

2) Calibration of 32K RC oscillator circuit

First set the bit2 of the PS_CR register to 0, and then set the bit2 of the PS_CR register to 1.

After calibration, the 32K RC oscillator will be relatively accurate. But if you want more accurate timing, it is recommended to use 40M clock frequency division

The obtained 32K clock will get the accurate RTC clock at this time, and the deviation will only be related to the crystal frequency deviation.

20.4 Register Description

20.4.1 Register List

Table 171 PMU register list

offset address	name	Abbreviated access	describe	reset value
0X0000	PMU Control Register PS_CR		RW is used to configure 32K calibration, configure 32K clock source, set the STANDBY function of the chip	0X0000_0002
0X0004	PMU Timer 0	TIMER0	RW Configure the timing value (in seconds), enable timing device	0X0000_0000
0X0008	Reserved			
0X0014	PMU interrupt source register INT_SRC		RW provides PMU interrupt flag	0X0000_0000

20.4.2 PMU Control Register

Table 172 PMU Control Register

bit access		Instructions	reset value

[31: 11] RO			24'b0
[10]	RW	<p>Wake-up key interrupt polarity selection:</p> <p>0: The interrupt is set when the wake-up button is at a high level;</p> <p>1: The interrupt is set when the wake-up button is at a high level;</p> <p>Only valid in Sleep interrupt;</p>	1'b0
[9:6]	RW	<p>The minimum hold time for key wake-up:</p> <p>Unit: 128ms;</p>	4'd01
[5]	RW	<p>DLDO_CORE reference voltage source selection</p> <p>1: ABG</p> <p>0: DBG</p>	1'b1
[4]	RW	<p>32K oscillator circuit BYPASS signal</p> <p>Active high, set to 1, 32K is divided into 40M clock frequency. •2</p>	1'b0
[3]	RW	<p>RC 32K oscillator calibration circuit start switch;</p> <p>1'b0: calibration circuit reset;</p> <p>1'b1: start the calibration circuit;</p> <p>To start the calibration function, this bit needs to be set to 0 first and then set to 1.</p>	1'b0
[2]	RW	<p>Enable button trigger to enter sleep function</p> <p>0: Disable;</p> <p>1: Press the io_wakeup button to reach the threshold time in active mode , will trigger io_sleep_flag is interrupted and reported to the MCU.</p>	1'b0
[1]	RW	<p>Sleep enable signal, active high.</p> <p>1'b0: Chip wake-up state</p>	1'b1

		<p>1'b1: The chip enters the Sleep state</p> <p>If the WAKEUP pin is inactive and the TIMER0/1 interrupt wake-up is not configured, this register</p> <p>When valid, the chip enters the Sleep state;</p> <p>If the wake-up interrupt is generated, the chip will switch from Sleep state to wake-up state, and the wake-up condition is satisfied,</p> <p>This bit is automatically cleared to 0.</p> <p>Wake-up source: WAKEUP pin, TIMER0/TIMER1, RTC</p> <p>1) WAKEUP pin, active high; in order for the chip to enter the Sleep state, the WAKEUP pin must be in low level. To wake up, pull up the WAKEUP pin to generate a wake-up interrupt and make the chip leave Sleep state.</p> <p>2) TIMER0, timer wake-up interrupt.</p> <p>When the WAKEUP pin is low, TIMER0 sets the timing time and enables it. When the timing time is up, a wakeup will be generated.</p> <p>The wake-up interrupt causes the chip to leave the Sleep state.</p> <p>3) RTC, timed to wake up</p> <p>When the WAKEUP pin is low and the RTC time is up, a wake-up interrupt will be generated to make the chip leave the Sleep state.</p> <p>state</p>	
[0]	RW	<p>STANDBY enable signal, active high.</p> <p>1'b0: Chip wake-up state</p> <p>1'b1: The chip enters the STANDBY state</p> <p>If the WAKEUP pin is inactive and the TIMER0/1 interrupt wake-up is not configured, this register</p> <p>When valid, the chip enters the STANDBY state;</p> <p>If the wake-up interrupt is generated, the chip will switch from STANDBY state to wake-up state, and the wake-up condition is full enough, this bit is automatically cleared to 0.</p>	1'b0

	<p>Wake-up source: WAKEUP pin, TIMER0/TIMER1, RTC</p> <p>4) WAKEUP pin, active high; in order for the chip to enter the STANDBY state, the WAKEUP pin must be in at low level. To wake up, pull up the WAKEUP pin to generate a wake-up interrupt and make the chip leave STANDBY state.</p> <p>5) TIMER0, timer wake-up interrupt.</p> <p>When the WAKEUP pin is low, TIMER0 sets the timing time and enables it. When the timing time is up, a wakeup will be generated. The wake-up interrupt causes the chip to leave the STANDBY state.</p> <p>6) RTC, timed to wake up</p> <p>When the WAKEUP pin is low and the RTC time is up, a wake-up interrupt will be generated, causing the chip to leave STANDBY status</p>	
--	---	--

20.4.3 PMU Timer 0

Table 173 PMU Timer 0 Register

bit access		Instructions	reset value
[31:17] RO reserved			15'b0
[16]	RW	Timer0 enable bit 1'b0: Bit enable. 1'b1: enable;	1'b0
[15: 0]	RW	Timer0 timing value, unit: second	16'b0

20.4.4 PMU Interrupt Source Register

Table 174 PMU Interrupt Source Register

bit access		Instructions	reset value
[31: 9]	R reserved		
[8]	RW shows the current power-on status: 1'b0: Power-on or reset start 1'b1: wake up from sleep state, write 1 to clear		1'b0
[7]	RO reserved		1'b0
[6]	RO reserved		1'b0
[5]	RW RTC timer interrupt flag bit: 1'b0: A timed interrupt is generated 1'b1: No timing interrupt is generated, write 1 to clear		1'b0
[4]	RW reserved		1'b0
[3]	RW reserved		1'b0
[2]	RW WAKEUP pin wakeup interrupt flag 1'b0: No WAKEUP wake-up interrupt is generated 1'b1: WAKEUP wake-up interrupt is generated, write 1 to clear		1'b0
[1]	RW reserved		1'b0
[0]	RW Timer0 timer interrupt flag bit: 1'b0: No Timer0 interrupt is generated 1'b1: Timer0 interrupt is generated, write 1 to clear		1'b0

Winner Micro

21 Real time clock module

21.1 Function overview

The RTC is a BCD counter/timer provided by the PMU module. The two 32-bit registers contain seconds, minutes, hours, days, months, and years.

The decimal format representation of the hexadecimal code (BCD) can automatically correct the months of 28, 29 (leap year), 30, and 31 days.

Under the corresponding software configuration, the RTC can not only provide the clock calendar function, but also can be used as a timer, when the timer reaches the set time

An RTC interrupt is then generated, which can be used to wake up the system from a sleep state.

The RTC module has two clock sources that can be configured: 40M clock frequency division and internal 32K clock. Which can be used by software configuration during normal work

clock source; only 32K clocks can be used in sleep state. If the RTC clock source is divided by 40M clock in normal working state, then

After entering the sleep state, it will automatically switch to the 32K clock, and the system will keep using the 32K clock after being woken up. So as long as the power supply voltage

Within the working range, the RTC module will not stop working regardless of whether the module is in a normal working state or a sleep state.

21.2 Main Features

- ÿ Provide timing function

- ÿ Provide timing function

- ÿ Provide timed interrupt

- ÿ Interrupt to wake up the system

21.3 Functional Description

21.3.1 Timing function

The initial value of day, hour, minute and second can be configured in RTC configuration register 1, and the initial value of year and month can be configured in RTC configuration register 2.

The timekeeping function is enabled in RTC Configuration Register 2.

After the RTC timing function is enabled, read the RTC configuration register 1 to get the current day, hour, minute and second values.

Register 2 can get the current year and month value.

21.3.2 Timing function

The day, hour, minute, and second timing values can be configured in RTC configuration register 1, and the year and month timing values can be configured in RTC configuration register 2.

The RTC configuration register 1 enables the timing function.

When the RTC timer reaches the timing time, an RTC interrupt will be generated. At this time, set the RTC interrupt bit of the PMU interrupt source register to 1 to clear

except the interrupted state.

When the system enters sleep mode, the interrupt generated by the RTC timer will wake up the system.

21.4 Register Description

21.4.1 Register List

The RTC module has a total of two 32-bit dedicated registers, and the RTC interrupt status needs to query the PMU interrupt source register.

Table 175 RTC register list

offset address	name	Abbreviated access	describe	reset value
0X000C	RTC Configuration Register 1	RTC_R1 RW	Configure RTC day, hour, minute and second value, configure enable timing 0X0000_0000	
0X0010	RTC Configuration Register 2	RTC_R2 RW	Configure RTC year and month value, configure enable timing 0X0000_0000	

21.4.2 RTC Configuration Register 1

Table 176 RTC Configuration Register 1

bit access		Instructions	reset value
[31]	RW	RTC timer interrupt function enable	1'b0

		1'b0: Disable 1'b1: enable	
[30:29]		Reserve	
[28:24] RW		Day start value/day time value	5'b0
[23:21]		Reserve	
[20:16] RW		Hour initial value/hour timing value	5'b0
[15:14]		Reserve	
[13: 8]		Reserve	
[7 : 6]		Reserve	
[5 : 0] RW		Second initial value/second timing value	6'b0

21.4.3 RTC Configuration Register 2

Table 177 RTC Configuration Register 2

bit access		Instructions	reset value
[31:17]		Reserve	
[16]	RW	RTC timing function enable bit 1'b0: Disable 1'b1: enable	1'b0
[15]		Reserve	
[14: 8] RW		Beginning of the year value/yearly fixed value	7'b0
[7 : 4]		Reserve	
[3 : 0] RW		Monthly value/monthly fixed value	4'b0

22 Watchdog module

22.1 Function overview

Implement the "watchdog" function. Designed for global reset in case of system crash.

"Watchdog" will generate a periodic interrupt. After the interrupt is generated, the system software will clear its interrupt flag. If it exceeds the set time, it will not be cleared.

Otherwise, a hard reset signal will be generated to reset the system.

22.2 Main Features

- ÿ Provide timing function
- ÿ Provide reset function
- ÿ Provide timed interrupt

22.3 Functional Description

22.3.1 Timing function

After setting the timing value to the register WD_LD, set the BIT0 of WDG_CTRL to 1 to start the timer, and the WDG module will generate the timer when the timing is up.

When a timed interrupt occurs, the notification program is processed. If the BIT0 of the register WD_CLR is not cleared, a timed interrupt will be generated periodically.

The value of WD_LD is based on the APB clock unit, and the APB clock is divided from the 160M clock.

22.3.2 Reset function

After setting the chip timing value WD_LD, start the timing and reset function (set BIT1/BIT0 of WDG_CTRL), and the WDG module starts the reverse operation.

Timing, when the timing time is up, WDG will generate a timing interrupt. At the same time, if the BIT0 of WD_CLR is not cleared, the chip will

The reset signal is generated in the next cycle.

22.4 Register Description

22.4.1 Register List

Table 178 WDG Register List

offset address	name	Abbreviated access		describe	reset value
0X0000	WDG timing load register WD_LD		RW	configures the timing value for repeated loading	0XFFFF_FFFF
0X0004	WDG current value register WD_VAL		RO	gets the value of the current timer	0XFFFF_FFFF
0X0008	WDG Control Register WD_CTRL	RW		Control Register	0X0000_0000
0X000C	WDG Interrupt Clear Register WD_CLR		WO	Interrupt Clear Register	0X0000_0000
0X0010	WDG interrupt source register WD_SRC		RO	Interrupt Source Register	0X0000_0000
0X0014	WDG Interrupt Output Register WD_STATE	RO		Interrupt Output Status Register	0X0000_0000

22.4.2 WDG Timing Value Load Register

Table 179 WDG Timing Value Load Register

bit access		Instructions	reset value
[31: 0] RW		<p>Configure timing values for repeated loading</p> <p>The value of this register is counted in APB clocks.</p> <p>For example: if the APB clock is 40MHZ, the maximum duration of the timing value is about 107s, that is 0xFFFFFFFF/40000000</p>	32'hffff_ffff

22.4.3 WDG Current Value Register

Table 180 WDG current value register

bit access		Instructions	reset value
[31: 0] RO		<p>Get the value of the current timer</p> <p>To calculate the remaining time, just read the current value.</p> <p>To calculate the elapsed time, simply subtract the value of register WD_VAL from the value of register WD_LD</p>	32'hffff_ffff

22.4.4 WDG Control Register

Table 181 WDG Control Register

bit access		Instructions	reset value
[31: 2]		Reserve	30'h0
[1]	RW	<p>reset enable bit</p> <p>1'b0: When the WDG reset condition occurs, no reset signal is generated</p> <p>1'b1: When the WDG reset condition occurs, the reset signal is generated</p>	1'b0
[0]	RW	<p>timing enable bit</p> <p>1'b0: Timer not working</p> <p>1'b1: The timer works and generates periodic interrupts</p>	1'b0

22.4.5 WDG Interrupt Clear Register

Table 182 WDG Interrupt Clear Register

bit access		Instructions	reset value
[31: 1]		Reserve	31'h0

[0]	WHERE	Interrupt status clear bit, write any value to clear the current interrupt status	1'b0
-----	-------	---	------

22.4.6 WDG Interrupt Source Register

Table 183 WDG Interrupt Source Register

bit access		Instructions	reset value
[31: 1]		Reserve	31'h0
[0]	RO	The interrupt source register, the timer function is turned on, will generate the interrupt at the same time	1'b0

22.4.7 WDG Interrupt Status Register

Table 184 WDG Interrupt Status Register

bit access		Instructions	reset value
[31: 1]		Reserve	31'h0
[0]	RO	Interrupt output status register. This interrupt is not generated after the timer is turned off, but WD_SRC may be 1	1'b0

23 PWM controller

23.1 Function overview

PWM is a method of digitally encoding analog signal levels. Through the use of a high resolution counter, the duty cycle of the square wave is modulated with

to encode the level of a specific analog signal. The PWM signal is still digital because at any given moment, the full-scale

The current supply is either completely present (ON) or completely absent (OFF). A voltage or current source is a repetitive pulse of ON or OFF.

