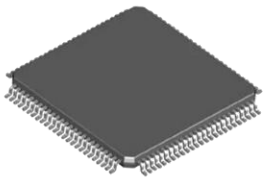
AG32 MCU

DATA SHEET

Revision:1.0

Features

AG32 MCU					
AGM Device	**AG32VF103**	**AG32VF107**	**AG32VF205**	**AG32VF303**	**AG32VF407**
Max Speed (MHz)	168	168	184	208	248
SRAM	128K	128K	128K	128K	128K
Flash	256K	1M	1M	1M	1M
CAN2.0	1	1	1	1	1
UART	5	5	5	5	5
I²C	2	2	2	2	2
Basic Timer	2	2	2	2	2
Advanced Timer	5	5	5	5	5
Support SDIO，Ethernet MAC	/	√	/	/	√
Support USB FS+OTG	√	√	√	√	√
Watchdog	√	√	√	√	√
3 x 12 bit,1M SPS ADCs(17 channels),2 DACs	√	√	√	√	√
2 x Comparator	√	√	√	√	√
RTC	√	√	√	√	√
SPI	√	√	√	√	√



LQFP100
14x14mm

LQFP64
10x10mm

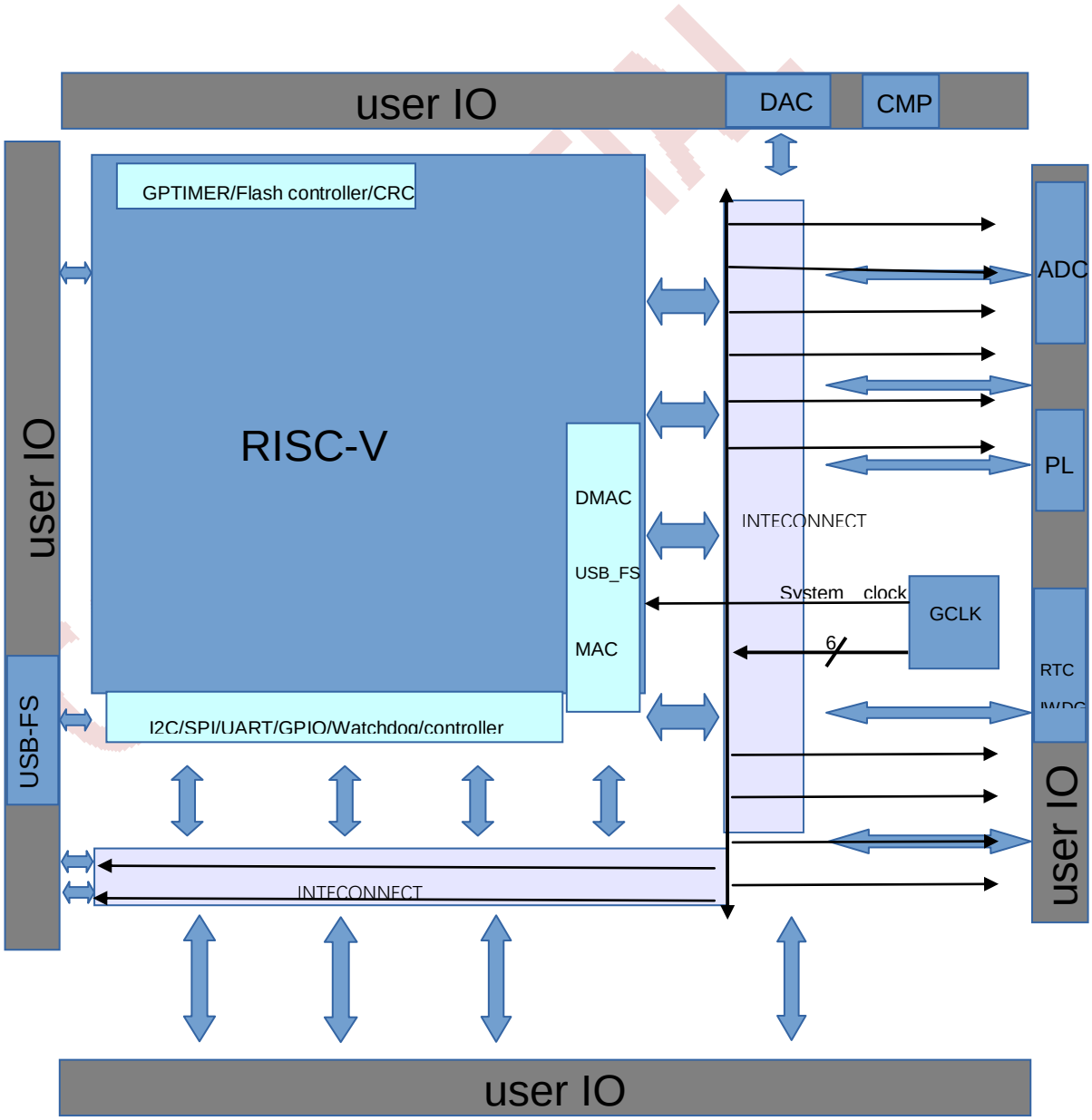
LQFP48
7x7mm

contents

1 Description

The AG32 family of 32-bit microcontrollers is designed to offer new degrees of freedom and rich compatible peripherals, and compatible pin and features to MCU users. AG32F103 product series offers supreme quality, stability, and exceptional pricing value.

1.1 Chip architecture

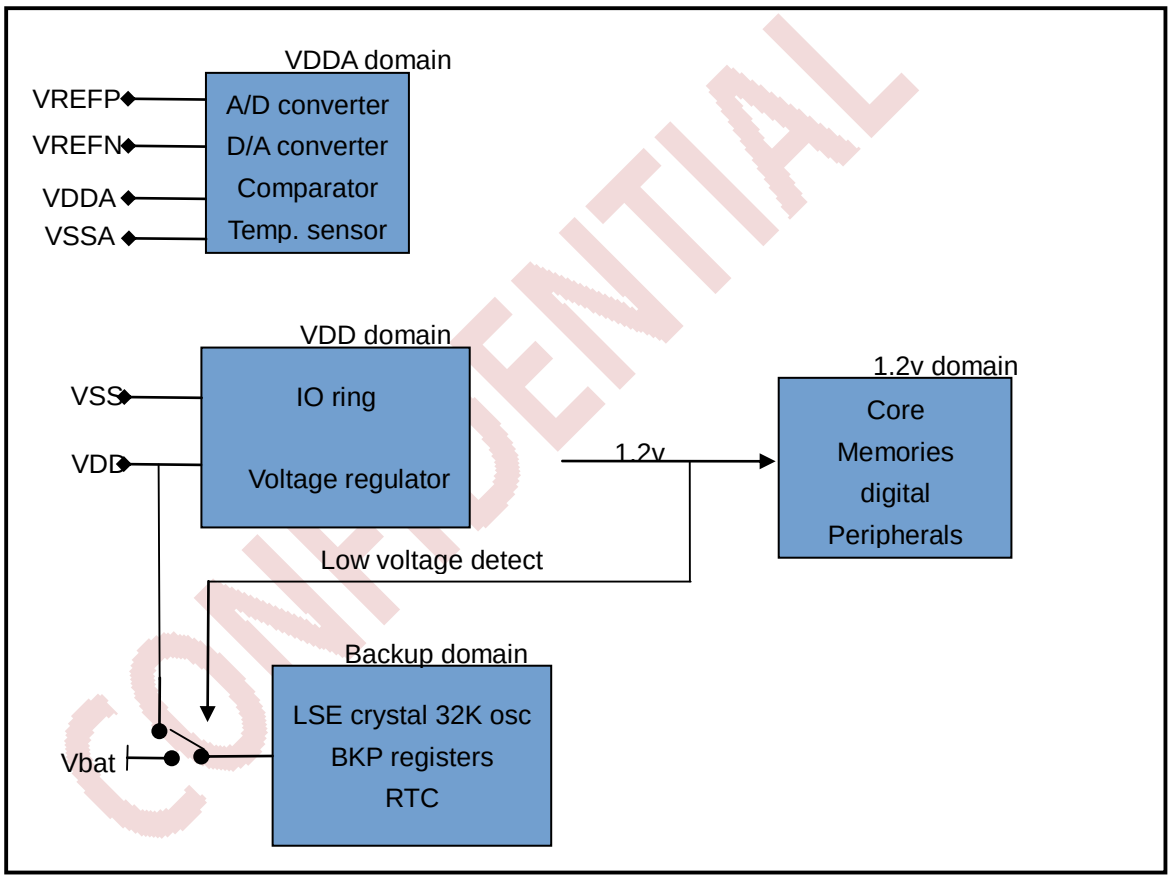


1.2 Power control

1.2.1 Power supplies

The AGRV2K requires a 3.0-to-3.6 V operating voltage supply (VDD). An embedded regulator is used to supply the internal 1.2V digital power.
The real-time clock (RTC) and backup registers can be powered from the VBAT voltage when the main VDD supply is powered off.

Power supply overview



1.2.2 Independent ADC and DAC converter supply and reference voltage

To improve conversion accuracy, the ADC and the DAC have an independent power supply which can be separately filtered and shielded from noise on the PCB.
(1) The ADC and DAC voltage supply input is available on a separate VDDA pin.
(2) An isolated supply ground connection is provided on pin VSSA.
When available (according to package), VREFN must be tied to VSSA.

To ensure a better accuracy on low-voltage inputs and outputs, the user can connect a separate external reference voltage on VREFP. VREFP is the highest voltage, represented by the full scale value, for an analog input (ADC) or output (DAC) signal. The voltage on VREFP can range from 3.0V to VDDA .

1.2.3 Battery backup domain

To retain the content of the Backup registers and supply the RTC function when VDD is turned off, VBAT pin can be connected to an optional standby voltage supplied by a battery or by another source.
The VBAT pin powers the RTC unit, the LSE oscillator and the PA5_RTC, OSC32_IN and OSC32_OUT Pins, allowing the RTC to operate even when the main digital supply (VDD) is turned off.
If no external battery is used in the application, it is recommended to connect VBAT externally to VDD with a 100nF external ceramic decoupling capacitor.
When the backup domain is supplied by VDD (analog switch connected to VDD), the following functions are available:
(1) OSC32_IN and OSC32_OUT can be used as LSE pins only.
(2) PA5_RTC can be used as GPIO, RTC Calibration Clock, RTC Alarm or second output.
When the backup domain is supplied by VBAT (analog switch connected to VBAT because VDD is not present), the following functions are available:
(1) OSC32_IN and OSC32_OUT can be used as LSE pins only.
(2) PA5_RTC can be used as RTC Alarm or Second output.

1.2.4 Voltage regulator

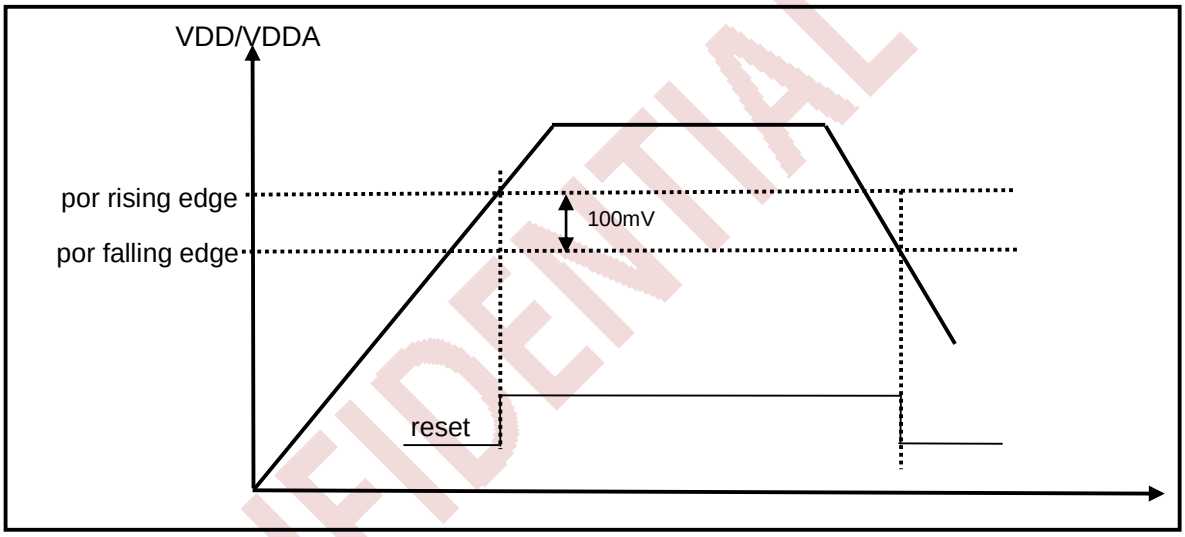
The voltage regulator is always enabled after Reset. It works in two different modes depending on the application modes.
(1) In Run and Stop modes, the regulator supplies full power to the 1.2V domain (core, memories,

digital peripherals and fpga core logic).
(2) In Standby Mode, the regulator is powered off. The contents of the registers and SRAM are lost except for the Standby circuitry and the Backup Domain.

1.2.5 Power on reset (POR)/power down reset (PDR)

The device has an integrated POR/PDR circuitry that allows proper operation starting from/down to 2.2V. The device remains in Reset mode when VDD/VDDA is below a specified threshold, VPOR/PDR, without the need for an external reset circuit.

Power on reset/power down reset waveform



1.2.6 Low-power modes

By default, the micro-controller is in Run mode after a system or a power Reset. Several low-power modes are available to save power when the CPU does not need to be kept running. It is up to the user to select the mode that gives the best compromise between low-power consumption, short startup time and available wakeup sources.
The AGRV2K devices feature three low-power modes:
(1) Sleep mode (CPU clock off, all peripherals including MCU core peripherals are kept running)
(2) Stop mode (all clocks are stopped)
(3) Standby mode (1.2V domain powered-off)
In addition, the power consumption in Run mode can be reduce by one of the following means:
(1) Slowing down the system clocks.
(2) Gating the clocks to the APB and AHB peripherals when they are unused.

Table 1. Low-power mode summary

Mode name	Entry	Wakeup	Effect on 1.2V domain clocks	Effect on VDD domain clocks	Voltage regulator
Sleep (Sleep now or Sleep-on -exit)	WFI	Any interrupt	CPU clock OFF no effect on other clocks or analog clock sources	None	ON
	WFE	Wakeup event			
Stop	PDDS and LPDS bits + SLEEPDEEP bit + WFI or WFE	Any EXTI line (configured in the EXTI registers)	All 1.2V domain clocks OFF	HSI and HSE oscillators OFF	ON
Standby	PDDS bit + SLEEPDEEP bit + WFI or WFE	WKUP pin rising edge, RTC alarm, external reset in NRST pin, IWDG reset			OFF

1.2.6.1 Slowing down system clocks

In Run mode the speed of the system clocks can be reduced. And also slow down peripherals before entering Sleep mode.

1.2.6.2 Peripheral clock gating

In Run mode, the clocks for individual peripherals and memories can be stopped at any time to reduce power consumption.
To further reduce power consumption in Sleep mode the peripheral clocks can be disabled prior to executing the WFI or WFE instructions.

1.2.6.3 Sleep mode

Entering Sleep mode
The Sleep mode is entered by executing the WFI (Wait For Interrupt) or WFE (Wait for Event) instructions. Two options are available to select the Sleep mode entry mechanism, depending on the SLEEPONEXIT bit in the System Control register:
(1) Sleep-now: if the SLEEPONEXIT bit is cleared, the MCU enters Sleep mode as soon as WFI or

WFE instruction is executed.

(2) Sleep-on-exit: if the SLEEPONEXIT bit is set, the MCU enters Sleep mode as soon as it exits the lowest priority ISR.

In the Sleep mode, all I/O pins keep the same state as in the Run mode.

Exiting Sleep mode

If the WFI instruction is used to enter Sleep mode, any peripheral interrupt acknowledged by the nested vectored interrupt controller (NVIC) can wake up the device from Sleep mode. If the WFE instruction is used to enter Sleep mode, the MCU exits Sleep mode as soon as an event occurs. The wakeup event can be generated either by:

(1) enabling an interrupt in the peripheral control register but not in the NVIC, and enabling the SEVONPEND bit in the System Control register. When the MCU resumes from WFE, the peripheral interrupt pending bit and the peripheral NVIC IRQ channel pending bit (in the NVIC interrupt clear pending register) have to be cleared.

(2) or configuring an external or internal EXTI line in event mode. When the CPU resumes from WFE, it is not necessary to clear the peripheral interrupt pending bit or the NVIC IRQ channel pending bit as the pending bit corresponding to the event line is not set. This mode offers the lowest wakeup time as no time is wasted in interrupt entry/exit.

Table 2. Sleep-now

Sleep-now mode	Description
Mode entry	WFI (Wait for Interrupt) or WFE (Wait for Event) while: – SLEEPDEEP = 0 and – SLEEPONEXIT = 0 Refer to the System Control register.
Mode exit	If WFI was used for entry: Interrupt: Refer to : Interrupt and exception vectors If WFE was used for entry Wakeup event: Refer to : Wakeup event management
Wakeup latency	None

Table 3. Sleep-on-exit

Sleep-on-exit	Description
Mode entry	WFI (wait for interrupt) while: – SLEEPDEEP = 0 and – SLEEPONEXIT = 1 Refer to the System Control register.
Mode exit	Interrupt: refer to: Interrupt and exception vectors.
Wakeup latency	None

1.2.6.4 Stop mode

The Stop mode is based on the MCU deep-sleep mode combined with peripheral clock gating.
In Stop mode, all clocks in the 1.2V domain are stopped, the PLL, the HSI and the HSE oscillators are disabled. SRAM and register contents are preserved.
In the Stop mode, all I/O pins keep the same state as in the Run mode.

Entering Stop mode
Refer to Table 4 for details on how to enter the Stop mode.
If Flash memory programming is ongoing, the Stop mode entry is delayed until the memory access is finished.
If an access to the APB domain is ongoing, The Stop mode entry is delayed until the APB access is finished.
In Stop mode, the following features can be selected by programming individual control bits:
(1) Independent watchdog (IWDG): the IWDG is started by writing to its enable register or by hardware option. Once started it cannot be stopped except by a Reset.
(2) Real-time clock (RTC): this is configured by the RTCEN bit in the Backup domain control register (RCC_BDCR).
(3) External 32.768 kHz oscillator (LSE OSC): this is configured by the LSEON bit in the Backup domain control register (RCC_BDCR).

The ADC or DAC can also consume power during the Stop mode, unless they are disabled before entering it.

Exiting Stop mode

Refer to Table 4 for more details on how to exit Stop mode.

When exiting Stop mode by issuing an interrupt or a wakeup event, the HSI RC oscillator is selected as system clock.

Table 4. Stop mode

Stop mode	Description
Mode entry	WFI (Wait for Interrupt) or WFE (Wait for Event) while: – Set SLEEPDEEP bit in System Control register – Clear PDDS bit in Power Control register (PWR_CR) Note: To enter Stop mode, all EXTI Line pending bits (in Pending register (EXTI_PR)), all peripheral interrupt pending bits, and RTC Alarm flag must be reset. Otherwise, the Stop mode entry procedure is ignored and program execution continues.
Mode exit	If WFI was used for entry: Any EXTI Line configured in Interrupt mode (the corresponding EXTI Interrupt vector must be enabled in the NVIC). Refer to: Interrupt and exception vectors. If WFE was used for entry: Any EXTI Line configured in event mode. Refer to: Wakeup event management
Wakeup latency	HSI RC wakeup time

1.2.6.5 Standby mode

The Standby mode allows to achieve the lowest power consumption. It is based on the deep-sleep mode, with the voltage regulator disabled. The 1.2V domain is consequently powered off. The PLL, the HSI oscillator and the HSE oscillator are also switched off. SRAM and register contents are lost except for registers in the Backup domain and Standby circuitry.

Entering Standby mode

Refer to Table 5 for more details on how to enter Standby mode.

In Standby mode, the following features can be selected by programming individual control bits:

(1) Independent watchdog (IWDG): the IWDG is started by writing to its enable register or by hardware option. Once started it cannot be stopped except by a reset.

(2) Real-time clock (RTC): this is configured by the RTCEN bit in the Backup domain control register (RCC_BDCR).

(3) External 32.768 kHz oscillator (LSE OSC): this is configured by the LSEON bit in the Backup domain control register (RCC_BDCR)

Exiting Standby mode

The micro-controller exits the Standby mode when an external reset (NRST pin), an IWDG reset, a

rising edge or falling edge on the WKUP pin or the rising edge of an RTC alarm occurs. All registers are reset after wakeup from Standby.
After waking up from Standby mode, program execution restarts in the same way as after a Reset. The SBF status flag in the Power control/status register (PWR_CSR) indicates that the MCU was in Standby mode.
Refer to Table 5 for more details on how to exit Standby mode.

Table 5. Standby mode

Stop mode	Description
Mode entry	WFI (Wait for Interrupt) or WFE (Wait for Event) while: – Set SLEEPDEEP in System Control register – Set PDDS bit in Power Control register (PWR_CR) – Clear WUF bit in Power Control/Status register (PWR_CSR) – No interrupt (for WFI) or event (for WFI) is pending
Mode exit	WKUP pin rising edge, RTC alarm event's rising edge, external Reset in NRST pin, IWDG Reset.
Wakeup latency	Reset phase

I/O states in Standby mode
In Standby mode, all I/O pins are high impedance except:
(1) Reset pin (still available)
(2) CLKRTCOUT pin if configured for calibration out
(3) WKUP pin, if enabled

1.2.6.6 Auto-wakeup (AWU) from low-power mode

The RTC can be used to wakeup the MCU from low-power mode without depending on an external interrupt (Auto-wakeup mode). The RTC provides a programmable time base for waking up from Stop or Standby mode at regular intervals. For this purpose, two of the three alternative RTC clock sources can be selected by programming the RTCSEL[1:0] bits in the Backup domain control register (RCC_BDCR):
(1) Low-power 32.768 kHz external crystal oscillator (LSE OSC).
This clock source provides a precise time base with very low-power consumption.
(2) Low-power internal RC Oscillator (LSI RC)
This clock source has the advantage of saving the cost of the 32.768 kHz crystal.

1.2.7 Power control registers

1.2.7.1 Power control register (PWR_CR)

Address offset: 0x00

Reset value: 0x0000 0000 (reset by wakeup from Standby mode)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
							DBP	reserved				CSBF	CWUF	PDDS	LPDS
							rw					rc_w1	rc_w1	rw	rw

Bit 8 DBP: Disable backup domain write protection.

In reset state, the RTC and backup registers are protected against parasitic write access. This bit must be set to enable write access to these registers.

0: Access to RTC and Backup registers disabled

1: Access to RTC and Backup registers enabled

Bit 3 CSBF: Clear standby flag.

This bit is always read as 0.

0: No effect

1: Clear the SBF Standby Flag (write).

Bit 2 CWUF: Clear wakeup flag.

This bit is always read as 0.

0: No effect

1: Clear the WUF Wakeup Flag after 2 System clock cycles. (write)

Bit 1 PDDS: Power down deep-sleep.

This bit is set and cleared by software. It works together with the LPDS bit.

0: Enter Stop mode when the CPU enters Deep-sleep. The regulator status depends on the LPDS bit.

1: Enter Standby mode when the CPU enters Deep-sleep.

Bit 0 LPDS: Low-power deep-sleep.

This bit is set and cleared by software. It works together with the PDDS bit.

0: Voltage regulator on during Stop mode

1: Voltage regulator in low-power mode during Stop mode

1.2.7.2 Power control/status register (PWR_CSR)

Address offset: 0x04
Reset value: 0x0000 0000 (not reset by wakeup from Standby mode)
Additional APB cycles are needed to read this register versus a standard APB read.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
reserved							EWUP	reserved						SBF	WUF
							rw							r	r

Bit 8 EWUP: Enable WKUP pin
This bit is set and cleared by software.
0: WKUP pin is used for general purpose I/O. An event on the WKUP pin does not wakeup the device from Standby mode.
1: WKUP pin is used for wakeup from Standby mode and forced in input pull down configuration (rising edge on WKUP pin wakes-up the system from Standby mode).
Note: This bit is reset by a system Reset.

Bit 1 SBF: Standby flag
This bit is set by hardware and cleared only by a POR/PDR (power on reset/power down reset) or by setting the CSBF bit in the Power control register (PWR_CR)
0: Device has not been in Standby mode
1: Device has been in Standby mode

Bit 0 WUF: Wakeup flag
This bit is set by hardware and cleared by hardware, by a system reset or by setting the CWUF bit in the Power control register (PWR_CR)
0: No wakeup event occurred
1: A wakeup event was received from the WKUP pin or from the RTC alarm
Note: An additional wakeup event is detected if the WKUP pin is enabled (by setting the EWUP bit) when the WKUP pin level is already high.

1.3 Backup registers (BKP)

1.3.1 BKP introduction

The backup registers are sixteen two 16-bit registers for storing 32 bytes of user application data. They are implemented in the backup domain that remains powered on by VBAT when the VDD power is switched off. They are not reset when the device wakes up from Standby mode or by a system reset or power reset.
In addition, the BKP control registers are used to manage the RTC calibration.
After reset, access to the Backup registers and RTC is disabled and the Backup domain (BKP) is protected against possible parasitic write access. To enable access to the Backup registers and the RTC, proceed as follows:
(1) enable the power and backup interface clocks by setting the PWREN and BKPEN bits in the RCC_APB1ENR register
(2) set the DBP bit the Power Control Register (PWR_CR) to enable access to the Backup registers and RTC.

1.3.2 BKP main features

(1) 32-byte data registers
(2) Calibration register for storing the RTC calibration value
(3) Possibility to output the RTC Calibration Clock, RTC Alarm pulse or Second pulse on pin PC13

1.3.3 RTC calibration

For measurement purposes, the RTC clock with a frequency divided by 64 can be output on the PC13 pin. This is enabled by setting the CCO bit in the RTC clock calibration register (BKP_RTCCR). The clock can be slowed down by up to 121 ppm by configuring CAL[6:0] bits.

1.3.4 BKP registers

1.3.4.1 Backup data register x (BKP_DRx) (x = 1 ..17)

Address offset:

Reset value: 0x0000 0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
D<15:0>															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits 15:0: D[15:0] Backup data
These bits can be written with user data.
Note: The BKP_DRx registers are not reset by a System reset or Power reset or when the device wakes up from Standby mode.
They are reset by a Backup Domain reset.

