interrupt4enable bit 0:prohibit 1:Enable
Bit 3	PIE3: External port interrupt3enable bit 0:prohibit 1:Enable
Bit 2	PIE2: External port interrupt2enable bit 0:prohibit 1:Enable
Bit 1	PIE1: External port interrupt1enable bit 0:prohibit 1:Enable
Bit 0	PIE0: External port interrupt0enable bit 0:prohibit 1:Enable

### 7. 6. 8 Interrupt Flag Register2(INTF2)

INTF2: Interrupt Flag Register2								
bit	7	6	5	4	3	2	1	0
name	T21MIF2	I2CIF	T21MIF1	T21MIF0	T21PIF	T21VIF	RXIF	TXIF
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R
POR	0	0	0	0	0	0	0	0

Bit 7	T21MIF2:T21capture/compare interrupt2flag bit 0: No interrupt request 1:T21multifunction interrupt2ask I2CIF:I2CS
Bit 6	Communication total interrupt flag bit 0: No communication interruption occurs 1: A communication interruption occurred
Bit 5	T21MIF1:T21capture/compare interrupt1flag bit 0: No interrupt request 1:T21multifunction interrupt1ask T21MIF0:T21
Bit 4	capture/compare interrupt0flag bit 0: No interrupt request 1:T21multifunction interrupt0ask
Bit3	T21PIF:T21Periodic Interrupt Flag 0: No interrupt request 1:T21Periodic Interrupt Request
Bit 2	T21VIF:T21overflow interrupt flag 0: No interrupt request

	1:T21overflow interrupt request
Bit 1	RXIF:UARTreceive interrupt flag 0: receive buffer empty (receive not complete) 1: Receive buffer full (receive complete), readRXB
Bit 0	clear TXIF:UARTsend interrupt flag 0: send buffer is full (send not completed) 1: send buffer empty (transmit complete), writeTXBclear

### 7. 6. 9 interrupt enable register2(INTE2)

INTE2: Interrupt Enable Register2								
bit	7	6	5	4	3	2	1	0
name	T21MIE2	I2CIE	T21MIE1	T21MIE0	T21PIE	T21VIE	RXIE	TXIE
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7	T21MIE2:T21capture/compare interrupt2enable bit 0:prohibit 1:Enable
Bit 6	I2CIE:I2CSCommunication total interrupt enable bit 0:prohibit 1:Enable
Bit 5	T21MIE1:T21capture/compare interrupt1enable bit 0:prohibit 1:Enable
Bit 4	T21MIE0:T21capture/compare interrupt0enable bit 0:prohibit 1:Enable
Bit 3	T21PIE:T21Periodic interrupt enable bit 0:prohibit 1:Enable
Bit 2	T21VIE:T21Overflow interrupt enable bit 0:prohibit 1:Enable
Bit 1	RXIE:UARTReceive interrupt enable bit 0:prohibit 1:Enable
Bit 0	TXIE:UARTtransmit interrupt enable bit 0:prohibit 1:Enable

## No.8CHIP CONFIGURATION WORD

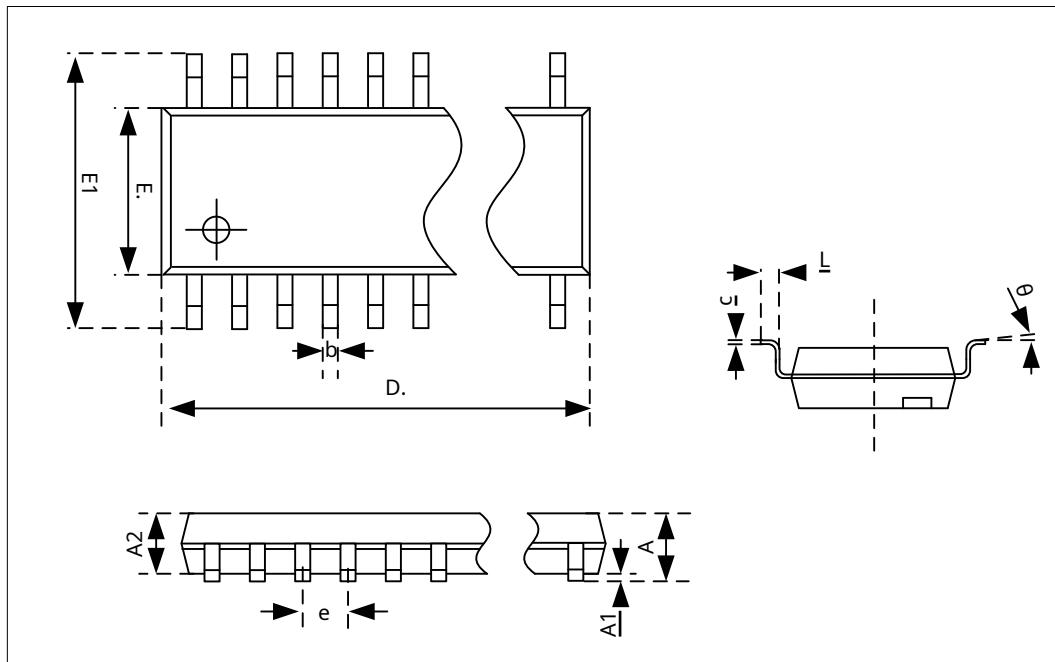
register name	Chip Configuration Words (CFG_WD0)	
address	8001h	
<b>OSCS&lt;2:0&gt;</b>	bit2-0	<p><b>Oscillator Select Bits</b></p> <p>000: reserved for unused 001: reserved for unused 010: reserved for unused 011:INTOSCIO 2MHzmodel,PB3forI/Opin 100: INTOSCIO 4MHzmodel,PB3forI/Opin 101: INTOSCIO 8MHzmodel,PB3forI/Opin 110: reserved for unused 111:INTOSCIO 16MHzmodel,PB3forI/Opin, the main system clock is INTHRC</p>
<b>WDTEN</b>	bit3	<p><b>Hardware watchdog enable bit</b></p> <p>0:prohibit 1:Enable</p>
<b>PWRTEB</b>	bit4	<p><b>Power-up/low-voltage timer enable bit</b></p> <p>whenPA3pin for external resetMRSTNhour 0: Enable (power-on delay approx.130ms) 1:prohibit  whenPA3When the pin is used for digital input and output, it is fixed as enable</p>
<b>MRSTEN</b>	bit5	<p><b>MRSTNPIN function select bit</b></p> <p>0:PA3Pins for digital input and output 1:PA3pin for external reset</p>
<b>BORVS</b>	bit7-6	<p><b>low voltage selection bit</b></p> <p>00:3.1V 01:2.5V 10:2.1V 11: reserved for unused</p>
—	Bit9-8	<b>fixed to full0</b>
<b>ICDEN</b>	bit10	<p><b>ICDDebug mode enable bit</b></p> <p>0:Enable 1:prohibit</p>
<b>INTVEN1</b>	bit11	<p><b>Interrupt mode select bit</b></p> <p>0: default interrupt mode 1: Vectored interrupt mode (control register bitINTVEN0(INTG&lt;2&gt;) must also be1)</p>
—	bit12	<b>fixed as0</b>
<b>FREN</b>	Bit13	<p><b>FLASHProgram storage area look-up instruction read enable bit</b></p> <p>0:prohibit 1:Enable</p>
—	Bit15-14	<b>fixed to full1</b>

Note1: The chip configuration word is configured through the programming interface;

## No.9Chapter chip package diagram

## 9. 1 20-pinPackage diagram

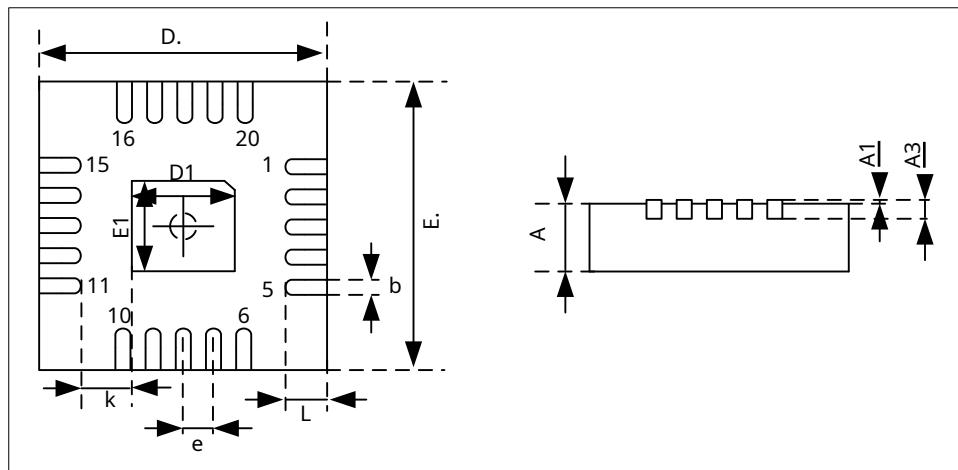
## 9.1.1 TSSOP20



label	Metric (mm)		
	MIN	NOM	MAX
A	—	—	1.20
A1	0.05	—	0.15
A2	0.80	—	1.00
b	0.19	—	0.30
c	0.09	—	0.20
D.	6.40	—	6.60
E.	4.30	—	4.50
E1	6.25	—	6.55
e	0.65BSC		
L	0.50	—	0.70
θ	1°	—	7°

Note: Dimension "D" does not include mold flash, protrusions or gate burrs. Mold flash, protrusions or gate burrs shall not exceed 0.15mm per side.