The pulse sequence is added to the simulated load. When it is on, the DC power supply is applied to the load, and when it is off, the power supply is disconnected.

when. Any analog value can be encoded using PWM as long as the bandwidth is sufficient.

23.2 Main Features

- ÿ Support 2-channel input signal capture function (two channels of PWM0 and PWM4)
- ÿ Input signal capture function supports interrupt interactive mode and DMA transfer mode; DMA mode supports word-by-word operation
- ÿ Support 5-channel PWM signal generation function
- ÿ 5-channel PWM signal generation supports one-shot generation mode and auto-load mode
- ÿ Support 5-channel braking function
- ÿ PWM output frequency range: 3Hz~160kHz
- ÿ The maximum precision of the duty cycle: 1/256, the width of the counter inserted in the dead zone: 8bit
- ÿ Support channel 0 channel 1 synchronization function, support channel 2 channel 3 synchronization function
- ÿ Support complementary and non-complementary modes of channel 0, channel 1, and complementary and non-complementary modes of channel 2 and channel 3
- ÿ Support 5-channel sync function

23.3 Functional Description

23.3.1 Input Signal Capture

The PWM controller supports the signal capture function of two channels, and the capture of channel 0 can be activated by setting Bit24 of the PWM_CTL register

function, the capture function of channel 4 can be activated by setting Bit1 of the PWM_CAP2CTL register. The level of the captured signal can also be

to set whether to flip the function. After the channel captures the corresponding signal, the capture number is updated to the corresponding capture register PWM_CAPDAT (pass

channel 0 capture number) and PWM_CAP2DAT (channel 4 capture number).

23.3.2 DMA Transfer Captures

After channel 0 or channel 4 enables the capture function, the count of the capture register can be quickly transferred to the memory through the DMA channel to speed up the user process.

23.3.3 Support for single-shot and autoload modes

The five output channels of the PWM controller all support one-shot output mode and auto-load mode. In single-load mode, the channel outputs the specified cycle

After the waveform, the PWM wave will no longer be output; in the automatic loading mode, after the channel outputs the specified cycle waveform, the cycle will be automatically reloaded number, so as to continue to generate PWM waves.

23.3.4 Multiple Output Modes

The PWM controller supports independent output mode, that is, each channel outputs independently without interfering with each other; it supports dual-channel synchronous mode, that is, one channel

The output is exactly the same as the output of another channel; the five-channel sync mode is supported, and the output of channel 1 to channel 4 is completely the same as the output of channel 0.

It supports dual-channel complementary output, that is, the waveform output by one channel is completely opposite to the waveform output by the other channel; supports complementary mode

Commonly used dead zone settings, the dead zone length can be set up to 256 clock cycles; support braking mode, when the braking port detects the specified power

After leveling, the output channel will output the set braking level.

A variety of output modes are flexible and configurable, which can satisfy users' various application scenarios related to PWM.

23.4 Register Description

23.4.1 PWM Register List

Table 185 PWM register list

offset address name		abbreviation	access	describe	reset value
0X0000 Clock divider register_01	PWM_CLKDIV01		RW	Divide the clock of channel 0 and channel 1 0X0000_0000	
0X0004 Clock frequency division register_23	PWM_CLKDIV23		RW	Divide the clock of channel 2 and channel 3 0X0000_0000	
0X0008 Control register	PWM_CTL		RW	is used to configure or control some configurable items 0X0000_0000	
0X000C Period register	PWM_PERIOD		RW	is used to set the period from channel 0 to channel 4 0X0000_0000	
0X0010 Cycle number register	PWM_PNUM		RW	is used to set the signal generation of channel 0 to channel 4 into cycles	0X0000_0000
0X0014 Compare register	PWM_CMPDAT		RW	is used to store the comparison value of channel 0 to channel 4 to produce different duty cycles	0X0000_0000
0X0018 Dead zone control register	PWM_DTCTL		RW	is used to configure or control the configurable set item	0X0000_0000
0X001C Interrupt Control Register	PWM_INTEN		RW	is used to enable and control related interrupts 0X0000_0000	
0X0020 Interrupt Status Register	PWM_INTSTS		RW	is used to query the status of related interrupts	0X0000_0000
0X0024 Channel 0 capture register	PWM_CAPDAT		RO	is used to capture and count the rising edge and falling edge to channel 0	0X0000_0000
0X0028 Brake control register	PWM_BRKCTL		RW	is used to control the braking mode	0X0000_0000
0X002C Clock frequency division register_4	PWM_CH4_reg1		RW	Divide the clock of channel 4	0X0000_0000

0X0030	Channel 4 control register_1 PWM_CH4_reg2 RW Set related configuration items of channel 4 0X0000_0000			
0X0034	Channel 4 capture register PWM_CAP2DAT	RO is used to capture and count the rising to channel 4 edge and falling edge		0X0000_0000
0X0038	Channel 4 control register_2 PWM_CAP2CTL RW Set related configuration items of channel 4 0X0000_0000			

23.4.2 Clock divider register_01

Table 186 PWM clock divider register_01

bit access		Instructions	reset value
[31:16] RW		<p>CLKDIV1</p> <p>CH1 frequency division counter</p> <p>The frequency division is determined by the counter value</p> <p>Note: The frequency division range is (0~65535). If frequency division is not required, enter 0 or 1.</p>	16'h0
[15:0] RW		<p>CLKDIV0</p> <p>CH0 frequency division counter</p> <p>Same as CH1</p>	16'h0

23.4.3 Clock divider register_23

Table 187 PWM clock divider register_23

bit access		Instructions	reset value
[31:16] RW		<p>CLKDIV3</p> <p>CH3 frequency division counter</p> <p>Same as CH1</p>	16'h0
[15:0] RW		CLKDIV2	16'h0

		CH2 frequency division counter Same as CH1	
--	--	---	--

23.4.4 Control Register

Table 188 PWM Control Register

bit access		Instructions	reset value
[31:27] RW		CNTEN Counter count enable 1'b0: stop counting 1'b1: start counting Note: Each bit controls each channel separately, and controls CH4, CH3, CH2, CH1 and CH0 in sequence from high to low	5'b0
[26]	-	Reserve	1'b0
[25]	RW	CAPINV Capture reverse enable flag 1'b0: Inverse of input signal in capture mode is invalid 1'b1: The input signal in the capture mode is valid in reverse, and the input signal is reversed	1'b0
[24]	RW	CPEN Capture function enable flag 1'b0: CH0 capture function is invalid, RCAPDAT and FCAPDAT values will not be updated; 1'b1: CH0 capture function is valid, capture and latch the PWM counter, respectively store in RCAPDAT (rising edge latch) and FCAPDAT (falling edge latch)	1'b0
[23:22] RW		CNTTYPE3	2'b0

		<p>CH3 Counter counting method</p> <p>2'b00: edge-aligned mode (counter counting is incremented, only for capture mode)</p> <p>2'b01: edge-aligned mode (counter counts down, only for PWM mode)</p> <p>2'b10: Center-aligned mode (PWM mode only)</p> <p>Note: In PWM mode, when the counter is set to edge-aligned mode, you need to set the counting method to decrement Way.</p>	
[21:20] RW		<p>CNTTYPE2</p> <p>CH2 Counter counting method</p> <p>Same as CH3</p>	2'b0
[19:18] RW		<p>CNTTYPE1</p> <p>CH1 Counter counting method</p> <p>Same as CH3</p>	2'b0
[17:16] RW		<p>CNTTYPE0</p> <p>CH0 Counter counting method</p> <p>Same as CH3</p>	2'b0
[15:14] RW		<p>TWOSYNCEN</p> <p>2-channel sync mode enable signal</p> <p>1'b0: 2-channel synchronization not allowed</p> <p>1'b1: Enable 2-channel synchronization,</p> <p>PWM_CH0 and PWM_CH1 have the same phase, and the phase is determined by PWM_CH0; PWM_CH2</p> <p>Has the same phase as PWM_CH3, and the phase is determined by PWM_CH2</p> <p>15bit control CH3 and CH2</p>	2'b0

		14bit control CH1 and CH0	
[13]	-- Reserve		1'b0
[12]	RW	<p>POINTS</p> <p>PWM pin output enable bit</p> <p>1'b0: PWM pin is set to output state</p> <p>1'b1: PWM pin is tri-stated</p> <p>Note: only for CH0</p>	1'b0
[11:8] RW		<p>CNTMODE</p> <p>PWM generation loop method</p> <p>1'b0: single shot mode</p> <p>1'b1: Autoload mode</p> <p>Note: During the change of CNTMODE, PWM_CMPDAT returns to zero; each bit controls each channel separately, from high</p> <p>To low control PW3, PW2, PW1 and PW0 in turn</p>	4'h0
[7]	-- Reserve		1'b0
[6]	RW	<p>ALLSYNCEN</p> <p>All-channel sync mode enable signal</p> <p>1'b0: All channels are not allowed to synchronize</p> <p>1'b1: All channels are allowed to synchronize, PWM_CH0, PWM_CH1, PWM_CH2 and PWM_CH3 have</p> <p>The same phase, and the phase is determined by PWM_CH0</p>	1'b0
[5:2] RW		<p>PINV</p> <p>PWM output signal polarity enable</p> <p>1'b0: PWM output polarity reversal disabled</p>	4'h0

		<p>1'b1: PWM output polarity inversion enable</p> <p>Note: Each bit controls each channel separately, and controls PW3, PW2, PW1 and PW0 in turn from high to low</p>	
[1:0] RW		<p>OUTMODE output mode</p> <p>1'b0: Non-complementary mode for every two channels</p> <p>1'b1: Complementary mode for every two channels</p> <p>BIT1 controls CH2 and CH3</p> <p>BIT0 controls CH0 and CH1</p>	2'b0

23.4.5 Period Register

Table 189 PWM Period Register

bit access		Instructions	reset value
[31:24] RW		<p>PERIOD3</p> <p>CH3 period register value (Note: period cannot be greater than 255)</p> <p>"Edge-aligned mode (counter counts down)":</p> <ul style="list-style-type: none"> ÿ PERIOD register value, period value is (PERIOD + 1) ÿ Duty cycle = (CMP+1)/(PERIOD + 1) ÿ CMP>=PERIOD: PWM output is fixed high ÿ CMP<PERIOD: PWM low level width is (PERIOD-CMP), high level width is (CMP+1) ÿ CMP=0: PWM low level width is PERIOD, high level width is 1; <p>"Center Alignment Mode":</p> <ul style="list-style-type: none"> ÿ PERIOD register value: period is 2*(PERIOD+1) 	8'h0

		<p>ÿ Duty cycle=(2*CMP+1)/(2*(PERIOD+1))</p> <p>ÿ CMP>PERIOD: PWM is continuously high</p> <p>ÿ CMP<=PERIOD: PWM low level=2*(PERIOD-CMP)+1, High level=(2*CMP)+1</p> <p>ÿ CMP=0: PWM low level width is 2*PERIOD+1, high level width is 1.</p> <p>Note: In "center-aligned mode", the number of cycles should not be 255.</p> <p>No matter which alignment mode is selected, the channel period is determined by the division number (N) and the number of periods (P),</p> <p>That is: the input clock is 40MHz, the clock frequency f_div after frequency division is: $f_{div} = 40MHz/N$, N is the division Frequency (16bit). The output frequency f_output is: $f_{output} = f_{div} / P$, where P is the number of cycles.</p> <p>Note: In PWM mode, when the counter is set to edge-aligned mode, you need to set the counting method to decrement Way.</p>	
[23:16] RW		PERIOD2 CH2 period register value (note: period cannot be greater than 255) Same as PERIOD3	8'h0
[15:8] RW		PERIOD1 CH1 period register value (Note: period cannot be greater than 255) Same as PERIOD3	8'h0
[7:0] RW		PERIOD0 CH0 period register value (Note: period cannot be greater than 255) Same as PERIOD3	8'h0

23.4.6 Cycle Number Register

Table 190 PWM Period Number Register

bit access		Instructions	reset value
[31:24] RW		<p>PNUM3</p> <p>PWM3 Generation Cycles</p> <p>Set the number of PWM3 cycles PNUM3, when the PWM generates PNUM3 PWM signals, stop generating signals.</p> <p>number, trigger the interrupt and set the interrupt status word at the same time</p>	8'h0
[23:16] RW		<p>PNUM2</p> <p>Number of PWM2 generation cycles</p> <p>Same PNUM3</p>	8'h0
[15:8] RW		<p>PNUM1</p> <p>Number of PWM1 generation cycles</p> <p>Same PNUM3</p>	8'h0
[7:0] RW		<p>PNUM0</p> <p>PWM0 generation cycle number</p> <p>Same PNUM3</p>	8'h0

23.4.7 Compare Register

Table 191 PWM Compare Register

bit access		Instructions	reset value
[31:24] RW		<p>CMP3</p> <p>PWM3 compare register value</p>	8'h0