1.3.4.2 RTC clock calibration register (BKP_RTCCR)

Address offset:
Reset value: 0x0000 0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
						ASOS	ASOE	COO	CAL<6:0>						
						rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bit 9 ASOS: Alarm or second output selection
When the ASOE bit is set, the ASOS bit can be used to select whether the signal output on the PC13 pin is the RTC Second pulse signal or the Alarm pulse signal:
0: RTC Alarm pulse output selected
1: RTC Second pulse output selected
Note: This bit is reset only by a Backup domain reset.

Bit 8 ASOE: Alarm or second output enable
Setting this bit outputs either the RTC Alarm pulse signal or the Second pulse signal on the PC13 pin depending on the ASOS bit.
0: output disable
1: output enable
The output pulse duration is one RTC clock period. The PC13 pin must not be enabled while the ASOE bit is set.
Note: This bit is reset only by a Backup domain reset.

Bit 7 CCO: Calibration clock output
0: No effect
1: Setting this bit outputs the RTC clock with a frequency divided by 64 on the PC13 pin.

Note: This bit is reset when the VDD supply is powered off.

Bit 6:0 CAL[6:0]: Calibration value
This value indicates the number of clock pulses that will be ignored every 2^20 clock pulses.
This allows the calibration of the RTC, slowing down the clock by steps of 1000000/2^20PPM.
The clock of the RTC can be slowed down from 0 to 121PPM.

1.4 Reset and clock control

1.4.1 Reset

There are three types of reset: system reset, power reset and backup domain reset.

1.4.1.1 System reset

A system reset sets all registers to their reset values except the reset flags in the clock controller CSR register and the registers in the Backup domain.
A system reset is generated when one of the following events occurs:
(1) A low level on the NRST pin (external reset)
(2) Window watchdog end of count condition (WWDG reset)
(3) Independent watchdog end of count condition (IWDG reset)
(4) A software reset (SW reset)
(5) Low-power management reset
The reset source can be identified by checking the reset flags in the Control/Status register, RCC_CSR.

Software reset
The SYSRESETREQ bit in MCU Application Interrupt and Reset Control Register must be set to force a software reset on the device.

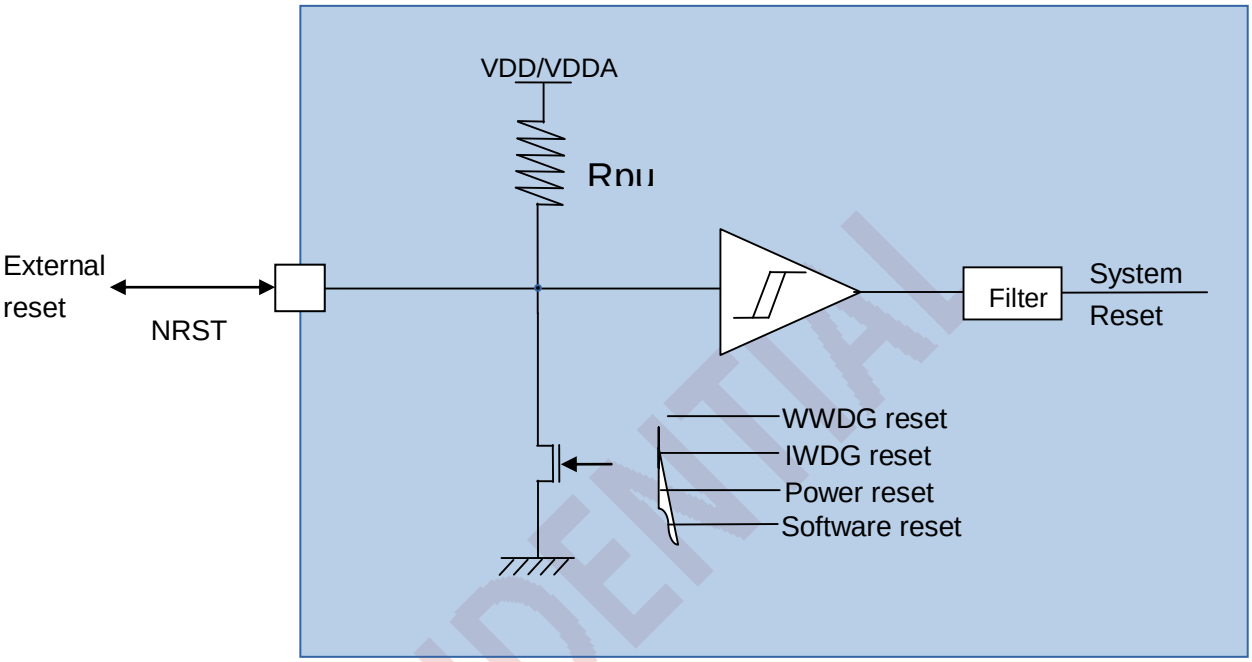
Low-power management reset
There are two ways to generate a low-power management reset:
(1) Reset generated when entering Standby mode:
This type of reset is enabled by resetting nRST_STDBY bit in User Option Bytes. In this case, whenever a Standby mode entry sequence is successfully executed, the device is reset instead of entering Standby mode.
(2) Reset when entering Stop mode:
This type of reset is enabled by resetting nRST_STOP bit in User Option Bytes. In this case, whenever a Stop mode entry sequence is successfully executed, the device is reset instead of entering Stop mode.

1.4.1.2 Power reset

A power reset is generated when one of the following events occurs:

(1) Power-on/power-down reset (POR/PDR reset)

(2) When exiting Standby mode



1.4.1.3 Backup domain reset

The backup domain has two specific resets that affect only the backup domain.

A backup domain reset is generated when one of the following events occurs:

(1) Software reset, triggered by setting the BDRST bit in the Backup domain control register (RCC_BDCR).

(2) VDD or VBAT power on, if both supplies have previously been powered off.

1.4.2 Clocks

Three different clock sources can be used to drive the system clock (SYSCLK):

(1) HSI oscillator clock

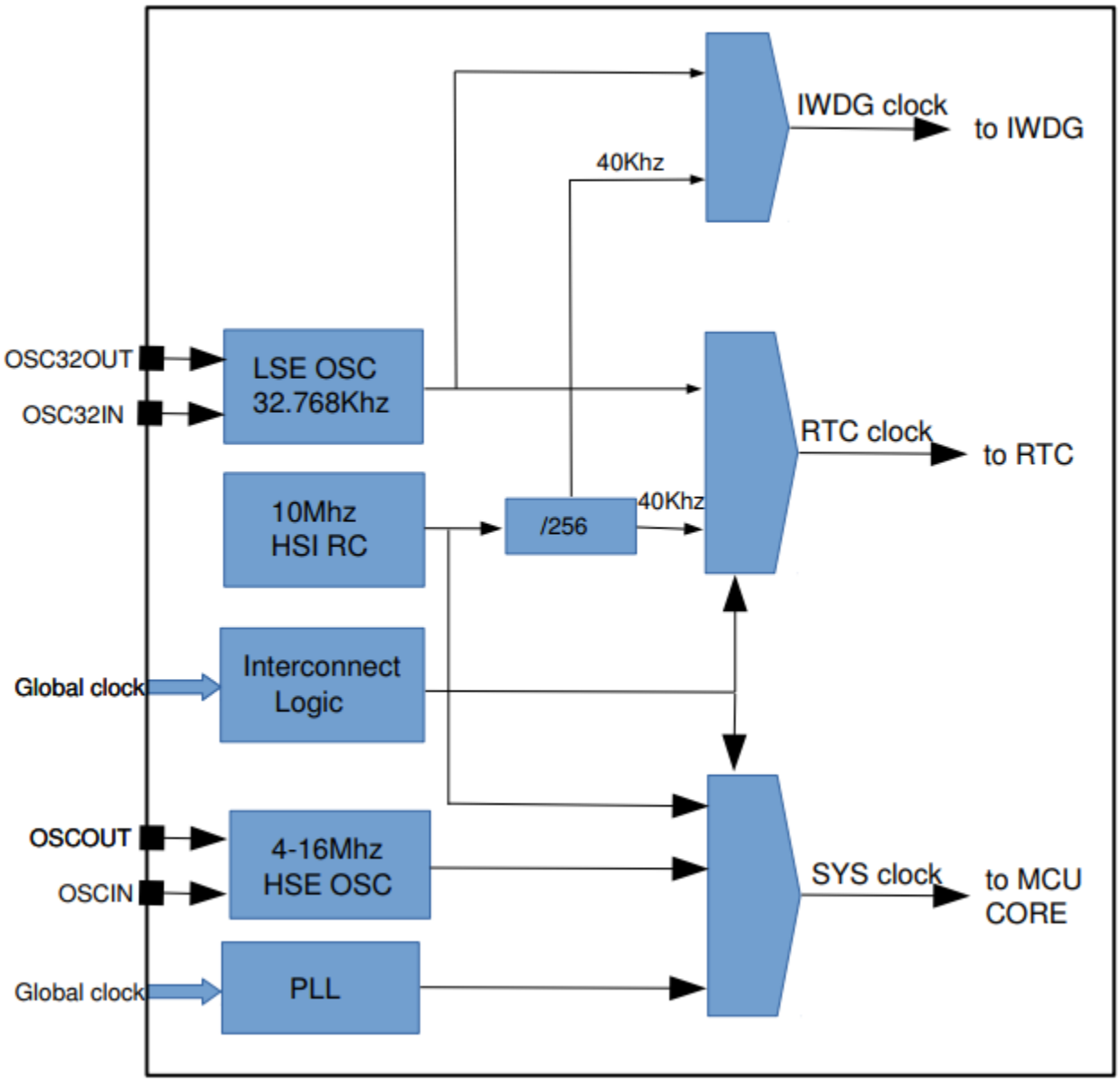
(2) HSE oscillator clock

(3) PLL clock

(4) FPGA global clocks

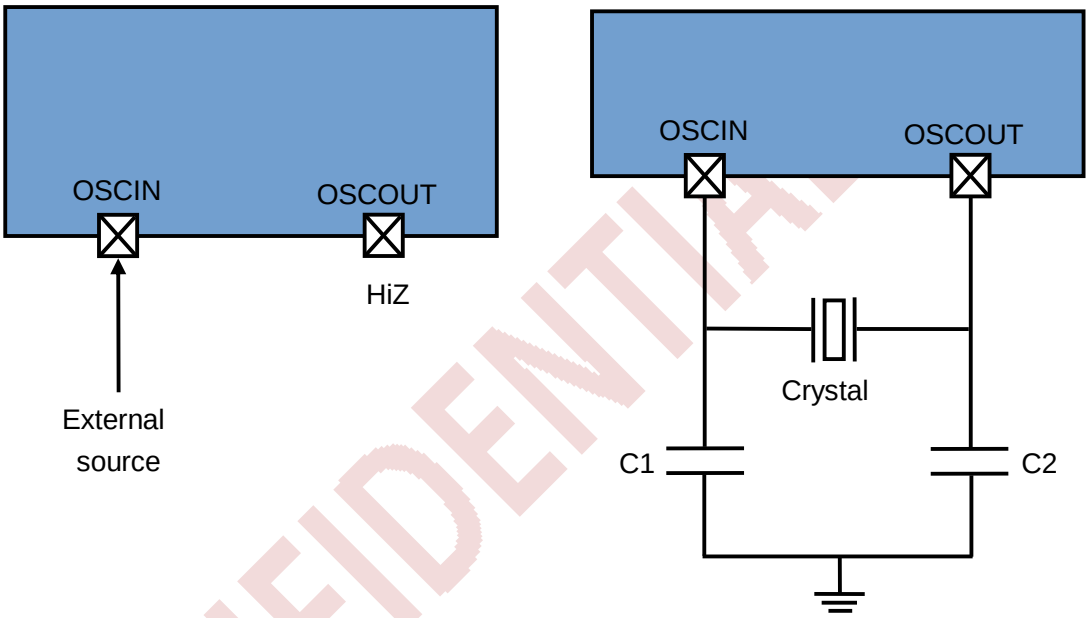
The devices have the following two secondary clock sources:
(1) 40 kHz low speed internal RC (LSI), which drives the independent watchdog and optionally the RTC used for Auto-wakeup from Stop/Standby mode.
(2) 32.768 kHz low speed external crystal (LSE crystal), which optionally drives the real-time clock (RTCCLK)
Each clock source can be switched on or off independently when it is not used, to optimize power consumption.

Clock tree



1.4.2.1 HSE clock

The high speed external clock signal (HSE) can be generated from two possible clock sources:
(1) HSE external crystal/ceramic resonator
(2) HSE user external clock
The resonator and the load capacitors have to be placed as close as possible to the oscillator pins in order to minimize output distortion and startup stabilization time. The loading capacitance values must be adjusted according to the selected oscillator.



External source (HSE bypass)
In this mode, an external clock source must be provided. It can have a frequency of up to 100 MHz. You select this mode by setting the HSEBYP and HSEON bits in the Clock control register (RCC_CR). The external clock signal (square, sinus or triangle) with ~50% duty cycle has to drive the OSC_IN pin while the OSC_OUT pin should be left hi-Z.

External crystal/ceramic resonator (HSE crystal)
The 4 to 16 MHz external oscillator has the advantage of producing a very accurate rate on the main clock. Refer to the electrical characteristics section of the data sheet for more details. The HSERDY flag in the Clock control register (RCC_CR) indicates if the high-speed external oscillator is stable or not. At startup, the clock is not released until this bit is set by hardware. An interrupt can be generated if enabled in the Clock interrupt register (RCC_CIR). The HSE Crystal can be switched on and off using the HSEON bit in the Clock control register (RCC_CR).

1.4.2.2 HSI clock

The HSI clock signal is generated from an internal Oscillator and can be used directly as a system clock.
The HSI internal oscillator has the advantage of providing a clock source at low cost (no external components). It also has a faster startup time than the HSE crystal oscillator.

1.4.2.3 PLL

The internal PLL can be used to multiply HSE crystal output clock frequency. An interrupt can be generated when the PLL is ready if enabled in the Clock interrupt register (RCC_CIR).
If the USB interface is used in the application, the PLL must be programmed to output 48 MHz. This is needed to provide a 48 MHz USBCLK.

1.4.2.4 LSE clock

The LSE crystal is a 32.768 kHz Low Speed External crystal or ceramic resonator. It has the advantage providing a low-power but highly accurate clock source to the real-time clock peripheral (RTC) for clock/calendar or other timing functions.
The LSE crystal is switched on and off using the LSEON bit in Backup domain control register (RCC_BDCR).
The LSERDY flag in the Backup domain control register (RCC_BDCR) indicates if the LSE crystal is stable or not. At startup, the LSE crystal output clock signal is not released until this bit is set by hardware. An interrupt can be generated if enabled in the Clock interrupt register (RCC_CIR).

External source (LSE bypass)
In this mode, an external clock source must be provided. It can have a frequency of up to 1 MHz. You select this mode by setting the LSEBYP and LSEON bits in the Backup domain control register (RCC_BDCR). The external clock signal (square, sinus or triangle) with ~50% duty cycle has to drive the OSC32_IN pin while the OSC32_OUT pin should be left Hi-Z.

1.4.2.5 LSI clock

The LSI clock is HSI divided by 256. It can be kept running in Stop mode for the independent watchdog (IWDG) and Auto-wakeup unit (AWU). The clock frequency is around 40 kHz (between 30 kHz and 60 kHz).

1.4.2.6 System clock (SYSCLK) selection

After a system reset, the HSI oscillator is selected as system clock. When a clock source is used directly or through the PLL as system clock, it is not possible to stop it.
A switch from one clock source to another occurs only if the target clock source is ready (clock stable after startup delay or PLL locked). If a clock source which is not yet ready is selected, the switch will occur when the clock source will be ready. Status bits in the Clock control register (RCC_CR) indicate which clock(s) is (are) ready and which clock is currently used as system clock.

1.4.2.7 RTC clock

The RTCCLK clock source can be either the CLKLOCAL(from fpga core logic), LSE or LSI clocks. This is selected by programming the RTCSEL[1:0] bits in the Backup domain control register (RCC_BDCR).
This selection cannot be modified without resetting the Backup domain.
The LSE clock is in the Backup domain, whereas the HSE and LSI clocks are not.
Consequently:
(1) If LSE is selected as RTC clock:
The RTC continues to work even if the VDD supply is switched off, provided the VBAT supply is maintained.
(2) If LSI is selected as Auto-Wakeup unit (AWU) clock:
The AWU state is not guaranteed if the VDD supply is powered off.
(3) If the CLKLOCAL is used as the RTC clock:
The RTC state is not guaranteed if the VDD supply is powered off or if the internal voltage regulator is powered off (removing power from the 1.2 V domain).
The DPB bit (disable backup domain write protection) in the Power controller register must be set to 1.

1.4.2.8 Watchdog clock

If the Independent watchdog (IWDG) is started by either hardware option or software access,
(1) Under run or stop mode
Select LSE or LSI clock source by setting the IWDG_STOP_CLKSEL bit in the Backup domain control register (******).
(2) Under Standby mode
HW will select LSE as clock source for IWDG.

1.4.3 Reset and clock control registers

1.4.3.1 Clock control register (RCC_CR)

Address offset:
Reset value:

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
reserved						PLL RDY	PLL ON	reserved					HSE BYP	HSE RDY	HSE ON
						r	rw						rw	r	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
reserved															

Bit 25 PLLRDY: PLL clock ready flag
Set by hardware to indicate that the PLL is locked.
0: PLL unlocked
1: PLL locked

Bit 24 PLLON: PLL enable
Set and cleared by software to enable PLL.
Cleared by hardware when entering Stop or Standby mode. This bit can not be reset if the PLL clock is used as system clock or is selected to become the system clock.
0: PLL OFF
1: PLL ON

Bit 18 HSEBYP: External high-speed clock bypass
Set and cleared by software to bypass the oscillator with an external clock. The external clock must be enabled with the HSEON bit set, to be used by the device. The HSEBYP bit can be written only if the HSE oscillator is disabled.
0: external 4-16 MHz oscillator not bypassed
1: external 4-16 MHz oscillator bypassed with external clock

Bit 17 HSERDY: External high-speed clock ready flag. Set by hardware to indicate that the HSE oscillator is stable. This bit needs 6 cycles of the HSE oscillator clock to fall down after HSEON reset.
0: HSE oscillator not ready
1: HSE oscillator ready
Bit 16 HSEON: HSE clock enable
Set and cleared by software. Cleared by hardware to stop the HSE oscillator when entering Stop

or Standby mode. This bit cannot be reset if the HSE oscillator is used directly or indirectly as the system clock.
0: HSE oscillator OFF
1: HSE oscillator ON

1.4.3.2 Clock configuration register (RCC_CFGR)

1.4.3.3 Clock interrupt register (RCC_CIR)

1.4.3.4 APB2 peripheral reset register (RCC_APB2RSTR)

1.4.3.5 APB1 peripheral reset register (RCC_APB1RSTR)

1.4.3.6 AHB peripheral clock enable register (RCC_AHBENR)

1.4.3.7 APB2 peripheral clock enable register (RCC_APB2ENR)

1.4.3.8 APB1 peripheral clock enable register (RCC_APB1ENR)

1.4.3.9 Backup domain control register (RCC_BDCR)

Address offset:
Reset value: 0x0000 0000, reset by Backup domain Reset.
The LSEON, LSEBYP, RTCSEL and RTCEN bits of the Backup domain control register (RCC_BDCR) are in the Backup domain. As a result, after Reset, these bits are write-protected and the DBP bit in the Power control register (PWR_CR) has to be set before these can be modified.
These bits are only reset after a Backup domain Reset. Any internal or external Reset will not have any effect on these bits.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
reserved															BDRST
															rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RTC EN	reserved					RTCSEL[1:0]		reserved					LSE BYP	LSE RDY	LSEON
rw						rw	rw						rw	r	rw

Bit 16 BDRST: Backup domain software reset
Set and cleared by software.
0: Reset not activated
1: Resets the entire Backup domain

Bit 15 RTCEN: RTC clock enable
Set and cleared by software.
0: RTC clock disabled
1: RTC clock enabled
Bits 9:8 RTCSEL[1:0]: RTC clock source selection
Set by software to select the clock source for the RTC. Once the RTC clock source has been selected, it cannot be changed anymore unless the Backup domain is reset. The BDRST bit can be used to reset them.
00: No clock
01: LSE oscillator clock used as RTC clock
10: LSI oscillator clock used as RTC clock
11: Local clock used as RTC clock

Bit 2 LSEBYP: External low-speed oscillator bypass
Set and cleared by software to bypass oscillator in debug mode. This bit can be written only when the external 32 kHz oscillator is disabled.
0: LSE oscillator not bypassed
1: LSE oscillator bypassed

Bit 1 LSERDY: External low-speed oscillator ready
Set and cleared by hardware to indicate when the external 32 kHz oscillator is stable. After the LSEON bit is cleared, LSERDY goes low after 6 external low-speed oscillator clock cycles.
0: External 32 kHz oscillator not ready
1: External 32 kHz oscillator ready

Bit 0 LSEON: External low-speed oscillator enable
Set and cleared by software.
0: External 32 kHz oscillator OFF
1: External 32 kHz oscillator ON

1.4.3.10 Control/status register (RCC_CSR)

Address: ****

Reset value: 0x0C00 0000, reset by system Reset, except reset flags by power Reset only.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
reserved	WWDGRSTF	IWDGRSTF	SFTRSTF	PORRSTF	PINRSTF	RMVF	reserved								
	rw	rw	rw	rw	rw	rw									

Bit 14 WWDGRSTF: Window watchdog reset flag
Set by hardware when a window watchdog reset occurs.
Cleared by writing to the RMVF bit.
0: No window watchdog reset occurred
1: Window watchdog reset occurred

Bit 13 IWDGRSTF: Independent watchdog reset flag
Set by hardware when an independent watchdog reset from VDD domain occurs.
Cleared by writing to the RMVF bit.
0: No watchdog reset occurred
1: Watchdog reset occurred
Bit 12 SFTRSTF: Software reset flag
Set by hardware when a software reset occurs.
Cleared by writing to the RMVF bit.
0: No software reset occurred
1: Software reset occurred

Bit 11 PORRSTF: POR/PDR reset flag
Set by hardware when a POR/PDR reset occurs.
Cleared by writing to the RMVF bit.
0: No POR/PDR reset occurred
1: POR/PDR reset occurred

Bit 10 PINRSTF: PIN reset flag
Set by hardware when a reset from the NRST pin occurs.
Cleared by writing to the RMVF bit.
0: No reset from NRST pin occurred
1: Reset from NRST pin occurred

Bit 8 RMVF: Remove reset flag

Set by software to clear the reset flags.
0: No effect
1: Clear the reset flags

1.5 Analog-to-digital converter (ADC)

1.5.1 ADC introduction

The 12-bit ADC is a successive approximation analog-to-digital converter. It has up to 17 multiplexed channels allowing it measure signals from sixteen external and one internal sources. A/D conversion of the various channels can be performed in single, continuous, scan or discontinuous mode.
The analog watchdog feature allows the application to detect if the input voltage goes outside the user-defined high or low thresholds.
The ADC input clock is generated from the interconnect logic clock and it must not exceed 13 MHz.