## 9.1.2 QFN20



label	Metric (mm)		
	MIN	NOM	MAX
A	0.7	0.8	0.9
A1	0.00	0.025	0.05
A2	—	0.203(REF)	—
b	0.15	—	0.25
D.	2.924	3.00	3.076
E.	2.924	3.00	3.076
E1	1.400	1.500	1.600
e	0.400 (BSC)		
L	0.324	0.4	0.476

Note: Dimension "D" does not include mold flash, protrusions or gate burrs. Mold flash, protrusions or gate burrs shall not exceed 0.15mm per side.

**appendix1Instruction Set****appendix1. 1 overview**

This chip provides 79 condensed instructions.

For the convenience of programmers, the assembly instructions are mostly composed of English abbreviations of instruction functions. After the program composed of these instructions is compiled and connected by the compiler, it will be converted into the corresponding instruction code. The converted instruction codes can be divided into opcodes (OP Code) with operands (Operand) in two parts. The opcode portion corresponds to the instruction itself.

The chip runs on 4MHz when oscillating the clock, the time of one machine cycle is 500ns.

Instructions can be divided into two-cycle instructions and single-cycle instructions according to the number of machine cycles executed by the instruction, where JUMP, AJMP, GOTO, CALL, LCALL, RET, RETIA, RETIE, TBR, TBR#1, TBR\_1, TBR1#, TBW, TBW#1, TBW\_1, TBW1# is a two-cycle instruction; when the jump condition is met, JBC, JBS, JCAIE, JCAIG, JCAIL, JCRAE, JCRAAG, JCRAL, JCCRE, JCCRG, JCCRL, JDEC, JINC The instruction is a two-cycle instruction, otherwise it is a single-cycle instruction; other instructions are single-cycle instructions.

**appendix1. 2 Register Manipulation Instructions**

serial number	instruction		Influence status bit	machine cycle	operate
1	SECTION	I<7:0>	—	1	I<7:0>->BKSRI<7:0>
2	PAGE	I<8:0>	—	1	I<1:0>->PCRHI<4:3>
3	ISTEP	I<7:0>	—	1	IAA+i->IAA(-128≤i≤127)
4	MOVI	I<7:0>	—	1	I<7:0>->(A)
5	MOV	R<7:0>,F	Z,N	1	(R)->(Target)
6	MOVA	R<7:0>	—	1	(A)->(R)
7	MOVAR	R<10:0>	—	1	(A)->(R)(RforGPR)
8	MOVRA	R<10:0>	—	1	(R)->(A)(RforGPR)

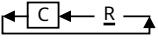
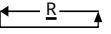
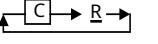
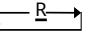
**appendix1. 3 program control instruction**

serial number	instruction		Affects status bits	machine cycle	operate
9	JUMP	I<7:0>	—	2	PC+1+i<7:0>->PC (-128≤i≤127)
10	AJMP	I<19:0>	—	2	I<12:0>->PC<12:0> I<12:8>->PCRH<4:0>
11	GOTO	I<10:0>	—	2	I<10:0>->PC<10:0>, PCRH<4:3>->PC<12:11>
12	CALL	I<10:0>	—	2	PC+1->TOS,I<10:0>->PC<10:0>, PCRH<4:3>->PC<12:11>
13	LCALL	I<19:0>	—	2	PC+1->TOS,I<12:0>->PC<12:0>

serial number	instruction		Affects status bits	machine cycle	operate
					I<12:8>->PCRH<4:0>
14	RCALL	R<7:0>	—	2	PC+1→TOS, (R)→PC<7:0>, PCRH<4:0>→PC<12:8>,
15	JBC	R<7:0>,B<2:0>	—	2or1	when R<B> = 0 skip next instruction
16	JBS	R<7:0>,B<2:0>	—	2or1	when R<B> = 1 skip next instruction
17	JCAIE	I<7:0>	—	2or1	when(A) = I skip next instruction
18	JCAIG	I<7:0>	—	2or1	when(A) > I skip next instruction
19	JCAIL	I<7:0>	—	2or1	when(A) < I skip next instruction
20	JCRAE	R<7:0>	—	2or1	when(R) = (A) skip next instruction
twenty one	JCRAG	R<7:0>	—	2or1	when(R) > (A) skip next instruction
twenty two	JCRAL	R<7:0>	—	2or1	when(R) < (A) skip next instruction
twenty three	JCCRE	R<7:0>,B<2:0>	—	2or1	when C = R(B) skip next instruction
twenty four	JCCRG	R<7:0>,B<2:0>	—	2or1	when C > R(B) skip next instruction
25	JCCRL	R<7:0>,B<2:0>	—	2or1	when C < R(B) skip next instruction
26	JDEC	R<7:0>,F	—	2or1	(R-1)->(target register), when the value of the target register is 0 skip the next instruction
27	JINC	R<7:0>,F	—	2or1	(R+1)->(target register), when the value of the target register is 0 skip the next instruction
28	NOP	—	—	1	no operation
29	POP	—	—	1	AS->A, PSWS->PSW, BKSR->BKSRS, PCRHS->PCRH
30	PUSH	—	—	1	A->AS, PSW->PSWS, BKSR->BKSRS, PCRH->PCRHS
31	RET	—	—	2	TOS->PC
32	RETIA	I<7:0>	—	2	I->(A), TOS->PC
33	RETIE	—	—	2	TOS->PC, 1->GIE
34	RST	—	all status bits are affected	1	Software reset command
35	CWDT	—	N_TO, N_PD	1	00h->WDT, 0->WDT Prescaler, 1->N_TO, 1->N_PD
36	IDLE	—	N_TO, N_PD	1	00h->WDT, 0->WDT Pre scaler, 1->N_TO, 0->N_PD

**appendix1. 4Arithmetic/Logical Operation Instructions**

serial number	instruction		Influence status bit	machine cycle	operate
37	ADD	R<7:0>,F	C, DC, Z, OV, N	1	(R)+(A)->(Target)
38	ADDC	R<7:0>,F	C, DC, Z, OV, N	1	(R)+(A)+C->(Target)
39	ADDCI	I<7:0>	C, DC, Z, OV, N	1	I+(A)+C->(A)
40	ADDI	I<7:0>	C, DC, Z, OV, N	1	I+(A)->(A)
41	AND	R<7:0>,F	Z, N	1	(A).AND.(R)->(Target)

serial number	instruction		Influence status bit	machine cycle	operate
42	ANDI	I<7:0>	Z,N	1	I.AND.(A)->(A)
43	BCC	R<7:0>,B<2:0>	—	1	0->R<B>
44	BSS	R<7:0>,B<2:0>	—	1	1->R<B>
45	BTT	R<7:0>,B<2:0>	—	1	(~R<B>)->R<B>
46	CLR	R<7:0>	Z	1	(R)=0
47	SETR	R<7:0>	—	1	FFh->(R)
48	NEG	R<7:0>	C,DC,Z,OV,N	1	~(R)+1 -> (R)
49	COM	R<7:0>,F	Z,N	1	(~R)->(Target)
50	DAR	R<7:0>,F	C	1	right(R)Decimal Adjustment -> (Target)
51	DAA	—	C	1	right(A)decimal adjustment ->(A)
52	DEC	R<7:0>,F	C,DC,Z,OV,N	1	(R-1)->(Target)
53	INC	R<7:0>,F	C,DC,Z,OV,N	1	(R+1)->(Target)
54	IOR	R<7:0>,F	Z,N	1	(A).OR.(R)->(Target)
55	IORI	I<7:0>	Z,N	1	I.OR.(A)->(A)
56	RLB	R<7:0>,F,B<2:0>	C, Z, N	1	 C<<R<7:0> (Rbring CTowards rotate left)
57	RLBNC	R<7:0>,F,B<2:0>	Z,N	1	 R<7> << R<7:0> (RWithoutC rotate left)
58	RRB	R<7:0>,F,B<2:0>	C, Z, N	1	 C>>R<7:0> (Rbring Crotate right)
59	RRBNC	R<7:0>,F,B<2:0>	Z,N	1	 R<7:0> >> R<0> (RWithoutC rotate right)
60	SUB	R<7:0>,F	C,DC,Z,OV,N	1	(R)-(A)->(Target)
61	SUBC	R<7:0>,F	C,DC,Z,OV,N	1	(R)-(A)- (~C)->(Target)
62	SUBCI	I<7:0>	C, DC, Z, OV, N	1	I-(A)- (~C)->(A)
63	SUBI	I<7:0>	C, DC, Z, OV, N	1	I-(A)->(A)
64	SSUB	R<7:0>,F	C, DC, Z, OV, N	1	(A)-(R)->(Target)
65	SSUBC	R<7:0>,F	C, DC, Z, OV, N	1	(A)-(R)- (~C)->(Target)
66	SSUBCI	I<7:0>	C, DC, Z, OV, N	1	(A)-I- (~C)->(A)
67	SSUBI	I<7:0>	C, DC, Z, OV, N	1	(A)-I->(A)
68	SWAP	R<7:0>,F	—	1	R<3:0>->(target)<7:4>, R<7:4>->(target)<3:0>
69	TBR	—	—	2	Pmem(FRA)->ROMD

serial number	instruction		Influence status bit	machine cycle	operate
70	TBR#1	—	—	2	Pmem(FRA) -> ROMD, FRA+1->FRA
71	TBR_1	—	—	2	Pmem(FRA) -> ROMD, FRA-1->FRA
72	TBR1#	—	—	2	FRA+1->FRA, Pmem(FRA) -> ROMD
73	TBW	—	—	2	ROMD->prog buffer
74	TBW#1	—	—	2	ROMD>prog buffer, FRA+1->FRA
75	TBW_1	—	—	2	ROMD->prog buffer, FRA-1->FRA
76	TBW1#	—	—	2	FRA+1->FRA, ROMD->prog buffer
77	XOR	R<7:0>, F	Z,N	1	(A).XOR.(R)->(Target)
78	XORI	I<7:0>	Z,N	1	I.XOR.(A)->(A)

Note: instruction set description

1:i- immediate value,f- Flag bit,A-registerA,R-registerR,B-registerRFFirstBbit or moveBbit. 2:C-

carry/borrow,DC- half carry/half borrow,Z- zero flag,OV- overflow flag,N- Negative flag. 3:TOS-

Top stack.