		<p>"Edge-aligned mode (counter counts down)":</p> <ul style="list-style-type: none"> ÿ PERIOD register value, period value is (PERIOD + 1) ÿ Duty cycle = (CMP+1)/(PERIOD + 1) ÿ CMP>=PERIOD: PWM output fixed bit high ÿ CMP<PERIOD: PWM low level width is (PERIOD-CMP), high level width is (CMP+1) ÿ CMP=0: PWM low level width is PERIOD, high level width is 1; <p>"Center Alignment Mode":</p> <ul style="list-style-type: none"> ÿ PERIOD register value: period is 2*(PERIOD+1) ÿ Duty ratio=(2*CMP+1)/2*(PERIOD+1) ÿ CMP>PERIOD: PWM is continuously high ÿ CMP<=PERIOD: PWM low level=2*(PERIOD-CMP)+1, high level=(2*CMP)+1 ÿ CMP=0: PWM low level width is 2*PERIOD+1, high level width is 1. <p>No matter which alignment mode is selected, the channel period is determined by the division number (N) and the number of periods (P), namely:</p> <p>The input clock is 40MHz, the clock frequency f_div after frequency division is: $f_{div} = 40\text{MHz}/N$, N is the frequency division number (16bit). The output frequency f_output is: $f_{output} = f_{div} / P$, where P is the number of cycles.</p> <p>Note: In PWM mode, when the counter is set to edge-aligned mode, you need to set the counting method to decrement Way.</p>	
[23:16] RW		CMP2 PWM2 compare register value Same as CMP3	8'h0
[15:8] RW		CMP1 PWM1 compare register value	8'h0

		Same as CMP3	
[7:0] RW		CMP0 PWM0 compare register value Same as CMP3	8'h0

23.4.8 Dead Time Control Register

Table 192 PWM Dead Time Control Register

bit access		Instructions	reset value
[31:22]	-- Reserve		10'h0
[21]	RW	<p>DTEN23</p> <p>Whether channel 2 and channel 3 can insert deadband valid flag</p> <p>Insert deadband valid signal is valid only after the complementary mode of the channel is turned on. And, if a valid signal is inserted</p> <p>If it is 0, the complementary signal output by the two channels has no dead zone insertion.</p> <p>1'b0: Invalid insertion dead zone</p> <p>1'b1: Insert dead zone valid</p>	1'b0
[20]	RW	<p>DTEN01</p> <p>Can channel 0 and channel 1 insert deadband valid flags?</p> <p>Same as DTEN23</p>	1'b0
[19:18]	-- Reserve		2'b0
[17:16] RW		<p>DTDIV</p> <p>Dead time clock divider control</p> <p>2'b00: Dead time clock equal to base clock (40MHz)</p>	2'b0

		<p>2'b01: Dead-band clock equal to base clock (40MHz) divided by two</p> <p>2'b10: Dead-band clock equal to base clock (40MHz) divided by four</p> <p>2'b11: Dead-band clock equal to base clock (40MHz) divided by eight</p>	
[15:8] RW		<p>DTCNT23</p> <p>Dead time interval for channel 3 and channel 2</p> <p>8bit determines the dead zone interval value, and the dead zone clock is determined by DTDIV</p>	8'h0
[7:0] RW		<p>DTCNT01</p> <p>Dead time interval for channel 1 and channel 0</p> <p>8bit determines the dead zone interval value, and the dead zone clock is determined by DTDIV</p>	8'h0

23.4.9 Interrupt Control Register

Table 193 PWM Interrupt Control Register

bit access		Instructions	reset value
[31:8]	-- Reserve		24'h0
[7]	RW	<p>DMA_request_EN</p> <p>DMA_request enable</p> <p>1'b0: DMA_request is invalid</p> <p>1'b1: DMA_request is valid</p>	1'b0
[6]	RW	<p>FLIEN</p> <p>Falling edge buffer interrupt enable bit</p> <p>1'b0: Falling edge buffer interrupt invalid</p> <p>1'b1: Falling edge buffer interrupt is valid</p>	1'b0

		Note: For CH0	
[5]	RW	<p>RLIEN</p> <p>Rising edge buffer interrupt enable bit</p> <p>1'b0: Rising cache interrupt invalid</p> <p>1'b1: Rising edge buffer interrupt is valid</p> <p>Note: For CH0</p>	1'b0
[4:0] RW		<p>SMALL</p> <p>PWM Period Interrupt Enable Bit</p> <p>1'b0: Period interrupt is invalid</p> <p>1'b1: Periodic interrupt is valid</p> <p>Note: When the counter counts to 0 and the number of PWM cycles meets PWM_PNUM, an interrupt is triggered.</p>	5'b0

23.4.10 Interrupt Status Register

Table 194 PWM Interrupt Status Register

bit access		Instructions	reset value
[31:10]	-- Reserve		12'h0
[9]	RW	<p>OVERFL</p> <p>Counter overflow flag</p> <p>1'b0: Capture mode, the counter does not overflow during the counting process</p> <p>1'b1: Capture mode, counter overflows during counter counting</p> <p>Note: When the user clears CFLIF or CRLIF, this bit is also cleared at the same time</p>	1'b0
[8]	RW	FLIFOV	1'b0

		<p>Falling edge delay interrupt flags overrun status</p> <p>1'b0: When CFILF is 1, no falling edge delay interrupt is generated</p> <p>1'b1: When CFILF is 1, another falling edge delay interrupt occurs</p> <p>Note: When the user clears CFILF, this bit is also cleared at the same time</p>	
[7]	RW	<p>RLIFOV</p> <p>Rising edge delay interrupt flag overrun status</p> <p>1'b0: When CRILF is 1, no rising edge delay interrupt is generated</p> <p>1'b1: When CRILF is 1, another rising edge delay interrupt occurs</p> <p>Note: When the user clears CRILF, this bit is also cleared at the same time</p>	1'b0
[6]	RW	<p>CFLIF</p> <p>Capture falling edge interrupt flag</p> <p>1'b0: No falling edge captured</p> <p>1'b1: When a falling edge is captured, this bit is set to 1</p> <p>Note: By writing 1, the flag is cleared;</p> <p>Note: For CH0</p>	1'b0
[5]	RW	<p>CRLIF</p> <p>Capture rising edge interrupt flag</p> <p>1'b0: no rising edge captured</p> <p>1'b1: When a rising edge is captured, this bit is set to 1</p> <p>Note: By writing 1, the flag is cleared;</p> <p>Note: For CH0</p>	1'b0
[4:0]	RW	BIP	5'b0

		<p>PWM Period Interrupt Flag</p> <p>When PWM generates a PWM signal with a specified period, the flag is set to 1; write 1 through software to clear the flag</p> <p>Note: Each bit identifies each channel, and controls PW4, PW3, PW2, PW1 and PW0 in sequence from high to low</p>	
--	--	---	--

23.4.11 Channel 0 Capture Register

Table 195 PWM Channel 0 Capture Register

bit access		Instructions	reset value
[31:16] RO		<p>PWM_FCAPDAT</p> <p>Capture falling edge register</p> <p>When there is a falling edge of the input signal, the current counter value is stored</p>	16'h0
[15:0] RO		<p>PWM_RCAPDAT</p> <p>Capture rising edge register</p> <p>When there is a rising edge of the input signal, the current counter value is stored</p>	16'h0

23.4.12 Brake Control Register

Table 196 PWM Brake Control Register

bit access		Instructions	reset value
[31:16]	-- Reserve		16'h0
[15:11] RW		<p>BRKCTL</p> <p>Brake Mode Enable</p> <p>1'b0: Braking mode disabled</p> <p>1'b1: Brake mode activated</p>	5'b0

		[7:3] Corresponding to CH4, CH3, CH2, CH1 and CH0 respectively	
[10:8]	-- Reserve		3'b0
[7:3] RW		<p>BKOD</p> <p>Brake output control register</p> <p>1'b0: When the braking mode is valid, the PWM output is low level</p> <p>1'b1: PWM output high level when braking mode is valid</p> <p>[7:3] Corresponding to CH4, CH3, CH2, CH1 and CH0 respectively</p>	5'b0
[2:0]	-- Reserve		3'b0

23.4.13 Clock divider register_4

Table 197 PWM clock divider register_4

bit access		Instructions	reset value
[31:16] RW		<p>CLKDIV4</p> <p>CH4 Frequency division counter</p> <p>The frequency division is determined by the counter value</p> <p>Note: The frequency division range is (0~65535). If frequency division is not required, enter 0 or 1.</p>	16'h0
[15:8] RW		<p>PERIOD4</p> <p>CH4 period register value (Note: period cannot be greater than 255)</p> <p>"Edge-aligned mode (counter counts down)":</p> <ul style="list-style-type: none"> ÿ PERIOD register value, period value is (PERIOD + 1) ÿ Duty cycle = (CMP+1)/(PERIOD + 1) ÿ CMP>=PERIOD: PWM output fixed bit high 	8'h0

		<ul style="list-style-type: none"> ÿ CMP<PERIOD: PWM low level width is (PERIOD-CMP), high level width is (CMP+1) ÿ CMP=0: PWM low level width is PERIOD, high level width is 1; "Center Alignment Mode": ÿ PERIOD register value: period is 2*(PERIOD+1) ÿ Duty cycle=(2*CMP+1)/(2*(PERIOD+1)) ÿ CMP>PERIOD: PWM is continuously high ÿ CMP<=PERIOD: PWM low level=2*(PERIOD-CMP)+1, high level=(2*CMP)+1 ÿ CMP=0: PWM low level width is 2*PERIOD+1, high level width is 1. <p>Note: In "center-aligned mode", the number of cycles should not be 255.</p> <p>No matter which alignment mode is selected, the channel period is determined by the division number (N) and the number of periods (P),</p> <p>That is: the input clock is 40MHz, the clock frequency f_div after frequency division is: $f_{div} = 40\text{MHz}/N$, N is the division Frequency (16bit). The output frequency f_output is: $f_{output} = f_{div} / P$, where P is the number of cycles.</p> <p>Note: In PWM mode, when the counter is set to edge-aligned mode, you need to set the counting method to decrement Way.</p>	
[7:0] RW		<p>CH4 Generation Cycles</p> <p>Set the number of PWM4 cycles to PNUM4, after the PWM generates PNUM4 PWM signals,</p> <p>Stop generating a signal while triggering an interrupt and setting the interrupt status word</p>	8'h0

23.4.14 Channel 4 Control Register_1

Table 198 PWM Channel 4 Control Register_1

bit access		Instructions	reset value
[31:16]	-- Reserve		16'h0

[15:8] RW	<p>CMP4</p> <p>CH4 period register value</p> <p>"Edge-aligned mode (counter counts down)":</p> <ul style="list-style-type: none"> ÿ PERIOD register value, period value is (PERIOD + 1) ÿ Duty cycle = (CMP+1)/(PERIOD + 1) ÿ CMP>=PERIOD: PWM output fixed bit high ÿ CMP<PERIOD: PWM low level width is (PERIOD-CMP), high level width is (CMP+1) ÿ CMP=0: PWM low level width is PERIOD, high level width is 1; <p>"Center Alignment Mode":</p> <ul style="list-style-type: none"> ÿ PERIOD register value: period is 2*(PERIOD+1) ÿ Duty ratio=(2*CMP+1)/2*(PERIOD+1) ÿ CMP>PERIOD: PWM is continuously high ÿ CMP<=PERIOD: PWM low level=2*(PERIOD-CMP)+1, high level=(2*CMP)+1 ÿ CMP=0: PWM low level width is 2*PERIOD+1, high level width is 1. <p>No matter which alignment mode is selected, the channel period is determined by the division number (N) and the number of periods (P), namely:</p> <p>The input clock is 40MHz, the clock frequency f_div after frequency division is: $f_{div} = 40\text{MHz}/N$, N is the frequency division number (16bit). The output frequency f_output is: $f_{output} = f_{div} / P$, where P is the number of cycles.</p> <p>Note: In PWM mode, when the counter is set to edge-aligned mode, you need to set the counting method to decrement Way.</p>	8'h0
[7:5]	-- Reserve	3'b0
[4:3] RW	<p>CNTTYPE4</p> <p>CH4 Counter counting method</p>	2'b0

		<p>2'b00: edge-aligned mode (counter counting is incremented, only for capture mode)</p> <p>2'b01: edge-aligned mode (counter counts down, only for PWM mode)</p> <p>2'b10: Center-aligned mode (PWM mode only)</p> <p>Note: In PWM mode, when the counter is set to edge-aligned mode, you need to set the counting method to decrement Way.</p>	
[2]	-- Reserve		1'b0
[1]	RW	<p>CNTMODE4</p> <p>CH4 generation cycle mode</p> <p>1'b0: single shot mode</p> <p>1'b1: Autoload mode</p> <p>Note: During CNTMODE changing, PWM_CMPDAT returns to zero</p>	1'b0
[0]	RW	<p>PINV4</p> <p>CH4 output signal polarity enable</p> <p>1'b0: PWM output polarity reversal disabled</p> <p>1'b1: PWM output polarity inversion enable</p>	1'b0

23.4.15 Channel 4 Capture Register

Table 199 PWM Channel 4 Capture Register

bit access		Instructions	reset value
[31:16] RO		<p>PWM_FCAP2DAT</p> <p>Capture falling edge register</p> <p>When there is a falling edge of the input signal, the current counter value is stored</p>	16'h0

[15:0] RO		PWM_RCAP2DAT Capture rising edge register When there is a rising edge of the input signal, the current counter value is stored	16'h0
-----------	--	--	-------

23.4.16 Channel 4 Control Register_2

Table 200 PWM Channel 4 Control Register_2

bit access		Instructions	reset value
[31:11]	-- Reserve		21'h0
[10]	RW	DMA_request2_mask DMA_request2 enable 1'b0: DMA_request2 is invalid 1'b1: DMA_request2 is valid Note: only for CH4	1'b0
[9]	RW	FLIEN2 Falling edge buffer interrupt enable bit 1'b0: Falling edge buffer interrupt invalid 1'b1: Falling edge buffer interrupt is valid Note: only for CH4	1'b0
[8]	RW	RLIEN2 Rising edge buffer interrupt enable bit 1'b0: Rising cache interrupt invalid 1'b1: Rising edge buffer interrupt is valid Note: only for CH4	1'b0