1.5.2 Characteristics

(1) ADC sampling rate: 1 MSPS for 12-bit resolution
(2) Programmable sampling time
(3) DMA support
(4) 16 external analog inputs and 1 channel for internal temperature sensor (VSENSE) or for VBAT
(5) Converts a single channel or scans a sequence of channels
(6) Single mode converts selected inputs once per trigger
(7) Continuous mode converts selected inputs continuously
(8) Discontinuous mode
(9) SYNC mode(the device with two or more ADCs)
(10) Analog watchdog
(11) Interrupt generation: at the end of regular and inserted group conversions or analog watchdog event
(12) ADC supply requirements: 3.0V to 3.6V, and typical power supply voltage is 3.3V
(13) ADC input range: VREFN ≤VIN ≤VREFP

1.5.3 Pins and internal signals

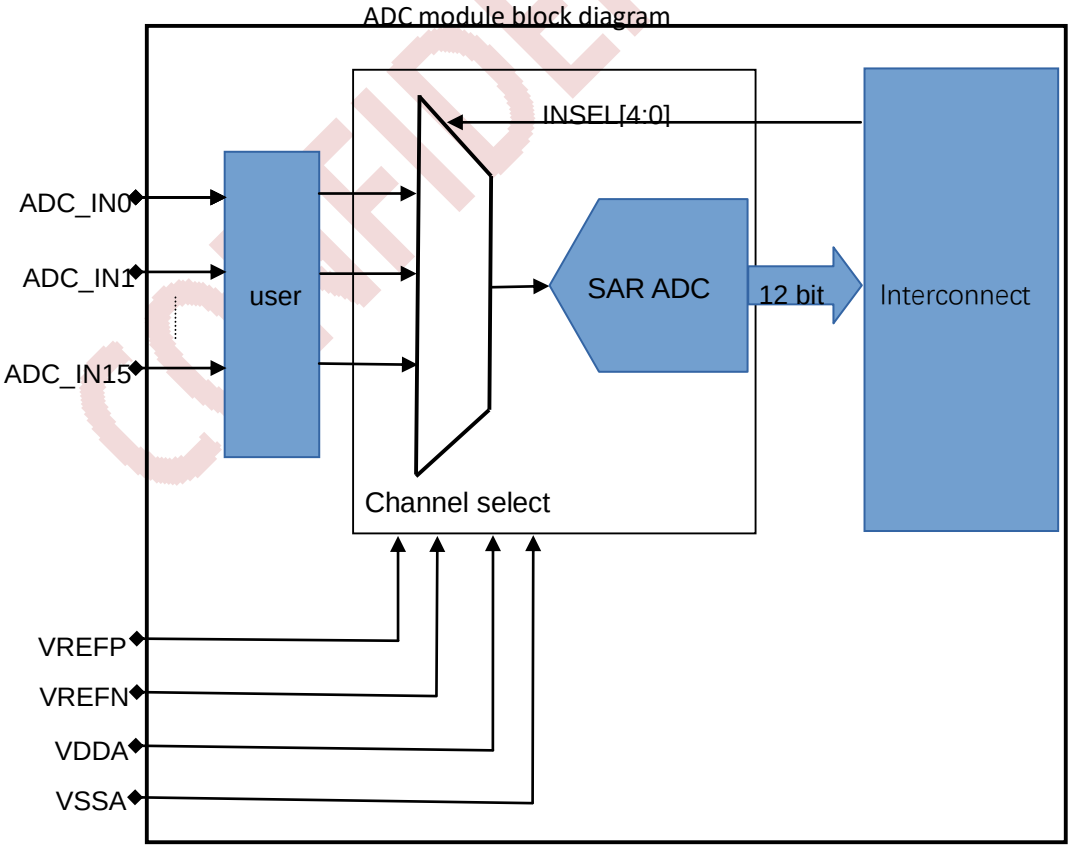
Table 4-1. ADC internal signals

Internal signal name	Signal type	Description
Vtemp-sense	input	Internal temperature sensor output voltage
Vbat	input	Vbat pin voltage

Table 4-2. ADC pins definition

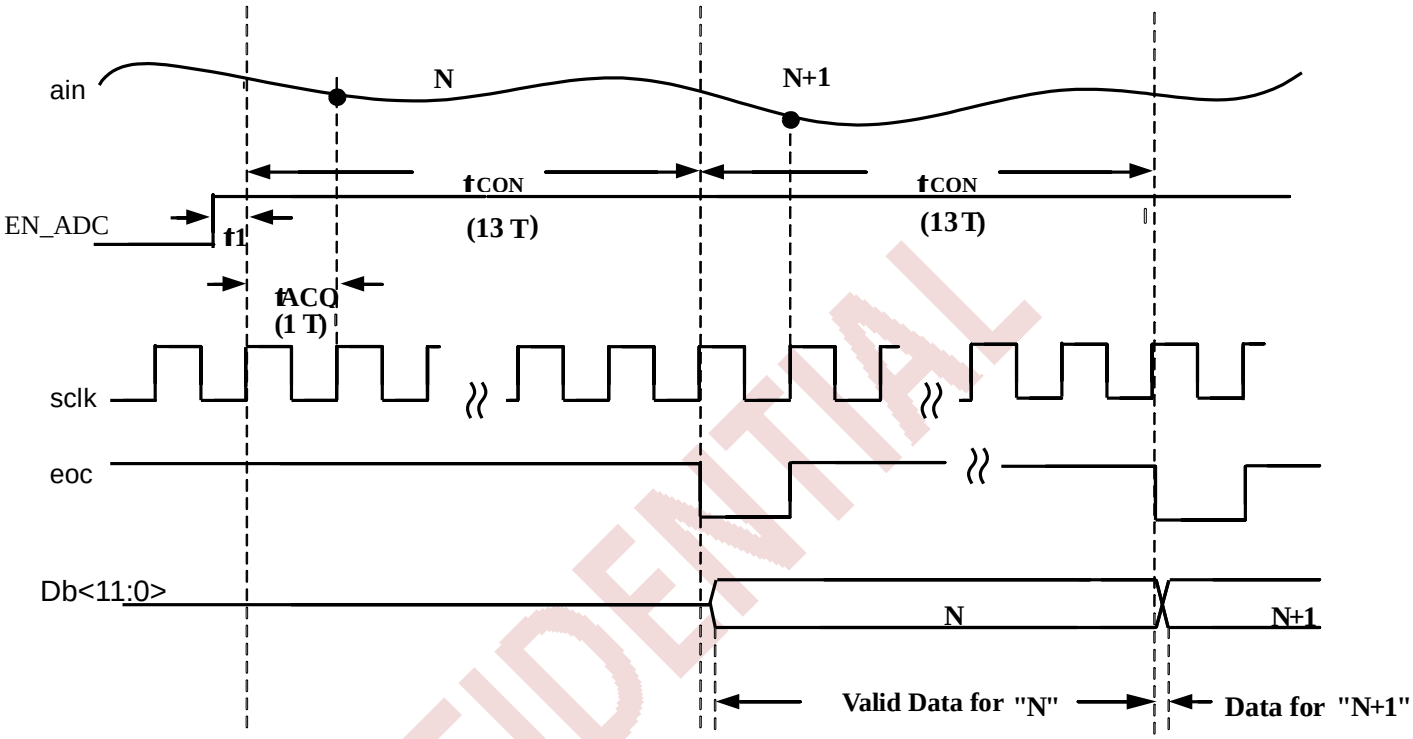
Name	Signal type	Remarks
VDDA	Input, analog power supply	Analog power supply equal to VDD and 3.0 V ≤ VDDA≤ 3.6 V
VSSA	Input, analog power ground	Ground for analog power supply equal to VSS
VREP	Input, analog reference positive	The positive reference voltage for the ADC, 3.0 V ≤ VREFP ≤ VDDA
VREFN	Input, analog reference negative	The negative reference voltage for the ADC,VREFN = VSSA
ADCx_IN[15:0]	Input, Analog signals	Up to 16 external channels

The ADCCLK clock provided by global clock from interconnect core logic. The maximum frequency is 14MHz.



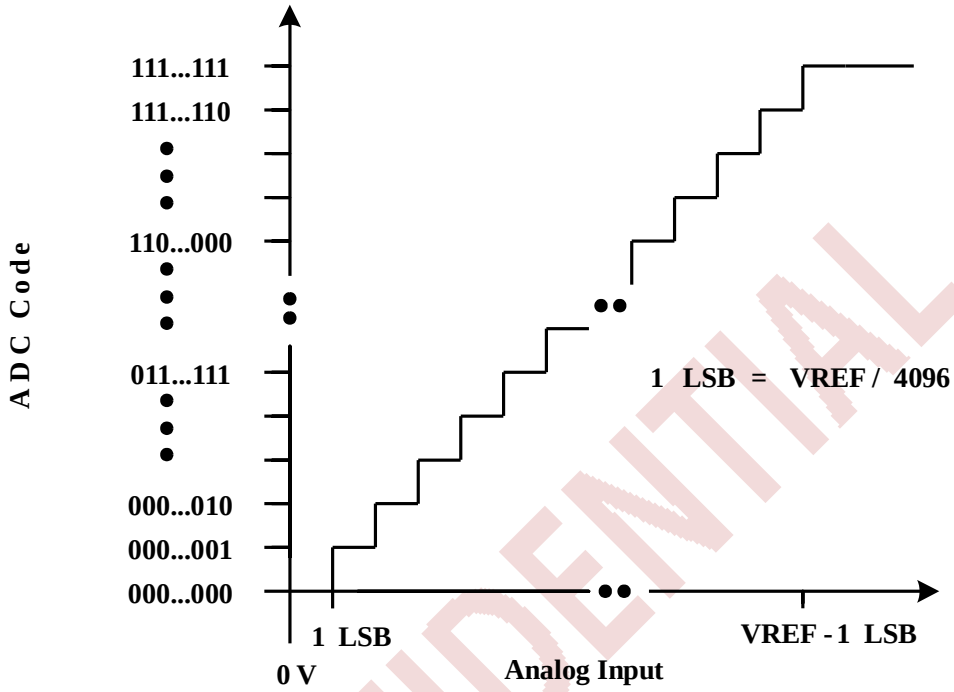
1.5.4 Timing diagram

After the start of ADC conversion and after 13 clock cycles, the EOC flag is set and the 12-bit Data is ready.



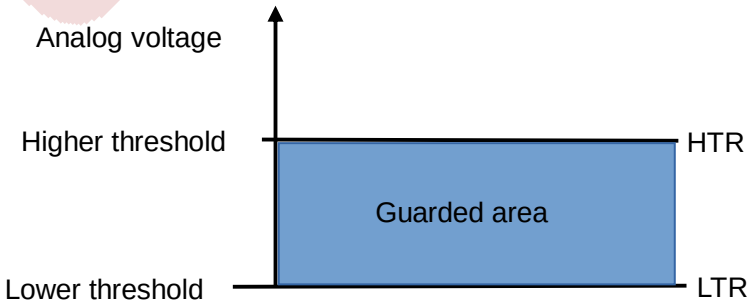
1.5.5 ADC output data

The output coding is straight binary. The designed code transitions occur at successive integer LSB values; that is, 1LSB, 2LSB, and so on. The LSB size is VREF/4096.



1.5.6 Analog watchdog

The AWD analog watchdog status bit is set if the analog voltage converted by the ADC is below a low threshold or above a high threshold.



1.5.7 Temperature sensor

The temperature sensor can be used to measure the ambient temperature of the device. The sensor output voltage can be converted into a digital value by ADC. The sampling time for the temperature sensor is recommended to be set to at least 20μs.
The output voltage of the temperature sensor changes linearly with temperature. Because there is an offset, which is up to 45°C and varies from chip to chip due to process variation, the internal temperature sensor is more suited for applications that detect temperature variations instead of absolute temperature.

1.6 Digital-to-analog converter (DAC)

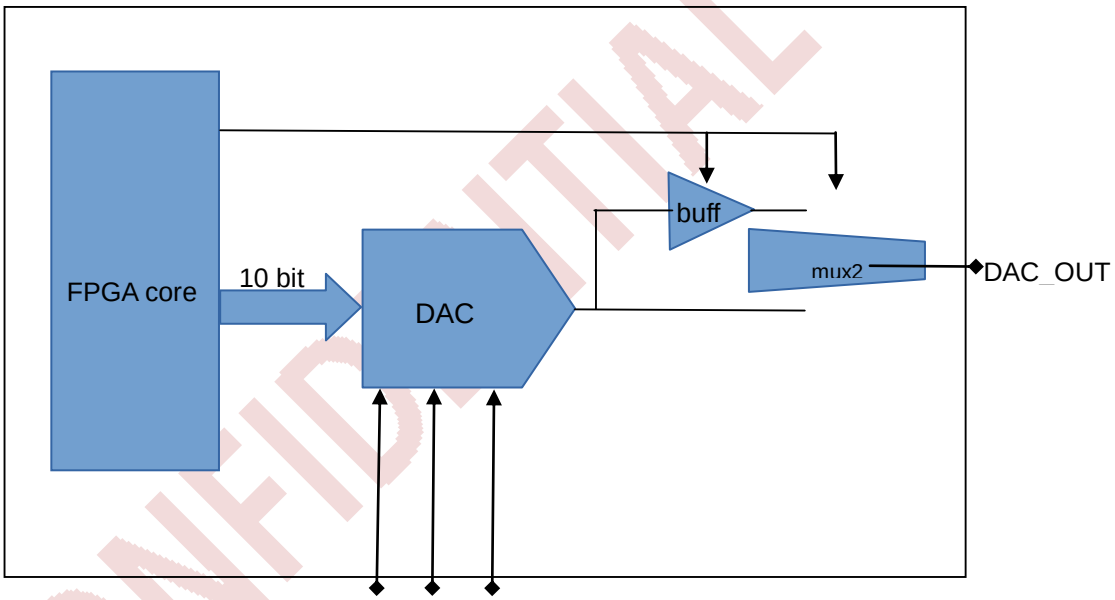
1.6.1 Overview

The Digital-to-analog converter converts 10-bit digital data to a voltage on the external pins. DMA can be used to update the digital data on external triggers. The output voltage can be optionally buffered for higher drive capability. The two DACs can work independently or concurrently.

1.6.2 DAC main features

(1) Two DAC converters: one output channel each
(2) Conversion triggered by external triggers
(3) Dual DAC channel independent or simultaneous conversions
(4) Configurable internal buffer
(5) DMA capability for each channel
(6) External triggers for conversion
(7) Input voltage reference VREFP

DAC block diagram



DAC pins

Name	Signal type	Remarks
VDDA	Input, analog power supply	Analog power supply equal to VDD and 3.0 V ≤ VDDA≤ 3.6 V
VSSA	Input, analog power ground	Ground for analog power supply equal to VSS
VREP	Input, analog reference positive	The positive reference voltage for the ADC, 3.0 V ≤ VREFP ≤ VDDA
DAC_OUTx	DACx analog output	Analog output signal

1.6.3 DAC output buffer enable

The DAC integrates two output buffers that can be used to reduce the output impedance, and to drive external loads directly without having to add an external operational amplifier.

1.6.4 DAC output voltage

The analog output voltage on the DAC pin is determined by the following equation:
DACoutput = VREFP ∗ DAC_Dout/1024
The digital input is linearly converted to an analog output voltage, its range is 0 to VREFP.

1.7 Comparator (CMP)

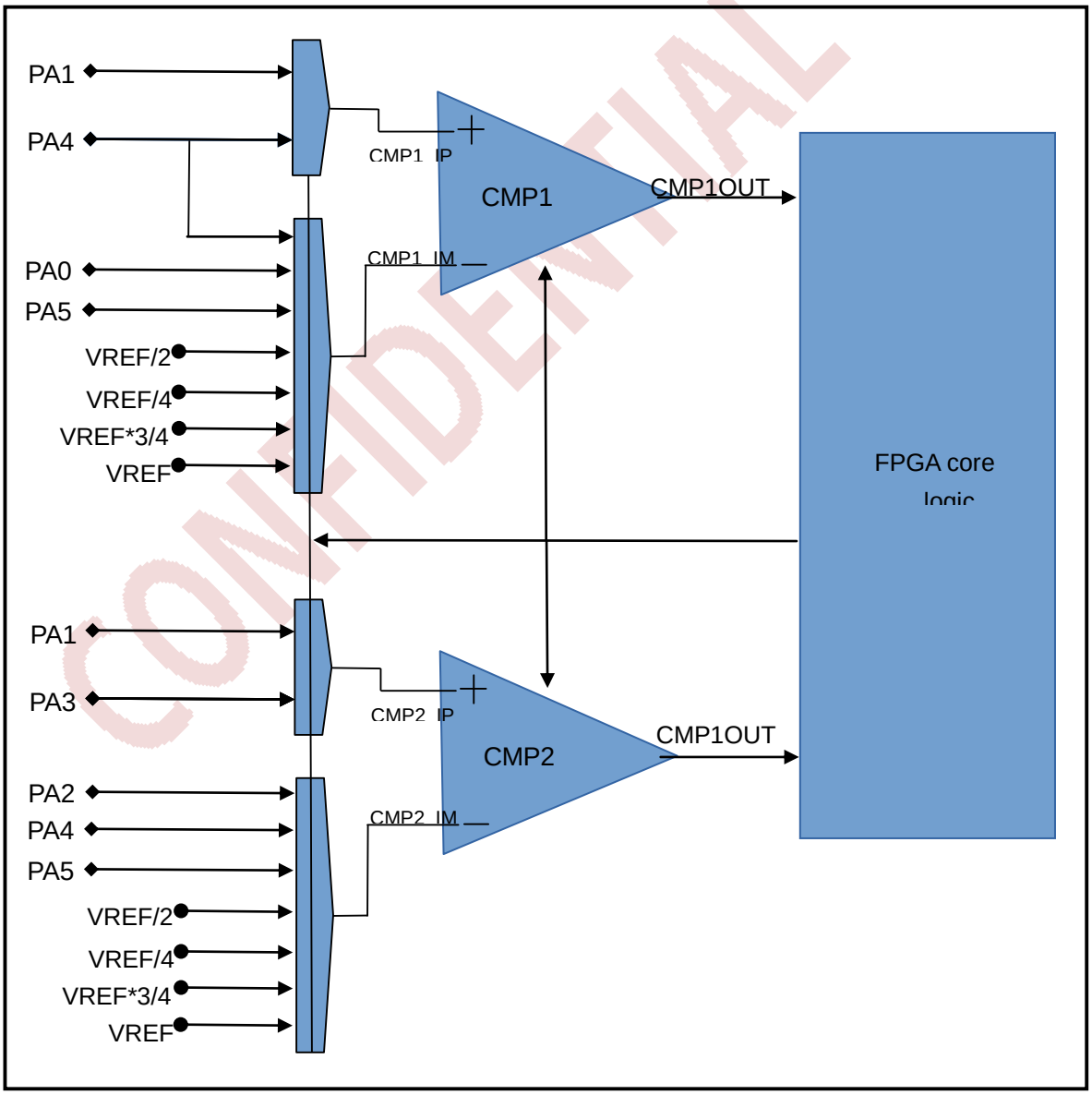
1.7.1 Overview

The general purpose comparators, CMP0 and CMP1, can work either standalone or together with the timers.
It could be used to wake up the MCU from low-power mode by an analog signal, provide a trigger source when an analog signal is in a certain condition, achieves some current control by working together with a PWM output of a timer and the DAC.

1.7.2 Characteristic

(1) Rail-to-rail comparators
(2) Configurable hysteresis
(3) Configurable speed and consumption
(4) Each comparator has configurable analog input source
(5) The whole or sub-multiple values of internal reference voltage Window comparator
(6) Outputs to I/O
(7) Outputs to timers for triggering

CMP block diagram



1.8 Real-time clock (RTC)

1.8.1 RTC introduction

The real-time clock is an independent timer. The RTC provides a set of continuously running counters which can be used, with suitable software, to provide a clock-calendar function. The counter values can be written to set the current time/date of the system.

The RTC core and clock configuration (RCC_BDCR register) are in the Backup domain, which means that RTC setting and time are kept after reset or wakeup from Standby mode. After reset, access to the Backup registers and RTC is disabled and the Backup domain (BKP) is protected against possible parasitic write access. To enable access to the Backup registers and the RTC, proceed as follows:

(1) Enable the power and backup interface clocks by setting the PWREN and BKPEN bits in the RCC_APB1ENR register

(2) Set the DBP bit the Power Control Register (PWR_CR) to enable access to the Backup registers and RTC.

1.8.2 RTC main features

- Programmable pre-scaler : division factor up to 2^20
- 32-bit programmable counter for long-term measurement
- The RTC clock source could be any of the following ones:

(1) CLKLOCAL from fpga core logic

(2) LSE oscillator clock

(3) LSI oscillator clock

- Two separate reset types:

(1) The APB1 interface is reset by system reset

(2) The RTC Core (Pre-scaler, Alarm, Counter and Divider) is reset only by a Backup domain reset.

- Three dedicate interrupt lines:

(1) Alarm interrupt, for generating a software programmable alarm interrupt.

(2) Seconds interrupt, for generating a periodic interrupt signal with a programmable period length (up to 1 second).

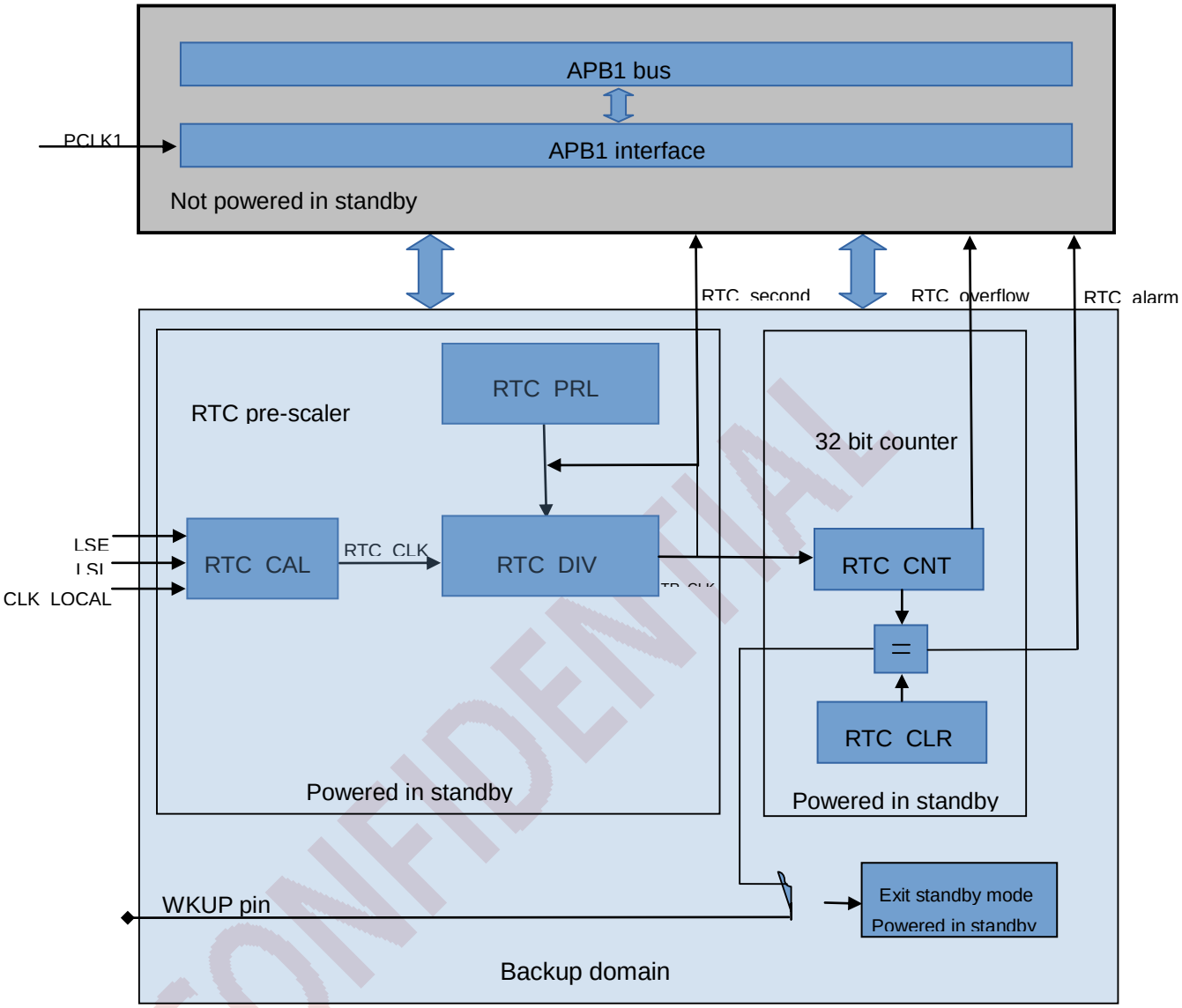
(3) Overflow interrupt, to detect when the internal programmable counter rolls over to zero.

1.8.3 RTC functional description

The RTC consists of two main units. The first one (APB1 Interface) is used to interface with the APB1 bus. This unit also contains a set of 16-bit registers accessible from the APB1 bus in read or write mode. The APB1 interface is clocked by the APB1 bus clock in order to interface with the APB1 bus.

The other unit (RTC Core) consists of a chain of programmable counters made of two main blocks. The first block is the RTC pre-scaler block, which generates the RTC time base TR_CLK that can be programmed to have a period of up to 1 second. It includes a 20-bit programmable divider (RTC Pre-scaler). Every TR_CLK period, the RTC generates an interrupt (Second Interrupt) if it is enabled in the RTC_CR register. The second block is a 32-bit programmable counter that can be initialized to the current system time. The system time is incremented at the TR_CLK rate and compared with a programmable date (stored in the RTC_ALR register) in order to generate an alarm interrupt, if enabled in the RTC_CR control register.

RTC simplified block diagram



Resetting RTC registers

All system registers are asynchronously reset by a System Reset or Power Reset, except for RTC_PRL, RTC_ALR, RTC_CNT, and RTC_DIV.

The RTC_PRL, RTC_ALR, RTC_CNT, and RTC_DIV registers are reset only by a Backup Domain reset.

Reading RTC registers

The RTC core is completely independent from the RTC APB1 interface.

Software accesses the RTC pre-scaler, counter and alarm values through the APB1 interface but the associated readable registers are internally updated at each rising edge of the RTC clock resynchronized by the RTC APB1 clock. This is also true for the RTC flags.

This means that the first read to the RTC APB1 registers may be corrupted (generally read as 0) if the APB1 interface has previously been disabled and the read occurs immediately after the APB1 interface is enabled but before the first internal update of the registers. This can occur if:

(1) A system reset or power reset has occurred

(2) The MCU has just woken up from Standby mode

(3) The MCU has just woken up from Stop mode

In all the above cases, the RTC core has been kept running while the APB1 interface was disabled (reset, not clocked or unpowered).

Consequently when reading the RTC registers, after having disabled the RTC APB1 interface, the software must first wait for the RSF bit (Register Synchronized Flag) in the RTC_CRL register to be set by hardware.

Note that the RTC APB1 interface is not affected by WFI and WFE low-power modes.

Configuring RTC registers

To write in the RTC_PRL, RTC_CNT, RTC_ALR registers, the peripheral must enter Configuration Mode. This is done by setting the CNF bit in the RTC_CRL register.

In addition, writing to any RTC register is only enabled if the previous write operation is finished. To enable the software to detect this situation, the RTOFF status bit is provided in the RTC_CR register to indicate that an update of the registers is in progress. A new value can be written to the RTC registers only when the RTOFF status bit value is '1'.

Configuration procedure

1. Poll RTOFF, wait until its value goes to '1'
2. Set the CNF bit to enter configuration mode
3. Write to one or more RTC registers
4. Clear the CNF bit to exit configuration mode
5. Poll RTOFF, wait until its value goes to '1' to check the end of the write operation.

The write operation only executes when the CNF bit is cleared; it takes at least three RTCCLK cycles to complete.

RTC flag assertion

The RTC Second flag (SECF) is asserted on each RTC Core clock cycle before the update of the RTC Counter.

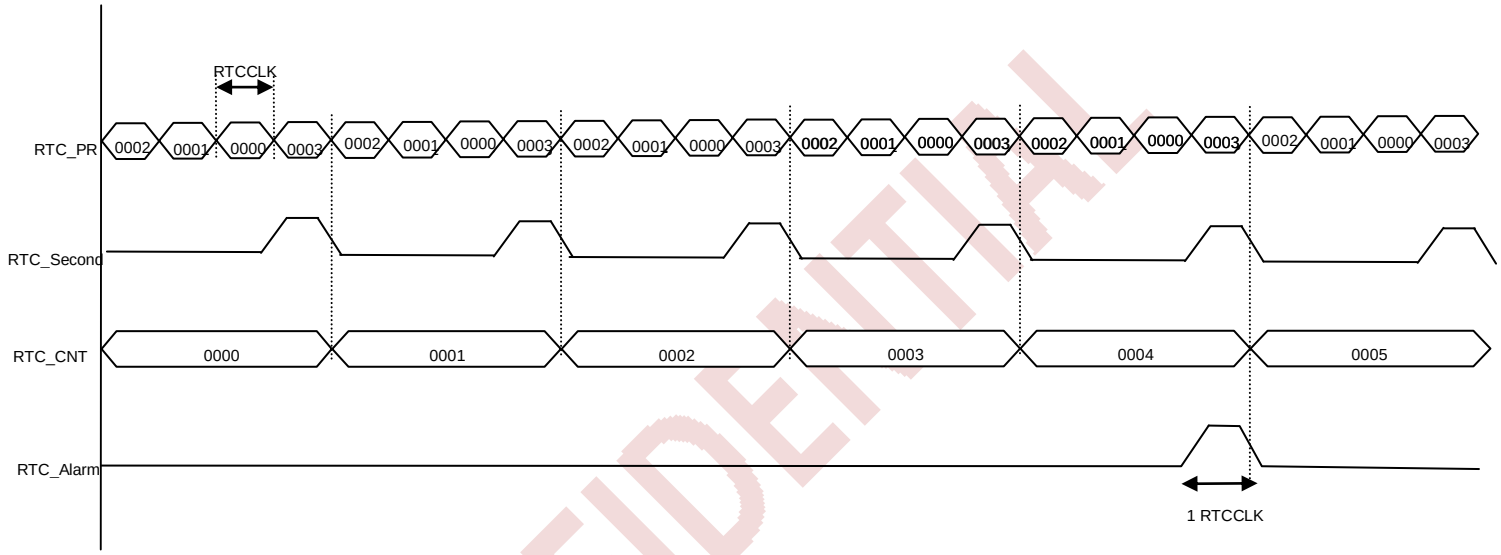
The RTC Overflow flag (OWF) is asserted on the last RTC Core clock cycle before the counter reaches 0x0000.