4: if the flag bitF = 0, then the target register is the registerA; if the flag bitF = 1, then the target register is the registerR. 5:79one of the instructionsNOPInstructions not described in the table above.

6:SECTIONInstruction,NThe number of digits depends on the actual chip. For this chip, general purpose data memoryGPRDivided into8bank group, soN

The number of digits is3bit.

7:PAGEInstruction,NThe number of digits depends on the actual chip. For this chip, noPCRUregister,NThe number of digits is2bit. 8:PCthe number of digits andPCRURegisters, depending on the actual chip. For this chip,PCThe number of digits is13bit, noPCRUregister.

appendix2Special Function Register Summary Table

address	name	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Power-on reset value	
FF80h	IAD	IAD<7:0>									0000 0000
FF81h	IAAL	IAAL<7:0>									0000 0000
FF82h	IAAH	IAAH<7:0>									0000 0000
FF83h	BKSR	—	—	—	—	—	DBKSR<2:0>			0000 0000	
FF84h	PSW	—	UF	OF	N	OV	Z	DC	C	x00 xxxx	
FF85h	AREG	AREG<7:0>									xxxx xxxx
FF86h	PCRL	PCRL<7:0>									0000 0000
FF87h	PCRH	—	—	—	PCRH<4:0>						0000 0000
FF88h	MULA/MULL	MULA<7:0> / MULL<7:0>									xxxx xxxx
FF89h	MULB/MULH	MULB<7:0> / MULH<7:0>									xxxx xxxx
FF8Ah	DIVEL/DIVQL	DIVEL<7:0>/DIVQL<7:0>									xxxx xxxx
FF8Bh	DIVEH/DIVQH	DIVEH<7:0>/DIVQH<7:0>									xxxx xxxx
FF8Ch	DIVS/DIVR	DIVS<7:0>/DIVR<7:0>									xxxx xxxx
FF8Dh	T31CH2RH	CH2R<15:8>									0000 0000
FF8Eh	T31CH3RL	CH3R<7:0>									0000 0000
FF8Fh	T31CH3RH	CH3R<15:8>									0000 0000
FF90h	FRAL	FRAL<7:0>									xxxx xxxx
FF91h	FRAH	FRAH<7:0>									xxxx xxxx
FF92h	ROMDL	ROMDL<7:0>									xxxx xxxx
FF93h	ROMDH	ROMDH<7:0>									xxxx xxxx
FF94h	ROMCL	—	IAPSEL<2:0>			FPEE	WREN	WR	—	0000 0000	
FF95h	ROMCH	ROMCH<7:0>									0000 0000
FF96h	INTG	GIE	GIEL	—	—	SOFTIF	INTVEN0	INTV<1:0>			0000 0000

address	name	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Power-on reset value
FF97h	INTP	IGP<7:0>								
FF98h	INTCO	PEG3<1:0>		PEG2<1:0>		PEG1<1:0>		PEG0<1:0>		0000 0000
FF99h	T31CH4RL	CH4R<7:0>								
FF9Ah	INTE0	SPI_GIEIE	—	—	—	PIE8	T31IE	T8NIE	ADIE	0000 0000
FF9Bh	INTF0	SPI_GEIF	—	—	—	PIF8	T31IF	T8NIF	ADIF	0000 0000
FF9Ch	INTE1	PIE7	PIE6	PIE5	PIE4	PIE3	PIE2	PIE1	PIEO	0000 0000
FF9Dh	INTF1	PIF7	PIF6	PIF5	PIF4	PIF3	PIF2	PIF1	PIFO	0000 0000
FF9Eh	INTE2	T21MIE2	I2CIE	T21MIE1	T21MIE0	T21PIE	T21VIE	RXIE	TXIE	0000 0000
FF9Fh	INTF2	T21MIF2	I2CIF	T21MIF1	T21MIF0	T21PIF	T21VIF	RXIF	TXIF	0000 0000
FFA0h	SPICON0	RXCLR	TXCLR	CKS<1:0>		RBIM<1:0>		TBIM<1:0>		0011 0000
FFA1h	SPICON1	DFS<1:0>		DRE	—	REN	MS	SPIRST	SPIEN	0000 0000
FFA2h	SPIIE	—	TBWEIE	NSSIE	IDIE	ROIE	TEIE	RBIE	TBIE	0000 0000
FFA3h	SPIIF	—	TBWEIF	NSSIF	IDIF	ROIF	TEIF	RBIF	TBIF	0000 0001
FFA4h	SPIRBR	RBR<7:0>								
FFA5h	SPITBW	TBW<7:0>								
FFA6h	PWRC	LPM<1:0>		—	—	N_TO	N_PD	N_POR	N_BOR	0001 1100
FFA7h	WDTC	—	—	—	—	WDTPRE	WDTPRS<2:0>			0000 1111
FFA8h	WKDC	WKDC<7:0>								
FFA9h	PWEN	ADVREFS2	—	MRSTF	PORLOST	BORFLT<1:0>		RCEN	HALT_PWM	0000 1011
FFAAh	PA	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0	xxxx xxxx
FFABh	PAT	PAT7	PAT6	PAT5	PAT4	PAT3	PAT2	PAT1	PATO	1111 1111
FFACh	PB	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0	xxxx xxxx
FFADh	PBT	PBT7	PBT6	PBT5	PBT4	PBT3	PBT2	PBT1	PBT0	1111 1111
FFAEh	PC	—	—	—	—	—	—	PC1	PC0	xxxx xxxx
FFAFh	PCT	—	—	—	—	—	—	PCT1	PCT0	0000 0011

address	name	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Power-on reset value
FFB0h	PAPU	PAPU7	PAPU6	PAPU5	PAPU4	PAPU3	PAPU2	PAPU1	PAPU0	0000 1000
FFB1h	PBPU	PBPU7	PBPU6	PBPU5	PBPU4	PBPU3	PBPU2	PBPU1	PBPU0	0000 0000
FFB2h	PCPU	—	—	—	—	—	—	PCPU1	PCPU0	0000 0000
FFB3h	T31CHBK	CHOE	AROE	BKPS	BKE	ROFFS	NOFFS	PROTS<1:0>		0000 0000
FFB4h	T31CH4RH	CH4R<15:8>							0000 0000	
FFB5h	PORTCTR	—	—	T31_CH4EN	T31_CH3EN	AD_ETR1EN	ADC_ETROEN	PBOD<1:0>		0000 0000
FFB6h	T31DLYT	DLYT<7:0>							0000 0000	
FFB7h	PAPD	PAPD7	PAPD6	PAPD5	PAPD4	PAPD3	PAPD2	PAPD1	PAPD0	0000 0000
FFB8h	PBDP	PBDP7	PBDP6	PBDP5	PBDP4	PBDP3	PBDP2	PBDP1	PBDP0	0000 0000
FFB9h	FRALN	FRALN<7:0>							0000 0000	
FFBAh	FRAHN	FRAHN<7:0>							0000 0000	
FFBBh	T8N	T8N<7:0>							0000 0000	
FFBCh	T8NC	T8NEN	T8NCLK	T8NM	T8NEG	T8NPRE	T8NPRS<2:0>			0000 0000
FFBDh	T31COL	RLB	CMC<1:0>		DIRS	SPME	UES	UED	T31EN	0000 0000
FFBEh	T31C0H	—	HTOE OFF	—	—	—	—	DFCKS<1:0>		0000 0000
FFBFh	T31C1L	—	ADTRGS<2:0>			—	CHCUS	—	CHCBE	0000 0000
FFC0h	T31C1H	—	ONS4	ONS3N	ONS3	ONS2N	ONS2	ONS1N	ONS1	0000 0000
FFC1h	T31C2L	MSM	TRGS<2:0>			COCE	T31SM<2:0>			0000 0000
FFC2h	T31C2H	ETEG	ECM2E	ETPRS<1:0>		ETFS<3:0>				0000 0000
FFC3h	T31IEL	BKIE	TRGIE	CHUIE	MIE4	MIE3	MIE2	MIE1	UPIE	0000 0000
FFC4h	T31IEH	—	—	—	OVIE4	OVIE3	OVIE2	OVIE1	—	0000 0000
FFC5h	T31IDL	BKID	TRGID	CHUID	MID4	MID3	MID2	MID1	UPID	0000 0000
FFC6h	T31IDH	—	—	—	OVID4	OVID3	OVID2	OVID1	—	0000 0000
FFC7h	T31IVSL	BKIS	TRGIS	CHUIS	MIS4	MIS3	MIS2	MIS1	UPIS	0000 0000
FFC8h	T31IVSH	—	—	—	OVIS4	OVIS3	OVIS2	OVIS1	—	0000 0000