[7]	RW	<p>OVERFL2</p> <p>Counter overflow flag</p> <p>1'b0: Capture mode, the counter does not overflow during the counting process</p> <p>1'b1: Capture mode, counter overflows during counter counting</p> <p>Note: When the user clears CFLIF or CRLIF, this bit is also cleared at the same time</p> <p>Note: only for CH4</p>	1'b0
[6]	RW	<p>FLIFOV2</p> <p>Falling edge delay interrupt flags overrun status</p> <p>1'b0: When CFILF is 1, no falling edge delay interrupt is generated</p> <p>1'b1: When CFILF is 1, another falling edge delay interrupt occurs</p> <p>Note: When the user clears CFILF, this bit is also cleared at the same time</p> <p>Note: only for CH4</p>	1'b0
[5]	RW	<p>RLIFOV2</p> <p>Rising edge delay interrupt flag overrun status</p> <p>1'b0: When CRILF is 1, no rising edge delay interrupt is generated</p> <p>1'b1: When CRILF is 1, another rising edge delay interrupt occurs</p> <p>Note: When the user clears CRILF, this bit is also cleared at the same time</p> <p>Note: only for CH4</p>	1'b0
[4]	RW	<p>CFLIF2</p> <p>Capture falling edge interrupt flag</p> <p>1'b0: No falling edge captured</p> <p>1'b1: When a falling edge is captured, this bit is set to 1</p>	1'b0

		<p>Note: By writing 1, this flag is cleared</p> <p>Note: only for CH4</p>	
[3]	RW	<p>CRLIF2</p> <p>Capture rising edge interrupt flag</p> <p>1'b0: no rising edge captured</p> <p>1'b1: When a rising edge is captured, this bit is set to 1</p> <p>Note: By writing 1, this flag is cleared</p> <p>Note: only for CH4</p>	1'b0
[2]	RW	<p>POEN2</p> <p>PWM pin output enable bit</p> <p>1'b0: PWM pin is set to output state</p> <p>1'b1: PWM pin is tri-stated</p> <p>Note: only for CH4</p>	1'b0
[1]	RW	<p>CPEN2</p> <p>Capture function enable flag</p> <p>1'b0: CH4 capture function is invalid, RCAPDAT and FCAPDAT values will not be updated;</p> <p>1'b1: CH4 capture function is valid, capture and latch the PWM counter, respectively store in RCAPDAT (rising edge latch) and FCAPDAT (falling edge latch)</p> <p>Note: only for CH4</p>	1'b0
[0]	RW	<p>CAPINV2</p> <p>Capture reverse enable flag</p> <p>1'b0: Inverse of input signal in capture mode is invalid</p>	1'b0

		<p>1'b1: The input signal in the capture mode is valid in reverse, and the input signal is reversed</p> <p>Note: only for CH4</p>	
--	--	---	--

24 QFLASH Controller

24.1 Function overview

W800 has a built-in QFLASH controller, which provides QFLASH read, write and erase operations in bus mode, and provides access to system bus and data bus.

Arbitration, to realize the QFLASH read operation in CACHE mode.

24.2 Main Features

- ÿ Provide bus access FLASH interface
- ÿ Support to directly send SPI commands to FLASH through register configuration
- ÿ Support 24-bit or 32-bit address access to FLASH
- ÿ Support automatic decryption and reading of code and data stored in FLASH

24.3 Functional Description

24.3.1 BUS ACCESS

The address of FLASH in the system starts from 0x08000000, and the address space can be read directly.

24.3.2 Register Access

Through the configuration of registers, SPI commands can be sent directly to FLASH to achieve more flexible access to FLASH, supporting most Sub-FLASH control commands.

24.3.3 Command configuration and startup

The control command code of the Flash is written into the command bit of the register FLASH_CMD, and the command bit of the register is configured according to the command format.

For other bits, for example, if there is an address bit in the command, set the address bit to 1. If there is data in the command, set the DAT bit with

Body reference register description.

After the command configuration is completed, set the command_b bit of the CMD_START register to 1, and the controller starts to send the command to the FLASH.

24.3.3.1 Communication mode and configuration

First send a command to FLASH to configure FLASH as QIO or QPI mode, then set the QIOM or QPIM of the FLASH controller

Set the bit to 1 to use QIO mode or QPI mode to communicate with FLASH.

24.3.3.2 Data Cache

When reading and writing data to FLASH through registers, use the storage space at the address of 0x4000 0000 ~ 0x4000 03ff as the data.

According to the data cache, this address space is also used as the data cache used by the RSA encryption and decryption module. When reading data, the controller will read from FLASH

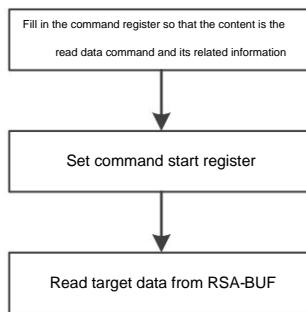
The output data is stored sequentially in the address space starting from 0x4000 0000 for caching. When writing data to FLASH, it is also from 0x4000 0000 Start to read data and send to flash.

24.3.3.3 READ OPERATION

Each read operation can read up to 256 bytes of data, and can read data from any position in QFlash.

After the read operation command is started, the data is stored in the cache without waiting or querying.

The operation flow is shown in the figure.



24.3.3.4 Write Operation

The following commands are all write operations:

WRSR: Write Status Register

PP: Page Programming

SE: sector erase

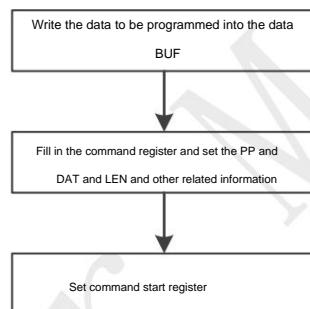
BE: 32K block erase

HBE: 64K block erase

CE: Full Chip Erase

The erase command does not require data, directly configure the command register to the corresponding command, and then start it.

The programming command process is as follows



24.4 Register Description

24.4.1 Register List

Table 201 QFLASH controller register list

offset address name		abbreviation	describe	Defaults
0X0000_0000 Command information register		CMD_INFO	Command and data words required for QFLASH operation <small>Festival</small>	0X0000_0000
0X0000_0004 Command start register		CMD_START	QFLASH command to start	0X0000_0000
0X0000_0008 Flash control register Flash_CR			QFLASH operating mode control	0X0000_0000
0X0000_000C Remap register		Remap	Remap option control	0X0000_0000
0X0000_0010 ADDR register		Flash_ADDR	QFLASH operation address	0X0000_0000
0X0000_0014 Decryption control register		DECRYPT_CR	Firmware confidential mode control	0X0000_0000
0X0000_0018 Decryption Status Register		DECRYPT_STA	firmware decryption status;	0X0000_0000
0X0000_0200 ... 0X0000_0600	Read and write data buffer	RSA_BUF	Used to cache QFLASH read and write data or RSA encryption and decryption data. This memory is shared between QFLASH controller and RSA module. Therefore, QFLASH read and write and RSA operations cannot be used at the same time, it needs to be implemented in the driver. Increase mutual exclusion protection	0X0000_0000

24.4.2 Command Information Register

Table 202 QFLASH command information register

bit access		Instructions	reset value

[31]	RW 1: Indicates that CMD_START carries the Address, and the CMD_START register contains the lower 20 bits of the Address Bit, the high 4 of the Address is fixed to 0, a total of 24 bits	1'b0
[30]	RW 1: Indicates that the CRM in the CMD_START register carries the content	1'b0
[29]	RW 1: Indicates that CMD_INFO carries the Dummy period	1'b0
[28:26]	RW indicates how many Dummy cycles CMD_INFO carries after its content +1	3'b0
[25:16]	RW indicates how many bytes of data CMD_INFO carries after its content +1	10'b0
[15]	RW 1: Indicates that CMD_INFO carries data	1'b0
[14]	RW 1: Indicates that the command filled in the current CMD_INFO command field is a read command, and the command includes RDSR, FR, QIOFR, RDPRID\RSR	1'b0
[13]	RW 1: Indicates that the command filled in the current CMD_INFO command field is the WRSR command	1'b0
[12]	RW 1: Indicates that the command filled in the current CMD_INFO command field is the PP command	1'b0
[11]	RW 1: Indicates that the command filled in the current CMD_INFO command field is the SE command	1'b0
[10]	RW 1: Indicates that the command filled in the current CMD_INFO command field is the BE command	1'b0
[9]	RW 1: Indicates that the command filled in the current CMD_INFO command field is the HBE command	1'b0
[8]	RW 1: Indicates that the command filled in the current CMD_INFO command field is a CE command	1'b0
[7 : 0]	RW CMD_INFO command field, fill in the QFlash command, please refer to the QFlash chip manual.	8'b0

24.4.3 Command Start Register

Table 203 QFLASH command start register

bit access		Instructions	reset value
[31:29] RO		Reserved	3'b0

[28]	When RW is set to 1, it means to start the command. After the command operation is completed, it will be cleared by hardware.	1'b0
[27: 8] RW	Address[19:0]: If the A flag in the command register is 1, the lower 20 bits of the Address are stored here. 20'b0	
[7 : 0] RW	CRM[7:0]: If the CRM flag in the command register is 1, the CRM content is stored here.	8'b0

24.5 Common Commands of QFLASH

Table 204 QFALSH common commands

command name	Command word	command abbr
Write Enable	06H	WREN
Write Disable	04H	WRDI
Read Status	05H	RDSR
Write Status	01H	WRSR
Fast Read	0BH	FR
Quad IO Fast Read	EBH	QIOFR
Page Program	02H	PP
Sector Erase	20H	I KNOW

For other commands, please refer to the related manuals of QFLASH.

Winner Micro

25 PSRAM interface controller

25.1 Function overview

W800 has a built-in PSRAM controller with SPI/QSPI interface, supports external PSRAM device access, and provides bus-based PSRAM reading

Write and erase operations. The maximum read and write speed is 80MHz.

25.2 Main Features

- ÿ Support read and write access to external PSRAM

- ÿ Configurable as SPI and QSPI

- ÿ SPI/QSPI clock frequency can be configured

- ÿ Support BURST INC mode access

- ÿ Support half sleep mode of PSRAM

25.3 Functional Description

The full name of PSRAM is Pseudo static random access memory, which refers to pseudo-static random access memory. Compared with traditional SRAM

It has the advantages of small package, large capacity and low cost, and is mainly used for data caching in IoT applications. The interface is mostly SPI, QSPI

Wait. The interface pins mainly include clock signal SCK, chip select signal CS and 4 bidirectional data IOs. The PSRAM controller provided by the W800 can

It is accessed by the bus mode of PSRAM supporting SPI/QSPI interface, the maximum working clock rate is 80MHz, and the maximum capacity supports 64Mb.

25.3.1 Pin Description

SCLK: SPI interface clock, the SCLK cycle is set by PSRAM_CTRL[7:4], the minimum frequency division value that can be set is 3, and the default is AHB

Divide by 4 of the clock.

CS: SPI interface chip select signal

SIO0: SPI mode data input, QSPI mode SD[0]

SIO1: SPI mode data output, QSPI mode SD[1]

SIO2: QSPI mode SD[2]

SIO3: QSPI mode SD[3]

25.3.2 Access Mode Settings

The PSRAM controller supports two access modes: SPI and four-wire QSPI to the external PSRAM, and the default is SPI. by setting

PSRAM_CTRL[1] configures the mode of the SPI.

PSRAM_CTRL[1] is 0 by default, that is, SPI mode. At this time, it takes 64 SCLK cycles to complete a write operation and 64 SCLK cycles to complete a read operation.

72 SCLK cycles are required.

If PSRAM works in SPI mode, when writing 1 to PSRAM_CTRL[1], the controller will send command 35H to PSRAM,

When PSRAM_CTRL[1] is read as 1, it means that the command is sent and the PSRAM enters the QPI mode.

16 SCLK cycles, 22 SCLK cycles are required to complete a read operation.

If PSRAM works in QPI mode, when writing 0 to PSRAM_CTRL[1], the controller will send command F5H to PSRAM,

When PSRAM_CTRL[1] is read as 0, it indicates that the command transmission is completed, and the PSRAM enters the SPI mode.

The PSRAM working mode must be set after the initialization operation is completed, but cannot be set at the same time.

25.3.3 PSRAM initialization

Before the first use, after the PSRAM is powered on and stabilized, write 1 to the register PSRAM_CTRL[0] to start the PSRAM reset initialization

operation, that is, send 66H and 99H commands to PSRAM. By default, SPI mode is used to send, that is, 8 SCLK + tCPH + 8

time of SCLK. After initialization, the hardware automatically clears PSRAM_CTRL[0].

The initialization operation will restore the PSRAM to SPI mode.

The recommended initialization process is:

ÿ Write PSRAM_CTRL[0] to 1

(2) Wait until PSRAM_CTRL[0] is automatically cleared

(3) Reset the PSRAM controller by soft reset

ÿ Reset other required parameters of PSRAM_CTRL

25.3.4 Access Methods of PSRAM

The way of reading and writing to PSRAM is the same as that of ordinary SRAM, that is, writing/reading data to the corresponding bus address.

25.3.5 BURST function

By setting PSRAM_CTRL[2], the controller can support the BURST initiated by the AHB bus, that is, when the HBURST on the AHB bus is

At 1/3/5/7, it means that the AHB bus starts a continuous read/write with an address increment. At this time, in order to improve the access speed of PSRAM, control the

After completing the read/write of a word, the processor will not pull CS high, but directly read/write the data of the next word.

The OVERTIMER register is used to set the maximum time for CS to be low, the unit is the number of cycles of HCLK, each time a BURST operation starts,

The internal counter starts counting from 0. When the counter value is greater than the set value, the controller stops automatically after completing the read/write of the current word.

BURST operation, directly change CS to high level, if there is still data on the AHB bus to read/write at this time, it will be treated as a separate WORD

conduct.

The PSRAM controller does not support BURST in the form of WRAP. If WRAP BURST is used to access PSRAM, please

PSRAM_CTRL[2] is set to 0.