The RTC_Alarm and RTC Alarm flag (ALRF) are asserted on the last RTC Core clock cycle before the counter reaches the RTC Alarm value stored in the Alarm register increased by one (RTC_ALR + 1). The write operation in the RTC Alarm and RTC Second flag must be synchronized by using one of the following sequences:

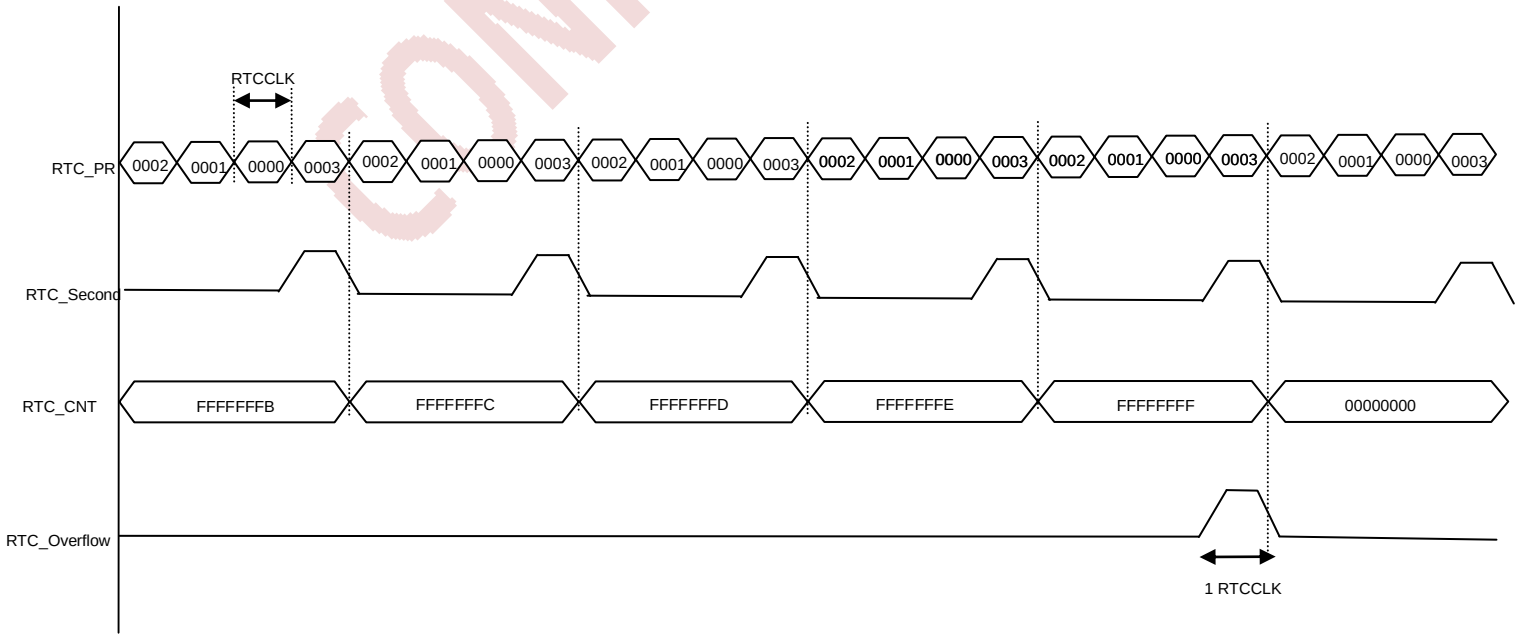
(1) Use the RTC Alarm interrupt and inside the RTC interrupt routine, the RTC Alarm and/or RTC Counter registers are updated.

(2) Wait for SECF bit to be set in the RTC Control register. Update the RTC Alarm and/or the RTC Counter register.

RTC second and alarm waveform example with PR=0003, ALARM=00004



RTC Overflow waveform example with PR=0003



1.8.4 RTC registers

RTC control register high (RTC_CRH)

Address offset: 0x00

Reset value: 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
reserved													OWIE	ALRIE	SECIE
													rw	rw	rw

Bit 2 OWIE: Overflow interrupt enable

0: Overflow interrupt is masked.

1: Overflow interrupt is enabled.

Bit 1 ALRIE: Alarm interrupt enable

0: Alarm interrupt is masked.

1: Alarm interrupt is enabled.

Bit 0 SECIE: Second interrupt enable

0: Second interrupt is masked.

1: Second interrupt is enabled.

RTC control register low (RTC_CRL)

Address offset: 0x04

Reset value: 0x0020

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
reserved										RTOFF	CNF	RSF	OWF	ALRF	SECF
										r	rw	rc_w0	rc_w0	rc_w0	rc_w0

Bit 5 RTOFF: RTC operation OFF

With this bit the RTC reports the status of the last write operation performed on its registers, indicating if it has been completed or not. If its value is '0' then it is not possible to write to any of the RTC registers. This bit is read only.

0: Last write operation on RTC registers is still ongoing.

1: Last write operation on RTC registers terminated.

Bit 4 CNF: Configuration flag

This bit must be set by software to enter in configuration mode so as to allow new values to be written in the RTC_CNT, RTC_ALR or RTC_PRL registers. The write operation is only executed

when the CNF bit is reset by software after has been set.
0: Exit configuration mode (start update of RTC registers).
1: Enter configuration mode.

Bit 3 RSF: Registers synchronized flag
This bit is set by hardware at each time the RTC_CNT and RTC_DIV registers are updated and cleared by software. Before any read operation after an APB1 reset or an APB1 clock stop, this bit must be cleared by software, and the user application must wait until it is set to be sure that the RTC_CNT, RTC_ALR or RTC_PRL registers are synchronized.
0: Registers not yet synchronized.
1: Registers synchronized.

Bit 2 OWF: Overflow flag
This bit is set by hardware when the 32-bit programmable counter overflows. An interrupt is generated if OWIE=1 in the RTC_CRH register. It can be cleared only by software. Writing '1' has no effect.
0: Overflow not detected
1: 32-bit programmable counter overflow occurred.

Bit 1 ALRF: Alarm flag
This bit is set by hardware when the 32-bit programmable counter reaches the threshold set in the RTC_ALR register. An interrupt is generated if ALRIE=1 in the RTC_CRH register. It can be cleared only by software. Writing '1' has no effect.
0: Alarm not detected
1: Alarm detected

Bit 0 SECF: Second flag
This bit is set by hardware when the 32-bit programmable pre-scaler overflows, thus incrementing the RTC counter. Hence this flag provides a periodic signal with a period corresponding to the resolution programmed for the RTC counter (usually one second). An interrupt is generated if SECIE=1 in the RTC_CRH register. It can be cleared only by software. Writing '1' has no effect.
0: Second flag condition not met.
1: Second flag condition met.

RTC pre-scaler load register (RTC_PRLH / RTC_PRLL)
The Pre-scaler Load registers keep the period counting value of the RTC pre-scaler. They are write-protected by the RTOFF bit in the RTC_CR register, and a write operation is allowed if the RTOFF value is '1'.

RTC pre-scaler load register high (RTC_PRLH)
Address offset: 0x08
Write only (see Section 18.3.4: Configuring RTC registers)
Reset value: 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
reserved												PRL[19:16]			
												w	w	w	w

Bits 3:0 PRL[19:16]: RTC pre-scaler reload value high
These bits are used to define the counter clock frequency according to the following formula:
f_TR_CLK = f_RTCCLK/(PRL[19:0]+1)

RTC pre-scaler load register low (RTC_PRLL)
Address offset: 0x0C
Write only
Reset value: 0x8000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PRL[15:0]															
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

Bits 15:0 PRL[15:0]: RTC pre-scaler reload value low
These bits are used to define the counter clock frequency according to the following formula:
f_TR_CLK = f_RTCCLK/(PRL[19:0]+1)
Caution: The zero value is not recommended. RTC interrupts and flags cannot be asserted correctly.
Note:If the input clock frequency (f_RTCCLK) is 32.768 kHz, write 7FFFh in this register to get a signal period of 1 second.

RTC pre-scaler divider register (RTC_DIVH / RTC_DIVL)
During each period of TR_CLK, the counter inside the RTC pre-scaler is reloaded with the value stored in the RTC_PRL register. To get an accurate time measurement it is possible to read the current value of the pre-scaler counter, stored in the RTC_DIV register, without stopping it. This register is read-only and it is reloaded by hardware after any change in the RTC_PRL or RTC_CNT registers.

RTC pre-scaler divider register high (RTC_DIVH)

Address offset: 0x10

Reset value: 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
reserved												RTC_DIV[19:16]			
												r	r	r	r

Bits 3:0 RTC_DIV[19:16]: RTC clock divider high

RTC pre-scaler divider register low (RTC_DIVL)

Address offset: 0x14

Reset value: 0x8000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RTC_DIV[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

Bits 15:0 RTC_DIV[15:0]: RTC clock divider low

RTC counter register (RTC_CNTH / RTC_CNTL)

The RTC core has one 32-bit programmable counter, accessed through two 16-bit registers; the count rate is based on the TR_CLK time reference, generated by the pre-scaler.

RTC_CNT registers keep the counting value of this counter. They are write-protected by bit RTOFF in the RTC_CR register, and a write operation is allowed if the RTOFF value is '1'. A write operation on the upper (RTC_CNTH) or lower (RTC_CNTL) registers directly loads the corresponding programmable counter and reloads the RTC Pre-scaler. When reading, the current value in the counter (system date) is returned.

RTC counter register high (RTC_CNTH)

Address offset: 0x18

Reset value: 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RTC_CNT[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits 15:0 RTC_CNT[31:16]: RTC counter high

Reading the RTC_CNTH register, the current value of the high part of the RTC Counter register is returned. To write to this register it is necessary to enter configuration mode.

RTC counter register low (RTC_CNTL)

Address offset: 0x1C

Reset value: 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RTC_CNT[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits 15:0 RTC_CNT[15:0]: RTC counter low

Reading the RTC_CNTL register, the current value of the lower part of the RTC Counter register is returned. To write to this register it is necessary to enter configuration mode.

RTC alarm register high (RTC_ALRH / RTC_ALRL)

When the programmable counter reaches the 32-bit value stored in the RTC_ALR register, an alarm is triggered and the RTC_alarm interrupt request is generated. This register is write-protected by the RTOFF bit in the RTC_CR register, and a write operation is allowed if the RTOFF value is '1'.

RTC alarm register high (RTC_ALRH)

Address offset: 0x20

Write only

Reset value: 0xFFFF

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RTC_ALR[31:16]															
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

Bits 15:0 RTC_ALR[31:16]: RTC alarm high

The high part of the alarm time is written by software in this register. To write to this register it is necessary to enter configuration mode.

RTC alarm register low (RTC_ALRL)

Address offset: 0x24

Write only

Reset value: 0xFFFF

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RTC_ALR[15:0]															
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

Bits 15:0 RTC_ALR[15:0]: RTC alarm low

The low part of the alarm time is written by software in this register. To write to this register it is necessary to enter configuration mode.

1.9 Independent watchdog (IWDG)

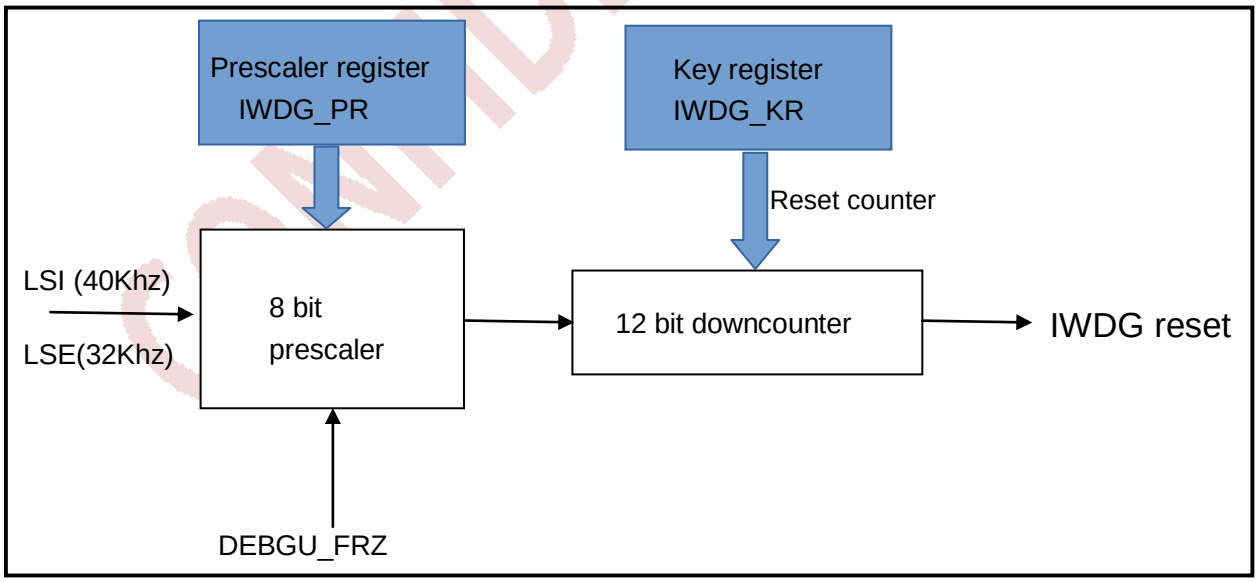
1.9.1 IWDG main features

(1) Free-running down-counter
(2) clocked from an LSI oscillator when Stop and normal modes and from LSE when Standby mode)
(3) Reset (if watchdog activated) when the down-counter value of 0x000 is reached

1.9.2 IWDG functional description

Figure 8.1 shows the functional blocks of the independent watchdog module.
When the independent watchdog is started, the counter starts counting down from the reset value of 0xFFF. When it reaches the end of count value (0x000) a reset signal is generated (IWDG reset).
Whenever the key value 1010 is written in the IWDG_KR register, the down-counter is initialized and the watchdog reset is prevented.

Figure 8.1 Independent watchdog block diagram



1.9.3 Debug mode

When the mcu enter debug mode, the IWDG counter either continues to work normally or stop, depending on DBG_IWDG_STOP configuration bit in DBG module.

IWDG timeout period (in ms) at 40 kHz (LSI)

Pre-scaler divider	PR[2:0] bits	timeout (ms)
/2	0	204.8
/4	1	409.6
/8	2	819.2
/16	3	1638.4
/32	4	3276.8
/64	5	6553.6
/128	6	13107.2
/256	7	26214.4

1.9.4 IWDG registers

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IWDG_KR[3:0]				reserved			IWDG_EN	reserved	IWDG_STOP_CKSEL	IWDG_STDBY_FRZ	IWDG_STOP_FRZ	reserved	IWDG_PR[2:0]		
rw	rw	rw	rw				rw		rw	rw	rw		rw	rw	rw

Bits 15:12 IWDG_KR<3:0>: Key value (write only, read 1010h)
These bits must be written by software at regular intervals with the key value 1010h, otherwise the watchdog generates a reset when the counter reaches 0.

Bits 8 IWDG_EN: IWDG enable control
0: IWDG disable
1: IWDG enable

Bits 6 IWDG_STOP_CKSEL: when STOP mode, select IWDG clock source.
0: select LSI
1: select LSE
note: when STANDBY mode, force to select LSI.

Bits 5 IWDG_STDBY_FRZ: when STANDBY mode, the IWDG counter either continues to work normally or stop.
0: normal work
1: stop

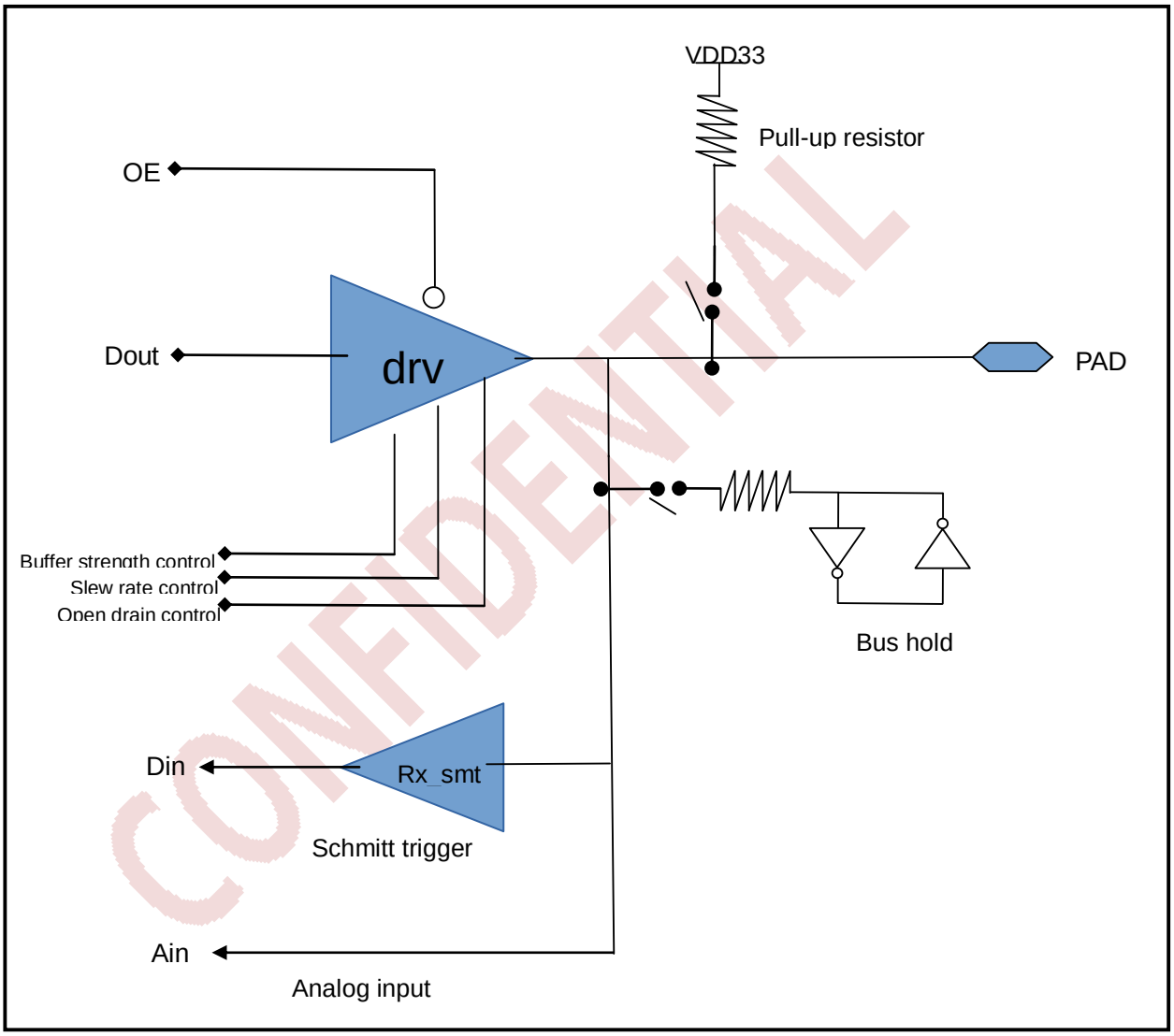
Bits 4 IWDG_STOP_FRZ: when STOP mode, the IWDG counter either continues to work normally or stop.
0: normal work
1: stop

Bits 2:0 IWDG_PR[2:0]: Pre-scaler divider
These bits are written by software to select the prescaler divider feeding the counter clock.
000: divider /2
001: divider /4
010: divider /8
011: divider /16
100: divider /32
101: divider /64
110: divider /128
111: divider /256

1.10 FPGA core

1.10.1 I/O

basic structure of a user I/O



I/O Features

IOs offers a range of programmable features for an I/O pin. These features increase the flexibility of I/O utilization and provide a way to reduce the usage of external discrete components, such as pull-up resistors.

Programmable Current Strength
The output buffer for each I/O pin has a programmable current strength control for certain I/O standards.
The LVTTL, LVCMOS standards have several levels of current strength that you can control.

Slew Rate Control
The output buffer for each I/O pin provides optional programmable output slew-rate control. However, these fast transitions may introduce noise transients in the system. A slower slew rate reduces system noise, but adds a nominal delay to rising and falling edges. Because each I/O pin has an individual slew-rate control, you can specify the slew rate on a pin-by-pin basis. The slew-rate control affects both the rising and falling edges.

Open-Drain Output
Each I/O pin provide an optional open-drain (equivalent to an open-collector) output. This open-drain output enables the device to provide system-level control signals (for example, interrupt and write enable signals) that are asserted by multiple devices in your system.

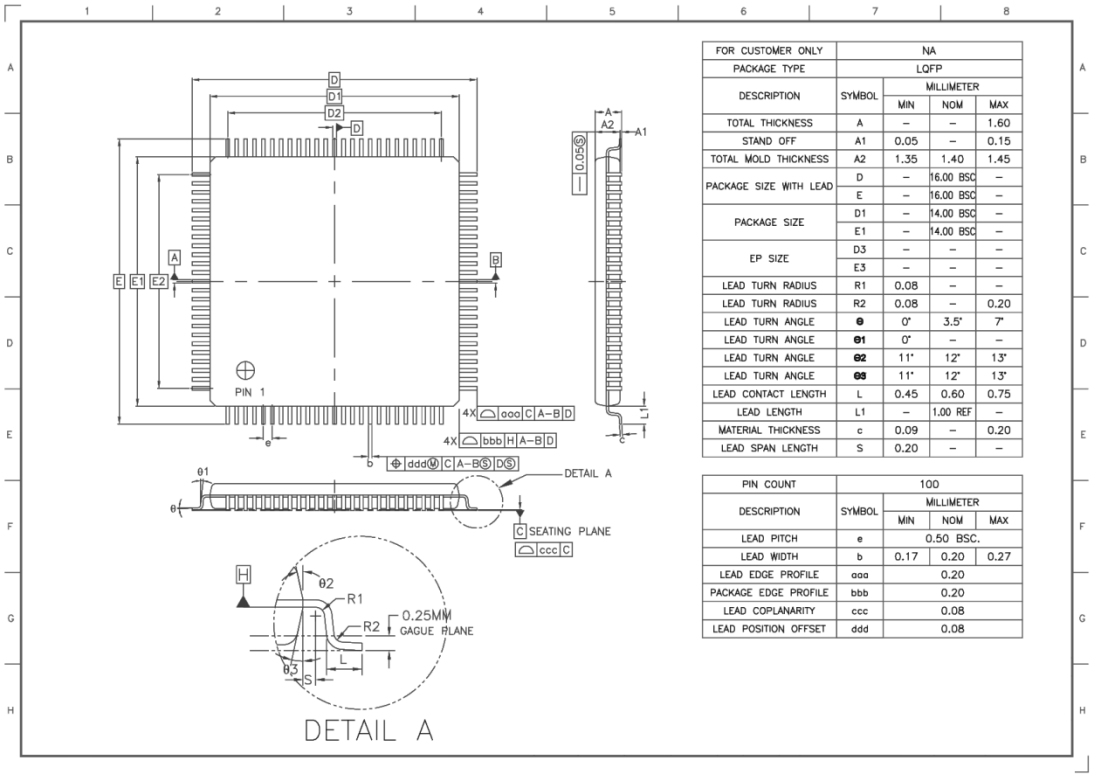
Bus Hold
Each user I/O pin provides an optional bus-hold feature. The bus-hold circuitry holds the signal on an I/O pin at its last-driven state. Because the bus-hold feature holds the last-driven state of the pin until the next input signal is present, an external pull-up or pull-down resistor is not necessary to hold a signal level when the bus is tri-stated.

Programmable Pull-Up Resistor
Each I/O pin provides an optional programmable pull-up resistor while in user mode. If you enable this feature for an I/O pin, the pull-up resistor holds the output to the VDD level.