address	name	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Power-on reset value							
FFC9h	T31IFL	BKIF	TRGIF	CHUIF	MIF4	MIF3	MIF2	MIF1	UPIF	0000 0000							
FFCAh	T31IFH	—	—	—	OVIF4	OVIF3	OVIF2	OVIF1	—	0000 0000							
FFCBh	T31IFML	BKIM	TRGIM	CHUIM	MIM4	MIM3	MIM2	MIM1	UPIM	0000 0000							
FFCCh	T31IFMH	—	—	—	OVIM4	OVIM3	OVIM2	OVIM1	—	0000 0000							
FFCDh	T31ICRL	BKIC	TRGIC	CHUIC	MIC4	MIC3	MIC2	MIC1	UPIC	0000 0000							
FFCEh	T31ICRH	—	—	—	OVIC4	OVIC3	OVIC2	OVIC1	—	0000 0000							
FFCFh	T31EVG	BKT	TRGT	CHUT	wxya	CH3T	CH2T	CH1T	UPT	0000 0000							
FFD0h	T31CH1C	CH1COCE	CH1OM<2:0>			CH1OBE	CH1OFE	CH1IOS<1:0>		0000 0000							
FFD0h	T31CH1C	CH1IFS<3:0>				CH1IM<1:0>		CH1IOS<1:0>		0000 0000							
FFD1h	T21L	T21<7:0>								0000 0000							
FFD2h	T21H	T21<15:8>								0000 0000							
FFD3h	T21PL	T21P<7:0>								1111 1111							
FFD4h	T21PH	T21P<15:8>								1111 1111							
FFD5h	T21ROL	T21R0<7:0>								0000 0000							
FFD6h	T21ROH	T21R0<15:8>								0000 0000							
FFD7h	T21CL	T21M<3:0>				CAP1S<1:0>		CAP0S<1:0>		0000 0000							
FFD8h	T21CM	CAP2S<1:0>		T21OM20	T21OM21	T21PRS<3:0>											
FFD9h	T21CH	T21EN	T21POS<6:0>														
FFDAh	ADCRL	ADCRL<7:0>								xxxx xxxx							
FFDBh	ADCRH	ADCRH<7:0>								xxxx xxxx							
FFDCh	ADCCL	ADCHS<3:0>				SMPON	SMPS	ADTRG	ADEN	1111 0100							
FFDDh	ADCCH	ADFM	ADCKS<2:0>			ADST<1:0>		ADVREFS<1:0>		0100 1000							
FFDEh	ANSL	ANSL7	ANSL6	ANSL5	ANSL4	ANSL3	ANSL2	ANSL1	ANSL0	0000 0000							
FFDFh	—	—	—	—	—	—	—	—	—	0000 0000							
FFE0h	RXB	RXB<7:0>								0000 0000							

address	name	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Power-on reset value
FFE1h	RXC	RXEN	RXM	—	—	—	OERR	FERR	RXR8	0000 000x
FFE2h	TXB				TXB<7:0>					0000 0000
FFE3h	TXC	TXEN	TXM	BRGH	—	—	—	TRMT	TXR8	0000 0010
FFE4h	BRR				BRR<7:0>					0000 0000
FFE5h	T21R1L				T21R1<7:0>					0000 0000
FFE6h	T21R1H				T21R1<15:8>					0000 0000
FFE7h	T21R2L				T21R2<7:0>					0000 0000
FFE8h	T21R2H				T21R2<15:8>					0000 0000
FFE9h	T21OC	T21TR	—	PT2EN<1:0>		T21OM22	PT1EN	PT0EN<1:0>		0000 0000
FFEAh	T31CH2C	CH2COCE		CH2OM<2:0>		CH2OBE	CH2OFE	CH2IOS<1:0>		0000 0000
FFEBh	T31CH2C			CH2IFS<3:0>		CH2IM<1:0>		CH2IOS<1:0>		0000 0000
FFEBh	T31CH3C	CH3COCE		CH3OM<2:0>		CH3OBE	CH3OFE	CH3IOS<1:0>		0000 0000
FFECh	T31CH3C			CH3IFS<3:0>		CH3IM<1:0>		CH3IOS<1:0>		0000 0000
FFECh	T31CH4C	CH4COCE		CH4OM<2:0>		CH4OBE	CH4OFE	CH4IOS<1:0>		0000 0000
FFEDh	T31PINCL	CH2NP	CH2NE	CH2P	CH2E	CH1NP	CH1NE	CH1P	CH1E	0000 0000
FFEEh	T31PINCH	CH4NP	—	CH4P	CH4E	CH3NP	CH3NE	CH3P	CH3E	0000 0000
FFEFh	I2CX16	—	—	—		I2CTX16<4:0>				0000 0000
FFF0h	I2CC	I2CTE	I2CPU	I2COD	I2CTAS	I2CANAE	I2CCSE	I2CRST	I2CEN	0000 0000
FFF1h	I2CSA				I2CSADDR<6:0>			I2CRW		0000 0000
FFF2h	I2CTB				I2CTB<7:0>					0000 0000
FFF3h	I2CRB				I2CRB<7:0>					0000 0000
FFF4h	I2CIEC	I2CWKUPEN	I2CNAIE	I2CROIE	I2CTEIE	I2CRBIE	I2CTBIE	I2CPIE	I2CSRIE	0000 0000
FFF5h	I2CIFC	—	I2CNAIF	I2CROIF	I2CTEIF	I2CRBIF	I2CTBIF	I2CPIF	I2CSRIF	1000 0100
FFF6h	T31CNTL				CNT<7:0>					0000 0000

address	name	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Power-on reset value
FFF7h	T31CNTH				CNT<15:8>					0000 0000
FFF8h	T31PRSL				PRS<7:0>					0000 0000
FFF9h	T31PRSH				PRS<15:8>					0000 0000
FFFAh	T31CNTLDL				CNTLD<7:0>					1111 1111
FFFFBh	T31CNTLDH				CNTLD<15:8>					1111 1111
FFFC <sub>h</sub>	T31POS				POS<7:0>					0000 0000
FFFD <sub>h</sub>	T31CH1RL				CH1R<7:0>					0000 0000
FFFE <sub>h</sub>	T31CH1RH				CH1R<15:8>					0000 0000
FFFF <sub>h</sub>	T31CH2RL				CH2R<7:0>					0000 0000

## appendix3electrical characteristics

## appendix3.1 Parameter characteristic table

-Maximum nominal value

parameter	symbol	condition	nominal value	unit
voltage	VDD	—	- 0.3 ~ 7.5	V
Input voltage	V <sub>IN</sub>	—	- 0.3 ~ VDD + 0.3	V
The output voltage	V <sub>out</sub>	—	- 0.3 ~ VDD + 0.3	V
storage temperature	T <sub>STG</sub>	—	- 55 ~ 125	°C
operating temperature	T <sub>OPR</sub>	VDD:2.3 ~ 5.5V	- 40 ~ 85	°C

- Chip power-on and power-off working conditions table (-40 ~ 85°C)

parameter	symbol	minimum value	maximum value	unit
VDDPower-on initial voltage	V <sub>start</sub>	0	0.1	V
VDDrate of ascent	T <sub>VDD</sub>	10	-	us/V
VDDrate of descent		20	-	us/V

- Chip power consumption characteristic parameter table

parameter	symbol	minimum value	typical value	maximum value	unit	working conditions
Chip supply voltage	VDD	2.3	—	5.5	V	- 40°C ~85°C
Chip Quiescent Current	I <sub>DD</sub>	—	900	—	μA	25°C,VDD = 5V,BOR not enabled, all I/O port input low level,MRSTN = 0,internal16MHz as system clock
IDLE0sleep mode Lower chip current	I <sub>PD0</sub>	—	6	—	μA	25°C,VDD = 5V,BOR Enable,WDTEnable.LDOs Sleep, the clock source stops oscillating
IDLE1sleep mode Lower chip current	I <sub>PD1</sub>	—	70	—	μA	25°C,VDD = 5V,BOR Enable, WDTenable, internal RCoscillator.LDOsNormal work operation, the clock source stops
IDLE2sleep mode Lower chip current	I <sub>PD2</sub>	—	300	—	μA	25°C,VDD = 5V,BOR Enable, WDTenable, internal RCoscillator.LDOsNormal work operation, the clock source does not stop
normal operating mode chip currentOP1	I <sub>OP1</sub>	—	2.5	—	mA	25°C,VDD = 5V, normal operating mode, the internal16MHz clock is the system clock,I/Oend Port output fixed level, no negative load.ADCThe module is enabled.

parameter	symbol	minimum value	typical value	maximum value	unit	working conditions
normal operating mode chip currentOP2	IOP2	—	1.5	—	mA	25°C, VDD = 5V, normal operating mode, the internal 8MHz clock is the system clock, I/O port Output fixed level, no load. ADC The module is enabled.
normal operating mode chip currentOP3	IOP3	—	1.2	—	mA	25°C, VDD = 5V, normal operating mode, the internal 4MHz clock is the system clock, I/O port Output fixed level, no load. ADC The module is enabled.
normal operating mode chip currentOP4	IOP4	—	1.0	—	mA	25°C, VDD = 5V, normal operating mode, the internal 2MHz clock is the system clock, I/O port Output fixed level, no load. ADC The module is enabled.
VDDpin input current	IMAXVDD	—	80	—	mA	25°C, VDD = 5V
VSSpin Maximum output current	IMAXVSS	—	200	—	mA	25°C, VDD = 5V
I/Oport sink current	IOL	twenty two	twenty four	26	mA	25°C, VDD = 5V V <sub>OL</sub> = 0.6V
		42	45	48	mA	25°C, VDD = 5V V <sub>OL</sub> = 1.4V
		17	19	twenty one	mA	25°C, VDD = 3.5V V <sub>OL</sub> = 0.6V
		27	30	33	mA	25°C, VDD = 3.5V V <sub>OL</sub> = 1.2V
I/Oport source current	IOH	—	10	—	mA	25°C, VDD = 5V V <sub>OH</sub> = 4.4V
		—	6	—	mA	25°C, VDD = 3.5V V <sub>OH</sub> = 3.0V

Note: I/Oport sink current I<sub>OL</sub>: The driving ability increases with the decrease of temperature, and the whole temperature range (-40~85°C) relative to the normal temperature change of about ±15%.