25.4 Register Description

25.4.1 Register List

Table 205 PSRAM Controller Register List

offset address name		abbreviation	describe	Defaults
0X0000_0000 Control Register		PSRAM_CTRL	PSRAM Controller Settings	0X0000_0000
0X0000_0004 Timeout Control Register		OVERTIMER	CS Timeout Control	0X0000_0000

25.4.2 Command Information Register

Table 206 PSRAM Control Setting Register

bit access		Instructions	reset value
[31:12] RO		RSV	'b0
[11]	RW HSM,	Halsleep mode enabled 1: Enable PSRAM half-sleep mode 0: Clear half sleep mode	1'b0
[10:8]	RW tCPH,	the shortest time setting of CS high level, the unit is the number of AHB clock cycles, must be greater than 1, the specific time depends on the Set it according to the instructions in the same psram manual, if you don't know, you can not modify the default value.	3'd6
[7:4]	RW SPI frequency divider setting	 It can only be configured to a value of 2 or more, and the written value is a multiple of the frequency division	4'd4
[3]	RO	RSV	
[2]	RW INC_EN	BURST function enable 1: Support BURST function on AHB bus 0: BURST function is not supported	1'b0

[1]	RW QUAD	Write 1 to enable QPI mode of PSRAM, write 0 to enable SPI mode Read this flag to know which mode the current PSRAM is in.	1'b0
[0]	RW PSRAM reset	Write 1 to start the reset operation to PSRAM, and it will be automatically cleared after reset.	1'b0

25.4.3 Timeout Control Register

Table 207 CS Timeout Control Register

bit access		Instructions	reset value
[11:0]	RW	Timeout register setting, sets the maximum time for CS to be low, for BURST mode	12'd0

26 ADC

26.1 ADC Function Overview

The acquisition module based on Sigma-Delta ADC completes the acquisition of up to 4 channels of analog signals. The sampling rate is controlled by an external input clock.

It can collect input voltage and chip temperature, and supports input calibration and temperature compensation calibration.

26.2 ADC Main Features

- ÿ Support up to 4 channels of data acquisition
- ÿ Support DMA module for data buffering;
- ÿ Support interrupt interaction mode;
- ÿ Support the function of comparing the collected data with the input data
- ÿ The highest sampling frequency is 1KHz;
- ÿ Support single-ended input mode and differential input mode;

26.3 ADC Function Description

26.3.1 Module Basic Structure

The ADC module is a PGA+SDADC structure. After the input signal is pre-amplified by the PGA, it is input to the SDADC for processing. The SDADC design

For 16bit ADC. It should be noted that the on-chip ADC is powered by a 2.5V LDO. In order to protect the internal circuit, the input signal must be fully

Foot Vinmax < 2.5V, other restrictions, see section 26.3.8.

26.3.2 Channel selection

Register 0x04[11:8] provides the channel selection function. The ADC of W800 provides 4-channel signal acquisition function. by setting the register

0x04[11:8] can select any of the 4 channels for single-ended signal data acquisition, or select 2 pre-paired differential pairs in the 4 channels

Differential signal data acquisition by channel. For specific channel selection, please refer to the register description.

In addition to 4 channels, the ADC module also provides temperature detection, voltage detection and offset calibration functions. corresponding to the channel selection

4'b1100|4'b1101|4'b1110.

If you need to use the corresponding function, configure the channel selection as the corresponding channel, and then follow the process of the above instructions.

26.3.3 Data Comparison

The ADC module provides data comparison function, the switch of this function can be set by BIT[5] of configuration register 0x10.

The user can configure the value to be compared through BIT[17:0] of result register 0x18. This value needs to be converted to a 18bit signed number, the most

The high bit is the sign bit. BIT[6] of ADC configuration register 0x10 can set the direction of data comparison, when this bit is 0, if ADC

If the collected value is greater than or equal to Comp_data, INT_CMP will be set to 1 and an interrupt will be generated; when this bit is 1, only

When the value collected by the ADC is less than Comp_data, INT_CMP will be set to 1 and an interrupt will be generated.

26.3.4 PGA Gain Adjustment Instructions

BIT[8:4] of register address 0X08, 5-bit gain control signal, among which

0X08[8:7]=GAIN_CTRL_PGA<4:3> (corresponding to GAIN2)

0X08[6:4]=GAIN_CTRL_PGA<2:0> (corresponding to GAIN1, 110 and 111 are not used)

PGA overall gain GAIN_PGA=GAIN1*GAIN2

PGA gain configuration table (represented by magnification)

[6:4]\[8:7]	00	01	10	11
000	1	2	3	4
001	16	32	48	64
010	32	64	96	128
011	64	128	192	256
100	128	256	384	512
101	256	512	768	1024

According to the simulation results, for the same magnification, it is recommended to use the configuration with the larger magnification of GAIN2.

26.3.5 Single-Ended Mode VCMIN Adjustment Description

The internal VCMIN voltage of the PGA, in single-ended mode, is used as the negative terminal of the op amp input.

The purpose of adjusting VCMIN is to adapt to various signal sources with different bias points and prevent PGA output from saturating at some too high or too low bias points.

and.

BIT[16:11] of register address 0X40000D20, default 100000, ie VCMIN default=1.311V, step \geq 39.1mV

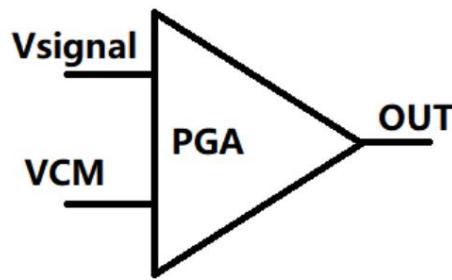


Figure 37 Principle of single-ended working mode

26.3.6 Bypass Mode

Bypass_pga register address 0X08 BIT[3], the default is 0, not bypassed, the PGA works normally; if it is 1, the PGA is bypassed at this time,

The external signal source is directly connected to the SDADC input, which is mainly used for internal module testing, and no bypass is required in normal use.

Bypass_ref register address 0X08 BIT[2], the default is 0, no bypass, SDADC works normally; if it is 1, then the SDADC

The partial ref is bypassed, and an external reference voltage needs to be added through the test pin. It is mainly used for internal module testing, and bypass is not required in normal use.

26.3.7 Supplementary description of SDADC output code

SDADC output result is stored in ADC_RESULT register, register address 0X00 BIT[17:0].

The output code of SDADC itself is an unsigned number, because the number processes the output code of SDADC (the highest bit is inverted), it becomes signed

number, and the 0 code of the SDADC output code is not fixed (and not exactly 10 0000 0000 0000 0000), so the register is directly used.

The value of the device for subsequent calculations is wrong.

When calculating the output result of SDADC, the value of register ADC_RESULT[17:0] (the read value is a hexadecimal number) requires software to

Convert back to an unsigned number (the highest bit ADC_RESULT[17] of the binary number is inverted), and then perform subsequent calculations.

The SDADC designed this time is a 16bit ADC (the design effective number of bits is 16bit), and the register data is 18bit, which is increased by a digital filter.

16bit number is converted to 18bit number (18bit number \geq 16bit number * 4), so the effective number of digits is the upper 16bit (ie

ADC_RESULT[17:2])

After converting the read value back to an unsigned number, the calculation involving the LSB also requires software to convert the 18-bit data into 16-bit data, that is, discard the most

The lower two bits of data (ADC_RESULT[1:0]), only the upper 16bits are reserved (returned to unsigned ADC_RESULT[17:2]), and then 16bits are used

The data is used for subsequent calculation. At this time, the simulation result of LSB is approximately equal to 68.5uV (the actual LSB of the chip needs to be tested to determine).

26.3.8 Input signal voltage range

Because the NTO version PGA and SDADC use 2.5V LDO power supply, in order to prevent internal circuit saturation, the input signal voltage range is limited system.

Assuming that the input differential signals are Vinp and Vinn (in single-ended mode, the signal Vinn is VCMIN), then the input differential voltage Vdiff=Vinp

Vinn, input common mode voltage Vcm=(Vinp+Vinn)/2. PGA first stage gain GAIN1, second stage gain GAIN2, PGA overall gain

GAIN_PGA=GAIN1*GAIN2 (see Section 2.2.2 for details).

The input signal must meet the following constraints at the same time:

0V < Vinp < 2.5V \geq 0V < Win < 2.5V \geq

0V < Vcm-(Vdiff*GAIN1)/2 \geq

Vcm+(Vdiff*GAIN1)/2 < 2.5V \geq

0V < 1.2V-(Vdiff*GAIN_PGA)/2 \geq

1.2V+(Vdiff*GAIN_PGA)/2 < 2.5V \geq

If the approximate range of the input signal is known, it can be directly substituted into the above formula to calculate and select the available gain setting.

For unknown input signal, you can first use the x1 gain setting to determine the approximate range, and then substitute it into the above formula to calculate and select the available gain setting.

26.4 Register Configuration for Typical Use Cases

The following configuration words are all hexadecimal numbers, and the high-order 0 has been omitted.

26.4.1 Offset measurement

1. Set the working state of the module, set register 0X04 to 0xE03, and enable channel selection (offset detection)/SDADC_CHOP to enable

enable/SDADC enable/LDO enable

2. Set module register 0X08=0x3, PGA is x1 gain configuration/PGA_CHOP enable/PGA enable. Among them, the gain configuration

0X08 BIT[8:4] should be modified according to the requirements of the test (that is, what gain configuration is used for the subsequent test, offset measurement

also use the same gain configuration)

3. Set the clock state 0X40000E14 BIT[15:8]=0x28, and confirm that the SDADC clock is configured as 1MHz. The default value after the chip is powered on

That is 0x28, if there is no change, this step can be omitted

4. Wait for 2ms, read the output result of ADC_RESULT register, refer to Section 26.3.7, need software to convert back to unsigned number (most

The high bit ADC_RESULT[17] is inverted), and then the lowest two bits are discarded, and only the high 16bit data is retained. Repeat the reading for a total of 10 times, the reading interval is 2ms,

Calculate the average number of codes and record it as code_offset

26.4.2 Differential Input Mode

1. First measure the offset according to Section 26.4.1 to get code_offset

2. Set the module register 0X04=0x803, channel selection (positive channel 0, negative channel 1)/SDADC_CHOP enable

/SDADC enable/LDO enable. The channel selects 0X04[11:8], which can be configured as the differential input of other channels, refer to 26.3.2

Festival:

3. Set module register 0X08=0x3, PGA is x1 gain configuration/PGA_CHOP enable/PGA enable. Among them, the gain configuration

0X08[8:4] Select the appropriate configuration according to the input signal voltage range, refer to Sections 26.3.4 and 26.3.8

4. Set the clock state 0X40000E14[15:8]=0x28, and confirm that the SDADC clock is configured as 1MHz. After the chip is powered on, the default value is

28. If there is no change, this step can be omitted

5. Wait 2ms, read the output result of ADC_RESULT register, register address 0X00[17:0]. Refer to Section 26.3.7, required

The software first converts back to an unsigned number (the highest bit ADC_RESULT[17] is inverted), and then discards the lowest two bits, and only retains the high 16bit data. 6.

Repeat the reading for a total of 10 times, the reading interval is 2ms, and the average value of the code number is calculated as code_out

7. Differential input signal Vindiff=(code_out-code_offset)*LSB/GAIN_PGA, where LSB is approximately equal to

68.5uV

26.4.3 SINGLE-ENDED INPUT MODE

1. First measure the offset according to Section 26.4.1 to get code_offset;

2. Set module register 0X04=0x3, channel selection (positive channel 0, negative VCMIN)/SDADC_CHOP enable/SDADC

Enable/LDO enable. The channel selects 0X04[11:8], which can be configured as the single-ended input of other channels, refer to Section 26.3.2

Set module register 0X408=0x3, PGA is x1 gain configuration/PGA_CHOP enable/PGA enable. Among them, VCMIN configuration

0X40000D20[16:11] and gain configuration 0X08[8:4], select the appropriate configuration according to the input signal voltage range, refer to 26.3.4,

Sections 26.3.5 and 26.3.8

3. Set the clock state 0X40000E14[15:8]=0x28, and confirm that the SDADC clock is configured as 1MHz. After the chip is powered on, the default value is

28. If there is no change, this step can be omitted;

4. Wait 2ms, read the output result of ADC_RESULT register, register address 0X00[17:0]. Refer to Section 26.3.7, required

The software first converts back to an unsigned number (the highest bit ADC_RESULT[17] is inverted), and then discards the lowest two bits, and only retains the high 16bit data. repeat

Read a total of 10 times, the reading interval is 2ms, and the average value of the code number is calculated as code_out;

5. Single-ended mode input signal vin_single=(code_out-code_offset)*LSB/ GAIN_PGA+VCMIN, where LSB according to

The simulation result is approximately equal to 68.5uV, and the default value of VCMIN is 1.311V;

26.4.4 Temperature detection mode

1. Set the module register 0X04=0xC03, channel selection (temperature detection)/SDADC_CHOP enable/SDADC enable/LDO enable

can;

2. Set module register 0X08=0x183, PGA is x4 gain configuration/PGA_CHOP enable/PGA enable;

3. Set the clock state 0X40000E14[15:8]=0x28, and confirm that the SDADC clock is configured as 1MHz. After the chip is powered on, the default value is

28. If there is no change, this step can be omitted;

4. Set the working state of tempsensor module 0X0C=0x1, TempSensor enable/cal_offset_temp12=0,

TempSensor is configured for x2 gain;

5. Wait for 2ms, read the output result of ADC_RESULT register, refer to Section 26.3.7, need software to convert back to unsigned number (most

The high-order ADC_RESULT[17] is inverted), recorded as output code code_out1;

6. Set the working state of tempsensor module 0X0C=0x3, TempSensor enable/cal_offset_temp12=1,

TempSensor is configured for x2 gain;

7. Wait 2ms, read the output result of ADC_RESULT register, refer to Section 26.3.7, need software to convert back to unsigned number (most

The high-order ADC_RESULT[17] is inverted), recorded as the output code code_out2;

8. Repeat steps 4 to 7 for a total of 10 times to calculate the average values of code_out1avg and code_out2avg of code_out1 and code_out2 respectively.

code_out2avg \bar{y}

9. Calculate Vtemp=(code_out1avg-code_out2avg)/16 from the above steps, and TempSensor output voltage=Vtemp

Code number* (LSB/4), where LSB is approximately equal to 68.5uV according to the simulation result (this LSB is the simulation result according to 16bit, the temperature detection above

In steps 5 and 7, the reading is 18bit data, 18bit number \bar{y} 16bit number * 4, then the least significant bit corresponding to 18bit \bar{y} LSB/4)

10. According to the test results, the fitting formula has been sorted out, which can directly estimate the chip temperature

Vtemp code=(15.548*Chip temperature)+4444.1

Substitute the Vtemp calculated in step 9 into the above formula to obtain the die temperature. (Note that the fitting formula is in decimal)

Also, add an optional measure to improve accuracy:

For different chips, assuming that the coefficient k in the fitting formula $y=kx+b$ is constant (both are 15.548), the coefficient b of different chips is different to some extent.