During and just after reset, the user IOs are configured in Input Floating mode.
The JTAG pins are in input PU/PD after reset:
JTDI in PU
JTCK in PD
JTMS in PU
NJTRST in PU

2 Package

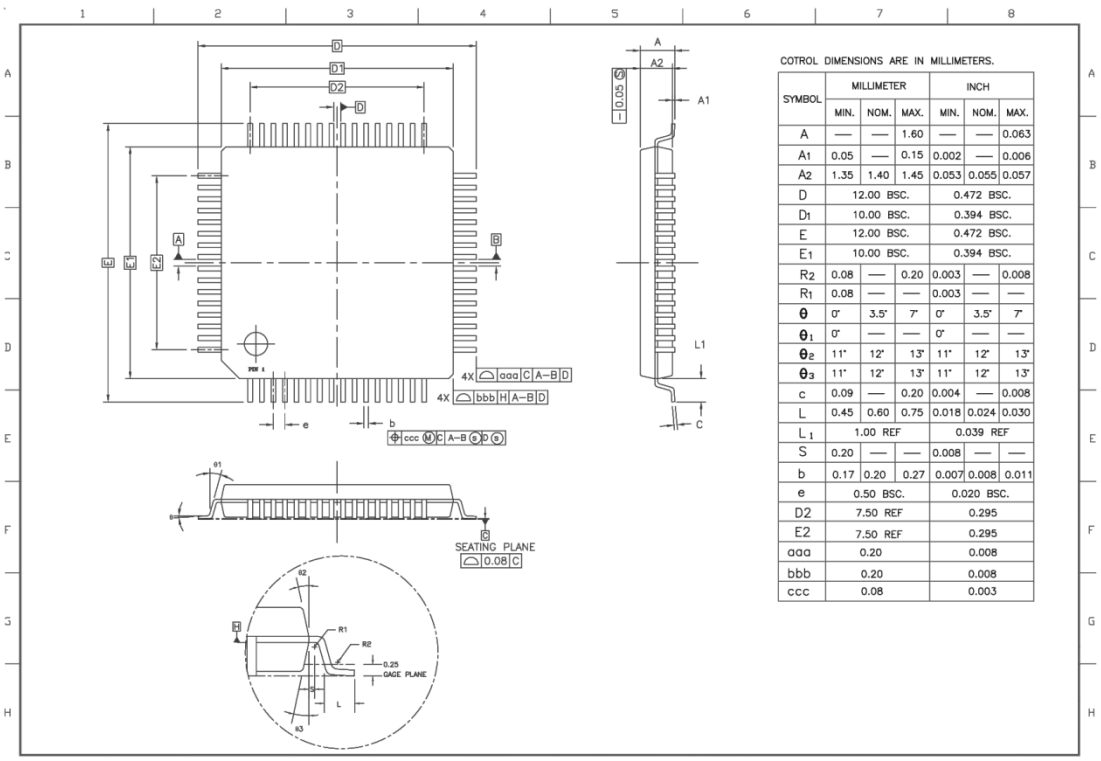
LQFP-100



FOR CUSTOMER ONLY		NA		
PACKAGE TYPE		LQFP		
DESCRIPTION	SYMBOL	MILLIMETER		
		MIN	NOM	MAX
TOTAL THICKNESS	A	–	–	1.60
STAND OFF	A1	0.05	–	0.15
TOTAL MOLD THICKNESS	A2	1.35	1.40	1.45
PACKAGE SIZE WITH LEAD	D	–	16.00 BSC	–
	E	–	16.00 BSC	–
PACKAGE SIZE	D1	–	14.00 BSC	–
	E1	–	14.00 BSC	–
EP SIZE	D3	–	–	–
	E3	–	–	–
LEAD TURN RADIUS	R1	0.08	–	–
LEAD TURN RADIUS	R2	0.08	–	0.20
LEAD TURN ANGLE	θ	0°	3.5°	7°
LEAD TURN ANGLE	θ1	0°	–	–
LEAD TURN ANGLE	θ2	11°	12°	13°
LEAD TURN ANGLE	θ3	11°	12°	13°
LEAD CONTACT LENGTH	L	0.45	0.60	0.75
LEAD LENGTH	L1	–	1.00 REF	–
MATERIAL THICKNESS	c	0.09	–	0.20
LEAD SPAN LENGTH	S	0.20	–	–

PIN COUNT		100		
DESCRIPTION	SYMBOL	MILLIMETER		
		MIN	NOM	MAX
LEAD PITCH	e	0.50 BSC.		
LEAD WIDTH	b	0.17	0.20	0.27
LEAD EDGE PROFILE	aaa	0.20		
PACKAGE EDGE PROFILE	bbb	0.20		
LEAD COPLANARITY	ccc	0.08		
LEAD POSITION OFFSET	ddd	0.08		

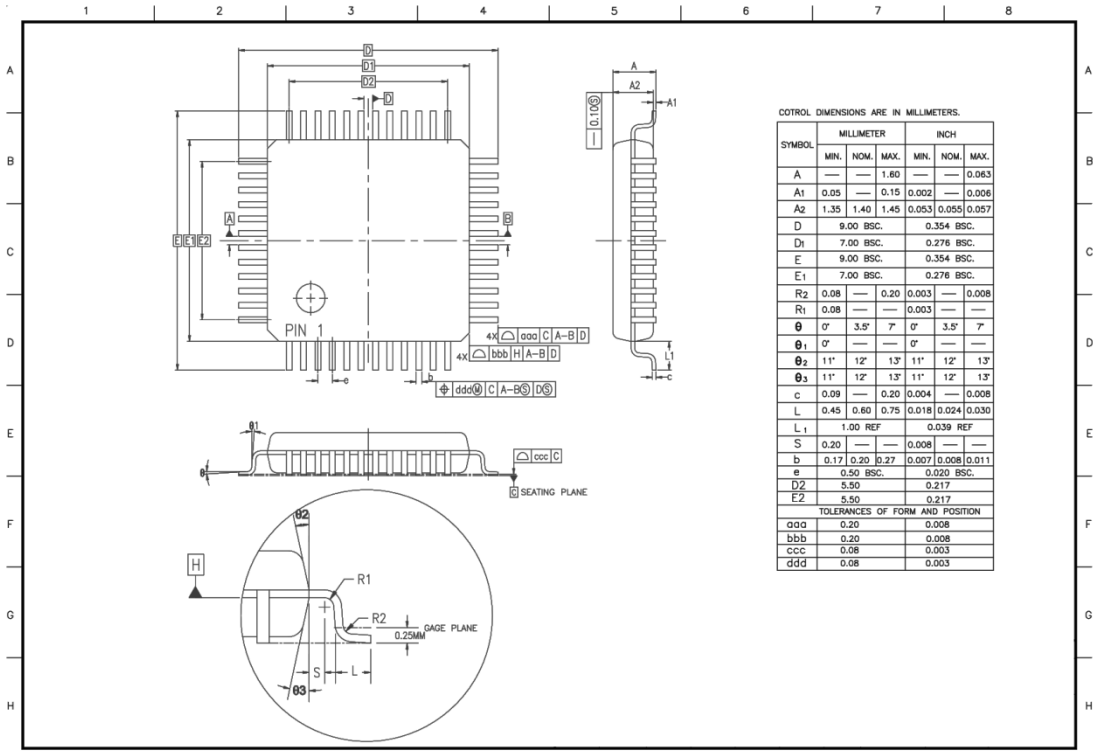
LQFP-64



COTROL DIMENSIONS ARE IN MILLIMETERS.

SYMBOL	MILLIMETER			INCH		
	MIN.	NOM.	MAX.	MIN.	NOM.	MAX.
A	—	—	1.60	—	—	0.063
A1	0.05	—	0.15	0.002	—	0.006
A2	1.35	1.40	1.45	0.053	0.055	0.057
D	12.00 BSC.			0.472 BSC.		
D1	10.00 BSC.			0.394 BSC.		
E	12.00 BSC.			0.472 BSC.		
E1	10.00 BSC.			0.394 BSC.		
R2	0.08	—	0.20	0.003	—	0.008
R1	0.08	—	—	0.003	—	—
θ	0°	3.5°	7°	0°	3.5°	7°
θ1	0°	—	—	0°	—	—
θ2	11°	12°	13°	11°	12°	13°
θ3	11°	12°	13°	11°	12°	13°
c	0.09	—	0.20	0.004	—	0.008
L	0.45	0.60	0.75	0.018	0.024	0.030
L1	1.00 REF			0.039 REF		
S	0.20	—	—	0.008	—	—
b	0.17	0.20	0.27	0.007	0.008	0.011
e	0.50 BSC.			0.020 BSC.		
D2	7.50 REF			0.295		
E2	7.50 REF			0.295		
aaa	0.20			0.008		
bbb	0.20			0.008		
ccc	0.08			0.003		

LQFP-48



COTROL DIMENSIONS ARE IN MILLIMETERS.

SYMBOL	MILLIMETER			INCH		
	MIN.	NOM.	MAX.	MIN.	NOM.	MAX.
A	—	—	1.60	—	—	0.063
A1	0.05	—	0.15	0.002	—	0.006
A2	1.35	1.40	1.45	0.053	0.055	0.057
D	9.00 BSC.			0.354 BSC.		
D1	7.00 BSC.			0.276 BSC.		
E	9.00 BSC.			0.354 BSC.		
E1	7.00 BSC.			0.276 BSC.		
R2	0.08	—	0.20	0.003	—	0.008
R1	0.08	—	—	0.003	—	—
θ	0°	3.5°	7°	0°	3.5°	7°
θ1	0°	—	—	0°	—	—
θ2	11°	12°	13°	11°	12°	13°
θ3	11°	12°	13°	11°	12°	13°
c	0.09	—	0.20	0.004	—	0.008
L	0.45	0.60	0.75	0.018	0.024	0.030
L1	1.00 REF			0.039 REF		
S	0.20	—	—	0.008	—	—
b	0.17	0.20	0.27	0.007	0.008	0.011
e	0.50 BSC.			0.020 BSC.		
D2	5.50			0.217		
E2	5.50			0.217		
TOLERANCES OF FORM AND POSITION						
aaa	0.20			0.008		
bbb	0.20			0.008		
ccc	0.08			0.003		
ddd	0.08			0.003		

3 Pin definitions

LQFP-100

Pin	Pin name	Function		Pin	Pin name	Function
1	PIN_1	IO		26	PIN_26	IO_ADC_IN3_CMP_PA3
2	PIN_2	IO		27	VSS33	GND
3	PIN_3	IO		28	VDD33	VDD33
4	PIN_4	IO		29	PIN_29	IO_ADC_IN4_CMP_PA4_DAC0
5	PIN_5	IO		30	PIN_30	IO_ADC_IN5_CMP_PA5_DAC1
6	VBAT	VBAT		31	PIN_31	IO_ADC_IN6
7	PIN_7	IO_RTC		32	PIN_32	IO_ADC_IN7
8	OSC32_IN	OSC32_IN		33	PIN_33	IO_ADC_IN14
9	OSC32_OUT	OSC32_OUT		34	PIN_34	IO_ADC_IN15
10	VSS33	GND		35	PIN_35	IO_ADC_IN8
11	VDD33	VDD33		36	PIN_36	IO_ADC_IN9
12	OSC_IN	OSC_IN		37	PIN_37	IO_BOOT1
13	OSC_OUT	OSC_OUT		38	PIN_38	IO
14	NRST	NRST		39	PIN_39	IO
15	PIN_15	IO_ADC_IN10		40	PIN_40	IO
16	PIN_16	IO_ADC_IN11		41	PIN_41	IO
17	PIN_17	IO_ADC_IN12		42	PIN_42	IO
18	PIN_18	IO_ADC_IN13		43	PIN_43	IO
19	NC	NC		44	PIN_44	IO
20	VSSA	GNDA		45	PIN_45	IO
21	VREFP	VREFP		46	PIN_46	IO
22	VDDA	VDDA		47	PIN_47	IO
23	PIN_23	IO_WKUP_ADC_IN0_CMP_PA0		48	PIN_48	IO
24	PIN_24	IO_ADC_IN1_CMP_PA1		49	NC	NC
25	PIN_25	IO_ADC_IN2_CMP_PA2		50	VDD33	VDD33

Pin	Pin name	Function		Pin	Pin name	Function
51	PIN_51	IO		76	PIN_76	IO_JTCK
52	PIN_52	IO		77	PIN_77	IO_JTDI
53	PIN_53	IO		78	PIN_78	IO
54	PIN_54	IO		79	PIN_79	IO
55	PIN_55	IO		80	PIN_80	IO
56	PIN_56	IO		81	PIN_81	IO
57	PIN_57	IO		82	PIN_82	IO
58	PIN_58	IO		83	PIN_83	IO
59	PIN_59	IO		84	PIN_84	IO
60	PIN_60	IO		85	PIN_85	IO
61	PIN_61	IO		86	PIN_86	IO
62	PIN_62	IO		87	PIN_87	IO
63	PIN_63	IO		88	PIN_88	IO
64	PIN_64	IO		89	PIN_89	IO_JTDO
65	PIN_65	IO		90	PIN_90	IO_JNTRST
66	PIN_66	IO		91	PIN_91	IO
67	PIN_67	IO		92	PIN_92	IO
68	PIN_68	IO_UART0_TX		93	PIN_93	IO
69	PIN_69	IO_UART0_RX		94	BOOT0	BOOT0
70	PIN_70	IO_USBDM		95	PIN_95	IO
71	PIN_71	IO_USBDP		96	PIN_96	IO
72	PIN_72	IO_JTMS		97	PIN_97	IO
73	NC	NC		98	PIN_98	IO
74	VSS33	GND		99	VSS33	GND
75	VDD33	VDD33		100	VDD33	VDD33

LQFP-64

Pin	Pin name	Function		Pin	Pin name	Function
1	VBAT	VBAT		33	PIN_33	IO
2	PIN_2	IO_RTC		34	PIN_34	IO
3	OSC32_IN	OSC32_IN		35	PIN_35	IO
4	OSC32_OUT	OSC32_OUT		36	PIN_36	IO
5	OSC_IN	OSC_IN		37	PIN_37	IO
6	OSC_OUT	OSC_OUT		38	PIN_38	IO
7	NRST	NRST		39	PIN_39	IO
8	PIN_8	IO_ADC_IN10		40	PIN_40	IO
9	PIN_9	IO_ADC_IN11		41	PIN_41	IO
10	PIN_10	IO_ADC_IN12		42	PIN_42	IO_UART0_TX
11	PIN_11	IO_ADC_IN13		43	PIN_43	IO_UART0_RX
12	VSSA	GNDA		44	PIN_44	IO_USBDM
13	VDDA	VDDA		45	PIN_45	IO_USBDP
14	PIN_14	IO_WKUP_ADC_IN0_CMP_PA0		46	PIN_46	IO_JTMS
15	PIN_15	IO_ADC_IN1_CMP_PA1		47	PIN_47	IO
16	PIN_16	IO_ADC_IN2_CMP_PA2		48	VDD33	VDD33
17	PIN_17	IO_ADC_IN3_CMP_PA3		49	PIN_49	IO_JTCK
18	VSS33	GND		50	PIN_50	IO_JTDI
19	VDD33	VDD33		51	PIN_51	IO
20	PIN_20	IO_ADC_IN4_CMP_PA4_DAC0		52	PIN_52	IO
21	PIN_21	IO_ADC_IN5_CMP_PA5_DAC1		53	PIN_53	IO
22	PIN_22	IO_ADC_IN6		54	PIN_54	IO
23	PIN_23	IO_ADC_IN7		55	PIN_55	IO_JTDO
24	PIN_24	IO_ADC_IN14		56	PIN_56	IO_JNTRST
25	PIN_25	IO_ADC_IN15		57	PIN_57	IO
26	PIN_26	IO_ADC_IN8		58	PIN_58	IO
27	PIN_27	IO_ADC_IN9		59	PIN_59	IO
28	PIN_28	IO_BOOT1		60	BOOT0	BOOT0
29	PIN_29	IO		61	PIN_61	IO
30	PIN_30	IO		62	PIN_62	IO
31	PIN_31	IO		63	VSS33	GND
32	VDD33	VDD33		64	VDD33	VDD33

LQFP-48

Pin	Pin name	Function		Pin	Pin name	Function
1	VBAT	VBAT		25	PIN_25	IO
2	PIN_2	IO_RTC		26	PIN_26	IO
3	OSC32_IN	OSC32_IN		27	PIN_27	IO
4	OSC32_OUT	OSC32_OUT		28	PIN_28	IO
5	OSC_IN	OSC_IN		29	PIN_29	IO
6	OSC_OUT	OSC_OUT		30	PIN_30	IO_UART0_TX
7	NRST	NRST		31	PIN_31	IO_UART0_RX
8	VSSA	GNDA		32	PIN_32	IO_USBDM
9	VDDA	VDDA		33	PIN_33	IO_USBDP
10	PIN_10	IO_WKUP_ADC_IN0_CMP_PA0		34	PIN_34	IO_JTMS
11	PIN_11	IO_ADC_IN1_CMP_PA1		35	PIN_35	IO
12	PIN_12	IO_ADC_IN2_CMP_PA2		36	VDD33	VDD33
13	PIN_13	IO_ADC_IN3_CMP_PA3		37	PIN_37	IO_JTCK
14	PIN_14	IO_ADC_IN4_CMP_PA4_DAC0		38	PIN_38	IO_JTDI
15	PIN_15	IO_ADC_IN5_CMP_PA5_DAC1		39	PIN_39	IO_JTDO
16	PIN_16	IO_ADC_IN6		40	PIN_40	IO_JNTRST
17	PIN_17	IO_ADC_IN7		41	PIN_41	IO
18	PIN_18	IO_ADC_IN8		42	PIN_42	IO
19	PIN_19	IO_ADC_IN9		43	PIN_43	IO
20	PIN_20	IO_BOOT1		44	BOOT0	BOOT0
21	PIN_21	IO		45	PIN_45	IO
22	PIN_22	IO		46	PIN_46	IO
23	VSS33	GND		47	VSS33	GND
24	VDD33	VDD33		48	VDD33	VDD33

4 Memorymapping

	Address	
ROM	0x0001 0000 - 0x0001 1FFF	
System Control	0x0300 0000 - 0x0300 0FFF	
PLIC	0x0C00 0000 - 0x0C20 FFFF	
SRAM	0x2000 0000 - 0x2001 FFFF	
FLASH (XIP)	0x8000 0000 - 0x80FF FFFF	
Option bytes	0x8100 0000 - 0x8100 003F	
RTC	0x4000 0000 - 0x4000 007F	
FLASH control	0x4000 1000 - 0x4000 1FFF	
APB Peripherals	0x4001 0000 - 0x40FF FFFF	
AHB Peripherals	0x4100 0000 - 0x41FF FFFF	
External AHB	0x6000 0000 - 0x7FFF FFFF	

5 Electricalcharacteristics

5.1 Absolute maximum ratings

The maximum ratings are the limits to which the device can be subjected without permanently damaging the device. Note that the device is not guaranteed to operate properly at the maximum ratings. Exposure to the absolute maximum rating conditions for extended periods may affect device reliability.

Table 1. Absolute maximum rating

Symbol	Parameter	Min	Max	Unit
VDD	External voltage range	VSS - 0.3	VSS + 3.6	V
VDDA	External analog supply voltage	VSSA - 0.3	VSSA + 3.6	V
VBAT	External battery supply voltage	VSS - 0.3	VSS + 3.6	V
VIN	Input voltage on I/O	VSS - 0.3	VSS + 3.6	V
Iio	Maximum current for GPIO pins	—	25	mA
Iinj	Injected current on I/O	—	±5	mA
TA	Operating temperature range	-40	+85	℃
TSTG	Storage temperature range	-55	+150	℃
TJ	Maximum junction temperature	—	125	℃

5.2 Recommended DC characteristics

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
VDD	Supply voltage	—	3.0	3.3	3.6	V
VDDA	Analog Supply voltage	—	3.0	3.3	3.6	V
VBAT	Battery supply voltage	—	2.2	—	3.6	V

5.3 Power consumption

Power consumption

The power measurements specified in the tables represent that code with data executing from on- chip Flash with the following specifications.

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
Idd	Supply current (Run mode)	VDD=VBAT=3.3V, HSE=8MHz, System clock=108 MHz, All peripherals enabled	—		—	mA
		VDD=VBAT=3.3V, HSE=8MHz, System clock =108 MHz, All peripherals disabled	—		—	mA
		VDD=VBAT=3.3V, HSE=8MHz, System clock =72MHz, All peripherals enabled	—		—	mA
		VDD=VBAT=3.3V, HSE=8MHz, System Clock =72 MHz, All peripherals disabled	—		—	mA
	Supply current (Sleep mode)	VDD=VBAT=3.3V, HSE=8MHz, CPU clock off, All peripherals enabled	—		—	mA
		VDD=VBAT=3.3V, HSE=8MHz, CPU clock off, All peripherals disabled	—		—	mA
	Supply current (Deep-Sleep mode)	VDD=VBAT=3.3V, All clock off, LSI on, RTC on, All IOs analog mode	—		——	mA
	Supply current (Standby mode)	VDD=VBAT=3.3V, LDO off, LSE off, LSI on, RTC on	—		—	μA
Ibat	Battery supply current (Standby mode)	VDD not available, VBAT=3.3V, LDO off, LSE on, LSI off, RTC on	—		—	μA
		VDD not available, VBAT=3.3 V, LDO off, LSE off, LSI on, RTC on	—		—	μA

5.4 Power up/down

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
Vpor	Power on reset threshold		2.0	2.2	2.4	V
Vpdr	power down reset threshold		1.8	2.0	2.2	V
Vhyst	PDR hysteresis		—	0.2	—	V
Trsttemp	Reset temporization		—	4	—	ms

5.5 External clock characteristics

High-speed external clock generated from a crystal/ceramic resonator. The high-speed external (HSE) clock can be supplied with a 4 to 16 MHz crystal/ceramic resonator oscillator. All the information given in this paragraph are based on characterization results obtained with typical external components specified in Table. In the application, the resonator and the load capacitors have to be placed as close as possible to the oscillator pins in order to minimize output distortion and startup stabilization time. Refer to the crystal resonator manufacturer for more details on the resonator characteristics (frequency, package, accuracy).

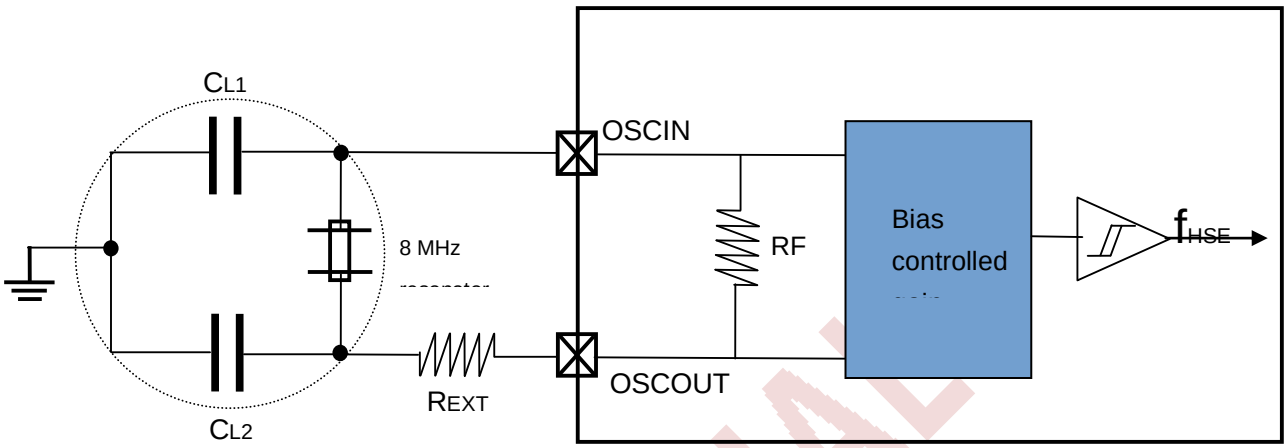
HSE 4-16 MHz oscillator characteristics

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
f_OSC_IN	Oscillator frequency	VDD=3.3V	4	8	16	MHz
RF	Feedback resistor	—	—	1	—	MΩ
C	Recommended load capacitance on OSC_IN and OSC_OUT	—	—	20	30	pF
gm	Oscillator transconductance	—	25	—	—	mA/V
Dosc_out	Oscillator oscillator duty cycle	—	45	50	55	%
T_su_hse	startup time	VDD is stabilized	—	2	—	mS

For CL1 and CL2, it is recommended to use high-quality external ceramic capacitors in the 5pF to 25pF range (typ.), designed for high-frequency applications, and selected to match the requirements of the crystal or resonator (see Figure 24). CL1 and CL2 are usually the same size. The crystal manufacturer typically specifies a load capacitance which is the series combination of

CL1 and CL2. PCB and MCU pin capacitance must be included (10pF can be used as a rough estimate of the combined pin and board capacitance) when sizing CL1 and CL2.

Typical application with an 8 MHz crystal

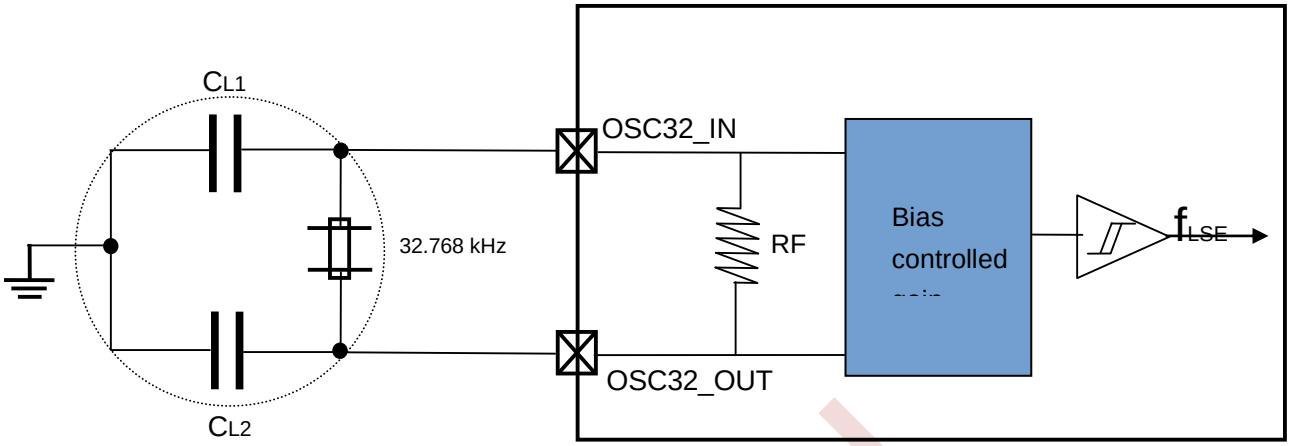


REXT value depends on the crystal characteristics.

Low-speed external clock generated from a crystal/ceramic resonator The low-speed external (LSE) clock can be supplied with a 32.768 kHz crystal/ceramic resonator oscillator. All the information given in this paragraph are based on characterization results obtained with typical external components specified in Table. In the application, the resonator and the load capacitors have to be placed as close as possible to the oscillator pins in order to minimize output distortion and startup stabilization time. Refer to the crystal resonator manufacturer for more details on the resonator characteristics (frequency, package, accuracy).