-Chip Input Port Characteristics Table

Chip operating temperature range: -40°C ~85°C						
parameter	symbol	minimum value	typical value	maximum value	unit	Test Conditions
I/OPort input high level (with Schmidt input characteristics)	VIH	0.8VDD	—	VDD	V	2.3V ≤ VDD ≤ 5.5V
master reset signal MRSTN input high level		0.8VDD	—	VDD	V	

(with Schmidt input characteristics)						
I/OPort input low level	$V_{IL}$	$V_{SS}$	—	0.18VDD	V	
master reset signalMRSTN input low level		$V_{SS}$	—	0.20VDD	V	
I/OPort Input Leakage Current	$I_{IL}$	—	—	$\pm 1$	$\mu A$	$2.3V \leq VDD \leq 5.5V$ $V_{pin} \leq VDD$ (The port is in a high-impedance state state)
Master Reset Port Leakage Current		—	—	5	$\mu A$	$VSS \leq Vpin \leq VDD$
I/OPort input weak pull-up block	$R_{WPu}$	16	18	20	k $\Omega$	$25^{\circ}C, VDD=5.0V$ $V_{pin} = VSS$
I/OPort input weak pull-down block	$R_{WPD}$	16	18	20	k $\Omega$	$25^{\circ}C, VDD=5.0V$ $V_{pin} = VDD$
I/Oinput portVDD/2 output	$V_{VDD/2}$	—	$\pm 3\%$	—		$25^{\circ}C, VDD=5V, weak$ Pull-up and weak pull-down at the same time Enable

Note:I/OPort input weak pull-up and weak pull-down resistors, over the full temperature range (-40~85°C) , relative to the normal temperature change in $\pm 10\%$ within.

- Chip output port characteristic table

Chip operating temperature range:-40°C ~85°C						
parameter	symbol	minimum value	typical value	maximum value	unit	Test Conditions
I/OPort output high level	$V_{Oh}$	$VDD-0.7$	—	—	V	$2.3V \leq VDD \leq 5.5V$ $oh = 6.0mA$
I/OPort output low level	$V_{OL}$	—	—	0.6	V	$2.3V \leq VDD \leq 5.5V$ $ol = 12 mA$

- ESDCharacteristic parameter table

parameter	symbol	grade	maximum value	unit	Test Conditions
ESDVoltage (Human Body Model)	$V_{ESDHBM}$	3A	4000	V	$25^{\circ}C, MIL-STD-883J$
ESDVoltage (machine model)	$V_{ESDMM}$	3	400	V	$25^{\circ}C, JESD22-A115$
ESDVoltage (charged device model)	$V_{ESDCDM}$	C3	1000	V	$25^{\circ}C, JEDEC JS-002-2014$
Latchupelectric current	$I_{LAT}$	I	$\pm 350$	mA	$25^{\circ}C, JESD78$

Note: the aboveESDThe characteristic parameter values are based on the theoretical design value and the test value of the tested sample, not the test value of batch products, and are only for reference in chip application.

- System Clock Requirements Table

parameter	symbol	minimum value	typical value	maximum value	unit	Test Conditions
system clock frequency	$f_{osc}$	—	—	16M	Hz	$2.3V \leq VDD \leq 5.5V$
system clock cycle	$T_{osc}$	62.5	—	—	ns	$2.3V \leq VDD \leq 5.5V$
machine cycle	$T_{inst}$	125	—	—	ns	—

parameter	symbol	minimum value	typical value	maximum value	unit	Test Conditions
External clock high and low time	T <sub>OSL</sub> ,T <sub>O SH</sub>	15	—	—	ns	—
external clock rising and fall time	T <sub>OSR</sub> ,T <sub>OSF</sub>	—	—	15	ns	—
WDTOverflow time (no frequency division)	T <sub>WDT</sub>	6.9 (37KHz)	8 (32KHz)	9.5 (27KHz)	ms	2.3V≤VDD≤5.5V - 40°C ~85°C

## -12bitADCFeature table

parameter	symbol	minimum value	typical value	maximum value	unit	Test Conditions
voltage	VDD	2.3	—	5.5	V	—
Working current	I <sub>ADC</sub>	—	650	—	uA	25°C,VDD=5.0V,ADC The conversion clock frequency is1MHz
resolution	R <sub>R</sub>	—	—	12	bit	—
Differential Linearity	DNL	—	—	±2	LSB	25°C,VDD=5.0V,ADC The conversion clock frequency is1MHz (Fosc/16) , the sampling time is 8T <sub>ADCLKhour</sub> ,VDDReference
Integral linearity	INL	—	—	±3	LSB	25°C,VDD=5.0V,ADC The conversion clock frequency is1MHz (Fosc/16) , the sampling time is 8T <sub>ADCLKhour</sub> ,VDDReference
Offset error	E. <sub>OFF</sub>	—	±2	±3	LSB	See mock up below table SignalADC offsetFeature table
external reference voltage	V <sub>REF</sub>	2.5	—	VDD	V	VDD=3.0V
		3.0	—	VDD	V	VDD=5.0V
Analog input voltage	V <sub>ADIN</sub>	—	—	V <sub>REF</sub>	V	—
input capacitance	C <sub>ADIN</sub>	—	—	40	Pf	—
input resistance	R <sub>ADIN</sub>	—	—	10	KΩ	—
Conversion Clock Frequency	f <sub>ADCLK</sub>	32KHz	—	8	MHz	ADconversion selectionVDDor external VREFPas a positive reference to press
conversion time (no including sampling time between)	T <sub>A DC</sub>	—	13	—	T <sub>ADCLK</sub>	—
sampling time	T <sub>ADS</sub>	250	—	—	ns	Recommended Use8T <sub>ADCLK</sub>

Note1: If the supply voltage is at3Vabove, it is recommendedADCThe conversion clock frequency is set at512KHz~2MHzBetween, the sampling time is set to8T<sub>ADCLK</sub>. Note2: If the supply voltage is at3VBelow, it is recommendedADCThe conversion clock frequency is set at256KHz~512KHzBetween, the sampling time is set to16T<sub>ADCLK</sub>.

## -Analog small signalADC offsetFeature table

Room temperature,ADCThe clock frequency is1MHz(Fosc/16), the sampling time is8TADCLKhour,VDDFor reference, corresponding to different analog inputVainvoltage small signal,ADC offsetTypical values for are as follows:

parameter	minimum value	typical value	maximum value	Test Conditions			
				analog input VoltageVain	reference voltage	ADC Clock frequency	voltage VDD
ADC offset	—	3LSB	—	0mV	VDD	1MHz	5.0V
	—	3LSB	—	4mV	VDD	1MHz	5.0V

## -ADCconversion time comparison table,ADCConvert the positive terminal reference voltage to choose asVDDor externalVREFP

A/Dclock source choose	working frequency			
	16M	8M	4M	1M
Fosc	not recommended	not recommended	not recommended	TADCLK= 1us
Fosc/2	not recommended	not recommended	TADCLK= 0.5us	TADCLK= 2us
Fosc/4	not recommended	TADCLK= 0.5us	TADCLK= 1us	TADCLK= 4us
Fosc/8	TADCLK= 0.5us	TADCLK= 1us	TADCLK= 2us	TADCLK= 8us
Fosc/16	TADCLK= 1us	TADCLK= 2us	TADCLK= 4us	TADCLK= 16us
Fosc/32	TADCLK= 2us	TADCLK= 4us	TADCLK= 8us	TADCLK= 32us
Fosc/64	TADCLK= 4us	TADCLK= 8us	TADCLK= 16us	not recommended

Note: The conversion frequency is not recommended in the table, because the frequency setting is too high or too low, it may causeADCConversion accuracy is reduced.