According to the above instructions, first calculate the Vtemp code at a known ambient temperature (eg room temperature 25°C), and substitute it into the Vtemp code

$=15.548*(\text{known ambient temperature}+11.8)+b$, get the coefficient b of different chips, save it in rom or flash, replace the above formula

4444.1. Then, the temperature is calculated according to the coefficient of the new formula, which can improve the temperature detection accuracy of different chips to a certain extent.

26.4.5 Voltage Detection Mode

1. First measure the offset according to Section 26.4.1 to get code_offset;
2. Set module register 0X04=0xD03, channel selection (voltage detection)/SDADC_CHOP enable/SDADC enable/LDO enable

can

3. Set module register 0X08=0x83, PGA is x2 gain configuration/PGA_CHOP enable/PGA enable.
4. Set the clock state 0X40000E14[15:8]=28, and confirm that the SDADC clock is configured as 1MHz. After the chip is powered on, the default value is 28.

If there is no change, this step can be omitted

5. Wait for 2ms, read the output result of ADC_RESULT register, refer to Section 26.3.7, need software to convert back to unsigned number (most

The high bit ADC_RESULT[17] is inverted), and then the lowest two bits are discarded, and only the high 16bit data is retained. Repeat the reading for a total of 10 times, the reading interval is 2ms,

Calculate the average number of codes and record it as code_out;

6. Chip power supply voltage $V_{power} = (code_out - code_offset) * LSB / 2 + 1.2V$, where LSB is approximately equal to the simulation result

68.5uV

26.5 ADC Register Descriptions

26.5.1 Register List

Table 208 ADC register list

offset address name		abbreviation	describe	reset value
0X0000	ADC result register	ADC_RESULT	Store ADC acquisition value and data comparison value	0X0000_0000
0X0004	ADC analog register	ADC_ANA_CTRL	Configure ADC related functions	0X0000_0004
0X0008	PGA Configuration Register	PGA_CTRL	Configure PGA related functions;	0x0000_0000
0X000c	TempSensor configuration register <small>device</small>	TEMP_CTRL	Configure temperature sensor related functions; 0x0000_0000	
0x0010	ADC Configuration Register	ADC_CTRL	Configure ADC module function;	0x0050_0500
0x0014	ADC Interrupt Register	ADC_INT_STATUS	ADC Module Interrupt Status Register; 0x0000_0000	
0x0018	Compare Value Register	CMP_VALUE	Compare threshold settings	0x0000_0000

26.5.2 ADC Result Register

Table 209 ADC result register

bit access		Instructions	reset value
[17:0] RW	ADC conversion	result value, valid bit width is 18bits. Signed number, the most significant bit is the sign bit.	1'b0

26.5.3 ADC ANALOG CONFIGURATION REGISTER

Table 210 ADC configuration registers

Bit Access	Operating Instructions		reset value
[11 : 8] RW	<p>ADC working channel selection signal:</p> <p>4'b0000: AIN0 channel works. In DMA mode, the corresponding DMA channel 0 works</p> <p>4'b0001: AIN1 channel works. In DMA mode, the corresponding DMA channel 1 works</p> <p>4'b0010: AIN2 channel works. In DMA mode, the corresponding DMA channel 2 works</p> <p>4'b0011: AIN3 channel works. In DMA mode, the corresponding DMA channel 3 works</p> <p>4'b0100~RSV</p> <p>4'b0101 ~ RSV</p> <p>4'b0110 ~ RSV</p> <p>4'b0111 ~ RSV</p> <p>4'b1000: AIN0/AIN1 differential signal input. In DMA mode, the corresponding DMA channel 0 works</p> <p>4'b1001: AIN2/AIN3 differential signal input. In DMA mode, the corresponding DMA channel 2 works</p> <p>4'b1010~RSV</p> <p>4'b1011~RSV</p> <p>4'b1100: Temperature sensor input, corresponding to DMA channel 2</p> <p>4'b1101: Voltage detection module input, corresponding to DMA channel 3</p> <p>4'b1110: offset detection input;</p> <p>4'b1111~RSV</p>		4'b1000
[7]	RO	RSV	1'b0
[6:5] RW		chop_enr	2'b00

		LDO chopper signal PD signal	
[4]	RW	<p>Chop_ens</p> <p>sdadc chopped signal PD signal</p> <p>0: enable</p> <p>1: Disable</p>	1'b0
[3]	RO	RSV	1'b0
[2]	RW	<p>Pd_sdadc</p> <p>sdadc analog module power down enable</p> <p>1: Power down</p> <p>0: work</p>	1'b1
[1]	RW	<p>Rstn_sdadc</p> <p>Reset signal of the digital logic part of the analog module</p> <p>0: reset</p> <p>1: normal work</p>	1'b0
[0]	RW	<p>en_ldo_sdadc</p> <p>ADC LDO enable</p> <p>0: Power down 1: Working</p>	1'b0

26.5.4 PGA CONFIGURATION REGISTERS

Table 211 PGA configuration registers

Bit Access	Operating Instructions		reset value
[8:4] RW	Gain_ctrl_pga PGA gain configuration; BIT[8:7] configure GAIN2, BIT[6:4] configure GAIN1, please refer to Section26.3.4 for the specific gain table		5'd0
[3]	RW Bypass_pga pga bypass signal 1: Bypass pga 0: do not bypass		1'b0
[2]	RW Bypass_ref Internal voltage reference bypass signal 1: Bypass the internal reference voltage 0: do not bypass		1'b0
[1]	RW Chop_enp PGA chop enable signal 1: enable 0: Disable		1'b0
[0]	RW En_pga Pga enable signal 1: enable 0: Disable		1'b0

26.5.5 TEMP CONFIGURATION REGISTER

Table 212 Temperature Sensor Configuration Register

Bit Access	Operating Instructions		reset value
[5:4] RW		Gain_temp temp gain control Coding Gain 00 -- 2 01 -- 4 10 -- 6 11 -- 8	2'd0
[3:2] RO		RSV	2'b0
[1]	RW	Cal_offset_temp12 TEMPSEN offset calibration function	1'b0
[0]	RW	ON_TEMP 1: temp enable 0: temp is not enabled	1'b0

26.5.6 ADC Function Configuration Register

Table 213 Temperature Sensor Configuration Register

Bit Access	Operating Instructions		reset value
[29:20] RW		ana_swth_time After the software switches the data channel, the time required for the analog circuit to remain stable, the default is 80 pclk, or 2us	10'h050

[19:18] RO		RSV	2'b0
[17:8] RW		<p>ana_init_time</p> <p>After the software starts adc_start, the time required for the analog circuit to remain stable, the default is 80 pclk, that is, 2us</p>	10'h050
[7]	RO	RSV	1'b0
[6]	RW	<p>Cmp_pol</p> <p>0: interrupt when adc_result >= cmp_value</p> <p>1: Interrupt when adc_result<cmp_value</p>	1'b0
[5]	RW	<p>Cmp_int_ena</p> <p>1: Compare interrupt enable</p> <p>0: Compare interrupt disabled</p>	1'b0
[4]	RW	<p>Adc_cmp_enable</p> <p>1: adc compare function is enabled</p> <p>0: adc comparison function is disabled</p>	1'b0
[3:2]	RO	Reserved	2'b0
[1]	RW	<p>Adc_int_ena</p> <p>1: ADC data conversion interrupt enable</p> <p>0: ADC data conversion interrupt disabled</p>	1'b0
[0]	RW	<p>Adc_dma_enable</p> <p>1: dma enable</p> <p>0: dma is not enabled</p>	1'b0

26.5.7 ADC Interrupt Status Register

Table 214 ADC Interrupt Status Register

Bit Access	Operating Instructions		reset value
[1]	RW	Cmp_int Compare interrupt flag bit, set by hardware, cleared by software by writing 1	1'b0
[0]	RW	Adc_int Data conversion complete interrupt, hardware set, software write 1 to clear	1'b0

26.5.8 Compare Threshold Register

Table 215 Compare Threshold Register

Bit Access	Operating Instructions		reset value
[17:0] R/W		Cmp_value value to compare	18'h00000

27 Touch Sensor

27.1 Overview of Module Functions

The basic functions of the module are as follows:

↳ Supports up to 16 Touch Sensor scans;

↳ Record the scanning results of each Touch Sensor;

↳ Reporting scan results through interrupts;

27.2 Function Instructions

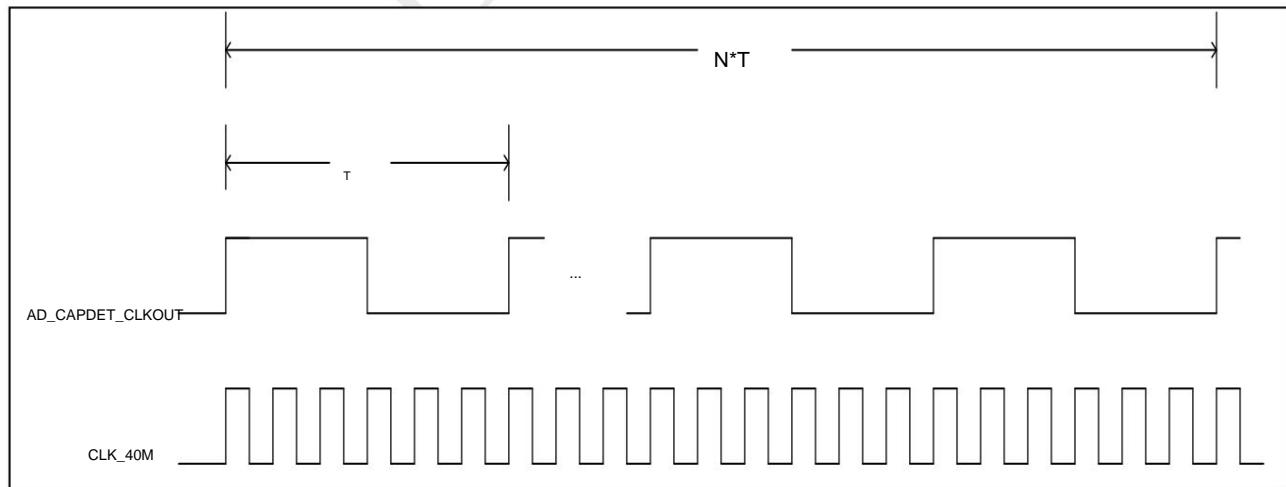
The touch button module is integrated in the W800 system. The basic principle is that when the button is touched, the capacitance value of the circuit on the button will change, so that the

Affects the output clock frequency in the module. By detecting the frequency of the output clock of the module, it can be judged whether the capacitance value has changed, so as to judge

The state of the button.

The basic function of this digital module is to scan the state of each touch key at regular intervals, count and record within the set time window.

the state of each button. If it exceeds the set threshold, it is judged that the button is touched, and it is reported to the MCU system through an interrupt.



27.2.1 Basic Workflow

1. Set the Touch_CR register, configure the scan cycle, scan window, and select the IO to be scanned. In Touch_CR

scan_period is the interval period of each scan, the unit is 16ms. That is, if the register is set to 10, every 160ms will be

Scan 16 touch keys. CAPDET_CNT is the window counted when scanning each IO state, which is N in the above figure. It should be noted that for

To avoid the jitter caused by switching the channel, the counting will start at the beginning of the third pulse after each switching of the scan IO, so if this register is set

is N, the actual count window is N-2.

2. Set the count threshold corresponding to each Touch IO. If the result of the scan exceeds the base + threshold, the button can be considered to be touched;

3. Enable Touch_CR[0], the module starts to work.

4. After the touch key module is enabled, the hardware will scan the selected IO one by one, and store the count value of each IO as the base.

Then scan the selected IO successively every scan_period, and store the currently scanned value. After all IO scans are completed, the

The count value of each IO is compared with the radix, if the current value > radix+threshold, the button is considered to be touched, in register 0x44

The corresponding bit in PAD_STATUS will be set to 1 and reported to the system through an interrupt. PAD_STATUS Write 1 to clear 0.