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
f_LSE	Oscillator frequency	VDD=VBAT=3.3V		32.768	1000	KHz
RF	Feedback resistor	—	—	10	—	MΩ
C	Recommended load capacitance on OSC32_IN and OSC32_OUT	—	—	—	15	pF
gm	Oscillator transconductance	—	10	—	—	uA/V
Dosc_out	Oscillator oscillator duty cycle	—	45	50	55	%
T_su_lse	startup time	VDD is stabilized	—	3	—	S

Typical application with a 32.768 kHz crystal



5.6 Internal clock source characteristics

High-speed internal (HSI) RC oscillator

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
$f_{\_HSI}$	Oscillator frequency	VDD=3.3V	10	20	40	MHz
D_{uty_HSI}	Duty cycle		45	50	55	%
$T_{\_su\_HSI}$	HSI oscillator startup time		1	—	2	μs

5.7 PLL characteristics

Symbol	Parameter	Min	Typ	Max	Unit
f_{PLL_IN}	PLL input clock	4	20	50	MHz
	PLL input clock duty cycle	40	50	60	%
f_{PLL_OUT}	PLL multiplier output clock	2	200	300	MHz
t_{LOCK}	PLL lock time	—	—	400	μs
Jitter	Cycle-to-cycle jitter	—	—	400	ps

5.8 emory characteristics

Flash memory characteristics

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
PE_{CYC}	Number of guaranteed program /erase cycles before failure (Endurance)	TA=-40 ℃ ~ +85 ℃	100	—	—	kcycles
t_{RET}	Data retention time	TA=125 ℃	20	—	—	years
t_{PROG}	Word programming time	TA=-40 ℃ ~ +85 ℃	—	2	3	ms
t_{ERASE}	Page erase time	TA=-40 ℃ ~ +85 ℃	—	8	20	ms
t_{MERASE}	Mass erase time	TA=-40 ℃ ~ +85 ℃	—	8	20	ms

5.9 IO characteristics

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
V_{IL}	Standard IO Low level input voltage	VDD ≥ 3.0V	-0.3	—	0.8	V
V_{IH}	Standard IO High level input voltage	VDD ≥ 3.0V	1.5	—	3.6	V
V_{OL}	Low level output voltage	VDD ≥ 3.0V	—	—	0.2	V
V_{OH}	High level output voltage	VDD ≥ 3.0V	2.8	—	—	V
R_{PU}	Internal pull-up resistor	VIN=VSS	30	40	50	kΩ
R_{PD}	Internal pull-down resistor	VIN=VDD	30	40	50	kΩ

5.10 ADC characteristics

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
VDDA	Operating voltage		3.0	3.3	3.6	V
VIN	ADC input voltage range		0	—	VREFP	V
f_{ADC}	ADC clock		0.5	—	13	MHz
f_s	Sampling rate		—	—	1	MHz
t_{conv}	ADC conversion time		1	—	20	μs
R_{ADC}	Input sampling switch resistance		—	—	0.5	kΩ
C_{ADC}	Input sampling capacitance		—	8	—	pF
tsu	Startup time		—	—	2	μs

5.11 DAC characteristics

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
VDDA	Operating voltage		3.0	3.3	3.6	V
VREFP	Reference supply voltage	VREFP should always below VDDA	3.0	3.3	3.6	V
R_{LOAD}	Load resistance	Resistive load vs. VSSA with buffer ON	5	—	—	kΩ
C_{LOAD}	Load capacitance	No pin/pad capacitance included	—	—	50	pF

5.12 Comparator characteristics

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
VDDA	Analog supply voltage	—	3.0	3.3	3.6	V
VIN	Comparator input voltage range	—	0	—	VDDA	V
t_{start}	Comparator startup time	VDDA ≥ 3.0 V	—	—	10	μs
t_D	Propagation delay for full range step with 100 mV overdrive	VDDA ≥ 3.0 V	—	—	40	ns
V_{OFFSET}	Comparator offset error	VDDA ≥ 3.0 V	—	—	±25	mV

5.13 I² C characteristics

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
f_{SCL}	SCL clock frequency	—	0	—	100	KHz
$t_{SCL(H)}$	SCL clock high time	—	4.0	—	0.6	ns
$t_{SCL(L)}$	SCL clock low time	—	4.7	—	1.3	ns

5.14 SPI characteristics

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
fSCK	SCK clock frequency	—	—	—	18	MHz
tSCK(H)	SCK clock high time	—	19	—	—	ns
tSCK(L)	SCK clock low time	—	19	—	—	ns
SPI master mode						
tV(MO)	Data output valid time	—	—	—	25	ns
tH(MO)	Data output hold time	—	2	—	—	ns
tSU(MI)	Data input setup time	—	5	—	—	ns
tH(MI)	Data input hold time	—	5	—	—	ns
SPI slave mode						
tSU(NSS)	NSS enable setup time	fPCLK=54MHz	74	—	—	ns
tH(NSS)	NSS enable hold time	fPCLK=54MHz	37	—	—	ns
tA(SO)	Data output access time	fPCLK=54MHz	0	—	55	ns
tDIS(SO)	Data output disable time	—	3	—	10	ns
tV(SO)	Data output valid time	—	—	—	25	ns
tH(SO)	Data output hold time	—	15	—	—	ns
tSU(SI)	Data input setup time	—	5	—	—	ns
tH(SI)	Data input hold time	—	4	—	—	ns

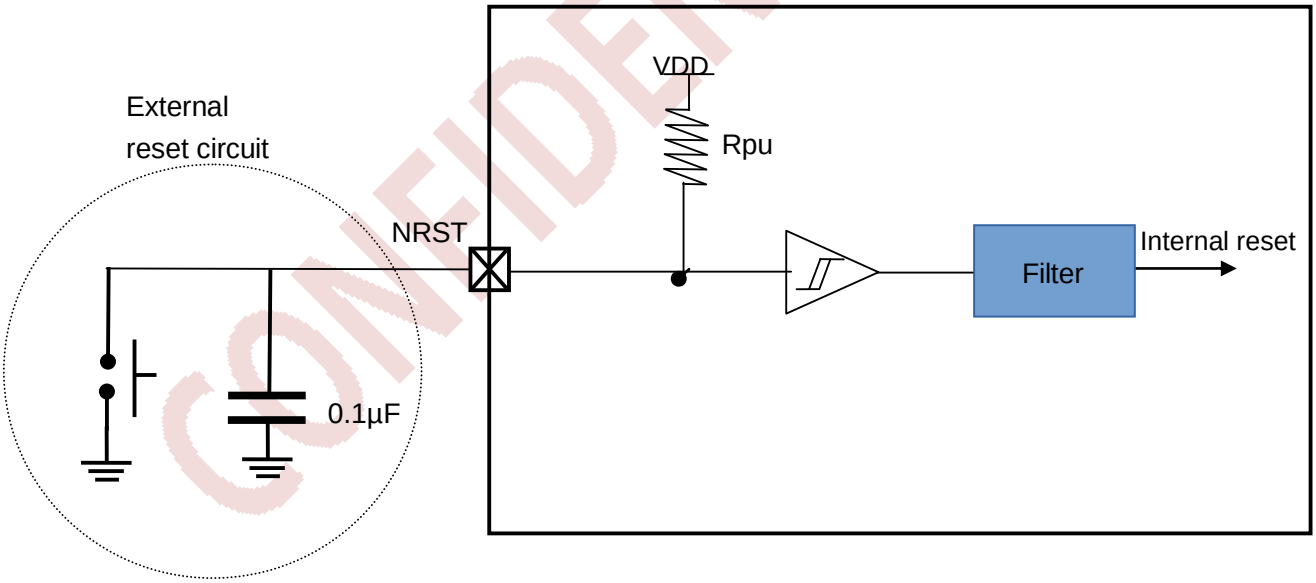
5.15 NRST pin characteristics

The NRST pin input driver uses CMOS technology. It is connected to a permanent pull-up resistor, RPU.

NRST pin characteristics

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
VIL(NRST)	NRST Input low level voltage	—	—	—	0.2VDD	V
VIH(NRST)	NRST Input high level voltage	—	0.5VDD	—	—	V
Vhys(NRST)	NRST Schmitt trigger voltage hysteresis	—	—	200	—	mV
RPU	Weak pull-up equivalent resistor	VIN = VSS	30	40	50	kΩ
VF(NRST)	NRST Input filtered pulse	—	—	—	100	ns
VNF(NRST)	NRST Input not filtered pulse	—	500	—	—	ns

Recommended NRST pin protection



Symbol	Parameter	Conditions	Min	Typ	Max	Unit
fPP	Clock frequency in data transfer mode	—	0	—	48	MHz
tW(CKL)	Clock low time	fpp = 48 MHz	10.5	11	—	ns
tW(CKH)	Clock high time	fpp = 48 MHz	9.5	10	—	ns
CMD, D inputs (referenced to CK) in MMC and SD HS mode						
tISU	Input setup time HS	fpp = 48 MHz	4	—	—	ns
tIH	Input hold time HS	fpp = 48 MHz	3	—	—	ns
CMD, D outputs (referenced to CK) in MMC and SD HS mode						
tOV	Output valid time HS	fpp = 48 MHz	—	—	13.8	ns
tOH	Output hold time HS	fpp = 48 MHz	12	—	—	ns
CMD, D inputs (referenced to CK) in SD default mode						
tISUD	Input setup time SD	fpp = 24 MHz	3	—	—	ns
tIHD	Input hold time SD	fpp = 24 MHz	3	—	—	ns
CMD, D outputs (referenced to CK) in SD default mode						
tOVD	Output valid default time SD	fpp = 24 MHz	—	2.4	2.8	ns
tOHD	Output hold default time SD	fpp = 24 MHz	0.8	—	—	ns

5.16 USART characteristics

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
fSCK	SCK clock frequency	fPCLK = 120 MHz	—	—	60	MHz
tSCK(H)	SCK clock high time	fPCLK = 120 MHz	7.5	—	—	ns
tSCK(L)	SCK clock low time	fPCLK = 120 MHz	7.5	—	—	ns

5.17 SDIO characteristics

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
fPP	Clock frequency in data transfer mode	—	0	—	48	MHz
tW(CKL)	Clock low time	fpp = 48 MHz	10.5	11	—	ns
tW(CKH)	Clock high time	fpp = 48 MHz	9.5	10	—	ns
CMD, D inputs (referenced to CK) in MMC and SD HS mode						
tISU	Input setup time HS	fpp = 48 MHz	4	—	—	ns
tIH	Input hold time HS	fpp = 48 MHz	3	—	—	ns
CMD, D outputs (referenced to CK) in MMC and SD HS mode						
tOV	Output valid time HS	fpp = 48 MHz	—	—	13.8	ns
tOH	Output hold time HS	fpp = 48 MHz	12	—	—	ns
CMD, D inputs (referenced to CK) in SD default mode						
tISUD	Input setup time SD	fpp = 24 MHz	3	—	—	ns
tIHD	Input hold time SD	fpp = 24 MHz	3	—	—	ns
CMD, D outputs (referenced to CK) in SD default mode						
tOVD	Output valid default time SD	fpp = 24 MHz	—	2.4	2.8	ns
tOHD	Output hold default time SD	fpp = 24 MHz	0.8	—	—	ns

5.18 USB characteristics

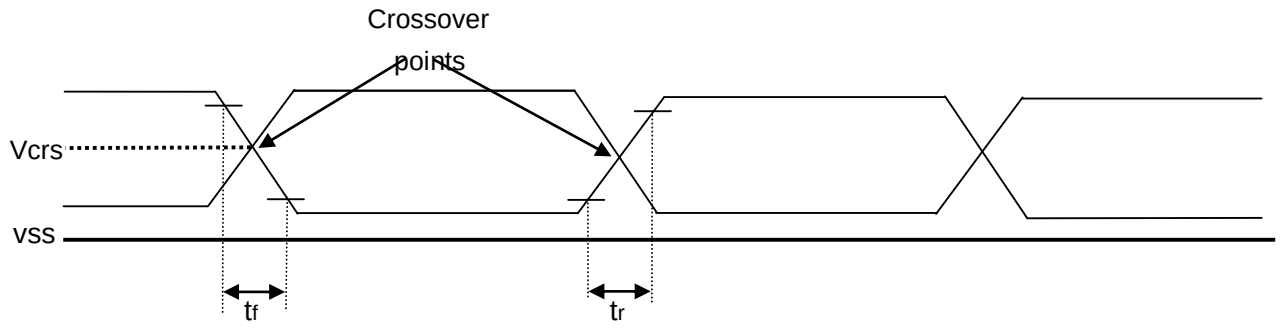
USB DC electrical characteristics

Symbol		Parameter	Conditions	Min	Typ	Max	Unit
Input levels	VDD	USB operating voltage	—	3	—	3.3	V
	VDI	Differential input sensitivity	I(USBDP, USBDM)	0.2	—	—	V
	VCM	Differential common mode range	Includes VDI range	0.8	—	2.5	V
	VSE	Single ended receiver threshold	—	1.3	—	2.0	V
Output Levels	VOL	Static output level low	RL of 1.5 kΩ to 3.6 V	—	—	0.3	V
	VOH	Static output level high	RL of 15 kΩ to VSS	2.8	3.3	3.6	V
tSTARTUP		USBFS startup time	—	—	—	1	μs

USB full speed-electrical characteristics

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
tR	Rise time	CL = 50 pF	4	—	20	ns
tF	Fall time	CL = 50 pF	4	—	20	ns
tRFM	Rise/ fall time matching	tR/tF	90	—	110	%
vCRS	Output signal crossover voltage	—	1.3	—	2.0	V

USB timings: definition of data signal rise and fall time



5.19 TIMER characteristics

TIMER characteristics

Symbol	Parameter	Conditions	Min	Max	Unit
tres	Timer resolution time	—	1	—	$t_{TIMERxCLK}$
		$f_{TIMERxCLK}$ = 120MHz	8.4	—	ns
fEXT	Timer external clock frequency	—	0	$f_{TIMERxCLK}/2$	MHz
		$f_{TIMERxCLK}$ = 120MHz	0	60	MHz
RES	Timer resolution	—	—	16	bit
$t_{COUNTER}$	16-bit counter clock period when internal clock is selected	—	1	65536	$t_{TIMERxCLK}$
		$f_{TIMERxCLK}$ = 120MHz	0.0084	546	μs
t_{MAX_COUNT}	Maximum possible count	—	—	65536x65536	$t_{TIMERxCLK}$
		$f_{TIMERxCLK}$ = 120MHz	—	35.7	s

5.20 CRC(Cyclic redundancy check calculation unit)

5.20.1 Introduction

The CRC (cyclic redundancy check) calculation unit is used to get a CRC code from 8-, 16- or 32-bit data word and a generator polynomial.

Among other applications, CRC-based techniques are used to verify data transmission or storage integrity. In the scope of the functional safety standards, they offer a means of verifying the Flash memory integrity. The CRC calculation unit helps compute a signature of the software during runtime, to be compared with a reference signature generated at link time and stored at a given memory location.

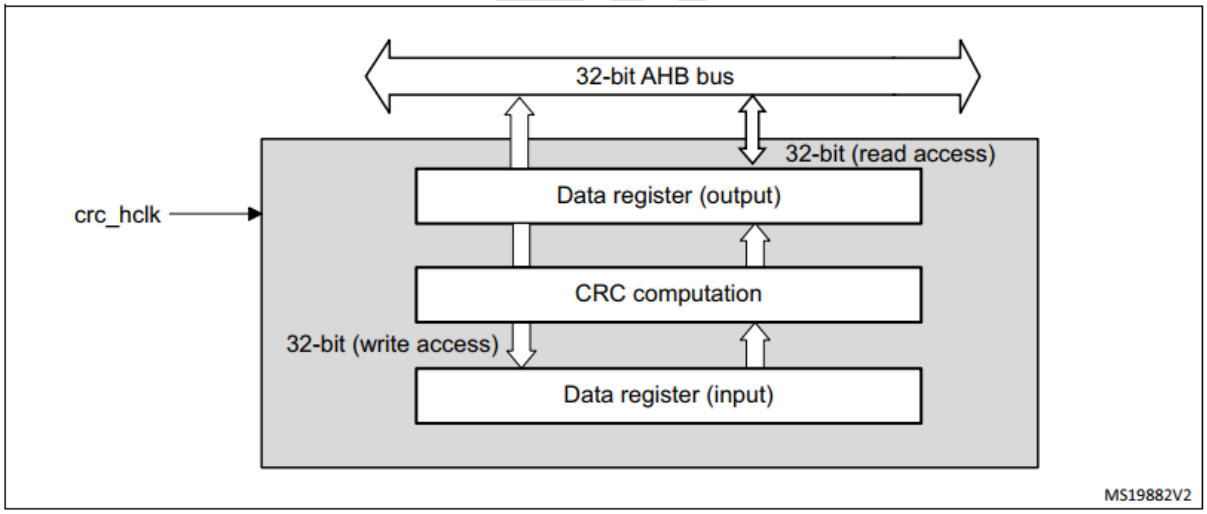
5.20.2 CRC main features

- Fully programmable polynomial with programmable size (7, 8, 16, 32 bits)
- Handles 8-,16-, 32-bit data size
- Programmable CRC initial value
- Single input/output 32-bit data register
- Input buffer to avoid bus stall during calculation
- CRC computation done in 4 AHB clock cycles (HCLK) for the 32-bit data size
- General-purpose 8-bit register (can be used for temporary storage)
- Reversibility option on I/O data

5.20.3 CRC functional description

5.20.3.1 CRC block diagram

Figure 7. CRC calculation unit block diagram



5.20.3.2 CRC internal signals

Table 15. CRC internal input/output signals

Signal name	Signal type	Description
crc_hclk	Digital input	AHB clock

5.20.3.3 CRC operation

The CRC calculation unit has a single 32-bit read/write data register (CRC_DR). It is used to input new data (write access), and holds the result of the previous CRC calculation (read access).

Each write operation to the data register creates a combination of the previous CRC value (stored in CRC_DR) and the new one. CRC computation is done on the whole 32-bit data word or byte by byte depending on the format of the data being written. The CRC_DR register can be accessed by word, right-aligned half-word and right-aligned byte. For the other registers only 32-bit access is allowed.
The duration of the computation depends on data width:

- 4 AHB clock cycles for 32-bit
- 2 AHB clock cycles for 16-bit
- 1 AHB clock cycles for 8-bit

An input buffer allows to immediately write a second data without waiting for any wait states due to the previous CRC calculation.
The data size can be dynamically adjusted to minimize the number of write accesses for a given number of bytes. For instance, a CRC for 5 bytes can be computed with a word write followed by a byte write.
The input data can be reversed, to manage the various endianness schemes. The reversing operation can be performed on 8 bits, 16 bits and 32 bits depending on the REV_IN[1:0] bits in the CRC_CR register.

For example: input data 0x1A2B3C4D is used for CRC calculation as:

0x58D43CB2 with bit-reversal done by byte

0xD458B23C with bit-reversal done by half-word

0xB23CD458 with bit-reversal done on the full word

The output data can also be reversed by setting the REV_OUT bit in the CRC_CR

register.

The operation is done at bit level: for example, output data 0x11223344 is converted into 0x22CC4488.

The CRC calculator can be initialized to a programmable value using the RESET control bit in the CRC_CR register (the default value is 0xFFFFFFFF).

The initial CRC value can be programmed with the CRC_INIT register. The CRC_DR register is automatically initialized upon CRC_INIT register write access.

The CRC_IDR register can be used to hold a temporary value related to CRC calculation. It is not affected by the RESET bit in the CRC_CR register.

- Polynomial programmability

The polynomial coefficients are fully programmable through the CRC_POL register, and the polynomial size can be configured to be 7, 8, 16 or 32 bits by programming the

POLYSIZE[1:0] bits in the CRC_CR register. Even polynomials are not supported.

If the CRC data is less than 32-bit, its value can be read from the least significant bits of the CRC_DR register.

To obtain a reliable CRC calculation, the change on-fly of the polynomial value or size can not be performed during a CRC calculation. As a result, if a CRC calculation is ongoing, the application must either reset it or perform a CRC_DR read before changing the polynomial.

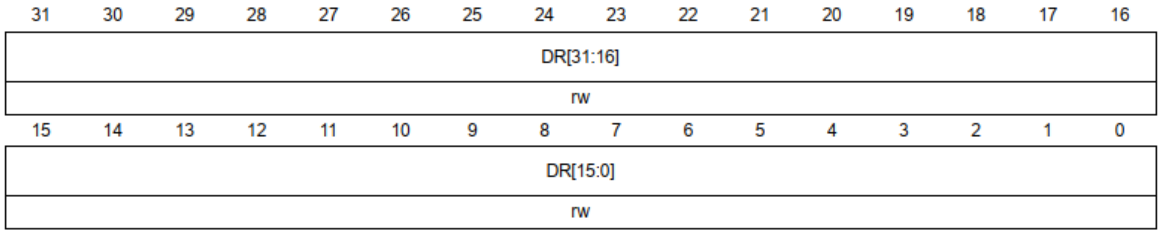
The default polynomial value is the CRC-32 (Ethernet) polynomial: 0x4C11D

5.20.4 CRC registers

5.20.4.1 Data register (CRC_DR)

Address offset: 0x00

Reset value: 0xFFFF FFFF



Bits 31:0 **DR[31:0]:** Data register bits

This register is used to write new data to the CRC calculator.

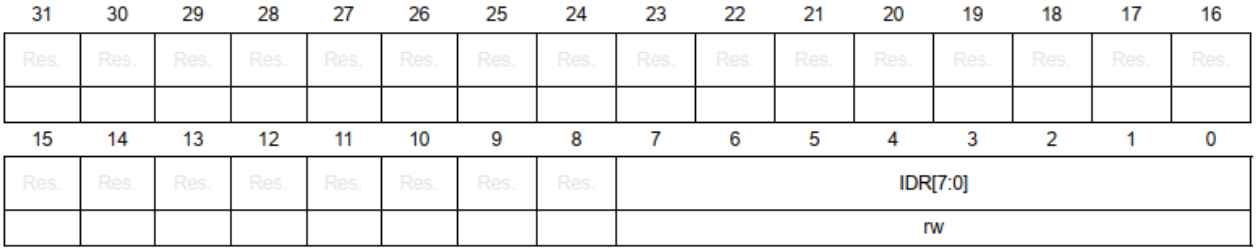
It holds the previous CRC calculation result when it is read.

If the data size is less than 32 bits, the least significant bits are used to write/read the correct value.

5.20.4.2 Independent data register (CRC_IDR)

Address offset: 0x04

Reset value: 0x0000 0000



Bits 31:8 Reserved, must be kept cleared.

Bits 7:0 **IDR[7:0]**: General-purpose 8-bit data register bits

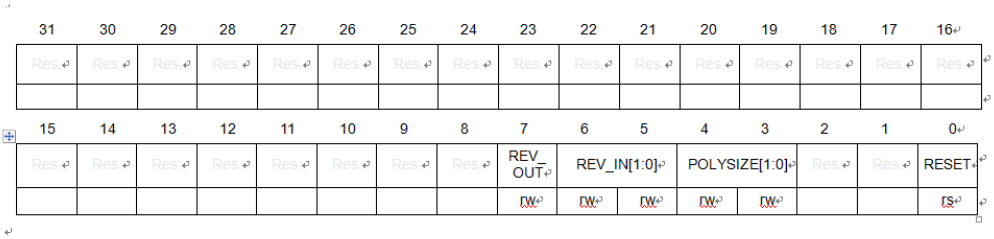
These bits can be used as a temporary storage location for one byte.

This register is not affected by CRC resets generated by the RESET bit in the CRC_CR register

5.20.4.3 Control register (CRC_CR)

Address offset: 0x08

Reset value: 0x0000 0000



Bits 31:8 Reserved, must be kept cleared.

Bit 7 **REV_OUT**: Reverse output data

This bit controls the reversal of the bit order of the output data. 0: Bit order not affected

1: Bit-reversed output format

Bits 6:5 **REV_IN[1:0]**: Reverse input data

These bits control the reversal of the bit order of the input data 00: Bit order not affected

01: Bit reversal done by byte

10: Bit reversal done by half-word 11: Bit reversal done by word

Bits 4:3 **POLYSIZE[1:0]**: Polynomial size

These bits control the size of the polynomial. 00: 32 bit polynomial

01: 16 bit polynomial

10: 8 bit polynomial

11: 7 bit polynomial

Bits 2:1 Reserved, must be kept cleared.

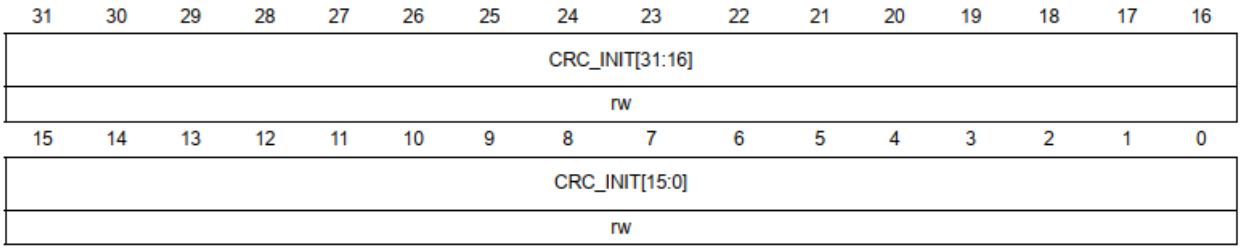
Bit 0 **RESET**: RESET bit

This bit is set by software to reset the CRC calculation unit and set the data register to the value stored in the CRC_INIT register. This bit can only be set, it is automatically cleared by

hardware

5.20.4.4 Initial CRC value (CRC_INIT)

Address offset: 0x10



Reset value: 0xFFFF FFFF

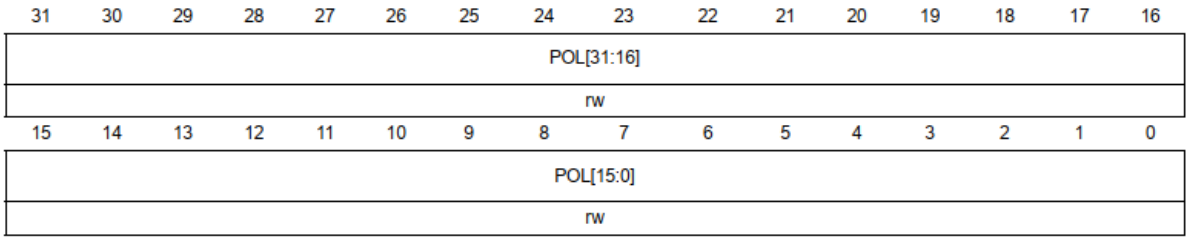
Bits 31:0 **CRC_INIT***: Programmable initial CRC value*

This register is used to write the CRC initial value.

5.20.4.5 CRC polynomial (CRC_POL)

Address offset: 0x14

Reset value: 0x04C11DB7



Bits 31:0 **POL[31:0]***:* Programmable polynomial

This register is used to write the coefficients of the polynomial to be used for CRC calculation.

If the polynomial size is less than 32 bits, the least significant bits have to be used to program the correct value.