## - Internal Clock Source Characteristics Table

parameter	symbol	minimum value	typical value	maximum value	unit	Test Conditions
INTHRCClock frequency	fINTHRC	15.84	16	16.16	MHz	25°C,VDD = 5V
		15.68	16	16.32	MHz	- 40°C~85°C, VDD=2.3V~5.5V
INTHRCWorking current	IINTHRC	—	150	—	uA	25°C,VDD=5.0V
INTLRCClock frequency	fINTLRC	30.4	32	33.6	KHz	25°C,VDD = 5.0V
		27	32	37	KHz	- 40°C~85°C, VDD=2.3V~5.5V

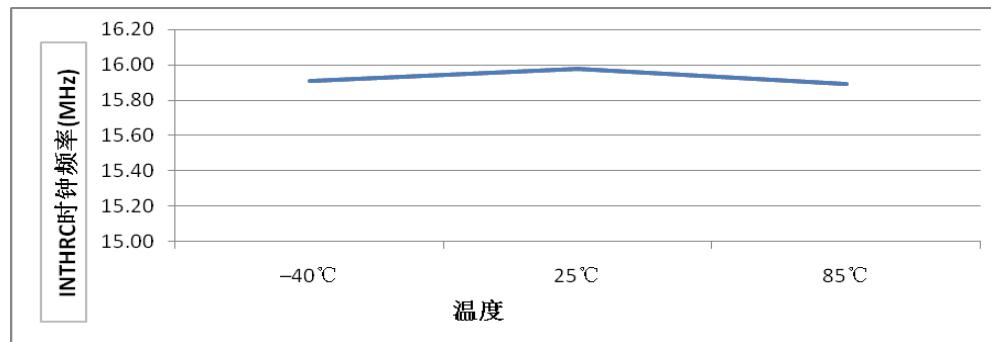
## - low voltage resetBORVoltage characteristics

parameter	symbol	minimum value	typical value	maximum value	unit	Test Conditions
BORlow voltage set voltage1	Vbor1	2.9	3.1	3.3	V	25°C
BORlow voltage set voltage2	Vbor2	2.3	2.5	2.7	V	25°C
BORlow voltage set voltage3	Vbor3	1.9	2.1	2.3	V	25°C
BORLow voltage reset pulse width	Tbor	-	220	-	us	Design theoretical value

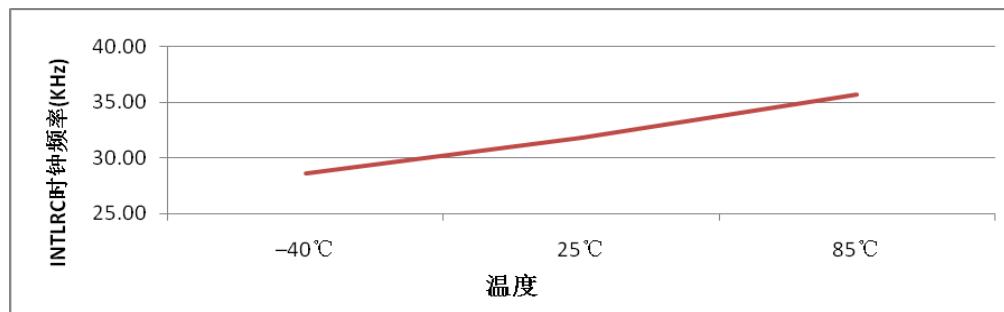
## appendix3.2Parameter characteristic diagram

The diagrams listed in this section are sample tests and are for design reference only. Some of the data listed in the illustrations have exceeded the specified operating range, and such information is for reference only, and the chip is only guaranteed to work normally within the specified range.

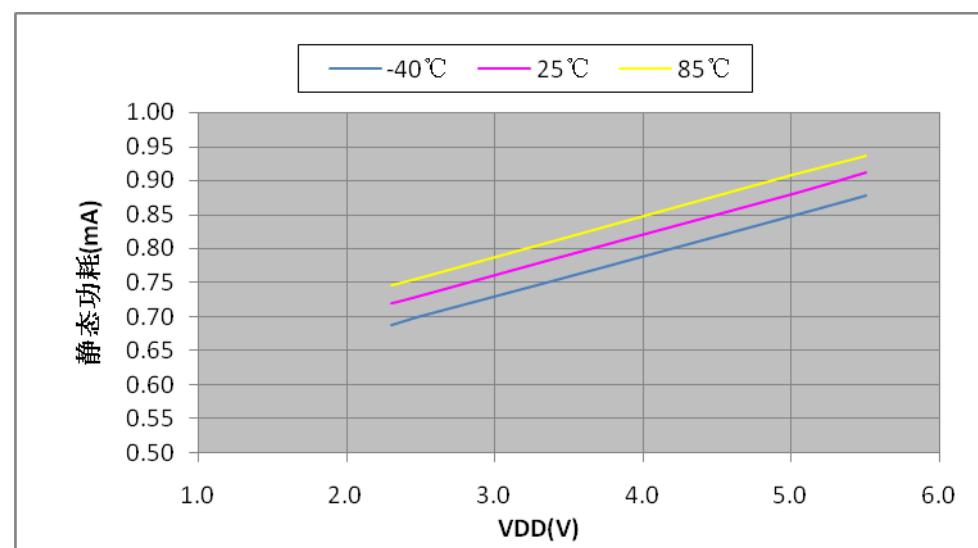
## -inside the chipINTHRClock Frequency vs. Temperature Variation Characteristics



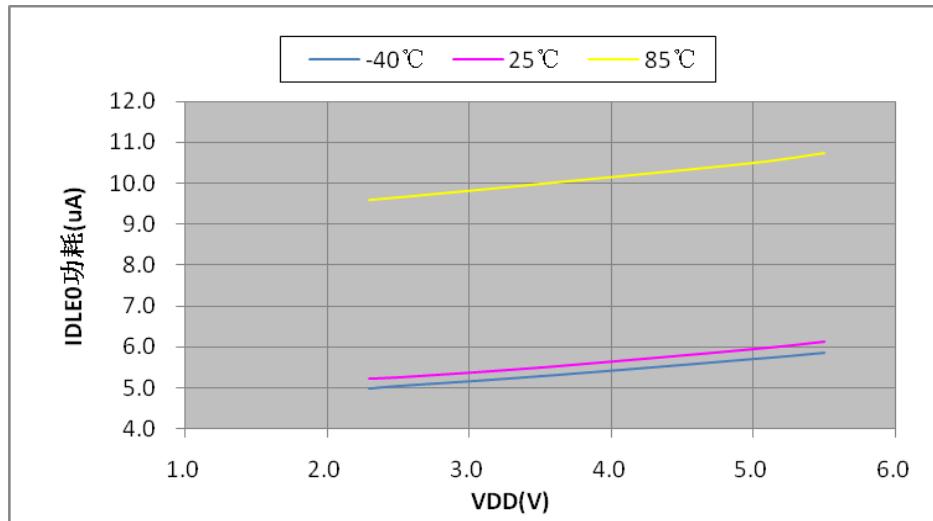
## - inside the chipINTLRCClock Frequency vs. Temperature Variation Characteristics



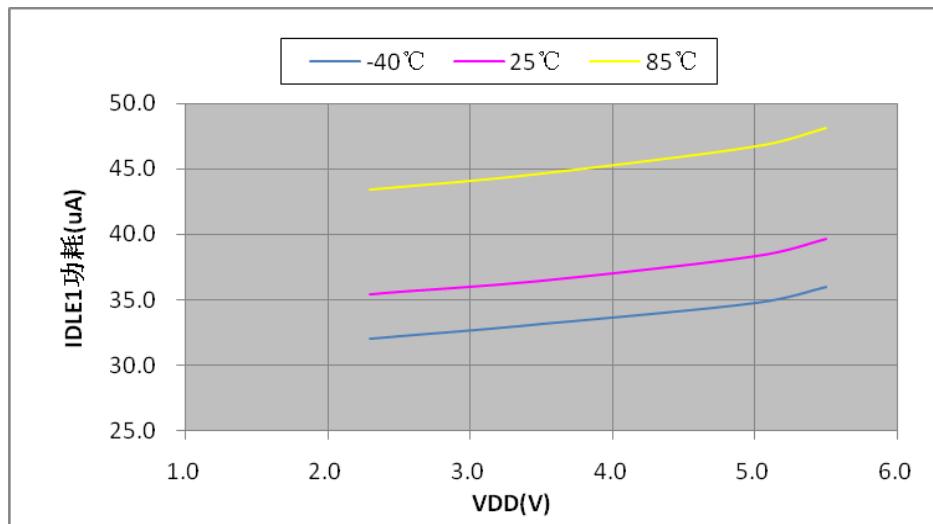
## - Chip quiescent current vs. chip voltage-temperature change characteristic diagram



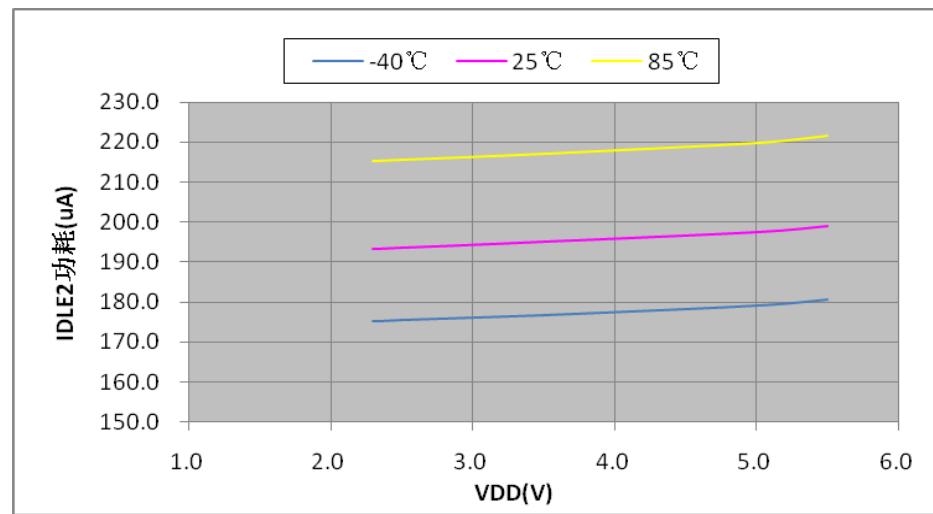
## -chipIDLE0Mode current vs. chip voltage-temperature variation characteristic diagram