27.3 Register List:

Table 216 Touch Sensor Controller Register List

offset address name		abbreviation	describe	Defaults
0X0000_0000 Control Register		Touch_CR	Touch Sensor Controller Settings	0X0000_0000

0X0000_0004			Threshold control and count value of each touch key 0X0000_0000	
-	One-way control of touch buttons	Touch_Sensor x		
0X0000_0040				
0x0000_0044 Interrupt	Controller	Int_Source	interrupt controller	0x0000_0000

27.3.1 Touch Sensor Control Register

Table 217 Touch Sensor Control Setting Register

bit access		Instructions	reset value
[31:26] RW	Scan_period	<p>Scan period, the unit is 16ms; for example, if scan_period is set to 6'd10, then scan every 160ms.</p> <p>key state;</p>	6'd10
[25:20] RW	CAPDET_CNT	<p>Select the number of CAPDET output pulses as the count window.</p> <p>Note: In order to avoid jitter caused by switching channels, counting will not start until the third pulse, so this register is set to N+2.</p> <p>If it is set to N, the actual count window is N-2. For example, if it is set to 6'd20, the cycle of 18 CAPDET pulses is the count window.</p>	6'd20
[19:4] RW	Touch Sensor	<p>key selection; scan the touch key state of the corresponding bit.</p> <p>0x0000: do not scan;</p> <p>0x0001: scan the first key;</p> <p>0x0002: scan the second key;</p>	16'd0

		0x0003: scan the first and second keys; ... 0xFFFF: scan all 16 keys;	
[3:1]	RW RSV		3'd0
[0]	RW Touch	key controller enable 1: Enable touch key scan; 0: Disable touch key scanning;	1'b0

27.3.2 Touch key single control register

Table 218 Touch key single channel setting register

bit access		Instructions	reset value
[22:8] RO		Touch Senor 1 count value	14'd0
[7]	RO	RSV	1'b0
[6:0]	RW Touch	Senor 1 Threshold	7'd50

27.3.3 Interrupt Control Register

Table 219 Touch key interrupt control register

bit access		Instructions	reset value
[31:16]	RW INT_EN	When the corresponding bit is 1, it means that the corresponding IO is triggered and an interrupt will be generated; When the corresponding bit is 0, it means that the corresponding IO is triggered and will not generate an interrupt;	16'd0

[15:0]	RW PAD_STATUS/INT_SOURCE	<p>When the bit is 1, it means that the corresponding PAD is triggered;</p> <p>When the bit is 0, it means that the corresponding PAD is not triggered;</p> <p>write 1 clear 0</p>	16'd0

28 W800 Security Architecture Design

28.1 Functional overview

XT804 expresses safety features of bus access via HPROT signal

	0	1
HPROT[3]	uncacheable	cacheable
HPROT[2]	un-security	security
HPROT[1]	User	super
HPROT[0]	Code	data

W800's support for security architecture is divided into security access control for sram and access control for peripherals.

28.1.1 SRAM Secure Access Controller (SASC)

Level 1 bus SRAM, Level 2 bus SRAM and FLASH each have a security access controller, each SASC controls the storage space have the following safety features

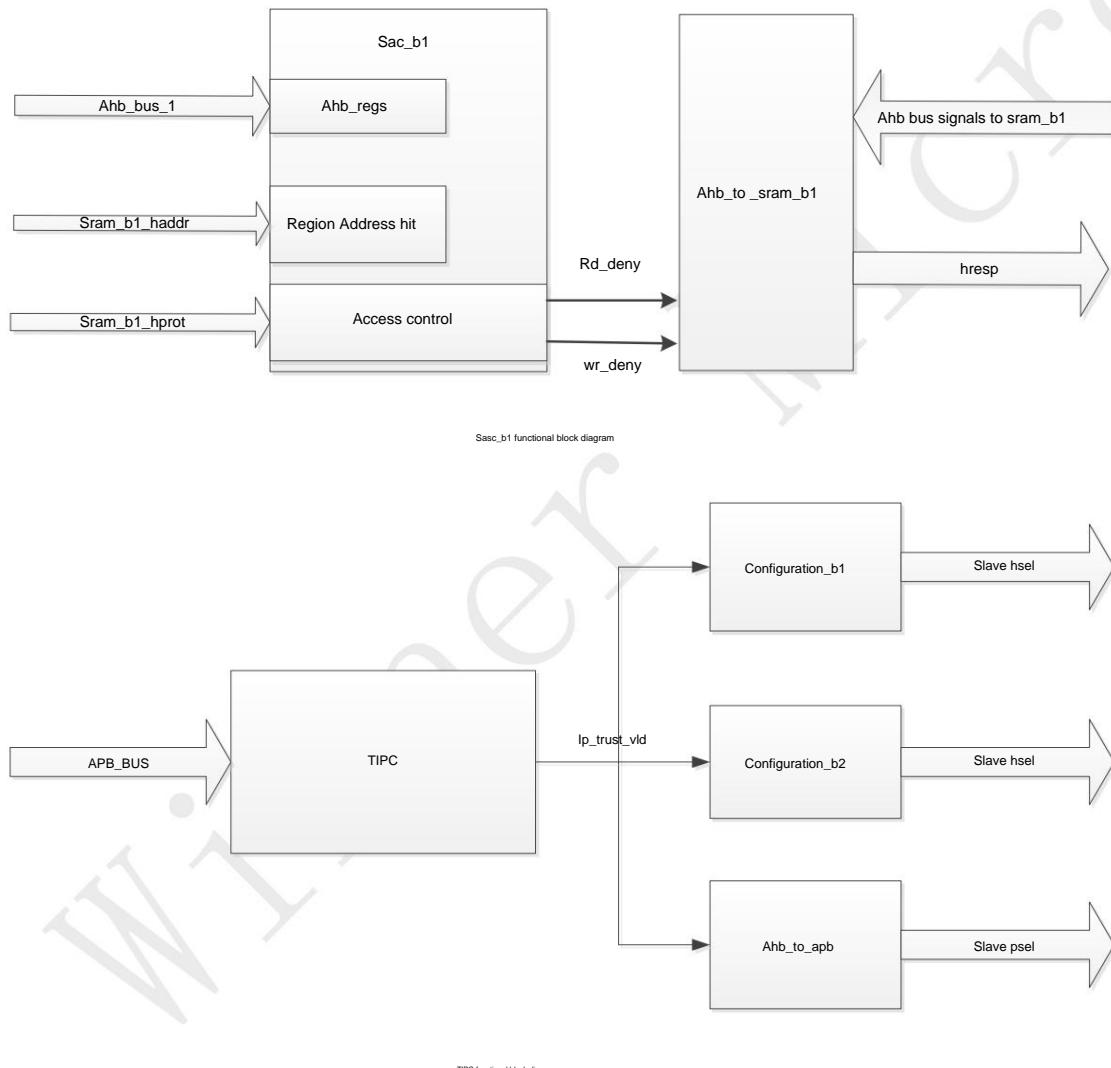
- ÿ 8 configurable security zones
- ÿ Each security area can be configured to a minimum of 4 bytes
- ÿ Security-super provides unrestricted access to all areas
- ÿ If the access of the CPU through the AHB bus is denied, it will generate a wrong response signal (HRESP=1), from An access exception is thrown instead.

ÿ No exception will be generated if the access of other master devices such as DMA on the bus is denied.

28.1.2 Trusted IP Controller (TIPC)

Hanging on the APB bus, it is used to configure the trust authority of all peripherals including the primary bus, the secondary bus and the APB bus. If the peripheral is configured as a trusted device (`ip_trust_vld=1`), then only access on the bus is security (HPROT[2]=1 or PPROT[2]=1) Only then can the peripherals be accessed normally, otherwise the read and write will be rejected.

28.2 Security Architecture Block Diagram



28.3 Register Description

28.3.1 SASC Register List

name offset	address	bit threshold	instruction	reset value
CAR	0x0	[31:16]	Reserved unused	-
		[15:14]	sram region7 configuration properties, same as region0 2'b00	
		[13:12]	sram region6 configuration properties, same as region0 2'b00	
		[11:10]	sram region5 configuration properties, same as region0 2'b00	
		[9:8]	sram region4 configuration properties, same as region0 2'b00	
		[7:6]	sram region3 configuration properties, same as region0 2'b00	
		[5:4]	sram region2 configuration properties, same as region0 2'b00	
		[3:2]	sram region1 configuration properties, same as region0 2'b00	
		[1:0]	sram region0 configuration property 00: un-security-user 01: un-security-super 10: security-user 11: security-super	2'b00
		31:8	unused reserve	-
CR	0x4	7	region7 attribute control register, same as region0	
		6	region6 attribute control register, same as region0	
		5	region5 attribute control register, same as region0	
		4	region4 attribute control register, same as region0	
		3	region3 attribute control register, same as region0	

		2	region2 attribute control register, same as region0	
		1	region1 attribute control register, same as region0	
		0	region0 attribute control register 0: Attribute invalid 1: Attribute valid	
	AP0 0x8	[31:16]	Reserved unused	
		[15:14]	region7 AP configuration for un-security user	
		[13:12]	region6 AP configuration for un-security user	
		[11:10]	region5 AP configuration for un-security user	
		[9:8]	region4 AP configuration for un-security user	
		[7:6]	region3 AP configuration for un-security user	
		[5:4]	region2 AP configuration for un-security user	
		[3:2]	region1 AP configuration for un-security user	
		[1:0]	region0 AP configuration for un-security 00: R / W 01: RO 10: WO 11: No	

			Access	
CD0	0xC	[31:16]	Reserved unused	
		[15:14]	region7 CD configuration for un-security user	
		[13:12]	region6 CD configuration for un-security user	
		[11:10]	region5 CD configuration for un-security user	
		[9:8]	region4 CD configuration for un-security user	
		[7:6]	region3 CD configuration for un-security user	
		[5:4]	region2 CD configuration for un-security user	
		[3:2]	region1 CD configuration for un-security user	
		[1:0]	region0 CD configuration, for un-security user 00: Date Access , Opcode Close 01: Data Access 10: Opcode Close 11: Data, Opcode all deny	

AP1	0x10	[31:16]	Reserved unused	
		[15:14]	region7 AP configuration for un-security super	
		[13:12]	region6 AP configuration for un-security super	
		[11:10]	region5 AP configuration for un-security super	
		[9:8]	region4 AP configuration for un-security super	
		[7:6]	region3 AP configuration for un-security super	
		[5:4]	region2 AP configuration for un-security super	
		[3:2]	region1 AP configuration for un-security super	
		[1:0]	region0 AP configuration for un-security super 00: R / W 01: RO 10: WO 11: No Access	
CD1	0x14	[31:16]	Reserved unused	
		[15:14]	region7 CD configuration for un-security super	

		[13:12] region6 CD configuration for un-security super	
		[11:10] region5 CD configuration for un-security super	
		[9:8] region4 CD configuration for un-security super	
		[7:6] region3 CD configuration for un-security super	
		[5:4] region2 CD configuration for un-security super	
		[3:2] region1 CD configuration for un-security super	
		[1:0] region0 CD configuration, for un-security super 00: Date Access , Opcode Close 01: Data Access 10: Opcode Close 11: Data, Opcode all deny	
AP2	0x18	[31:16] Reserved unused	
		[15:14] region7 AP configuration for security-user	
		[13:12] region6 AP configuration for security-user	
		[11:10] region5 AP configuration for security-user	

		[9:8]	region4 AP configuration for security-user	
		[7:6]	region3 AP configuration for security-user	
		[5:4]	region2 AP configuration for security-user	
		[3:2]	region1 AP configuration for security-user	
		[1:0]	region0 AP configuration for security-user 00: R / W 01: RO 10: WO 11: No Access	
CD2	0x1C	[31:16]	Reserved unused	
		[15:14]	region7 CD configuration for security-user	
		[13:12]	region6 CD configuration for security-user	
		[11:10]	region5 CD configuration for security-user	
		[9:8]	region4 CD configuration for security-user	
		[7:6]	region3 CD configuration for security-user	
		[5:4]	region2 CD configuration, for security-user	
		[3:2]	region1 CD configuration, for security-user	
		[1:0]	region0 CD configuration, for security-user 00: Date Access , Opcode Close 01: Data Access 10: Opcode Close 11: Data, Opcode all deny	
REGION0	0x20	31:n+1	reserved	0
		n:8	BAddr0, region0 base address, n and sram are larger	

			little related, sram=32kB ñ=20 sram=64kB, n=21	
		7:5 reserved		3'b000
		4:0	RSizeëregion0 size 00101: 4B 00110: 8B 10001: 16KB 10010: 32KB 	
	REGION1 0x24		31:n+1 reserved	0
		n:8	Baddr1, region1 base address	
		7:5 reserved		3'b000
		4:0	region1 size	
	REGION2 0x28		31:n+1 reserved	0
		n:8	region2 base address	
		7:5 reserved		3'b000
		4:0	region2 size	
	REGION3 0x2C		31:n+1 reserved	0
		n:8	region3 base address	
		7:5 reserved		3'b000

		4:0	region3 size	
REGION4 0x30		31:n+1 reserved		0
		n:8	region4 base address	
		7:5 reserved		3'b000
		4:0	region4 size	
REGION5 0x34		31:n+1 reserved		0
		n:8	region5 base address	
		7:5 reserved		3'b000
		4:0	region5 size	
REGION6 0x38		31:n+1 reserved		0
		n:8	region6 base address	
		7:5 reserved		3'b000
		4:0	region6 size	
REGION7 0x3C		31:n+1 reserved		0
		n:8	region7 base address	
		7:5 reserved		3'b000
		4:0	region7 size	

28.3.2 TIPC register

name offset	address	bit threshold	instruction	reset value
ip_trust_vld0 0x0		31:18 unused		0
		17	BT modem trusted attributes 1: Trusted 0: Untrustworthy	0
		16	I2S Trusted Properties	0
		15	PWM Trusted Properties	0
		14	LCD driver trusted attributes	0
		13	RF controller trusted attributes	0
		12	timer trusted attribute	0
		11	watch dog trusted attributes	0
		10	PORTB Trusted Properties	0
		9	PORTA Trusted Properties	0
		8	UART5 Trusted Properties	0
		7	UART4 Trusted Properties	0
		6	UART3 Trusted Properties	0
		5	UART2 Trusted Properties	0
		4	UART1 Trusted Properties	0
		3	UART0 Trusted Properties	0

		2	SPI MASTER Trusted Properties	0
		1	SAR ADC Trusted Properties	0
		0	I2C Trusted Properties	0
ip_trust_vld1 0x4		31:18 unused		0
		17	RF BIST Trusted Attribute Configuration 1: Trusted 0: Untrustworthy	0
		16	SDIO Wrapper Trusted Properties	0
		15	SPI_HS Trusted Properties	0
		14	SDIO Trusted Properties	0
		13	unused	0
		12	SEC Trusted Properties	0
		11	MAC Trusted Properties	0
		10	BBP Trusted Properties	0
		9	MMU Trusted Properties	0
		8	Clock reset control module trusted attribute 0	
		7	PMU Trusted Properties	0
		6	BT Trusted Properties	0
		5	GPSEC Trusted Properties	0
		4	DMA Trust Properties	0
		3	RSA Trusted Properties	0
		2	PSRAM Controller Trusted Properties	0
		1	FLASH Controller Trusted Properties	0

		0	SDIO HOST Trusted Properties	0
--	--	---	------------------------------	---

28.4 Instructions for use

28.4.1 Safe Memory Access (SASC)

28.4.1.1 ACCESS TO REGISTERS

Access to the SASC register follows the following principles:

- ÿ The CAR register can only be read and written when the cpu is security-super.
- ÿ When the cpu is in un-security-super, you can access the region-related registers configured as un-security-user

device position.