5.20.4.6 CRC register map

Table 16. CRC register map and reset values

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	CRC_DR	DR[31:0]																															
	Reset value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0x04	CRC_IDR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	IDR[7:0]							
	Reset value																									0	0	0	0	0	0	0	0
0x08	CRC_CR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	REV_OUT	REV_IN[1:0]		POLYSIZE[1:0]		Res	Res	RESET
	Reset value																									0	0	0	0	0			0
0x10	CRC_INIT	CRC_INIT[31:0]																															
	Reset value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0x14	CRC_POL	Polynomial coefficients																															
	Reset value	0x04C11DB7																															

5.21 DMA-Summary of registers

Summary of registers

Table 3-1 lists the DMAC registers.

Table 3-1 Register summary

Name	Address (base+)	Type	Reset value	Description
DMACIntStatus	0x000	RO	0x00	See *Interrupt Status Register* on page 3-10
DMACIntTCStatus	0x004	RO	0x00	See *Interrupt Terminal Count Status Register* on page 3-10
DMACIntTCClear	0x008	WO	-	See *Interrupt Terminal Count Clear Register* on page 3-11
DMACIntErrorStatus	0x00C	RO	0x00	See *Interrupt Error Status Register* on page 3-11
DMACIntErrClr	0x010	WO	-	See *Interrupt Error Clear Register* on page 3-12
DMACRawIntTCStatus	0x014	RO	-	See *Raw Interrupt Terminal Count Status Register* on page 3-13
DMACRawIntErrorStatus	0x018	RO	-	See *Raw Error Interrupt Status Register* on page 3-13
DMACEnbldChns	0x01C	RO	0x00	See *Enabled Channel Register* on page 3-14
DMACSoftBReq	0x020	R/W	0x0000	See *Software Burst Request Register* on page 3-14
DMACSoftSReq	0x024	R/W	0x0000	See *Software Single Request Register* on page 3-15
DMACSoftLBReq	0x028	R/W	0x0000	See *Software Last Burst Request Register* on page 3-16
DMACSoftLSReq	0x02C	R/W	0x0000	See *Software Last Single Request Register* on page 3-16
DMACConfiguration	0x030	R/W	0b000	See *Configuration Register* on page 3-18
DMACSync	0x34	R/W	0x0000	See *Synchronization Register* on page 3-19
	0x38 – 0x0EC-			Reserved
From 0x100, 8 channel of each below				
DMACC0SrcAddr	0x100	R/W	0x00000000	See *Channel Source Address Registers* on page 3-21
DMACC0DestAddr	0x104	R/W	0x00000000	See *Channel Destination Address Registers* on page 3-21

DMACC0LLI	0x108	R/W	0x00000000	See *Channel Linked List Item Registers* on page 3-22
DMACC0Control	0x10C	R/W	0x00000000	See *Channel Control Registers* on page 3-23
DMACC0Configuration	0x110	R/W	0x00000	See *Channel Configuration Registers* on page 3-27
Name	**Address (base+)**	**Type**	**Reset value**	**Description**
DMACC1SrcAddr	0x120	R/W	0x00000000	See Channel Source Address Registers on page 3-21
DMACC1DestAddr	0x124	R/W	0x00000000	See Channel Destination Address Registers on page 3-21
DMACC1LLI	0x128	R/W	0x00000000	See Channel Linked List Item Registers on page 3-22
DMACC1Control	0x12C	R/W	0x00000000	See Channel Control Registers on page 3-23
DMACC1Configuration	0x130	R/W	0x00000	See Channel Configuration Registers on page 3-27
DMACC2SrcAddr	0x140	R/W	0x00000000	See Channel Source Address Registers on page 3-21
DMACC2DestAddr	0x144	R/W	0x00000000	See Channel Destination Address Registers on page 3-21
DMACC2LLI	0x148	R/W	0x00000000	See Channel Linked List Item Registers on page 3-22
DMACC2Control	0x14C	R/W	0x00000000	See Channel Control Registers on page 3-23
DMACC2Configuration	0x150	R/W	0x00000	See Channel Configuration Registers on page 3-27
DMACC3SrcAddr	0x160	R/W	0x00000000	See Channel Source Address Registers on page 3-21
DMACC3DestAddr	0x164	R/W	0x00000000	See Channel Destination Address Registers on page 3-21
DMACC3LLI	0x168	R/W	0x00000000	See Channel Linked List Item Registers on page 3-22
DMACC3Control	0x16C	R/W	0x00000000	See Channel Control Registers on page 3-23
DMACC3Configuration	0x170	R/W	0x00000	See Channel Configuration Registers on page 3-27
DMACC4SrcAddr	0x180	R/W	0x00000000	See Channel Source Address Registers on page 3-21
DMACC4DestAddr	0x184	R/W	0x00000000	See Channel Destination Address Registers on page 3-21

DMACC4LLI	0x188	R/W	0x00000000	See Channel Linked List Item Registers on page 3-22
DMACC4Control	0x18C	R/W	0x00000000	See Channel Control Registers on page 3-23
Name	**Address (base+)**	**Type**	**Reset value**	**Description**
DMACC4Configuration	0x190	R/W	0x00000	See Channel Configuration Registers on page 3-27
DMACC5SrcAddr	0x1A0	R/W	0x00000000	See Channel Source Address Registers on page 3-21
DMACC5DestAddr	0x1A4	R/W	0x00000000	See Channel Destination Address Registers on page 3-21
DMACC5LLI	0x1A8	R/W	0x00000000	See Channel Linked List Item Registers on page 3-22
DMACC5Control	0x1AC	R/W	0x00000000	See Channel Control Registers on page 3-23
DMACC5Configuration	0x1B0	R/W	0x00000	See Channel Configuration Registers on page 3-27
DMACC6SrcAddr	0x1C0	R/W	0x00000000	See Channel Source Address Registers on page 3-21
DMACC6DestAddr	0x1C4	R/W	0x00000000	See Channel Destination Address Registers on page 3-21
DMACC6LLI	0x1C8	R/W	0x00000000	See Channel Linked List Item Registers on page 3-22
DMACC6Control	0x1CC	R/W	0x00000000	See Channel Control Registers on page 3-23
DMACC6Configuration	0x1D0	R/W	0x00000	See Channel Configuration Registers on page 3-27
DMACC7SrcAddr	0x1E0	R/W	0x00000000	See Channel Source Address Registers on page 3-21
DMACC7DestAddr	0x1E4	R/W	0x00000000	See Channel Destination Address Registers on page 3-21
DMACC7LLI	0x1E8	R/W	0x00000000	See Channel Linked List Item Registers on page 3-22
DMACC7Control	0x1EC	R/W	0x00000000	See Channel Control Registers on page 3-23
DMACC7Configuration	0x1F0	R/W	0x00000	ee Channel Configuration Registers on page 3-27
DMACPeriphID0	0xFE0	RO	0x80	See DMACPeriphID0 Register on page 3-30

DMACPeriphID1	0xFE4	RO	0x10	See DMACPeriphID1 Register on page 3-31
DMACPeriphID2	0xFE8	RO	0x04	See DMACPeriphID2 Register on page 3-31
DMACPeriphID3	0xFEC	RO	0x0A	See DMACPeriphID3 Register on page 3-32
DMACPCellID0	0xFF0	RO	0x0D	See DMACPCellID0 Register on page 3-35
Name	**Address (base+)**	**Type**	**Reset value**	**Description**
DMACPCellID1	0xFF4	RO	0xF0	See DMACPCellID1 Register on page 3-35
DMACPCellID2	0xFF8	RO	0x05	See DMACPCellID2 Register on page 3-36
DMACPCellID3	0xFFC	RO	0xB1	See DMACPCellID3 Register on page 3-36
DMACITCR	0x500	R/W	0x0	See Test Control Register on page 4-4
DMACITOP1	0x504	R/W	0x0000	See Integration Test Output Register 1 on page 4-5
DMACITOP2	0x508	R/W	0x0000	See Integration Test Output Register 2 on page 4-5
DMACITOP3	0x50C	R/W	0x0	See Integration Test Output Register 3 on page 4-6

5.22 Interrupt Controller

5.22.1 Interrupt overview

5.22.1.1 Local interrupts

4 local interrupts (LOCAL_INT0-3) are connected directly to the core and have lower latencies. They have fixed priorities.

5.22.1.2 External interrupts

External interrupts are routed through the Platform-Level Interrupt Controller (PLIC). They have programmable priority levels and a threshold. The interrupt numbrs are listed below:

Interrupt Name	Interrupt Number	Comment
FLASH	1	
RTC	2	
FCB0	3	
WATCHDOG0	4	
SPI0	5	
SPI1	6	
GPIO0	7	
GPIO1	8	
GPIO2	9	
GPIO3	10	
GPIO4	11	
GPIO5	12	
GPIO6	13	
GPIO7	14	
GPIO8	15	
GPIO9	16	
TIMER0	17	
TIMER1	18	
GPTIMER0	19	
GPTIMER1	20	
GPTIMER2	21	
GPTIMER3	22	
GPTIMER4	23	
UART0	24	
UART1	25	

UART2	26	
UART3	27	
UART4	28	
CAN0	29	
I2C0	30	
I2C1	31	
DMAC0_INTR	32	DMA combined interrupt
DMAC0_INTTC	33	DMA terminal count interrupt
DMAC0_INTERR	34	DMA error interrupt
USB0	35	
MAC0	36	
EXT_INT0	37	
EXT_INT1	38	
EXT_INT2	39	
EXT_INT3	40	
EXT_INT4	41	
EXT_INT5	42	
EXT_INT6	43	
EXT_INT7	44	

5.22.1.3 Machine software interrupt

Can be generated and cleared by software.

5.22.1.4 Machine timer interrupt

Can be generated from the 64-bit machine systick timer (MTIME) and the 64-bit timer compare register (MTIMECP). An interrupt is generated when machine timer interrupt is enabled and MTIME >= MTIMECP.

5.22.1.5 Overall priority

From highest priority to lowest:

- LOCAL_INT3
- LOCAL_INT2
- LOCAL_INT1
- LOCAL_INT0
- External interrupts from PLIC
- Machine software interrupt
- Machine timer interrupt

5.22.1.6 Interrupt enable

- The machine interrupt enable (MIE) bit of the RISC-V machine status register (mstatus) must be set as a global enable for all interrupts.
- Corresponding bits in the RISC-V machine interrupt enable register (mie) must be set for each type of interrupt to work:
  - The machine external interrupt enable (MEIE) bit for external interrupts.
  - The machine software interrupt enable (MSIE) bit for machine software interrupt.
  - The machine timer interrupt enable (MTIE) bit for machine timer interrupt.
  - Bits 16-19 for LOCAL_INT0-3, respectively.

5.22.2 Interrupt registers

- Machine software interrupt pending (MSIP)
  - Address: 0x2000000
  - Bit 0:
    - Write 1 to trigger machine software interrupt
    - Write 0 to clear the pending status
- Machine timer compare low (MTIMECMP_LO)
  - Address: 0x2004000
  - Bit [31:0]: Lower 32 bits of the machine timer compare register
- Machine timer compare high (MTIMECMP_HI)
  - Address: 0x2004004
  - Bit [31:0]: Higher 32 bits of the machine timer compare register
- Machine timer low (MTIME_LO)
  - Address: 0x200bFF8
  - Bit [31:0] Lower 32 bits of the machine timer register
- Machine timer high (MTIME_HI)
  - Address: 0x200bFFC
  - Bit [31:0] Higher 32 bits of the machine timer register
- External interrupt priority (PRIORITY)
  - Address: 0xC000000 + (interrupt number * 4)

- Each priority registers holds the priority level of the corresponding interrupt
- The valid range of priority level is from 0 (lowest, interrupt disabled) to 15 (highest).

- External interrupt pending (PENDING)
  - Address: 0xC001000
  - Each interrupt has 1 bit pending status. The bit offset is decided by the interrupt number.
  - The bit is set automatically by hardware when the corresponding interrupt is triggered and is cleared automatically by reading the CLAIM_COMPLETE register when the corresponding interrupt has the highest priority.
- External interrupt enable (ENABLE)
  - Address: 0xC002000
  - Each interrupt has 1 bit enable. The bit offset is decided by the interrupt number.
  - Each bit can be set or cleared by software.
- External interrupt threshold (THRESHOLD)
  - Address: 0xC200000
  - Bit [3:0]: Can be set by software to determine the external interrupt threshold. Only those external interrupts that have higher priority than THRESHOLD will trigger an interrupt to the CPU core.
- External interrupt claim and complete (CLAIM_COMPLETE)
  - Address: 0xC200004
  - Reading this register will return the highest priority pending interrupt number and clear the corresponding pending bit (only for enabled interrupts with above threshold priority). Since Interrupts are numbered starting from 1, a read value of 0 means no active interrupt. A write to this register will complete the interrupt and make the written interrupt number ready to respond again

5.23 Timer-Summary of registers

Table 3-1 Summary of registers

Address	Type	Width	Reset value	Name	Description
Base+0x00	Read/write	32	0x00000000	Timer1Load	See *Load Register, TimerXLoad* on page 3-4
Base+0x04	Read	32	0xFFFFFFFF	Timer1Value	See Current Value Register, TimerXValue on page 3-5
Base+0x08	Read/write	8	0x20	Timer1Control	See Control Register, TimerXControl on page 3-5
Base+0x0C	Write	-	-	Timer1IntClr	See Interrupt Clear Register. TimerXIntClr on page 3-6
Base+0x10	Read	1	0x0	Timer1RIS	See Raw Interrupt Status Register, TimerXRIS on page 3-6
Base+0x14	Read	1	0x0	Timer1MIS	See Masked Interrupt Status Register, TimerXMIS on page 3-7
Base+0x18	Read/write	32	0x00000000	Timer1BGLoad	See Background Load Register, TimerXBGLoad on page 3-7
Base+0x20	Read/write	32	0x00000000	Timer2Load	See Load Register, TimerXLoad on page 3-4
Base+0x24	Read	32	0xFFFFFFFF	Timer2Value	See Current Value Register, TimerXValue on page 3-5
Base+0x28	Read/write	8	0x20	Timer2Control	See Control Register, TimerXControl on page 3-5
Base+0x2C	Write	-	-	Timer2IntClr	See Interrupt Clear Register. TimerXIntClr on page 3-6
Base+0x30	Read	1	0x0	Timer2RIS	See Raw Interrupt Status Register, TimerXRIS on page 3-6
Base+0x34	Read	1	0x0	Timer2MIS	See Masked Interrupt Status Register, TimerXMIS on page 3-7
Base+0x38	Read/write	32	0x00000000	Timer2BGLoad	See Background Load Register, TimerXBGLoad on page 3-7

5.24 UART-Summary of registers

Table 3-1 Register summary

Offset	Type	Width	Reset value	Name	Description
0x000	RW	12/8	0x---	UARTDR	*Data register, UARTDR* on page 3-5
0x004	RW	4/0	0x0	UARTRSR/ UARTECR	Receive status register/error clear register, UARTRSR/UARTECR on page 3-6
0x008-0x014	-	-	-	-	Reserved
0x018	RO	9	0b-10010---	UARTFR	Flag register, UARTFR on page 3-8
0x01C-0x020	-	-	-	-	Reserved
0x024	RW	16	0x0000	UARTIBRD	Integer baud rate register, UARTIBRD on page 3-10
0x028	RW	6	0x00	UARTFBRD	Fractional baud rate register, UARTFBRD on page 3-10
0x02C	RW	8	0x00	UARTLCR_H	Line control register, UARTLCR_H on page 3-12
0x030	RW	16	0x0300	UARTCR	Control register, UARTCR on page 3-15
0x034	RW	6	0x12	UARTIFLS	Interrupt FIFO level select register, UARTIFLS on page 3-17
0x038	RW	11	0x000	UARTIMSC	Interrupt mask set/clear register, UARTIMSC on page 3-17
0x03C	RO	11	0x00-	UARTRIS	Raw interrupt status register, UARTRIS on page 3-19
0x040	RO	11	0x00-	UARTMIS	Masked interrupt status register, UARTMIS on page 3-20
0x044	WO	11	-	UARTICR	Interrupt clear register, UARTICR on page 3-21
0x048	RW	3	0x00	UARTDMACR	DMA control register, UARTDMACR on page 3-22

5.25 WatchDog-Summary of registers

Table 3-1 Summary of Watchdog module registers

Address	Type	Width	Reset value	Name	Description
Base + 0x00	Read/write	32	0xFFFFFFFF	WdogLoad	See *Load Register,WdogLoad* on page 3-4
Base + 0x04	Read-only	32	0xFFFFFFFF	WdogValue	See *Value Register, WdogValue* on page 3-4
Base + 0x08	Read/write	2	0x0	WdogControl	See *Control register, WdogControl* on page 3-4
Base + 0x0C	Write-only	-	-	WdogIntClr	See *Interrupt Clear Register, WdogIntClr* on page 3-5
Base + 0x10	Read-only	1	0x0	WdogRIS	See *Raw Interrupt Status Register, WdogRIS* on page 3-5
Base + 0x14	Read-only	1	0x0	WdogMIS	See *Masked Interrupt Status Register, WdogMIS* on page 3-5
Base + 0x18-0xBFC	-	-	-	-	reserved
Base + 0xC00	Read/write	32	0x0	WdogLock	See Lock Register, WdogLock on page 3-5

5.26 flash-SPIcontrol

5.26.1 Overview

5.26.1.1 Characteristics of this spi controller

The FlashSpi module is an spi master controller that can be configured through AHB bus. It is suitable for wifi chips based on AHB bus architecture like S902, and is used to read and write off-chip flash chips such as S25FL116K of Spansion, W25Q20CL of Winbond or similar off-chip flash chips that provide spi slave interface. This module has the following features:

1) provide a set of AHB slave interfaces, a set of DMA Single request interfaces and an interrupt request in the chip, and provide spi master interfaces outside the chip.

2) It can provide spi clock with the fastest half of the system clock frequency.

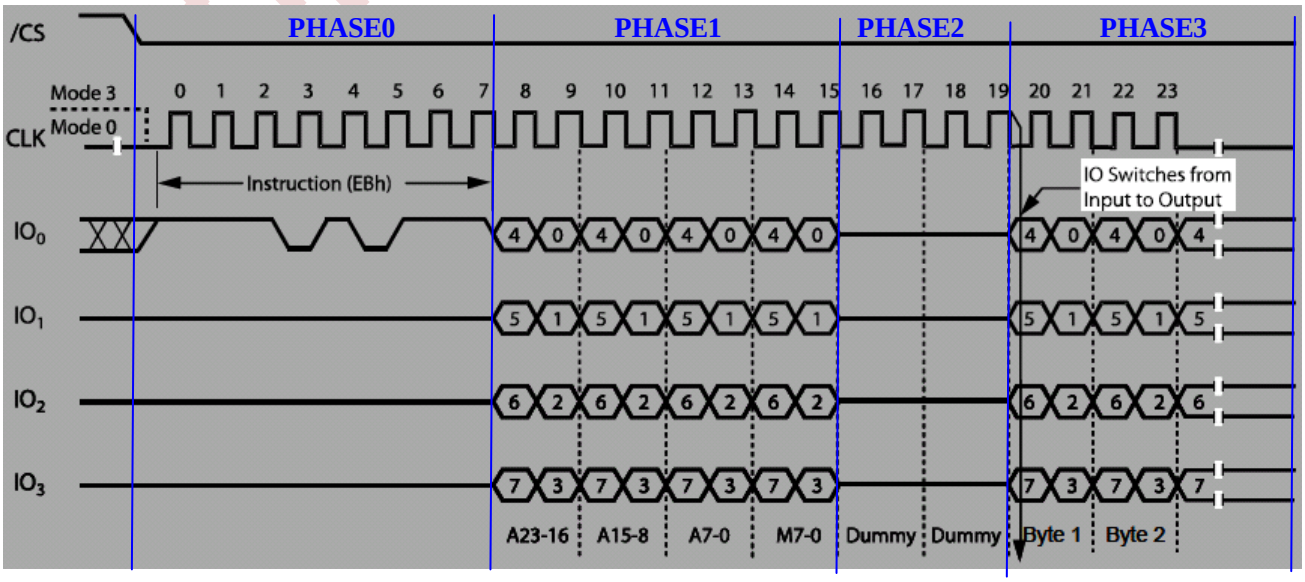
3) For a communication, it starts automatically after the register is configured. After the communication is completely finished, the register flag bit is displayed, and the completion interrupt can also be generated.

4) One communication can contain up to 8 independently configurable phase, which is enough to flexibly correspond to various situations contained in one spi communication of flash chip.

5) The whole module adopts synchronous clock design, and all signals belong to CLK clock domain.

5.26.1.2 the concept of PHASE

The following figure is a typical timing diagram of reading FLASH data through SPI.

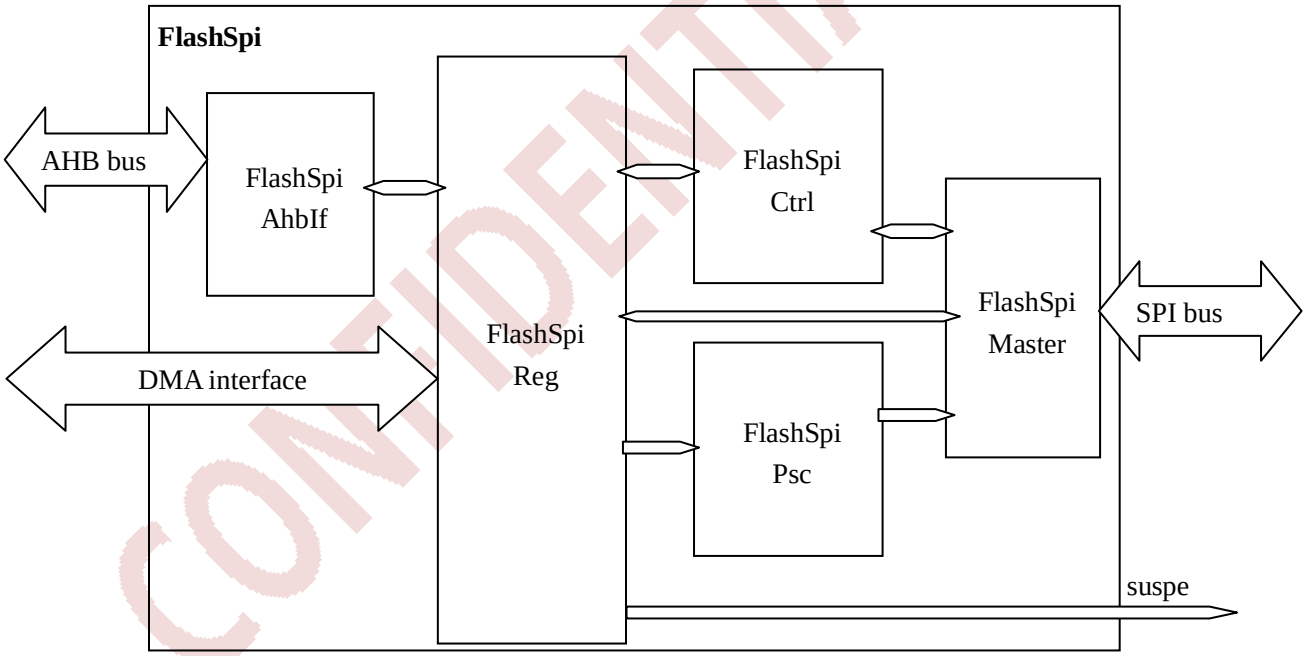


The process from CSn falling to CSn rising is called an SPI communication. In the SPI

communication shown above, Instruction, A, M and Dummy should be sent in turn, and then data should be received. And Instruction only uses one line, while the subsequent processes all use four lines. Aiming at this kind of timing, this module puts forward the concept of PHASE. A communication can contain up to 8 PHASEs, and each phase can independently set parameters such as read-write operation, 1/2/4 line mode, communication data volume, etc. After the first SPI communication starts, this module will finish the tasks set by each PHASE in the order of PHASE0→PHASE7, and then end the SPI communication. For example, for the timing of the above figure, it can be divided into 4 PHASE.

5.26.1.3 Module block diagram

The following figure is a block diagram of this spi controller:



FlashSpiAhbIf: an AHB Slave interface controller, which is responsible for converting the AHB bus signal into the internal read-write signal of the module and interacting with the register module FlashSpiReg to complete the read-write operation.

Flaspireg: register module of SPI controller, in which all registers are located. At the same time, the generation and processing of DMA interface signals and the generation of interrupts are also in this module.

Flaspictrl: the core control module of SPI controller. When it is detected that the SPI_START bit in FlashSpiReg is written as 1, the phase set in the register is analyzed to generate control and data for FlashSpiDataPump.

FlashSpiPsc：spi sck clock controller, used to control the frequency of sck clock.

FlashSpimaster: Spimaster interface controller, which controls the spi bus to send and receive according to the control signal sent by FlashSpiCtrl.