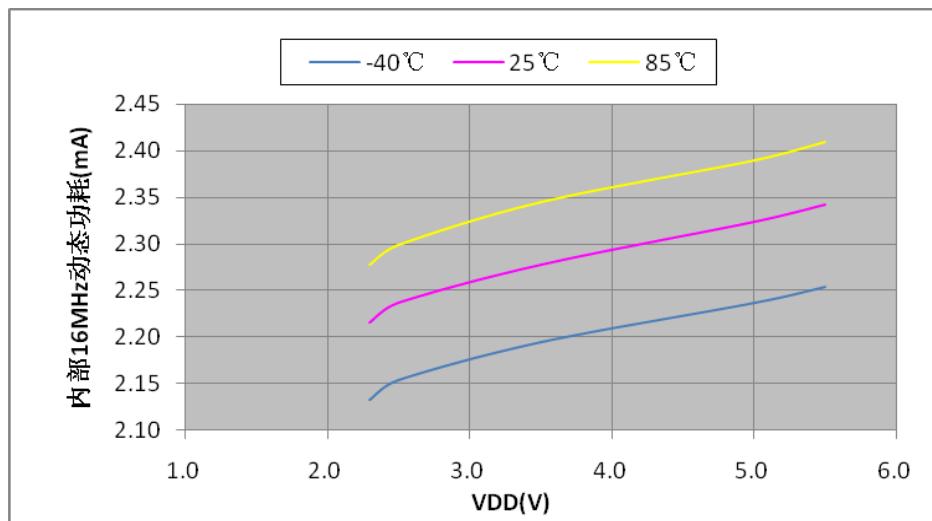
- chip IDLE1Mode current vs. chip voltage-temperature variation characteristic diagram



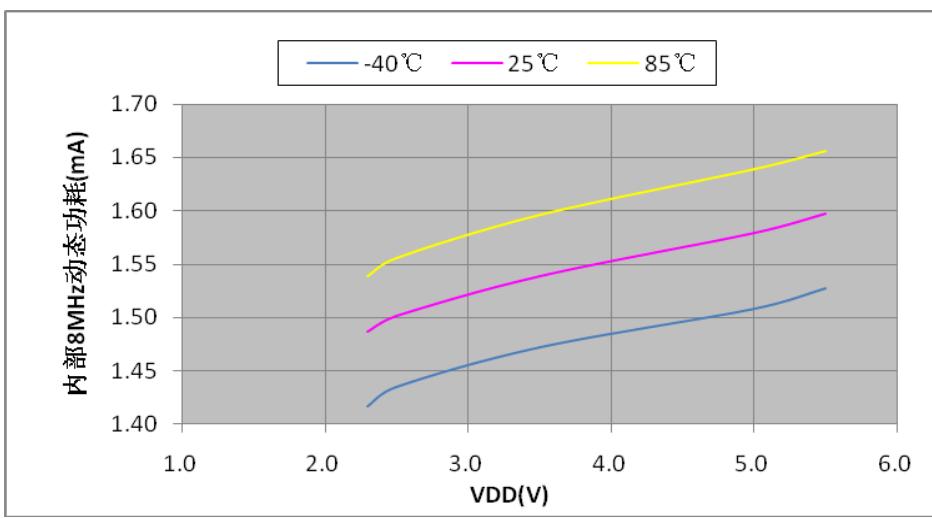
- chip IDLE2Mode current vs. chip voltage-temperature variation characteristic diagram



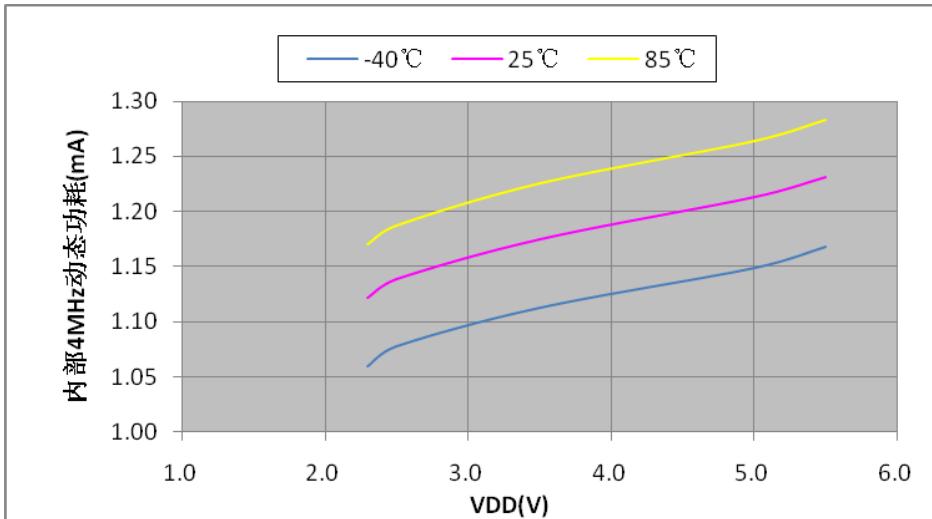
- Working frequency is 16MHz When , the dynamic current of the chip changes with the chip voltage-temperature characteristic diagram



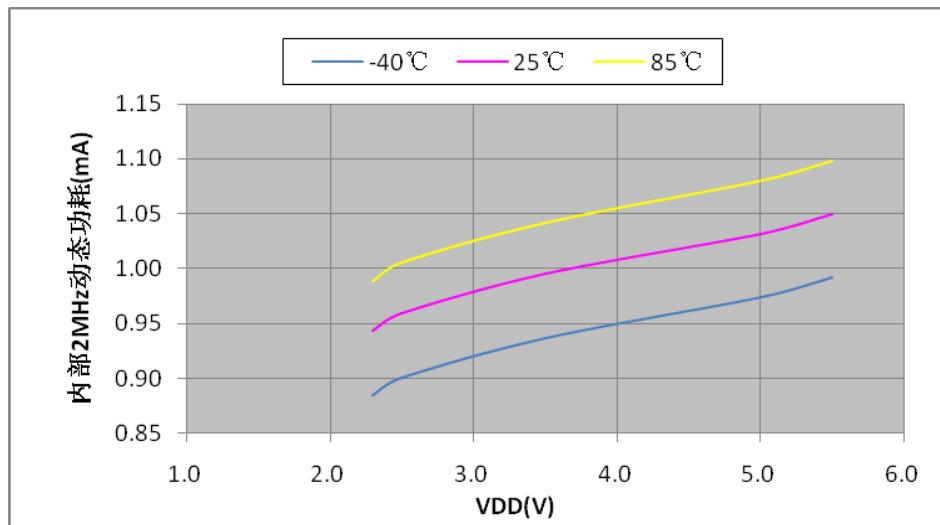
- Working frequency is 8MHzWhen , the dynamic current of the chip changes with the chip voltage-temperature characteristic diagram



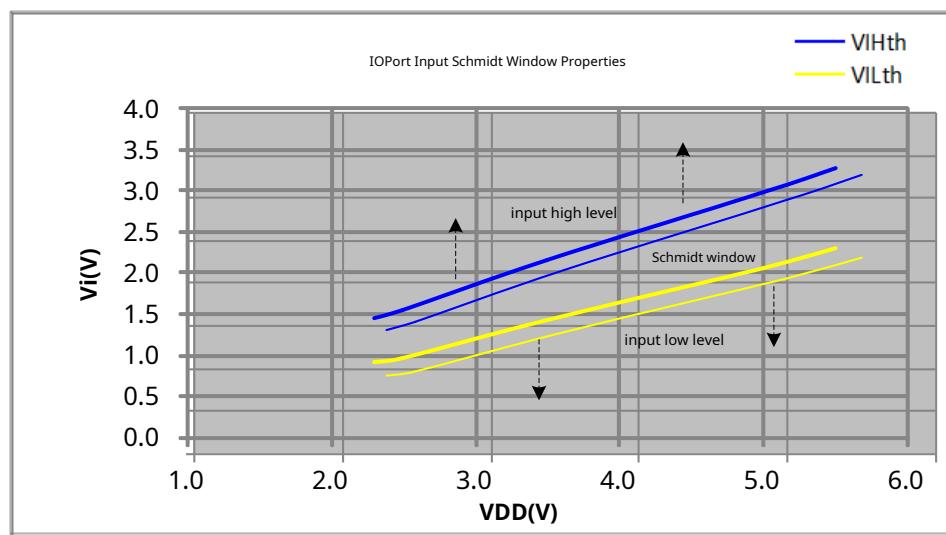
- Working frequency is 4MHzWhen , the dynamic current of the chip changes with the chip voltage-temperature characteristic diagram



- Working frequency is 2MHzWhen , the dynamic current of the chip changes with the chip voltage-temperature characteristic diagram



-I/O Port signal input characteristic diagram (room temperature 25°C)