- ÿ When the cpu is security-super, all registers can be read and written.

- ÿ The registers are otherwise inaccessible

For example, when CAR is set to 16'he4e4, it means that region0 and region4 are set to un-security-user. At this time, if the cpu

In un-security-super, the cpu can read and write registers REGION0 and REGION4, as well as CR[0], CR[4], APx[1:0],

APx[9:8], CDx[1:0], CDx[9:8]ÿ

28.4.1.2 Setting of protection area address

Each SASC supports 8 configurable memory protection range regions, and the base address in REGIONx[31:8] is the one you want to configure

25 to 2 bits of the actual physical address of the storage area. At the same time, the size and base address of Size need to meet the following requirements:

SIZE

00101: 4B;

00110: 8B; Baddrx[0] = 0

00111: 16B; Baddrx[1:0] = 0

01000: 32B; Baddrx[2:0] = 0

01001: 64B; Baddrx[3:0] = 0

01010: 128B; Baddrx[4:0] = 0

01011: 256B; Baddrx[5:0] = 0

01100: 512B; Baddrx[6:0] = 0

01101: 1KB; Baddrx[7:0] = 0

01110: 2KB; Baddrx[8:0] = 0

01111: 4KB; Baddrx[9:0] = 0

10000: 8KB; Baddrx[10:0] = 0

10001: 16KB; Baddrx[11:0] = 0

For example, if 20000100 is used as the base address of region0, then the maximum size of region0 can be set to 256B.

region0 is set to 128B, and the REGION0 register should be filled with 0x0000400a. If 20040000 is used as the base address, the region

The maximum can be set to 64KB. If a 64KB region is to be defined, the register should be filled with 0x01000013.

28.4.1.3 Access rights to memory

Priority of the four authority for mem:

request mem	Security super	Security user	Un-Security super	Un-Security user
Security super	✓ +	✓ +	✓ +	✓ +
Security user	X	✓	X	X
Un-Security super	X	X	✓	✓ +
Un-Security user	X	X	X	✓

Note: ✓ + all access, include read, write, data access, opcode fetch

✓ access based on the current attribute

X no access

According to the picture above

ÿ The bus access of security-super can access sram at will

ÿ The bus access of un-security-super can be freely configured as un-security-user

region

ÿ The bus access of security-user can only access security-user according to the current AP and CD properties

region

ÿ The bus access of un-security-super can access un by current AP and CD properties

security-super ÿ region

ÿ The bus access of un-security-user can only access un according to the current AP and CD attributes

security-user ÿ region

APx and CDx registers are used to set the access attributes when each region corresponds to different permissions, where CDx registers

The register is only valid for read operations during bus access, that is, to set whether to allow read data and read commands. APx registers are used for

Set whether to allow read/write operations.

For example, if CAR[1:0] is set to 00, and CR[0] is 1, it means REGION 0 is un-security-user, and

And enable permission protection, at this time, if the bus access is security-super or un-security-super, you can follow the

intended access to REGION 0; if the bus access is Security-user, any access will be denied; if the bus access is

Ask is un-security-user, AP0[1:0] is 01, which means read-only, CD0[1:0] is 01, which means data access, then

REGION 0 allows the bus to read data, other operations are denied.

If the AHB bus accesses the inaccessible area or the permission is wrong, the sasc module will give read deny

or write deny signal, thus returning an incorrect HRESP response signal, if the CPU is the one that initiated the bus access, then

A hardware exception will be generated, and if it is other devices, access will only be denied.

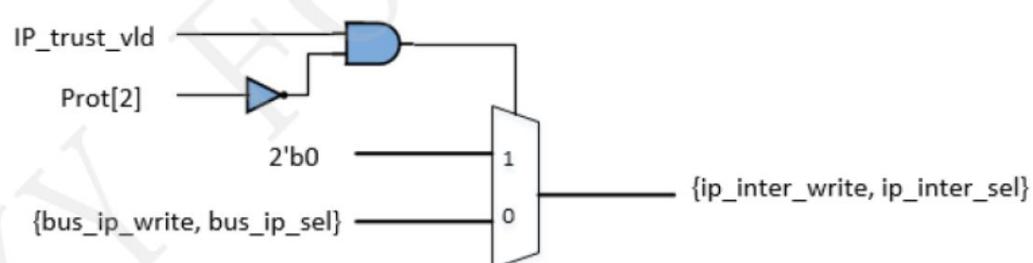
28.4.2 Trusted Access to Peripherals

The TIPC module can only be written when the PROT[2] signal is 1, that is, only in the trusted world can the IP address of each IP be written.

The letter attribute configuration register. The lp_trust_vld register is 0 by default, that is, all peripherals are untrusted. At this time, the peripherals

It can be accessed at will. If the ip_trust_vld corresponding to the peripheral is set, that is, the peripheral is set as a trusted device, only

Commands from the trusted world can access this peripheral.



29 Appendix 1. Chip Pin Definition

29.1 Chip Pinout

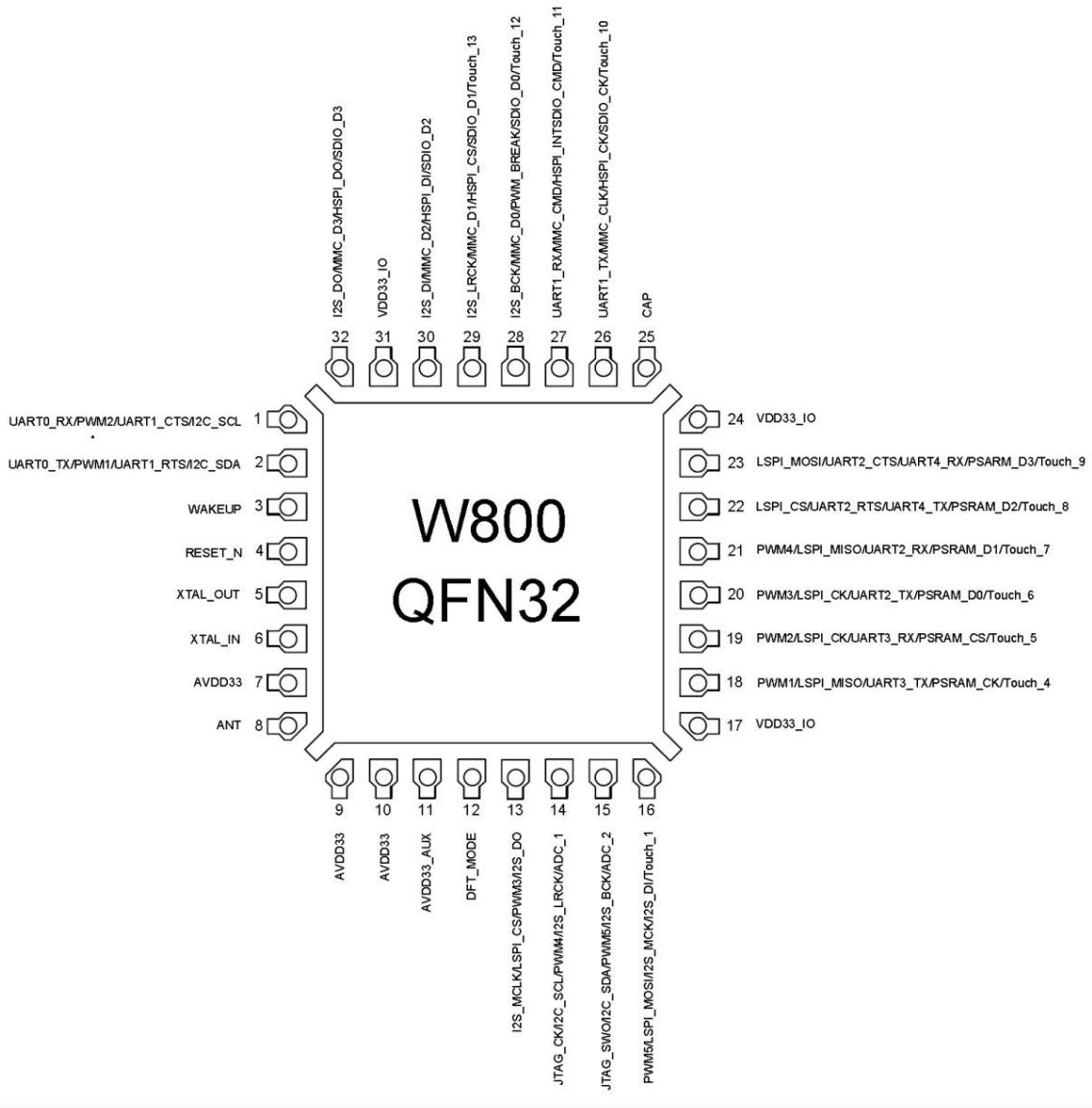


Figure 38 W800 chip pinout

Winner Micro

29.2 Chip pin multiplexing relationship

Table 220 Chip pin multiplexing relationship

Number	name	type	Pin function after reset	Multiplexing function	Maximum frequency pull-up and pull-down	capability drive capability
1	PB_20	I/O UART	RX	UART0_RX/PWM2/UART1_CTS/I2C_SCL	10MHz UP/DOWN	12mA
2	PB_19	I/O UART	TX	UART0_TX / PWM1 / UART1_RTS / I2C_SDA	10MHz UP/DOWN	12mA
3	WAKEUP	.	WAKEUP wake-up function		DOWN	
4	RESET	.	RESET reset		UP	
5	XTAL_OUT	O External	crystal output			
6	XTAL_IN	I External	crystal input			
7	AVDD33	P chip power supply, 3.3V				
8	ANT	I/O RF Antenna				
9	AVDD33	P chip power supply, 3.3V				
10	AVDD33	P chip power supply, 3.3V				
11	AVDD33_AUX	P chip power supply, 3.3V				
12	TEST	I Test function configuration pins				
13	BOOTMODE	I/O BOOTMODE		I2S_MCLK/LSPI_CS/PWM3/I2S_DO	20MHz UP/DOWN	12mA
14	PA_1	I / O JTAG_CK		JTAG_CK/I2C_SCL/PWM4/I2S_LRCK/ADC_1	20MHz UP/DOWN	12mA
15	PA_4	I / O JTAG_SWO		JTAG_SWO/I2C_SDA/PWM5/I2S_BCK/ADC_2	20MHz UP/DOWN	12mA
16	PA_7	I/O GPIO, input, high impedance		PWM5/LSPI_MOSI/I2S_MCK/I2S_DI/Touch_1	20MHz UP/DOWN	12mA

17	VDD33IO	P	IO power supply, 3.3V			
18	PB_0	I/O GPIO, input, high impedance	PWM1/LSPI_MISO/UART3_TX/PSRAM_CK/Touch_4 80MHz UP/DOWN			12mA
19	PB_1	I/O GPIO, input, high impedance	PWM2/LSPI_CK/UART3_RX/PSRAM_CS/Touch_5	80MHz UP/DOWN		12mA
20	PB_2	I/O GPIO, input, high impedance	PWM3/LSPI_CK/UART2_TX/PSRAM_D0/Touch_6	80MHz UP/DOWN		12mA
21	PB_3	I/O GPIO, input, high impedance	PWM4/LSPI_MISO/UART2_RX/PSRAM_D1/Touch_7 80MHz UP/DOWN			12mA
22	PB_4	I/O GPIO, input, high impedance	LSPI_CS/UART2_RTS/UART4_TX/PSRAM_D2/Touch_8	80MHz	UP/DOWN	12mA
23	PB_5	I/O GPIO, input, high impedance	LSPI_MOSI / UART2_CTS / UART4_RX / PSARM_D3 / Touch_9	80MHz	UP/DOWN	12mA
24	VDD33IO	P	IO power supply, 3.3V			
25	CAP	I External capacitor, 1μF				
26	PB_6	I/O GPIO, input, high impedance	UART1_TX/MMC_CLK/HSPI_CK/SDIO_CK/Touch_10 50MHz UP/DOWN			12mA
27	PB_7	I/O GPIO, input, high impedance	UART1_RX/MMC_CMD/HSPI_INT/SDIO_CMD/Touch_11	50MHz	UP/DOWN	12mA
28	PB_8	I/O GPIO, input, high impedance	I2S_BCK/MMC_D0/PWM_BREAK/SDIO_D0/Touch_12 50MHz UP/DOWN			12mA
29	PB_9	I/O GPIO, input, high impedance	I2S_LRCK/MMC_D1/HSPI_CS/SDIO_D1/Touch_13	50MHz UP/DOWN		12mA
30	PB_10	I/O GPIO, input, high impedance	I2S_DI/MMC_D2/HSPI_DI/SDIO_D2	50MHz UP/DOWN		12mA
31	VDD33IO	P	IO power supply, 3.3V			
32	PB_11	I/O GPIO, input, high impedance	2S_DO/MMC_D3/HSPI_DO/SDIO_D3	50MHz UP/DOWN		12mA
33	GND	P ground				

statement

Beijing Lianshengde Microelectronics Co., Ltd. reserves the right to modify and update Lianshengde Microelectronics products or various documents and materials at any time. Place

Updates will be released through the official channels of Lianshengde Microelectronics. Users must ensure that the correct information is obtained through official channels. United Sidley

Microelectronics does not undertake to notify everyone of updated information.