5.26.1.4 Top port

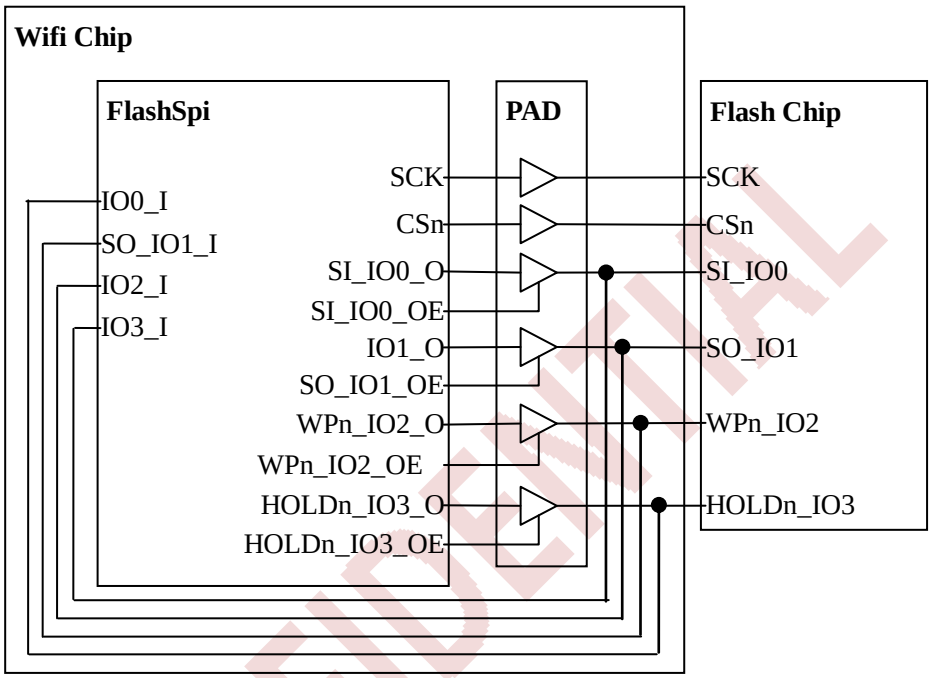
The following is a list of top-level ports and functions of this module:

Signal name	direction	bit wide	Connecting objects	explain
CLK	I	1	ClockGen	Module clock
RST_n	I	1	ResetGen	Module reset
HSEL	I	1	BusMatrix	AHB Slave bus signal
HWRITE	I	1	BusMatrix	AHB Slave bus signal
HADDR	I	32	BusMatrix	AHB Slave bus signal
HTRANS	I	2	BusMatrix	AHB Slave bus signal
HSIZE	I	3	BusMatrix	AHB Slave bus signal
HWDATA	I	32	BusMatrix	AHB Slave bus signal
HREADYIN	I	1	BusMatrix	AHB Slave bus signal
DMA_TX_SREQ_CLR	I	1	Dma	Clear TX_DMA request signal
DMA_RX_SREQ_CLR	I	1	Dma	Clear RX_DMA request signal
IO0_I	I	1	Pad	Spiio0 input of bus
SO_IO1_I	I	1	Pad	Spiso _ io1 input of bus
IO2_I	I	1	Pad	Spiio2 input of bus
IO3_I	I	1	Pad	Spiio3 input of bus
HREADYOUT	O	1	BusMatrix	AHB Slave bus signal
HRDATA	O	32	BusMatrix	AHB Slave bus signal
HRESP	O	2	BusMatrix	AHB Slave bus signal
DMA_TX_SREQ	O	1	Dma	TX_DMA request signal
DMA_RX_SREQ	O	1	Dma	RX_DMA request signal
SPI_DONE_INT	O	1	Cpu	SPI completion interrupt
SCK	O	1	Pad	Spisck signal of bus
CSn	O	1	Pad	Spicsn bus signal
SI_IO0_O	O	1	Pad	SPI bus SI_IO0 output data
SI_IO0_OE	O	1	Pad	SPI bus SI_IO0 output enable
IO1_O	O	1	Pad	Spiio1 bus output data
SO_IO1_OE	O	1	Pad	Spiio1 output enable for bus
WPn_IO2_O	O	1	Pad	Spiwpn _ io2 bus output data
WPn_IO2_OE	O	1	Pad	Spiwpn _ io2 bus output enable
HOLDn_IO3_O	O	1	Pad	Spiholdn _ io3 bus output data
HOLDn_IO3_OE	O	1	Pad	Spiholdn _ io3 bus output enable

5.26.2 Instructions for use of the module

5.26.2.1 System integration method

The integration of AHB bus, DMA interface and interrupt interface is relatively simple, which will not be described here. The integration of SPI bus should be equivalent to the following logic:

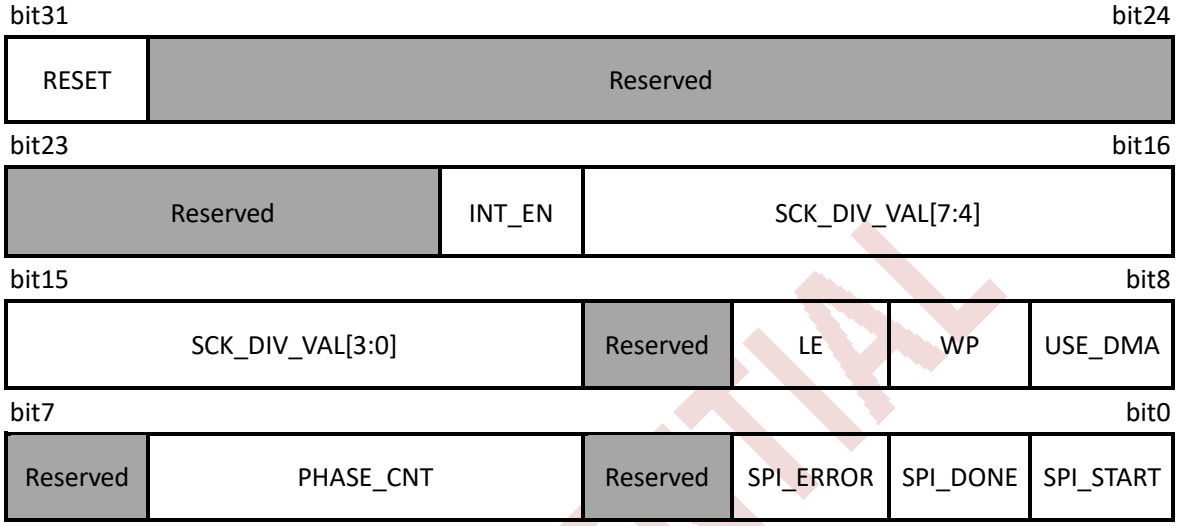


In addition, in order to prevent the input from floating, it is best to add a pull up resistor between Wifi Chip and Flash Chip.

5.26.2.2 register description

5.26.2.2.1 SPCR register (address: BASE_ADDR+8'h00)

SPCR is the global control register of SPI communication, and its bit configuration is as follows:

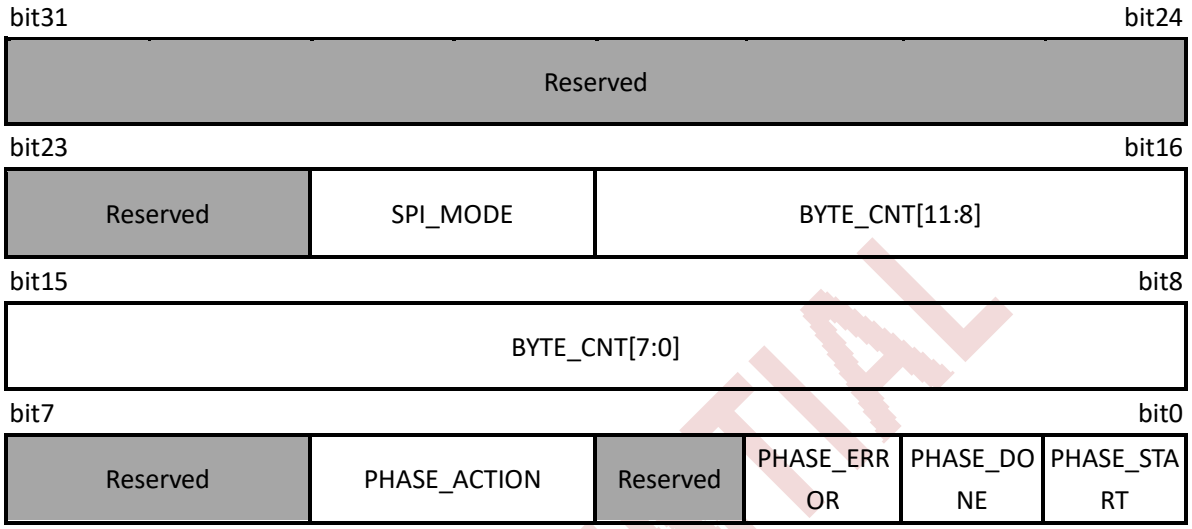


The function of each bit is defined in the following table:

bit	Bitname	initial value	read and write	describe
31	RESET	1'b0	W/R	Software reset of SPI controller. 1, all registers and all internal circuits except this bit are reset.
30-21	Reserved	10'h000	R	Keep.
20	INT_EN	1'b0	W/R	Interrupt enable. When set to 1, if SPI_DONE is 1, an interrupt will be sent to the CPU.
19-12	SCK_DIV_VAL	8'h08	W/R	SPI communication frequency setting. SPI communication rate is the frequency division of this value of the system clock. Even number must be filled in. Fill in 8'h00 to represent 256 frequency division.
11	Reserved	1'b0	R	Keep.
10	LE	1'b0	W/R	Small start. Because the data register is 32 bits, and SPI sends one byte at a time, when LE is set to 1, SPI will first send and receive [7:0] bits of the data register, then [15:8] until [31: 24]; When LE is set to 0, the order is reversed.
9	WP	1'b1	W/R	When SPI communication is in single or dual mode, WPn is valid when WP is 1, that is, 0; WPn is 1 when WP is 0.
8	USE_DMA	1'b0	W/R	Use DMA transfer for data of the last phase. Note: If there is only one phase in one spi communication, it is forbidden to set USE_DMA to 1.
7	Reserved	1'b0	R	Keep.
6-4	PHASE_CNT	3'h0	W/R	Number of PHASE included in one spi communication. 0: contains 1 phase. 1: contains 2 phase. …… 7: contains 8 phase.
3	Reserved	1'b0	R	Keep.
2	SPI_ERROR	1'b0	W0/R	When a communication is over, if any of the phase is wrong, this bit is set to one. Clear condition: software writes 0, or software writes SPI_START to 1 (that is, it is automatically cleared when the next communication starts).
1	SPI_DONE	1'b0	W0/R	SPI transmission and DMA transmission of the first communication have all ended. Clear condition: software writes 0, or software writes SPI_START to 1 (that is, it is automatically cleared when the next communication starts).
0	SPI_START	1'b0	W/R	Communication begins. Please set the ratio close to 1 after all the phase configurations of one communication. Clear condition: automatically clear after SPI communication ends. Please do not write 0 in the software.

5.26.2.2.2 phase _ ctrl0 ~ phase _ ctrl7 registers (address: base _ addr+8' h10 ~ 8' h2c)

The PHASE_CTRL register is used to individually configure each phase in an SPI communication. There are eight PHASE_CTRL registers, phase _ ctrl0 ~ phase _ ctrl7, which control each phase in turn. The bit configuration of the PHASE_CTRL register is as follows:



The function of each bit is defined in the following table:

bit	Bitname	initial value	read and write	describe
31-22	Reserved	10'h000	R	Keep.
21-20	SPI_MODE	2'h0	W/R	Current SPI bus mode of PHASE: 2' H0: Single mode 2' H1: Dual mode 2' H2: quad mode 2'h3: setting is prohibited.
19-8	BYTE_CNT	12'h000	W/R	Number of data byte in current PHASE communication. The value is invalid when PHASE_ACTION is set to POLL.
7-6	Reserved	1'b0	R	Keep.
5-4	PHASE_ACTION	2'h0	W/R	Action of current PHASE SPI: 2'h0：TX 2'h1：DUMMY TX 2'h2：RX 2'h3：POLL Please refer to section 2.3 for details.
3	Reserved	1'b0	R	Keep.
2	PHASE_ERROR	1'b0	R	PHASE_ACTION is POLL, and it exceeds the number of attempts that need to be unread. Clear condition: SPCR writes SPI_START to 1 (that is, it is automatically cleared when the next communication starts).
1	PHASE_DONE	1'b0	R	The current PHASE has been completed. Clear condition: SPCR writes SPI_START to 1 (that is, it is automatically cleared when the next communication starts).
0	PHASE_START	1'b0	R	PHASE is currently in progress. The hardware is automatically set and cleared according to the running situation.

5.26.2.2.3 phase _ data0 ~ phase _ data7 registers (address: base _ addr+8' h30 ~ 8' h4c)

Data register for each PHASE. The bit definition of the PHASE_DATA register is related to the PHASE _ action set by the current phase.
The details are as follows:
When PHASE_ACTION is POLL, it is used to save the configuration related to POLL:

bit31	bit23	bit15	bit7	bit0
POLL_LIMIT	POLL_MASK	POLL_EXPECT	POLL_READ	

When (POLL _ read & poll _ mask) = = poll _ expect, and the number of attempts is less than POLL_LIMIT, poll succeeds; Otherwise, the POLL fails and the PHASE_ERROR is set to one.

When the PHASE_ACTION is not POLL, the PHASE_DATA register is used to store the sent/received data, with a maximum of 4 byte.

If SPCR.LE is 0, the functions are as follows:

bit31	bit23	bit15	bit7	bit0
data byte 0	data byte 1	data byte 2	data byte 3	

If SPCR.LE is 1, the functions are as follows:

bit31	bit23	bit15	bit7	bit0
data byte 3	data byte 2	data byte 1	data byte 0	

That is, if sending data, SPI will first send data byte0, then data byte 1, until data byte 3. On the contrary, when receiving data, the position of data byte 0 will be written first until data byte 3.

5.26.2.3 description of PHASE_ACTION

When SPCR.SPI_START=1, the hardware will automatically install PHASE0 → phase1 ... phase7 to execute the transactions in each phase until the number specified in SPCR.PHASE_CNT is completed. According to the functions of common flash chips, there are four kinds of ACTION that SPI needs to perform in this module: TX, DUMMY TX, RX and POLL. Here's a detailed description of the definition of each operation.

5.26.2.3.1 TX

When PHASE_ACTION is TX, SPI will circularly send the data in PHASE_DATA in the order of data byte0 ~ data byte3 according to the bus mode configured by SPI_MODE until all the specified phase _ byte _ CNTs are sent. Therefore, when the PHASE_BYTE_CNT is less than 5, the software can directly match the data into the PHASE_DATA；. Otherwise, it is recommended to configure DMA, so that SPI will automatically call DMA to write new data into PHASE_DATA at the end of a cycle.

5.26.2.3.2 DUMMY TX

DUMMY TX is similar to TX except that SPI will no longer send data in PHASE_DATA, but will send 8'hFF. Therefore, even if the PHASE_BYTE_CNT is greater than 4, there is no need to configure DMA.

5.26.2.3.3 RX

When PHASE_ACTION is RX, SPI will receive data according to the bus mode configured by SPI_MODE, and write it into PHASE_DATA circularly in the order of data byte0 ~ data byte3. Therefore, when the PHASE_BYTE_CNT is less than 5, the software can read the data in the PHASE_DATA after the SPI communication. Otherwise, it is recommended to configure DMA, so that SPI will automatically call DMA to send away the data in PHASE_DATA at the end of a cycle to avoid being overwritten by new data.

5.26.2.3.4 POLL

When PHASE_ACTION is POLL, SPI will continue to receive data according to the bus mode configured by SPI_MODE, and do the comparison operation of (poll _ read & poll _ mask) = = poll _ expect. If it is successful within the number of times specified in POLL_LIMIT, the POLL operation is completed; Otherwise, after the specified number of times of POLL_LIMIT is reached, the POLL operation is forcibly completed, and the PHASE_ERROR is set to one.

Note: If the POLL_LIMIT is set to 8'h00, the infinite POLL mode will be entered. SPI will perform the POLL operation indefinitely until the comparison is successful, otherwise it cannot be stopped. The only way to stop is SOFT_RESET.

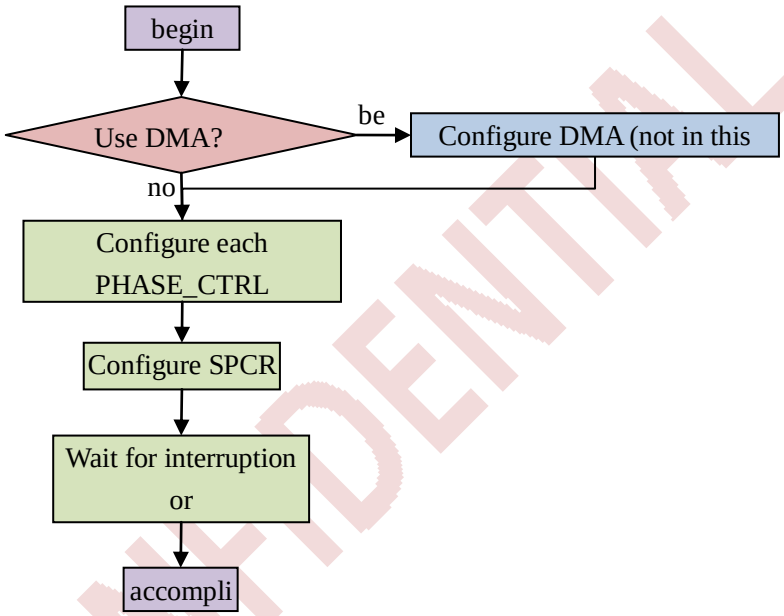
5.26.2.3.5 Pay attention.

1)RX and POLL must be the last PHASE of an SPI communication and not the first PHASE.

2) If the PHASE is set to use DMA, its PHASE_ACTION cannot be POLL.
3) If it is set to the PHASE using DMA, and its PHASE_ACTION is RX, it may happen that SPI communication has been completed, but DMA has not yet been completed. At this time, SPCR.SPI_START will be cleared, but SPCR.SPI_DONE will not be set. SPCR.SPI_DONE will not be set until the DMA transfer is complete.

5.26.2.4 Software configuration sequence

The software can configure the registers in the following order during each SPI communication:

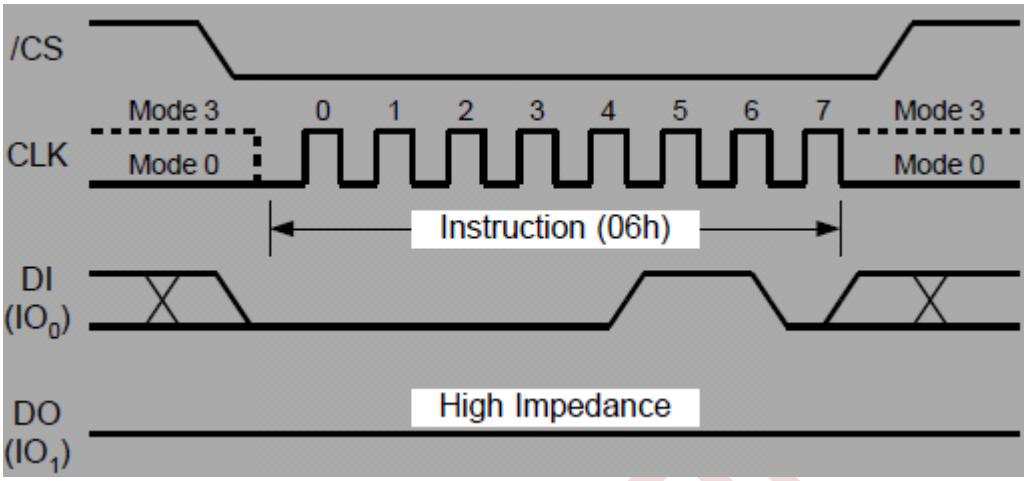


5.26.2.5 Example of software configuration

This section will introduce how to configure this module in combination with the common operations of common FLASH chips.

5.26.2.5.1 Send the Write Enable(06h) command.

The format of this command is as follows:



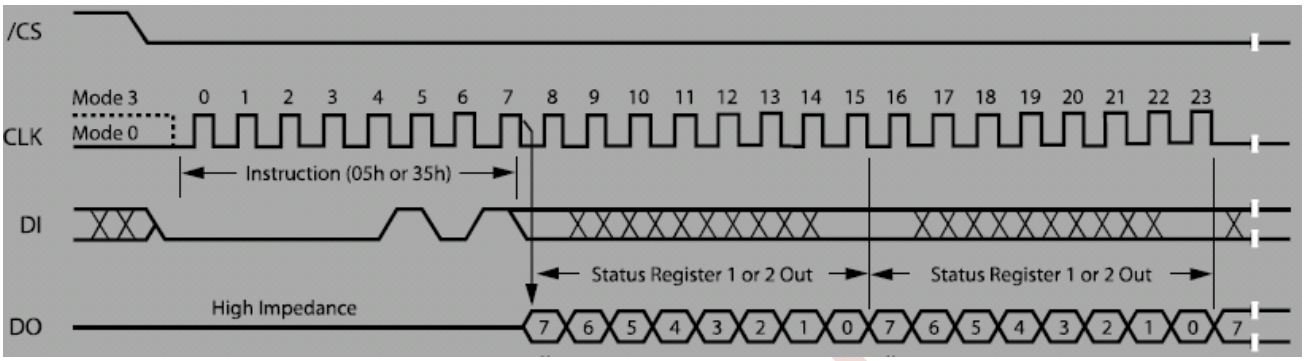
The following configuration is recommended:

sequence	Register name	Configuration value
one	PHASE_CTRL0	SPI_MODE=2'h0 BYTE_CNT=12'h001 PHASE_ACTION=2'h0
2	PHASE_DATA0	32'h0600_0000(SPCR.LE=0)/32'h0000_0006(SPCR.LE=1)
3	SPCR	USE_DMA=1'b0 PHASE_CNT=3'h0 WP=1'b0 SPI_START=1'b1

After configuration, SPI will start communication, and software can wait for interrupt or SPCR.SPI_START=0.

5.26.2.5.2 Send the Read Status Register-1(05h) command.

The sequence of this command is as follows:



Until CSn becomes 1, the value of Read Status Register-1 will be read repeatedly.

Example: read the value of Read Status Register-1 four times.

The following configuration is recommended:

sequence	Register name	Configuration value
one	PHASE_CTRL0	SPI_MODE=2'h0 BYTE_CNT=12'h001 PHASE_ACTION=2'h0
2	PHASE_DATA0	32'h0500_0000(SPCR.LE=0)/32'h0000_0005(SPCR.LE=1)
3	PHASE_CTRL1	SPI_MODE=2'h0 BYTE_CNT=12'h004 PHASE_ACTION=2'h2
4	SPCR	USE_DMA=1'b0 PHASE_CNT=3'h1 WP=1'b1 SPI_START=1'b1

After configuration, SPI will first run PHASE0, send 05h, then run PHASE1, read 4 byte, and store the read value in the PHASE_DATA1 register. The software can read the value in PHASE_DATA1 after waiting for SPI_START=0.

Example: wait for the 0th bit of Read Status Register-1 to be 0, but try to read it 100 times at most.

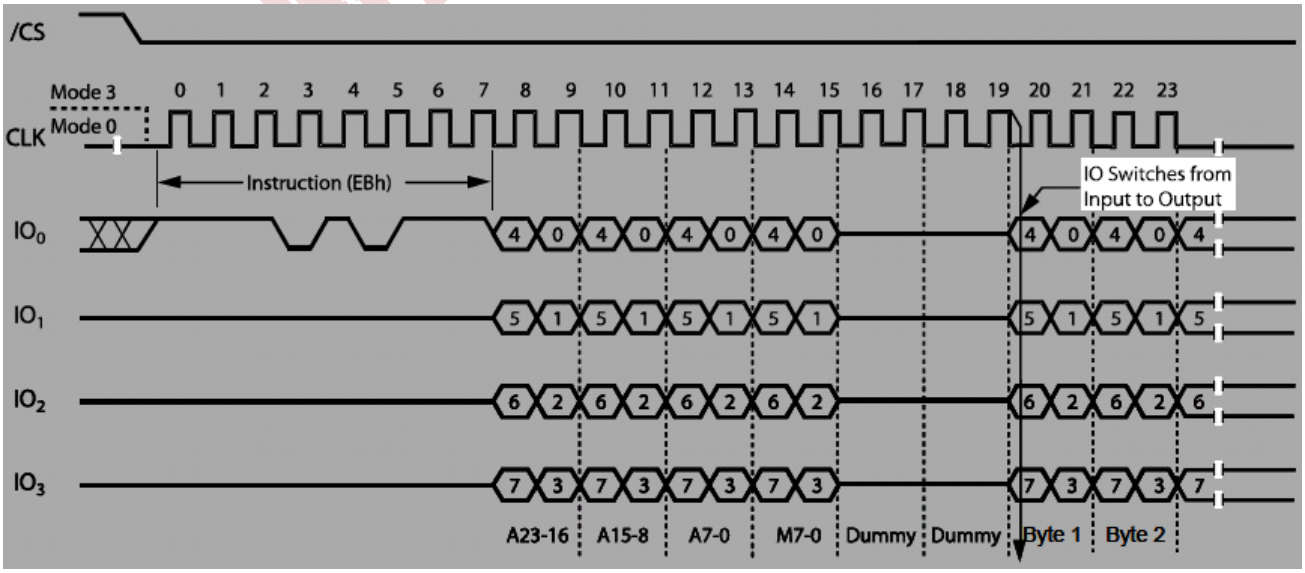
The following configuration is recommended:

sequence	Register name	Configuration value
1	PHASE_CTRL0	SPI_MODE=2'h0 BYTE_CNT=12'h001 PHASE_ACTION=2'h0
2	PHASE_DATA0	32'h0500_0000(SPCR.LE=0)/32'h0000_0005(SPCR.LE=1)
3	PHASE_CTRL1	SPI_MODE=2'h0 PHASE_ACTION=2'h3
4	PHASE_DATA1	32'h64010000
5	SPCR	USE_DMA=1'b0 PHASE_CNT=3'h1 WP=1'b1 SPI_START=1'b1

After the configuration is completed, spi will first run PHASE0, send 05h, then run phase1, constantly read data through SPI, compare the result with 0x00 and then compare it with 0x00 until the comparison is successful or exceeds 0x64 times. After waiting for SPI_START=0, the software can judge whether SPI_ERROR is 1 or not until the waiting is successful.

5.26.2.5.3 Read data with Fast Read Quad IO(EBh) command.

The timing diagram of this command is as follows:



Suppose the address is A=24'h123456 and M=78, and 100 data are read.

The recommended configuration is as follows:

sequence	Register name	Configuration value
1	DMA source address	PHASE_DATA3
2	PHASE_CTRL0	SPI_MODE=2'h0 BYTE_CNT=12'h001 PHASE_ACTION=2'h0
3	PHASE_DATA0	32'hEB00_0000(SPCR.LE=0)/32'h0000_00EB(SPCR.LE=1)
4	PHASE_CTRL1	SPI_MODE=2'h2 BYTE_CNT=12'h004 PHASE_ACTION=2'h0
5	PHASE_DATA1	32'h1234_5678(SPCR.LE=0)/32'h7856_3412(SPCR.LE=1)
6	PHASE_CTRL2	SPI_MODE=2'h2 BYTE_CNT=12'h002 PHASE_ACTION=2'h1
7	PHASE_CTRL3	SPI_MODE=2'h2 BYTE_CNT=12'h064 PHASE_ACTION=2'h2
8	SPCR	USE_DMA=1'b1 PHASE_CNT=3'h3 WP=1'b1 SPI_START=1'b1

After the communication starts, SPI first communicates with PHASE0 and sends 0xEB. Then send 0x12345678 of PHASE1. When configuring here, send A and M in one PHASE, because for SPI, A and M are the same sending data. Then send two dummy byte of PHASE2. At last, the one running PHASE3 charges 100 byte, and since SPCR.USE_DMA is set to 1, when PHASE_DATA3 is written, a DMA request will be sent to read the data in PHASE_DATA3.