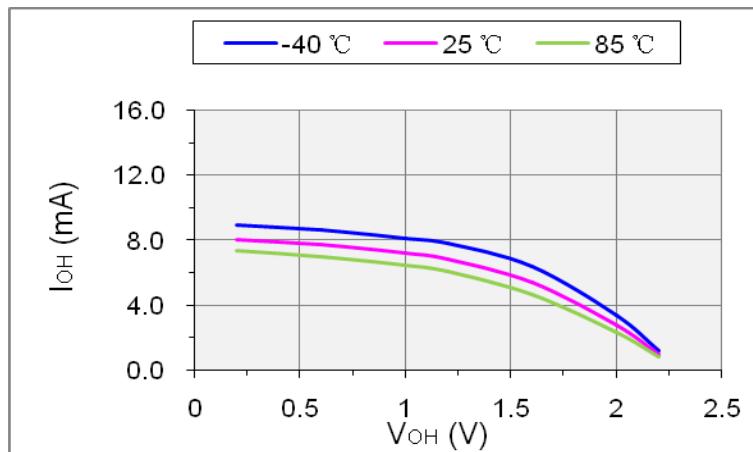
Note1:VIHthis the upper threshold level of the Schmidt window, and the input level greater than this threshold is high;

note2:VILthis the lower threshold level of the Schmidt window, and the input level less than the threshold is low;

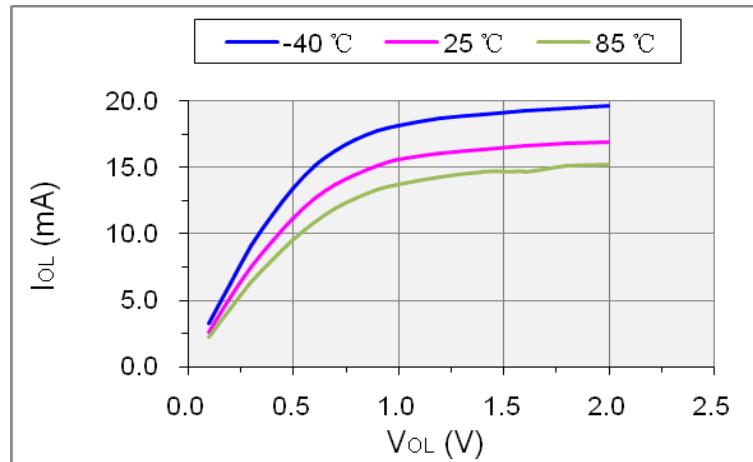
Note3:VIHandVILThere is a Schmidt window between them, and the input level in the window is uncertain, which may be high or low.

-I/O Port signal output characteristic diagram

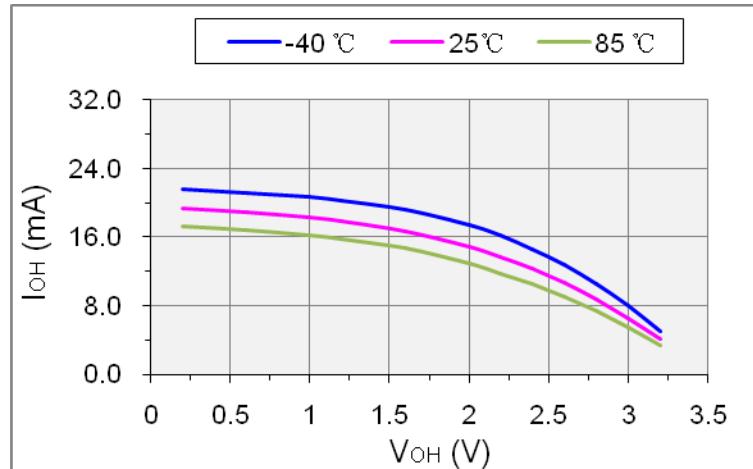
A:  $V_{OH}$  vs  $I_{OH}$  @ VDD = 2.3V



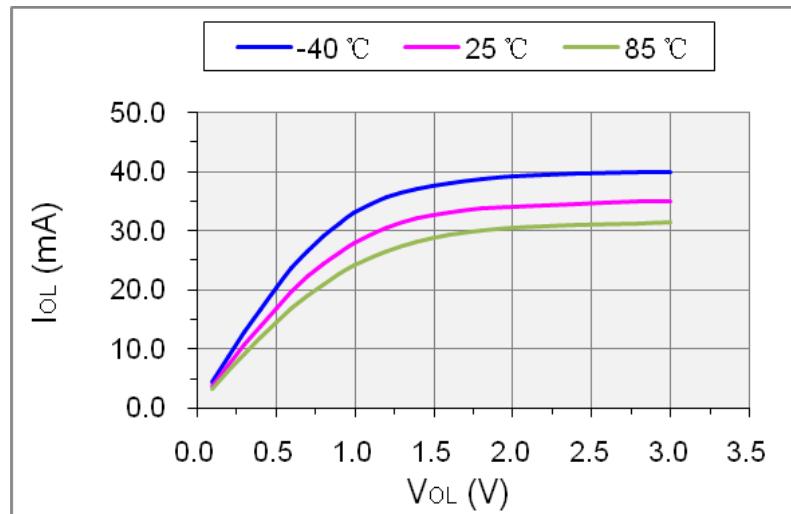
B:  $V_{OL}$  vs  $I_{OL}$  @ VDD = 2.3V



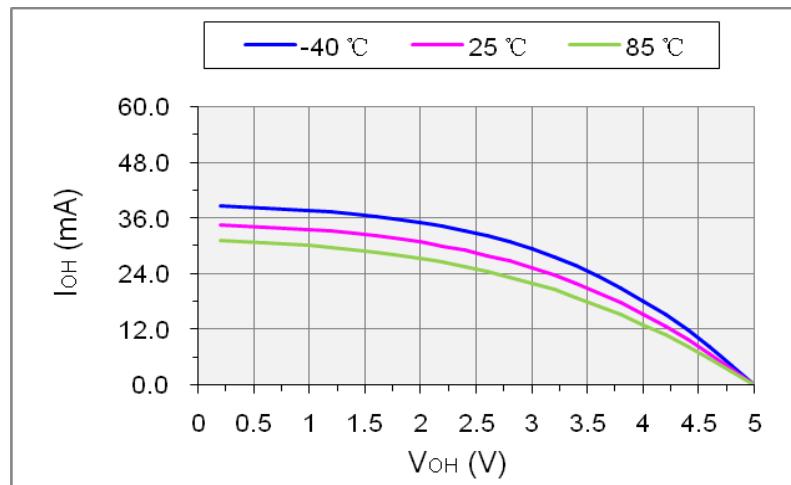
C:  $V_{OH}$  vs  $I_{OH}$  @ VDD = 3.5V



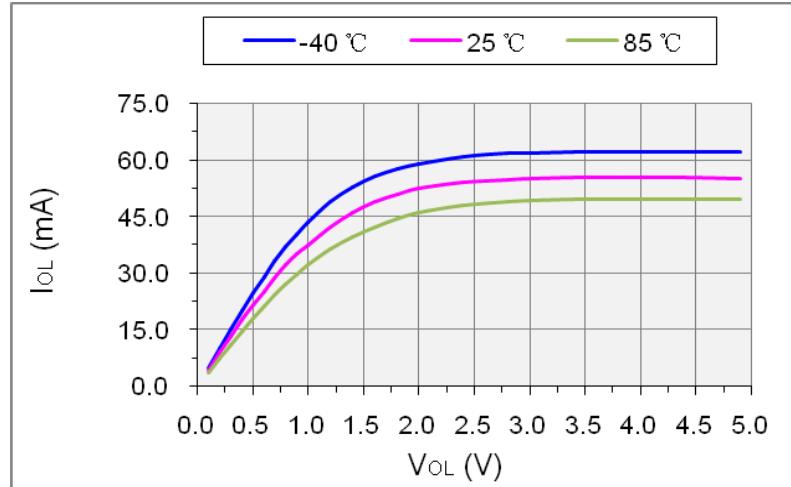
D:  $V_{OL}$  vs  $I_{OL}$  @ VDD = 3.5V



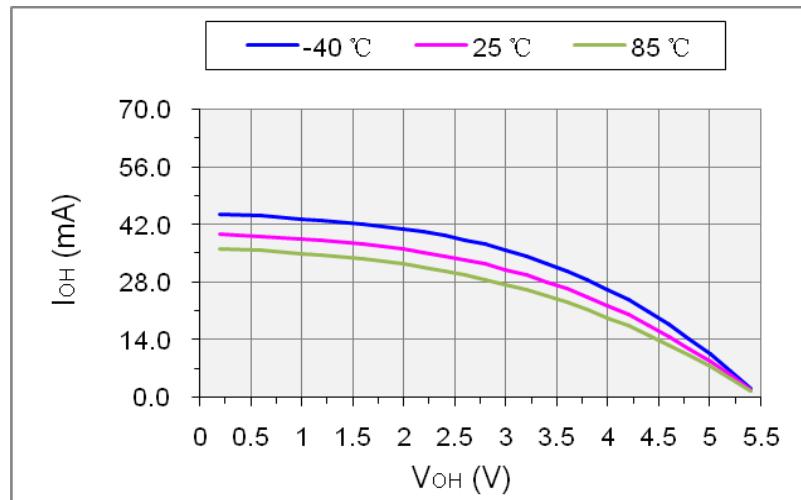
E: V<sub>O</sub>H vs I<sub>O</sub>H@VDD=5.0V



F: V<sub>O</sub>L vs I<sub>O</sub>L@VDD=5.0V



G:  $V_{OH}$  vs  $I_{OH}$  @ VDD = 5.5V



H:  $V_{OL}$  vs  $I_{OL}$  @ VDD = 5.5V

